Парадигма развития науки

Методологическое обеспечение

А.Е. Кононюк

ОСНОВЫ ТЕОРИИ ОПТИМИЗАЦИИ

Книга 2

Безусловная оптимизации

Часть 1

Киев
Освіта України
2011

УДК 51 (075.8)
ББК В161.я7
К 213

Рецензент: *Н.К.Печурин* - д-р техн. наук, проф. (Национальный авиационный университет).

Кононюк А.Е.
К65 Основы теории оптимизации. Безусловная оптимизация К.2.ч.1. Киев:"Освіта України", 2011. - 544 с.
ISBN 978-966-7599-50-8

Настоящая работа является систематическим изложением базовой теории оптимизации для конечномерных задач. Основное внимание уделяется идейным основам методов, их сравнительному анализу и примерам использования. Охвачен широкий круг задач — от безусловной минимизации до условной минимизации. Обсуждается методика постановки и решения прикладных проблем оптимизации. Приводятся условия экстремума, теоремы существования, единственности и устойчивости решения для основных классов задач. Исследуется влияние помех, негладкости функций, вырожденности минимума. Работа предназначена для магистров, аспирантов, докторантов, инженеров, экономистов, статистиков, вычислителей и всех тех, кто сталкивается с задачами оптимизации.

ББК В161.я7

ISBN 978-966-7599-50-8

Часть I
Методы безусловной оптимизации

Оглавление

1. Введение в теорию безусловной оптимизации …….6
1.1. Задачи оптимизации ……10
1.2. Краткий обзор методов оптимизации ……...17
1.3. Задача безусловной оптимизации ……..26
2. Методы одномерной оптимизация ……..42
2.1. Введение в одномерную оптимизацию ……42
2.2. Одномерная оптимизация ……..56
3. Методы одномерной минимизации нулевого порядка (прямые методы) ……..63
3.1. Общая характеристика методов нулевого порядка…… 70
3.2. Нелокальная линейная аппроксимация……71
3.3. Квадратичная аппроксимация……74
3.4. Метод перебора ……..76
3.5. Метод поразрядного поиска ……..78
3.6. Методы исключения отрезков……80
3.7. Метод Фибоначчи ……117
3.8. Метод конфигураций. ……..146
3.9. Метод деформируемого многогранника ……154
3.10. Метод прямого поиска (метод Хука-Дживса) ……170
3.11. Метод вращающихся координат (метод Розенброка)…….172
3.12. Метод параллельных касательных (метод Пауэлла) ……..175
3.13. Краткий обзор других методов ……177
4. Методы одномерной минимизации первого порядка ……178
4.1. Минимизация функций. Основные положения ……178
4.2. Метод парабол ……183
4.3. Градиентный метод как классический метод оптимизации .187
4.4. Метод наискорейшего спуска ……..195
4.5. Метод градиентного спуска…… 198
4.6. Градиентный метод с дроблением шага……..209
4.7. Метод сопряженных градиентов ……212
4.8. Методы оврагов ……225
4.9. Метод Флетчера-Ривса……..228

4.10. Минимизация неквадратичной целевой функции…… 235
4.11. Метод Дэвидона — Флетчера — Пауэлла (ДФП)……236
4.12. Некоторые методы первого порядка в иной интерпретации ……237
5. Методы минимизации второго порядка ……248
5.1. Особенности методов второго порядка ……248
5.2. Методы линейной аппроксимации……...250
5.3. Интерполяция кубическими сплайнами ……260
5.4. Метод Ньютона ……269
5.5. Метод касательных (Ньютона)……274
5.6. Метод Коши…… 290
5.7. Метод Марквардта ……..292
5.8. Связь методов Ньютона и сопряженных градиентов…….. 294
5.9. Сравнение методов одномерного поиска ……..305
5.10. Многошаговые методы ……310
5.11. Краткий анализ методов одномерной минимизации……. 319
6. Методы многомерной безусловной оптимизации ……327
6.1. Введение в методы многомерной оптимизации ……..327
6.2. Постановка задачи многомерной оптимизации. ……330
6.3. Критерий оптимальности для функции многих переменных 335
6.4. Квадратичная функция аргумента $\vec{x}$ ……341
6.5. Рельеф поверхности целевой функции *f(x)*. ……..342
6.6. Введение в методы безусловной минимизации функций многих переменных ……344
6.7. Многомерный поиск без использования производных…….364
6.8. Методы минимизации первого порядка. ……... 381
6.9. Методы второго порядка ……409
7. Методы анализа многомерной безусловной оптимизации ……415
7.1. Анализ методов прямого поиска ……416
7.2. Анализ методов первого и второго порядков…….. 430
7.3. Обобщённый алгоритм…… 441
8. Методы оптимизации овражных функций…… 442
9. Влияние помех на поведение методов безусловной минимизации 452
9.1. Источники и типы помех ……453
9.2. Градиентный метод при наличии помех…… 455
9.3. Другие методы минимизации при наличии помех…… 459
9.4. Прямые методы ……462
9.5. Оптимальные методы при наличии помех ……466
9.6. Псевдоградиентный метод с возмущением на входе для нестационарной задачи безусловной оптимизации ……..472

10. Стратегия оптимизационного исследования ……482
10.1. Построение модели ……482
10.2. Реализация модели ……484
10.3. Преодоление вычислительных трудностей……..486
10.4. Анализ модели ……487
10.5. Методы поиска и оценки решений…… 489
Приложения……495
Список обозначений…… .545
Литература. ……547

1. Введение в теорию безусловной оптимизации

Оптимизация как раздел математики существует достаточно давно. Оптимизация - это выбор, т.е. то, чем постоянно приходится заниматься в повседневной жизни. Термином "оптимизация" в литературе обозначают процесс или последовательность операций, позволяющих получить уточненное решение. Хотя конечной целью оптимизации является отыскание наилучшего или "оптимального" решения, обычно приходится довольствоваться улучшением известных решений, а не доведением их до совершенства. Поэтому под оптимизацией понимают скорее стремление к совершенству, которое, возможно, и не будет достигнуто.

Необходимость принятия наилучших решений так же стара, как само человечество. Испокон веку люди, приступая к осуществлению своих мероприятий, раздумывали над их возможными последствиями и принимали решения, выбирая тем или другим образом зависящие от них параметры - способы организации мероприятий. Но до поры, до времени решения могли приниматься без специального математического анализа, просто на основе опыта и здравого смысла.

Возьмем пример: человек вышел утром из дому, чтобы ехать на работу. По ходу дела ему приходится принять целый ряд решений: брать ли с собой зонтик? В каком месте перейти улицу? Каким видом транспорта воспользоваться? И так далее. Разумеется, все эти решения человек принимает без специальных расчетов, просто опираясь на имеющийся у него опыт и на здравый смысл. Для обоснования таких решений никакая наука не нужна, да вряд ли понадобится и в дальнейшем.

Однако возьмем другой пример. Допустим, организуется работа городского транспорта. В нашем распоряжении имеется какое-то количество транспортных средств. Необходимо принять ряд решений, например: какое количество и каких транспортных средств направить по тому или другому маршруту? Как изменять частоту следования машин в зависимости от времени суток? Где разместить остановки? И так далее.

Эти решения являются гораздо более ответственными, чем решения предыдущего примера. В силу сложности явления последствия каждого из них не столь ясны; для того, чтобы представить себе эти последствия, нужно провести расчеты. А главное, от этих решений гораздо больше зависит. В первом примере неправильный выбор решения затронет интересы одного человека; во

втором - может отразиться на деловой жизни целого города. Конечно, и во втором примере при выборе решения можно действовать интуитивно, опираясь на опыт и здравый смысл. Но решения окажутся гораздо более разумными, если они будут подкреплены количественными, математическими расчетами. Эти предварительные расчеты помогут избежать длительного и дорогостоящего поиска правильного решения "на ощупь".

Наиболее сложно обстоит дело с принятием решений, когда речь идет о мероприятиях, опыта в проведении которых еще не существует и, следовательно, здравому смыслу не на что опереться, а интуиция может обмануть. Пусть, например, составляется перспективный план развития вооружения на несколько лет вперед. Образцы вооружения, о которых может идти речь, еще не существуют, никакого опыта их применения нет. При планировании приходится опираться на большое количество данных, относящихся не столько к прошлому опыту, сколько к предвидимому будущему. Выбранное решение должно по возможности гарантировать нас от ошибок, связанных с неточным прогнозированием, и быть достаточно эффективным для широкого круга условий. Для обоснования такого решения приводится в действие сложная система математических расчетов.

Вообще, чем сложнее организуемое мероприятие, чем больше вкладывается в него материальных средств, чем шире спектр его возможных последствий, тем менее допустимы так называемые "волевые" решения, не опирающиеся на научный расчет, и тем большее значение получает совокупность научных методов, позволяющих заранее оценить последствия каждого решения, заранее отбросить недопустимые варианты и рекомендовать те, которые представляются наиболее удачными.

Практика порождает все новые и новые задачи оптимизации причем их сложность растет. Требуются новые математические модели и методы, которые учитывают наличие многих критериев, проводят глобальный поиск оптимума. Другими словами, жизнь заставляет развивать математический аппарат оптимизации.

Реальные прикладные задачи оптимизации очень сложны. Современные методы оптимизации далеко не всегда справляются с решением реальных задач без помощи человека. Нет пока такой теории, которая учла бы любые особенности функций, описывающих постановку задачи. Следует отдавать предпочтение таким методам, которыми проще управлять в процессе решения задачи.

Историческая справка

Задачи линейного программирования были первыми, подробно изученными задачами поиска экстремума функций при наличии ограничений типа неравенств. В 1820 г. Ж. Фурье и затем в 1947 г. Дж. Данциг предложил метод направленного перебора смежных вершин в направлении возрастания целевой функции — симплекс-метод, ставший основным при решении задач линейного программирования.

Присутствие в названии дисциплины термина «программирование» объясняется тем, что первые исследования и первые приложения линейных оптимизационных задач были в сфере экономики, так как в английском языке слово «programming» означает планирование, составление планов или программ. Вполне естественно, что терминология отражает тесную связь, существующую между математической постановкой задачи и её экономической интерпретацией (изучение оптимальной экономической программы). Термин «линейное программирование» был предложен Дж. Данцигом в 1949 г. для изучения теоретических и алгоритмических задач, связанных с оптимизацией линейных функций при линейных ограничениях. Поэтому наименование «Математическое программирование» связано с тем, что целью решения задач является выбор оптимальной программы действий.

Выделение класса экстремальных задач, определяемых линейным функционалом на множестве, задаваемом линейными ограничениями, следует отнести к 30-м годам XX столетия. Одними из первых, исследовавшими в общей форме задачи линейного программирования, были: Джон фон Нейман, знаменитый математик и физик, доказавший основную теорему о матричных играх и изучивший экономическую модель, носящую его имя; лауреат Нобелевской премии (1975 г.) Л. В. Канторович, сформулировавший ряд задач линейного программирования и предложивший (1939 г.) метод их решения (метод разрешающих множителей), незначительно отличающийся от симплекс-метода.

В 1931 г. венгерский математик Б. Эгервари рассмотрел математическую постановку и решил задачу линейного программирования, имеющую название «проблема выбора», метод решения получил название «венгерского метода».

Л. В. Канторовичем совместно с М. К. Гавуриным в 1949 г разработан метод потенциалов, который применяется при решении

транспортных задач. В последующих работах Л. В. Канторовича, В. С. Немчинова, В. В. Новожилова, А. Л. Лурье, А. Брудно, А. Г. Аганбегяна, Д. Б. Юдина, Е. Г. Гольштейна и других математиков и экономистов получили дальнейшее развитие как математическая теория линейного и нелинейного программирования, так и приложение её методов к исследованию различных экономических проблем. Методам линейного программирования посвящено много работ зарубежных ученых. В 1941 г. Ф. Л. Хитчкок поставил транспортную задачу. Основной метод решения задач линейного программирования — симплекс-метод — был опубликован в 1949 г Дж. Данцигом. Дальнейшее развитие методы линейного и нелинейного программирования получили в работах Г. Куна (англ.), А. Таккера (англ.), Гасса (Gass S. I.), Чарнеса (Charnes A.), Била (Beale E. M.) и др.

Одновременно с развитием линейного программирования большое внимание уделялось задачам нелинейного программирования, в которых либо целевая функция, либо ограничения, либо то и другое нелинейны. В 1951 г была опубликована работа Куна и Таккера, в которой приведены необходимые и достаточные условия оптимальности для решения задач нелинейного программирования. Эта работа послужила основой для последующих исследований в этой области.

Начиная с 1955 г опубликовано много работ, посвященных квадратическому программированию (работы Била, Э. Баранкина (Barankin E.) и Дорфмана (Dorfman R.), Франка (Frank M.) и Вольфа (Wolfe P.), Г. Марковица и др.). В работах Денниса (Dennis J. B.), Розена (Rosen J. B.) и Зонтендейка (Zontendijk G.) разработаны градиентные методы решения задач нелинейного программирования.

В настоящее время для эффективного применения методов математического программирования и решения задач на компьютерах разработаны алгебраические языки моделирования, представителями которыми являются AMPL и LINGO.

1.1. Задачи оптимизации

1.1.1. Обозначения.

Всюду ниже **R** — множество вещественных, **N** — натуральных, а **C** — комплексных чисел. С самого начала мы будем использовать векторные обозначения. Всегда через $\mathbf{R}^m$ обозначается m-мерное вещественное линейное пространство. При этом мы всегда считаем, что в $\mathbf{R}^m$ фиксирован базис и отождествляем $\mathbf{R}^m$ с арифметическим m-мерным пространством (пространством упорядоченных наборов m вещественных чисел). Буква Θ будет обозначать нуль пространства $\mathbf{R}^m$. Индекс внизу всегда обозначает координату вектора, например, x_i — это i-ая координата вектора x. Последовательности мы обычно будем обозначать индексом вверху: $\{x^n\}$.

Через $(\cdot\,,\cdot)$ обозначается каноническое скалярное произведение в $\mathbf{R}^m$: $(x,\ y) = \sum^m_{i=1} x_i y_i$. Если не оговорено противное, порожденную скалярным произведением: $\|\cdot\| = (\sum^m_{i=1} x_i^2)^{1/2}$.

Обозначение $B(x_0,\ r)$ закреплено для шара в пространстве $\mathbf{R}^m$ с центром в x_0 радиуса r: $B(x_0, r) = \{x \in \mathbf{R}^m: \|x - x_0\| \le r\}$.

Если $A=\{a_{ij}\}^{n,\,m}_{i=1,\,j=1}$ — $n\times m$-матрица, то через A также обозначается и линейный оператор из $\mathbf{R}^n$ в $\mathbf{R}^m$, задаваемый этой матрицей.

Для двух векторов $x, y \in \mathbf{R}^m$ мы будем писать $x \le y$, если $x_i \le y_i$ при всех $i = 1, ..., m$; здесь x_i и y_i — i-е координаты векторов x и y, соответственно.

Мы будем различать обозначение f: $X \to Y$ отображения, действующего из множества X во множество Y, и обозначение f: $x \to y$ (или $x \to f(x)$) отображения, переводящего точку x в точку $f(x)$, а также обозначение f отображения и обозначение $f(x)$ значения отображения f в точке x.

1.1.2. Задача наилучшего приближения.

Если рассматривать систему n линейных уравнений с m неизвестными

$$Ax = b$$

в случае, когда она переопределена, то иногда оказывается естественной задача о нахождении вектора x, который "удовлетворяет этой системе наилучшим образом", т. е. из всех "не решений" является лучшим. Например, бывает полезной задача о нахождении вектора x, для которого разность правой и левой частей системы (невязка) минимальна, т. е. минимальна функция

$$f(x) = ||Ax - b||. \qquad (1)$$

Эту задачу символически записывают в виде

$$f(x) \to \min$$

Норму в (1) можно брать разную. Например, если взята евклидова норма, то получается *задача о наилучшем квадратичном приближении*

$$\left(\sum_{i=1}^{n} \left| \sum_{j=1}^{m} a_{ij}x_j - b_i \right|^2 \right)^{1/2} \to \min,$$

или, что эквивалентно,

$$\sum_{i=1}^{n} \left\| \sum_{j=1}^{m} a_{ij}x_j - b_i \right\|^2 \to \min,$$

Геометрически эта задача интерпретируется как задача о нахождении на гиперплоскости $A(\mathbf{R}^m)$ в пространстве $\mathbf{R}^n$ точки, ближайшей к точке $b = (b_1, ..., b_n)$.

1.1.3. Задача Штейнера.

Классическая *задача Штейнера* формулируется так: требуется найти точку $x \in \mathbf{R}^m$, сумма расстояний от которой до заданных точек $x^1, ..., x^n \in \mathbf{R}^m$ минимальна. Эта задача типично оптимизационная:

$$f(x) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \|x - x^i\| \to \min$$

Приведенные выше задачи представляют собой задачи *безусловной оптимизации* — на искомое решение не налагается никаких дополнительных условий, кроме того, что оно должно доставлять минимум некоторой функции (другими словами, минимум функции ищется на всем пространстве — области определения функции). Чаще встречаются задачи *условной оптимизации*, примеры которых мы приводим ниже.

1.1.4. Задача о рационе.

Пусть имеется n различных пищевых продуктов, содержащих m различных питательных веществ. Обозначим через a_{ij} содержание (долю) j-го питательного вещества в i-ом продукте, через b_j — суточную потребность организма в j-ом питательном веществе, через c_i — стоимость единицы i-го продукта. Требуется составить суточный рацион питания минимальной стоимости, удовлетворяющий потребность во всех питательных веществах. Если обозначить через x_i суточное потребление i-го продукта, то эта задача может быть формализована следующим образом. Нужно минимизировать функцию

$$f(x_1, ..., x_n) = \sum_{i=1}^{n} c_i x_i \quad \text{(стоимость рациона)}$$

при условиях

$$\sum_{i=1}^{n} a_{ij}x_i \geq b_j, \quad j = 1, ..., m$$

(рацион должен содержать не менее суточной потребности в каждом из питательных веществ).

Очевидно, также следует требовать, чтобы

$x_i \geq 0, \quad i = 1, ..., n.$

В векторных обозначениях задача о рационе может быть записана так: минимизировать функцию

$f(x) = (c, x),$

где $c = (c_1, ..., c_n) \in \mathbf{R}^n$; эту задачу, как обычно, записывают в виде

$(c, x) \rightarrow \min,$

при ограничениях

$Ax \geq b,$

$x \geq \Theta.$

В них первое неравенство связывает два вектора Ax и b из $\mathbf{R}^m$, а второе – два вектора x и Θ из $\mathbf{R}^n$.

По легенде одним из первых приложений задачи о рационе к реальной жизни была попытка рассчитать оптимальный рацион для американской армии во время второй мировой войны. Результат был неожиданным: солдат в день должен выпивать литр уксуса и съедать килограм бобов (цифры и продукты условные).

1.1.5. Транспортная задача.

Эта задача — классическая задача линейного программирования. К ней сводятся многие оптимизационные задачи. Формулируется она так. На m складах находится груз, который нужно развезти n потребителям. Пусть a_i ($i = 1, ..., n$) — количество груза на i-ом складе, а b_j ($j = 1, ..., m$) — потребность в грузе j-го потребителя, c_{ij} — стоимость перевозки единицы груза с i-го склада j-му потребителю. Требуется минимизировать стоимость перевозок. Если обозначить через x_{ij} объем перевозок с i-го склада j-му потребителю, то транспортная задача формализуется так:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \to \min,$$

$$\sum_{i=1}^{n} x_{ij} = b_j, \quad j = 1, ..., m$$

(все потребители должны быть удовлетворены),

$$\sum_{j=1}^{m} x_{ij} = a_i, \quad i = 1, ..., n$$

(весь груз должен быть доставлен потребителю),

$x_{ij} \geq 0$

(нельзя перевозить груз от потребителя на склад).

Это были примеры линейных задач условной оптимизации. Приведем один пример нелинейной задачи.

1.1.6. Задачи о распределении ресурсов.

Общий смысл таких задач — распределить ограниченный ресурс между потребителями оптимальным образом. Рассмотрим простейший пример — *задачу о режиме работы энергосистемы.* Пусть m

электростанций питают одну нагрузку мощности p. Обозначим через x_j активную мощность, генерируемую j-ой электростанцией. Техническими условиями определяются возможный минимум μ_j и максимум M_j вырабатываемой j-ой электростанцией мощности. Допустим затраты на генерацию мощности x на j-ой электростанции равны $e_j(x)$. Требуется сгенерировать требуемую мощность p при минимальных затратах. В наших обозначениях

$$f(x) \stackrel{\text{def}}{=} \sum_{j=1}^{m} e_j(x_j) \to \min,$$

$$\sum_{j=1}^{m} x_j = p,$$

$\mu_j \le x_j \le M_j, \quad j = 1, ..., m.$

Если обозначить $\sum_{j=1}^{m} e_j(x_j)$через $f(x)$, $\sum_{j=1}^{m} x_j$– p через $g(x)$, а $\{x \in \mathbf{R}^m: \mu \le x \le M\}$ через Ω, то эта задача переписывается так

$f(x) \to \min,$

$g(x)=0,$

$x \in \Omega.$

1.1.7. О классификации задач оптимизации.

Один из классификационных признаков делит оптимизационные задачи на два класса: *задачи безусловной оптимизации* и *задачи условной оптимизации*. Первые из них характеризуются тем, что минимум функции f: $\mathbf{R}^m \to \mathbf{R}$ ищется на всем пространстве:

$$f(x) \to \min, \quad x \in \mathbf{R}^m. \tag{2}$$

В задачах же второго класса поиск минимума идет на некотором собственном подмножестве Ω пространства $\mathbf{R}^m$:

$$f(x) \to \min, \quad x \in \Omega. \tag{3}$$

Множество Ω часто выделяется *ограничениями типа равенств*

$$g_0(x) = \Theta, \tag{4}$$

где g_0: $\mathbf{R}^m \to \mathbf{R}^k$, и/или *ограничениями типа неравенств*

$$g_1(x) \leq \Theta, \tag{5}$$

где g_1: $\mathbf{R}^m \to \mathbf{R}^l$.

Другой классификационный признак задач оптимизации — свойства функций f и множеств Ω. Например задачи (2) и (3) называются *линейными* (часто говорят о *задачах линейного программирования*), если функция f — аффинная, а множество Ω — многогранное (множество Ω называется *многогранным*, если оно выделяется ограничениями вида (4) и (5) с аффинными функциями g_0 и g_1).

Замечание. Линейная задача безусловной оптимизации (1) имеет решение (причем обязательно неединственное) в том и только том случае, если $f(x) \equiv \text{const}$.

Если функции f, g_0 и g_1 квадратичные, то говорят о *задачах квадратичного программирования* или о *квадратичных задачах оптимизации* (условных или безусловных). Если эти функции выпуклые, то говорят о *задачах выпуклого программирования* (если множество Ω задается каким-либо другим образом, а не только ограничениями типа (4) и (5), то в задачах выпуклого программирования требуют его выпуклость). Наконец, в общем случае говорят о *задачах нелинейного программирования*. В таких задачах обычно предполагается гладкость фигурирующих в них функций.

1.2. Краткий обзор методов оптимизации

При решении конкретной задачи оптимизации исследователь прежде всего должен выбрать математический метод, который приводил бы к конечным результатам с наименьшими затратами на вычисления или же давал возможность получить наибольший объем информации об искомом решении. Выбор того или иного метода в значительной степени определяется постановкой оптимальной задачи, а также используемой математической моделью объекта оптимизации.

Для решения оптимальных задач применяют в основном следующие методы:

- методы исследования функций классического анализа;
- методы, основанные на использовании неопределенных множителей Лагранжа;
- вариационное исчисление;
- динамическое программирование;
- принцип максимума;
- линейное программирование;
- нелинейное программирование.
- геометрическое программирование.

Как правило, нельзя рекомендовать какой-либо один метод, который можно использовать для решения всех без исключения задач, возникающих на практике. Одни методы в этом отношении являются более общими, другие - менее общими. Наконец, целую группу методов (методы исследования функций классического анализа, метод множителей Лагранжа, методы нелинейного программирования) на определенных этапах решения оптимальной задачи можно применять в сочетании с другими методами, например динамическим программированием или принципом максимума.

Отметим также, что некоторые методы специально разработаны или наилучшим образом подходят для решения оптимальных задач с математическими моделями определенного вида. Так, математический аппарат линейного программирования, специально создан для решения задач с линейными критериями оптимальности и линейными ограничениями на переменные и позволяет решать большинство задач, сформулированных в такой постановке. Так же и геометрическое программирование предназначено для решения оптимальных задач, в которых критерий оптимальности и ограничения представляются специального вида функциями позиномами.

Динамическое программирование хорошо приспособлено для решения задач оптимизации многостадийных процессов, особенно тех, в которых состояние каждой стадии характеризуется относительно небольшим числом переменных состояния. Однако при наличии значительного числа этих переменных, т. е. при высокой размерности каждой стадии, применение метода динамического программирования затруднительно вследствие ограниченных быстродействия и объема памяти вычислительных машин.

Видимо наилучшим путем при выборе метода оптимизации, наиболее пригодного для решения соответствующей задачи, следует признать исследование возможностей и опыта применения различных методов оптимизации. Ниже приводится краткий обзор математических методов решения оптимальных задач и примеры их использования. Здесь же дана лишь краткая характеристика указанных методов и областей их применения, что до некоторой степени может облегчить выбор того или иного метода для решения конкретной оптимальной задачи.

Методы исследования функций классического анализа представляют собой наиболее известные методы решения несложных оптимальных задач, которые известны из курса математического анализа. Обычной областью использования данных методов являются задачи с известным аналитическим выражением критерия оптимальности, что позволяет найти не очень сложное, также аналитическое выражение для производных. Полученные приравниванием нулю производных уравнения, определяющие экстремальные решения оптимальной задачи, крайне редко удается решить аналитическим путем, поэтому, как, правило, применяют вычислительные машины. При этом надо решить систему конечных уравнений, чаще всего нелинейных, для чего приходится использовать численные методы, аналогичные методам нелинейного программирования. Дополнительные трудности при решении оптимальной задачи методами исследования функций классического анализа возникают вследствие того, что система уравнений, получаемая в результате их применения, обеспечивает лишь необходимые условия оптимальности. Поэтому все решения данной системы (а их может быть и несколько) должны быть проверены на достаточность. В результате такой проверки сначала отбрасывают решения, которые не определяют экстремальные значения критерия оптимальности, а затем среди остающихся экстремальных решений выбирают решение, удовлетворяющее условиям оптимальной задачи,

т. е. наибольшему или наименьшему значению критерия оптимальности в зависимости от постановки задачи.

Методы исследования при наличии ограничений на область изменения независимых переменных можно использовать только для отыскания экстремальных значений внутри указанной области. В особенности это относится к задачам с большим числом независимых переменных (практически больше двух), в которых анализ значений критерия оптимальности на границе допустимой области изменения переменных становится весьма сложным.

Метод множителей Лагранжа применяют для решения задач такого же класса сложности, как и при использовании обычных методов исследования функций, но при наличии ограничений типа равенств на независимые переменные. К требованию возможности получения аналитических выражений для производных от критерия оптимальности при этом добавляется аналогичное требование относительно аналитического вида уравнений ограничений. В основном при использовании метода множителей Лагранжа приходится решать те же задачи, что и без ограничений. Некоторое усложнение в данном случае возникает лишь от введения дополнительных неопределенных множителей, вследствие чего порядок системы уравнений, решаемой для нахождения экстремумов критерия оптимальности, соответственно повышается на число ограничений. В остальном, процедура поиска решений и проверки их на оптимальность отвечает процедуре решения задач без ограничений. Множители Лагранжа можно применять для решения задач оптимизации объектов на основе уравнений с частными производными и задач динамической оптимизации. При этом вместо решения системы конечных уравнений для отыскания оптимума необходимо интегрировать систему дифференциальных уравнений.

Следует отметить, что множители Лагранжа используют также в качестве вспомогательного средства и при решении специальными методами задач других классов с ограничениями типа равенств, например, в вариационном исчислении и динамическом программировании. Особенно эффективно применение множителей Лагранжа в методе динамического программирования, где с их помощью иногда удается снизить размерность решаемой задачи.

Методы вариационного исчисления обычно используют для решения задач, в которых критерии оптимальности представляются в виде **функционалов** и **решениями которых служат неизвестные функции.** Такие задачи возникают обычно при статической оптимизации процессов с распределенными параметрами или в задачах динамической оптимизации. Вариационные методы позволяют в этом случае свести решение оптимальной задачи к интегрированию системы дифференциальных уравнений Эйлера, каждое из которых является нелинейным дифференциальным уравнением второго порядка с граничными условиями, заданными на обоих концах интервала интегрирования. Число уравнений указанной системы при этом равно числу неизвестных функций, определяемых при решении оптимальной задачи. Каждую функцию находят в результате интегрирования получаемой системы. Уравнения Эйлера выводятся как необходимые условия экстремума функционала. Поэтому полученные интегрированием системы дифференциальных уравнений функции должны быть проверены на экстремум функционала.

При наличии ограничений типа равенств, имеющих вид функционалов, применяют множители Лагранжа, что дает возможность перейти от условной задачи к безусловной. Наиболее значительные трудности при использовании вариационных методов возникают в случае решения задач с ограничениями типа неравенств.

Заслуживают внимания прямые методы решения задач оптимизации функционалов, обычно позволяющие свести исходную вариационную задачу к задаче нелинейного программирования, решить которую иногда проще, чем краевую задачу для уравнений Эйлера.

Динамическое программирование служит эффективным методом решения задач оптимизации **дискретных многостадийных процессов**, для которых критерий оптимальности задается как **аддитивная функция** критериев оптимальности отдельных стадий. Без особых затруднений указанный метод можно распространить и на случай, когда критерий оптимальности задан в другой форме, однако при этом обычно увеличивается размерность отдельных стадий.

По существу метод динамического программирования представляет собой алгоритм определения оптимальной стратегии управления на всех стадиях процесса. При этом закон управления на каждой стадии находят путем решения частных задач оптимизации последовательно

для всех стадий процесса с помощью методов исследования функций классического анализа или методов нелинейного программирования. Результаты решения обычно не могут быть выражены в аналитической форме, а получаются в виде таблиц. Ограничения на переменные задачи не оказывают влияния на общий алгоритм решения, а учитываются при решении частных задач оптимизации на каждой стадии процесса. При наличии ограничений типа равенств иногда даже удается снизить размерность этих частных задач за счет использования множителей Лагранжа. Применение метода динамического программирования для оптимизации процессов с распределенными параметрами или в задачах динамической оптимизации приводит к решению дифференциальных уравнений в частных производных. **Вместо решения таких уравнений зачастую значительно проще представить непрерывный процесс как дискретный с достаточно большим числом стадий**. Подобный прием оправдан особенно в тех случаях, когда имеются ограничения на переменные задачи и прямое решение дифференциальных уравнений осложняется необходимостью учета указанных ограничений.

При решении задач методом динамического программирования, как правило, используют вычислительные машины, обладающие достаточным объемом памяти для хранения промежуточных результатов решения, которые обычно получаются в табличной форме.

Принцип максимума применяют для решения задач оптимизации процессов, описываемых системами дифференциальных уравнений. Достоинством математического аппарата принципа максимума является то, что решение может определяться в виде **разрывных функций;** это свойственно многим задачам оптимизации, например задачам оптимального управления объектами, описываемыми линейными дифференциальными уравнениями. Нахождение оптимального решения при использовании принципа максимума сводится к задаче интегрирования системы дифференциальных уравнений процесса и сопряженной системы для вспомогательных функций при граничных условиях, заданных на обоих концах интервала интегрирования, т. е. к решению **краевой задачи**. На область изменения переменных могут быть наложены ограничения. Систему дифференциальных уравнений интегрируют, применяя обычные программы на цифровых вычислительных машинах. Принцип максимума для процессов, описываемых дифференциальными уравнениями, при некоторых предположениях является достаточным условием оптимальности. Поэтому

дополнительной проверки на оптимум получаемых решений обычно не требуется.

Для дискретных процессов принцип максимума в той же формулировке, что и для непрерывных, вообще говоря, несправедлив. Однако условия оптимальности, получаемые при его применении для многостадийных процессов, позволяют найти достаточно удобные алгоритмы оптимизации.

Линейное программирование представляет собой математический аппарат, разработанный для решения оптимальных задач с линейными выражениями для критерия оптимальности и линейными ограничениями на область изменения переменных. Такие задачи обычно встречаются при решении вопросов оптимального планирования производства с ограниченным количеством ресурсов, при определении оптимального плана перевозок (транспортные задачи) и т. д.

Для решения большого круга задач линейного программирования имеется практически универсальный алгоритм - ***симплексный метод***, позволяющий за конечное число итераций находить оптимальное решение подавляющего большинства задач. Тип используемых ограничений (равенства или неравенства) не сказывается на возможности применения указанного алгоритма. Дополнительной проверки на оптимальность для получаемых решений не требуется. Как правило, практические задачи линейного программирования отличаются весьма значительным числом независимых переменных. Поэтому для их решения обычно используют вычислительные машины, необходимая мощность которых определяется размерностью решаемой задачи.

Методы нелинейного программирования применяют для решения оптимальных задач с ***нелинейными функциями цели***. На независимые переменные могут быть наложены ограничения также в виде нелинейных соотношений, имеющих вид равенств или неравенств. По существу методы нелинейного программирования используют, если ни один из перечисленных выше методов не позволяет сколько-нибудь продвинуться в решении оптимальной задачи. Поэтому указанные методы иногда называют также ***прямыми методами*** решения оптимальных задач.

Для получения численных результатов важное место отводится нелинейному программированию и в решении оптимальных задач такими методами, как динамическое программирование, принцип максимума и т. п. на определенных этапах их применения.

Названием “методы нелинейного программирования” объединяется большая группа численных методов, многие из которых приспособлены для решения оптимальных задач соответствующего класса. Выбор того или иного метода обусловлен сложностью вычисления критерия оптимальности и сложностью ограничивающих условий, необходимой точностью решения, мощностью имеющейся вычислительной машины и т.д. Ряд методов нелинейного программирования практически постоянно используется в сочетании с другими методами оптимизации, как, например, ***метод сканирования*** в динамическом программировании. Кроме того, эти методы служат основой построения ***систем автоматической оптимизации*** - оптимизаторов, непосредственно применяющихся для управления производственными процессами.

Геометрическое программирование есть метод решения одного специального класса задач нелинейного программирования, в которых критерий оптимальности и ограничения задаются в виде **позиномов - выражений, представляющих собой сумму произведений степенных функций от независимых переменных**. С подобными задачами иногда приходится сталкиваться в проектировании. Кроме того, некоторые задачи нелинейного программирования иногда можно свести к указанному представлению, используя аппроксимационное представление для целевых функций и ограничений.

Специфической особенностью методов решения оптимальных задач (за исключением методов нелинейного программирования) является то, что до некоторого этапа оптимальную задачу решают аналитически, т. е. находят определенные аналитические выражения, например, системы конечных или дифференциальных уравнений, откуда уже отыскивают оптимальное решение. В отличие от указанных методов при использовании методов нелинейного программирования, которые, как уже отмечалось выше, могут быть названы прямыми, применяют информацию, получаемую при вычислении критерия оптимальности, изменение которого служит оценкой эффективности того или иного действия.

Важной характеристикой любой оптимазационной задачи является ее размерность *n,* равная числу переменных, задание значений которых необходимо для однозначного определения состояния оптимизируемого объекта. Как правило, решение задач высокой размерности связано с необходимостью выполнения большого объема вычислений. Ряд методов (например, динамическое программирование и дискретный принцип максимума) специально предназначен для решения задач оптимизации процессов высокой размерности, которые могут быть представлены как многостадийные процессы с относительно невысокой размерностью каждой стадии.

В таблице 1 дана характеристика областей применения различных методов оптимизации, при этом за основу положена сравнительная оценка эффективности использования каждого метода для решения различных типов оптимальных задач.

Классификация задач проведена по следующим признакам:

- вид математического описания процесса;
- тип ограничений на переменные процесса
- число переменных.

Предполагается, что решение оптимальной задачи для процессов, описываемых системами конечных уравнений, определяется как конечный набор значений управляющих воздействий (статическая оптимизация процессов с сосредоточенными параметрами), а для процессов, описываемых системами обыкновенных дифференциальных уравнений, управляющие воздействия характеризуются функциями времени (динамическая оптимизация процессов с сосредоточенными параметрами) или пространственных переменных (статическая оптимизация процессов с распределенными параметрами).

ТАБЛИЦА 1. Области применения методов оптимизации

Вид описания процесса		Конечные уравнения						Дифференциальные уравнения					
Тип ограничений на переменные		Нет		Равенства		Неравенства		Нет		Равенства		Неравенства	
Число переменных *n*		?3	>3	?3	>3	?3	>3	?3	>3	?3	>3	?3	>3
Тип метода	Методы классического анализа	1	2	4	4	4	4	3	4	4	4	4	4
	Множители Лагранжа	-	-	1	2	-	-	-	-	2	3	-	-
	Вариационное исчисление	-	-	-	-	-	-	2	3	2; 7	3; 7	-	-
	Динамическое программирование	1; 5	3; 5	1;5;7	3; 5; 7	1; 5	3; 5	2	3	3	3	3	3
	Принцип максимума	2; 5	1; 5	2; 5	2; 5	2; 5	2; 5	1	1	2	2	2	2
	Линейное программирование	-	-	-	2; 6	2; 6	1; 6	-	-	-	-	-	-
	Методы нелинейного программирования	2	1	2	.1	2	1	4	4	4	4	4	4
	Геометрическое программирование	2; 8	2; 8	-	-	2; 8	2; 8	-	-	-	-	-	-

Примечания:

1. Эффективное применение метода.
2. Используется.
3. Возможно применение.
4. Используется как вспомогательный метод.
5. Многостадийные процессы (размерность указывается для отдельной стадии).
6. Задачи с линейными критериями оптимальности и линейными ограничениями.
7. Используются множители Лагранжа.
8. Задачи с критериями и ограничениями в форме позиномов.

Классификация задач по группам с числом независимых переменных, большим и меньшим трех или равным трем как характеристика размерности задач с большим и малым числом переменных, разумеется, весьма условна и в данном случае выбрана скорее из соображений наглядности графического изображения пространства изменения переменных задачи - фазового пространства (при числе переменных большем трех графическое изображение фазового пространства обычными приемами отсутствует). Тем не менее, такая классификация до некоторой степени все же отражает действительные трудности, возникающие при решении задач с размерностью выше трех.

1.3. Задача безусловной оптимизации

Здесь мы введем основные понятия и проведем теоретическое исследование задачи безусловной оптимизации. Отметим, что эта задача в теоретическом плане достаточно полно изложена в первой книге насточщей работы. Мы лишь повторим важнейшие факты, обращая внимание на "оптимизационную" специфику.

Определения.

Мы будем рассматривать *задачу безусловной оптимизации*

$$f(x) \to \min, \tag{1}$$

где f: $\mathbf{R}^m \to \mathbf{R}$. Точка $x^* \in \mathbf{R}^m$ называется *решением задачи* (1) (или *точкой глобального безусловного минимума функции f*), если

$$f(x^*) \le f(x) \tag{2}$$

при всех $x \in \mathbf{R}^m$. Если неравенство (2) выполнено лишь для x, лежащих в некоторой окрестности V_{x^*} точки x^*, то точка x^* называется *локальным решением задачи* (1), или *точкой локального безусловного минимума функции f*. Если неравенство (2) строгое при всех $x \ne x^*$, то говорят о *строгом глобальном* и, соответственно, *строгом локальном минимумах*. Решение задачи (1) иногда обозначают $\operatorname{argmin} f(x)$ (или, более полно, $\operatorname{argmin}_{x \in \mathbf{R}m} f(x)$; когда речь идет о задачах безусловной оптимизации в обозначениях $\operatorname{argmin}_{x \in \mathbf{R}m} f(x)$ и $\min_{x \in \mathbf{R}m} f(x)$ мы будем всегда опускать индекс "$x \in \mathbf{R}^m$"). Обычно из контекста ясно о каком минимуме (локальном, глобальном и т. д.) идет речь.

Аналогичные понятия (максимумов) определяются для задачи

$$f(x) \to \max.$$

Замечание. Точка x^* является точкой глобального безусловного (соответственно, локального, строгого) максимума функции f в том и только том случае, когда она является точкой глобального безусловного (соответственно, локального, строгого) минимума

функции $-f$. Поэтому всюду в дальнейшем мы будем заниматься только задачами о минимумах, все время помня, что задачи о максимумах к ним сводятся. Таким образом, слово "оптимизация" в нашем контексте будет всегда синонимом слова "минимизация".

О линейных операторах в $\mathbf{R}^m$.

Напомним, что линейный оператор A в $\mathbf{R}^m$ называется *самосопряженным* или *симметричным*, если при всех $x, y \in \mathbf{R}^m$

$$(Ax, y) = (x, Ay).$$

Известно, что оператор A симметричен в том и только том случае, когда его матрица симметрична (т. е. переходит в себя при транспонировании).

Оператор A называется *невырожденным*, если у него нулевое *ядро* $\ker A$, т. е. если он переводит в нуль только нуль. Другими словами, уравнение $Ax = \Theta$ имеет только нулевое решение. Из курса алгебры известно, что *оператор A невырожден в том и только том случае, если определитель его матрицы отличен от нуля.*

Оператор A называется *положительно определенным* (часто пишут $A > 0$), если

$$(Ax, x) > 0$$

при всех ненулевых $x \in \mathbf{R}^m$. В соответствии с **критерием Сильвестра** *оператор A положительно определен в том и только том случае, если все главные диагональные миноры матрицы оператора A положительны.* Наконец, оператор A называется *неотрицательно определенным* (пишут $A \geq 0$), если при всех $x \in \mathbf{R}^m$

$$(Ax, x) \geq 0.$$

Аналогично определяются понятия *отрицательно* и *неположительно определенных операторов*.

Если оператор $A - \lambda I$, где I — тождественный оператор на $\mathbf{R}^m$, а $\lambda \in \mathbf{R}$, положительно (неотрицательно) определен, то часто пишут $A > \lambda$ (соответственно, $A \geq \lambda$). Аналогично определяются записи $A < \lambda$ и $A \leq \lambda$.

Из курса алгебры известно, что *симметричный оператор A удовлетворяет неравенствам*

$$\lambda \leq A \leq \Lambda,$$

в том и только том случае, если все точки спектра $\sigma(A)$ *оператора A лежат на отрезке* $[\lambda, \Lambda]$:

$$\lambda \leq \lambda_i \leq \Lambda. \tag{3}$$

В частности, поскольку норму в $\mathbf{R}^m$ мы считаем евклидовой, для симметричных операторов A имеют место утверждения

$$\|A\| = \max_{\lambda_i \in \sigma(A)} \{|\lambda_i|\} \leq \max\{|\lambda|, |\Lambda|\}. \tag{4}$$

О дифференцируемости функций на $\mathbf{R}^m$.

Напомним ряд понятий и фактов из курса математического анализа, которые потребуются нам в дальнейшем.

Вектор $a \in \mathbf{R}^m$ такой, что

$$f(x + h) - f(x) - (a, h) = o(h)$$

при всех $h \in \mathbf{R}^m$ называется *производной* или *градиентом функции f в точке x*. Здесь и ниже символ $o(h)$ обозначает произвольную функцию, обладающую свойством

$$\frac{o(h)}{\|h\|} \to 0 \text{ при } h \to \Theta.$$

Функция f называется при этом *дифференцируемой в точке x*. Градиент обычно обозначается $f'(x)$, или $\operatorname{grad} f(x)$, или $\nabla f(x)$. Известно, что в координатной форме градиент имеет вид

$$f'(x) = \left(\frac{\partial f(x)}{\partial x_1}, ..., \frac{\partial f(x)}{\partial x_m} \right) = \left(\frac{\partial f(x_1, ..., x_m)}{\partial x_1}, ..., \frac{\partial f(x_1, ..., x_m)}{\partial x_m} \right).$$

Функция f: $\mathbf{R}^m \to \mathbf{R}^m$ дифференцируемая в каждой точке называется *дифференцируемой*.

Если дополнительно найдется линейный *самосопряженный* оператор A: $\mathbf{R}^m \to \mathbf{R}^m$ такой, что при всех $h \in \mathbf{R}^m$

$$f(x + h) - f(x) - (f'(x), h) - \frac{1}{2}(Ah, h) = o(h^2),$$

где запись $o(h^2)$ означает, что

$$\frac{o(h^2)}{\|h\|^2} \to \text{ при } h \to \Theta,$$

то f называется *дважды дифференцируемой в точке x*, а оператор A называется *второй производной функции f в точке x* и обозначается $f''(x)$ либо $\nabla^2 f(x)$. Матрицей, отвечающей оператору $A = f''(x)$, служит, как нетрудно видеть, так называемая *матрица Гессе* или *гессиан функции f*:

$$A = \begin{bmatrix} \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_m} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f(x)}{\partial x_m \partial x_1} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_m \partial x_m} \end{bmatrix}.$$

Замечание. Если A — линейный самосопряженный оператор в $\mathbf{R}^m$,и $b \in \mathbf{R}^m$, $c \in \mathbf{R}$ и $f(x) = (Ax, x)/2 + (b, x) + c$, то можно доказать, что $f'(x) = Ax + b$, и $f''(x) = A$.

Если функция F: $\mathbf{R}^m \to \mathbf{R}^k$, то линейный оператор A: $\mathbf{R}^m \to \mathbf{R}^k$ такой, что

$$F(x + h) - F(x) - Ah = o(h)$$

называется *производной функции F в точке x* и обозначается $F'(x)$ (это обобщение понятия градиента на случай функций со значениями в $\mathbf{R}^k$).

Если функция F: $\mathbf{R}^m \to \mathbf{R}$ дифференцируема, то ее градиент можно рассматривать как функцию из $\mathbf{R}^m$ в $\mathbf{R}^m$: каждому $x \in \mathbf{R}^m$ ставится в соответствие точка из $f'(x) \in \mathbf{R}^m$.

Замечание А. Можно доказать, что $[f'(x)]' = f''(x)$. Поясним: здесь $[f'(x)]'$ — производная функции $x \to f'(x)$, действующей из $\mathbf{R}^m$ в $\mathbf{R}^m$, а $f''(x)$ — вторая производная функции f: $\mathbf{R}^m \to \mathbf{R}^m$.

Приведем еще одно понятие. Функция F: $\mathbf{R}^m \to \mathbf{R}^k$ по определению *удовлетворяет условию Липшица с константой* Λ, если при всех $x, y \in \mathbf{R}^m$

$||F(x) - F(y)|| \le \Lambda\, ||x - y||$.

Замечание Б. Пусть F: $\mathbf{R}^m \to \mathbf{R}^k$ дифференцируема. тогда существует доказательство, что F удовлетворяет условию Липшица с константой Λ, в том и только том случае, если $||F'(x)|| \le \Lambda$ при всех x.

Ниже нам потребуется следующее утверждение. *Если* f: $\mathbf{R}^m \to \mathbf{R}$ — *дважды непрерывно дифференцируемая функция, то для того, чтобы ее градиент* f' *удовлетворял условию Липшица с константой* Λ *необходимо и достаточно, чтобы при всех* $x \in \mathbf{R}^m$ *выполнялось неравенство* $f'' \le \Lambda$. Действительно, в силу замечания А, при всех $t \in \mathbf{R}$ и $x, h \in \mathbf{R}^m$

$$(f'(x + th), th) - (f'(x), th) = (f''(x)th, th) + (o(th), th).$$

Но тогда в силу условия Липшица для f'

$$(f''(x)h, h) \le \frac{1}{t^2} |(f'(x + th) - f'(x), th)| + \frac{|o(th, th)|}{t^2} \le$$

$$\le \frac{\Lambda ||th||^2}{t^2} + \frac{||o(th)|| \cdot ||th||}{t^2} = \Lambda\, ||h||^2 + \frac{||o(th)||}{t}\, ||h||.$$

Устремляя t к 0, получим неравенство

$$(f''(x)h, h) \le \Lambda\, ||h||^2, \qquad (5)$$

эквивалентное нужному неравенству $f''(x) \le \Lambda$.

В заключение пункта еще одно обозначение. Мы будем писать $f \in C$, $f \in C^1$ и $f \in C^2$, если f соответственно непрерывна, непрерывно дифференцируема и дважды непрерывно дифференцируема.

Необходимое условие локального экстремума.

Такое условие дает хорошо известная из курса математического анализа теорема.

Теорема Ферма. *Если f — дифференцируемая функция и x^* — ее локальный минимум, то $f'(x^*) = 0$.*

Напомним доказательство теоремы. Допустим противное: $f'(x^*) \neq \Theta$. Положим $x_t = x^* - tf'(x^*)$ для всех $t > 0$. Тогда, во-первых, очевидно, $x_t - x^* \to \Theta$ при $t \to 0$ и, во-вторых, по определению градиента,

$$f(x_t)=f(x^*)+(f'(x^*),x_t-x^*)+o(x_t-x^*)=$$

$$= f(x^*) + (f'(x^*), -tf'(x^*)) + o(-tf'(x^*)) = \tag{6}$$

$$= f(x^*) - [\|f'(x^*)\|^2 + \frac{o(-tf'(x^*))}{t}].$$

Поскольку $\|f'(x^*)\| > 0$, а

$$\frac{o(-tf'(x^*))}{t} = \|f'(x^*)\| \cdot \frac{o(-tf'(x^*))}{\|(-tf'(x^*)\|} \to 0 \text{ при } t \to 0,$$

выражение в квадратных скобках в правой части (6) при всех достаточно малых t положительно и поэтому при всех достаточно малых положительных t

$$f(x_t) < f(x^*),$$

что противоречит тому, что $x^* = \operatorname{argmin} f(x)$.

Из доказательства следует, что, *двигаясь из заданной точки в направлении, противоположном градиенту (говорят* в направлении антиградиента*), мы локально уменьшаем значение функции.* Это замечание потребуется нам в дальнейшем.

Таким образом, минимум функции может достигаться *только* в тех точках, в которых ее производная обращается в нуль, и поэтому уравнение

$$f'(x) = 0, \tag{7}$$

или, что то же самое, система m (вообще говоря, нелинейных) уравнений с m неизвестными

$$\frac{\partial f(x_1, ..., x_m)}{\partial x_i} = 0, \quad i = 1, ..., m,$$

определяет точки "подозрительные на минимум". Точки, удовлетворяющие уравнению (7), называются *стационарными точками функции f.*

Стационарная точка x^* функции f может быть либо точкой локального минимума, либо точкой локального максимума, либо не быть ни той, ни другой (см. рис. 1).

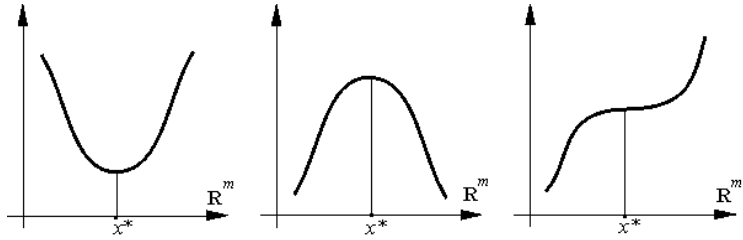


Рис. 1.

Точка (x^*, y^*) называется *седловой точкой функции f*: $\Omega_1 \times \Omega_2 \to \mathbf{R}$ ($\Omega_1 \in \mathbf{R}^n$, $\Omega_2 \in \mathbf{R}^m$), если при всех $(x, y) \in \Omega_1 \times \Omega_2$ выполнены неравенства

$f(x^*, y) \le f(x^*, y^*) \le f(x, y^*)$

(см. рис. 2). Если эти неравенства выполняется лишь для x достаточно близких к x^* и y достаточно близких к y^*, то, естественно, добавляется эпитет *локальная*.

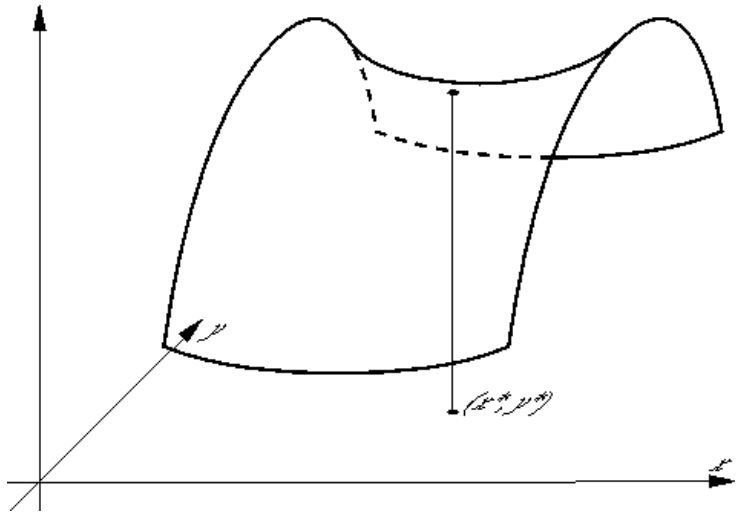


Рис. 2.

Легко доказать, что седловая точка непрерывно дифференцируемой функции всегда является стационарной точкой и, очевидно, никогда не является точкой экстремума.

Теорема о локальном минимуме (необходимые и достаточные условия второго порядка).

Пусть x^* — стационарная точка дважды дифференцируемой функции f. Для того, чтобы точка x^* была точкой (локального) минимума функции f необходимо, чтобы оператор $f''(x^*)$ был неотрицательно определен и достаточно, чтобы он был положительно определен.

Доказательство. ***Необходимость***. Пусть x^* — точка минимума и h — произвольный вектор из $\mathbf{R}^m$. Поскольку (в силу теоремы Ферма) x^* — стационарная точка,

$$0 < f(x^* + th) - f(x^*) = \frac{1}{2}(f''(x^*)th, th) + o((th)^2)$$

при всех достаточно малых $t \in \mathbf{R}$. Отсюда при всех $t \neq 0$

$$(f''(x^*)h, h) + \frac{o((th)^2)}{t^2} > 0.$$

Переходя в полученном неравенстве к пределу при $t \to 0$ и учитывая, что как легко видеть, $o((th)^2)/t^2 \to 0$ при $t \to 0$, получим нужное неравенство

$$(f''(x^*)h, h) \geq 0.$$

Достаточность. Пусть $f''(x^*)$ положительно определен, а стационарная точка x^* не является точкой локального минимума. Последнее означает наличие последовательности $x^n \to x^*$ при $n \to \infty$ такой, что $f(x^n) < f(x^*)$. Положим $h^n = x^n - x^*$. По определению второй производной, учитывая, что x^* стационарна,

$$0 > f(x^* + th^n) - f(x^*) = \frac{1}{2}(f''(x^*)h^n, h^n) + o((h^n)^2).$$

Если теперь обозначить $h^n/\|h^n\|$ через g^n, то последнее неравенство (поделив его на $\|h^n\|^2$) можно переписать в виде

$$(f''(x^*)g^n, g^n) + \frac{o((h^n)^2)}{\|h^n\|^2} < 0. \qquad (8)$$

Поскольку $\|g^n\| = 1$, а сфера в $\mathbf{R}^m$ компактна, последовательность $\{g^n\}$, не ограничивая общности, можно считать сходящейся к некоторому лежащему на ней (и следовательно, отличному от нуля) вектору g^0. Предельный при $n \to \infty$ переход в неравенстве (8) приводит к противоречащему положительной определенности оператора $f''(x^*)$ неравенству

$$(f''(x^*)g^0, g^0) \le 0.$$

Теорема доказана.

З а д а ч а . Исследуйте на экстремум функцию f: $\mathbf{R}^2 \to \mathbf{R}$, задаваемую формулой $f(x_1, x_2) = x_1^2/a + x_2^2/b$, при различных a и b.

Замечания о существовании решений.

Из курса математического анализа известно, что *задача о существовании минимума непрерывной функции на компактном множестве всегда имеет по крайней мере одно решение* (**теорема Вейерштрасса**). В нашем случае — случае некомпактной области определения — нужны дополнительные условия.

В следующей теореме приводится одно из таких возможных дополнительных условий.

Теорема о разрешимости задачи безусловной оптимизации. *Пусть функция f непрерывна и при некотором $\alpha \in \mathbf{R}^m$ множество $S_\alpha = \{x \in \mathbf{R}^m: f(x) \le \alpha\}$ непусто и ограничено. Тогда задача* (1) *имеет по крайней мере одно решение.*

Д о к а з а т е л ь с т в о. Множество S_α замкнуто.

Поэтому S_α — компактное подмножество $\mathbf{R}^m$. В силу теоремы Вейерштрасса, очевидно, функция f достигает на S_α своего минимума: $x^* = \operatorname{argmin}_{x \in S\alpha} f(x)$. Очевидно, x^* — решение задачи (1), поскольку $f(x^*) \le \alpha$ в S_α, а вне S_α функция f принимает значения бóльшие α.

Замечания о единственности решений.

Вопрос о единственности (как, впрочем, и о существовании) решений весьма важен в теоретическом плане. Например, если x^* — единственное решение задачи (1) и $\{x^k\} \in \mathbf{R}^m$ — ограниченная последовательность такая, что $f(x^k) \to f(x^*) = \min f(x)$ при $k \to \infty$, то $x^k \to x^* = \operatorname{argmin} f(x)$ при $k \to \infty$. Такое свойство бывает полезным при исследовании приближенных методов решения оптимизационных задач.

Точка x^* локального минимума дважды дифференцируемой функции f называется *невырожденной*, если оператор $f''(x^*)$ невырожден. Она называется *локально единственной*, если в некоторой ее окрестности V_{x^*} нет других точек локального минимума функции f.

Теорема о локальной единственности решений. *Невырожденная точка локального минимума локально единственна.*

Д о к а з а т е л ь с т в о. Допустим противное: x^* не является локально единственной точкой минимума, т. е. найдется сходящаяся к x^* последовательность $\{x^n\}$ локальных минимумов функции f. Тогда

$$f'(x^n) - f'(x^*) = f''(x^*)(x^n - x^*) + o(x^n - x^*).$$

Поскольку x^n и x^* — локальные минимумы и, следовательно, стационарные точки, $f'(x^n) = f'(x^*) = \Theta$. Далее, положим (как мы уже делали) $g^n = (x^n - x^*)/\|x^n - x^*\|$. Тогда, очевидно,

$$f''(x^*)g^n = \frac{o(x^n - x^*)}{\|x^n - x^*\|}. \qquad (9)$$

Далее рассуждения стандартны: $\{g^n\}$ лежит на единичной сфере в $\mathbf{R}^m$, поэтому ее можно считать сходящейся к некоторому пределу $g^0 \neq \Theta$. Переходя к пределу в (9), получаем

$$f''(x^*)g^0 = \Theta,$$

что противоречит невырожденности оператора $f''(x^*)$.

Выпуклые функции на $\mathbf{R}^m$.

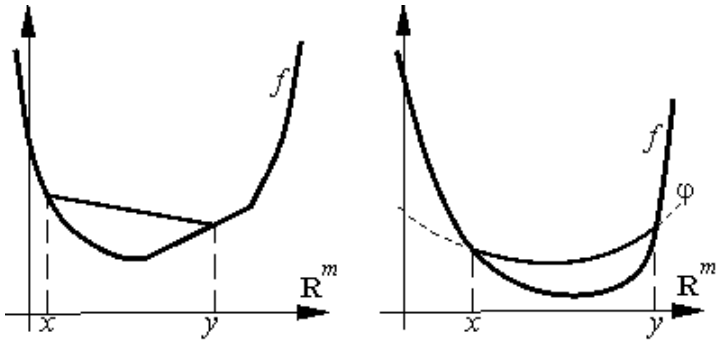
Особенно легко вопросы существования и единственности решаются для выпуклых функций. Эти функции являются очень важным объектом в теории оптимизационных задач. Начнем с определений.

Функция f: $\mathbf{R}^m \to \mathbf{R}$ называется *выпуклой*, если при всех $x, y \in \mathbf{R}^m$ и $\lambda \in (0, 1)$

$$f(\lambda x + (1 - \lambda)y) \le \lambda f(x) + (1 - \lambda)f(y). \qquad (10)$$

Если неравенство (10) строгое, то f называется *строго выпуклой*. Геометрически выпуклость означает, что график функции на интервале (x, y), соединяющем любые точки x и y, лежит не выше прямой, соединяющей точки $(x, f(x))$ и $(y, f(y))$ (см. рис. 3,*а*). Функция f *сильно выпукла* (*с константой* $c > 0$), если неравенство (10) выполняется в следующей более сильной форме

$$f(\lambda x + (1 - \lambda)y) \le \lambda f(x) + (1 - \lambda)f(y) + \frac{c}{2}\lambda(1 - \lambda)\|x - y\|^2. \qquad (11)$$



а) *б*)

Рис. 3.

Геометрически это понятие можно интерпретировать так. Пусть точки отрезка $[x, y]$, соединяющего точки x и y, параметризованы

параметром λ: $\lambda \to \lambda x + (1 - \lambda)y$. Правая часть неравенства (11) определяет на этом отрезке полином φ второго порядка (от λ). График сильно выпуклой функции над отрезком $[x, y]$ должен лежать ниже параболы — графика этого полинома (см. рис. 3,*б*).

Критерий выпуклости дифференцируемой функции. *Для того, чтобы дифференцируемая функция f была выпуклой необходимо и достаточно выполнения при всех $x,y \in \mathbf{R}^m$ неравенства*

$$f(x) - f(y) \geq (f'(y), x - y). \tag{12}$$

Действительно, определим на отрезке $[0, 1]$ функцию φ, положив

$$\varphi(\lambda) = f(\lambda x + (1 - \lambda)y).$$

Очевидно функция φ выпукла одновременно с функцией f. Кроме того, легко показать, что

$$\varphi'(\lambda) = (f'(\lambda x + (1 - \lambda)y), x - y).$$

Неравенство (12) в новых обозначениях переписывается в виде

$$\varphi(1) - \varphi(0) \geq \varphi'(0),$$

или, если воспользоваться формулой Лагранжа,

$$\varphi'(\tau) \geq \varphi'(0), \tag{13}$$

где τ — некоторая точка интервала $(0, 1)$. Из курса математического анализа известно, что для дифференцируемых функций выпуклость эквивалентна монотонности производной. Поэтому, если f выпукла, то φ' монотонна. Следовательно, имеет место эквивалентное (12) неравенство (13).

Геометрически доказанное утверждение означает, что значения функции $f(x)$ "лежит выше" гиперплоскости

$$H_y = \{(x, \xi) \in \mathbf{R}^m \times \mathbf{R}: \xi = f(y) + (f'(y), x - y)\},$$

касательной в точке $(y, f(y))$ к *графику* $\mathrm{Gr}\, f = \{(x, \xi) \in \mathbf{R}^m \times \mathbf{R}: \xi = f(x)\}$ при всех $y \in \mathbf{R}^m$ (см. рис. 4).

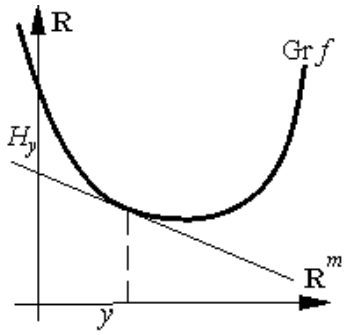


Рис. 4.

Строгая выпуклость дифференцируемой функции, как легко видеть, *эквивалентна строгому при $x \neq y$ неравенству* (12). *Сильная же выпуклость функции f эквивалентна выполнению при всех x и y неравенства*

$$f(x) - f(y) \geq (f'(y), x - y) + c\|x - y\|^2. \qquad (14)$$

Замечание. Функция $f \in C^2$ сильно выпукла с константой c в том и только том случае, если $f''(x) \geq c$ при всех $x \in \mathbf{R}^m$.

Теорема о разрешимости для сильно выпуклой функции.

Задача (1) *с дифференцируемой сильно выпуклой функцией разрешима.*

Доказательство. Неравенство (14) для $y = \Theta$ и произвольного x имеет вид

$$f(x) \geq f(\Theta) + (f'(\Theta), x) + c\|x\|^2. \qquad (15)$$

Для $\alpha = f(\Theta)$ множество $S_\alpha = \{x \in \mathbf{R}^m: f(x) \leq \alpha\}$, во-первых, непусто, поскольку содержит точку Θ, а во-вторых, ограничено, поскольку вне шара $B(\Theta, \|f'(\Theta)\|/c)$

$f(x) > \alpha$.

Действительно, продолжая (15), при $\|x\| > \|f'(\Theta)\|/c$ имеем

$$f(x) \geq f(\Theta) + (f'(\Theta), x) + c\|x\|^2 \geq \alpha - |(f'(\Theta), x)| + c\|x\|^2 \geq$$

$$\geq \alpha + c\|x\|^2 - \|f'(\Theta)\| \cdot \|x\| = \alpha + \|x\|(c\|x\| - \|f'(\Theta)\|) > \alpha.$$

Заключение теоремы теперь следует из теоремы о разрешимости задачи безусловной оптимизации.

Замечание. Для выпуклой (и даже для строго выпуклой) функции утверждение теоремы в общем случае не верно.

Теорема единственности для строго выпуклой функции.

Задача (1) *со строго выпуклой функцией не может иметь более одного решения.*

Д о к а з а т е л ь с т в о. В предположении существования двух точек минимума x^* и x^{**} (очевидно тогда, что $f(x^*) = f(x^{**})$), в силу строгой выпуклости, получим противоречащее равенству $x^* = \operatorname{argmin} f(x)$ неравенство

$$f\left(\frac{x^* + x^{**}}{2}\right) < \frac{f(x^*)}{2} + \frac{f(x^{**})}{2} = f(x^*).$$

2. Методы одномерной оптимизации

2.1. Введение в одномерную оптимизацию

2.1.1. Основные определения

Явно или неявно мы встречаемся с оптимизацией в любой сфере человеческой деятельности от сугубо личного до самого высокого общегосударственного уровня. Экономическое планирование, управление, распределение ограниченных ресурсов, анализ производственных процессов, проектирование сложных объектов всегда должно быть направлено на поиск наилучшего варианта с точки зрения намеченной цели. Это - важнейшее условие научно-технического прогресса.

При небывалом разнообразии задач оптимизации только математика может дать общие методы их решения. Однако для того, чтобы воспользоваться математическим аппаратом, необходимо сначала сформулировать интересующую нас проблему как математическую задачу, придав количественные оценки возможным вариантам, количественный смысл словам "лучше", "хуже".

Многие задачи оптимизации сводятся к отысканию наименьшего (или наибольшего) значения некоторой функции, которую, как мы уже говорили, принято называть *целевой функцией* или *критерием качества.*

Постановка задачи и методы исследования существенно зависят от свойств целевой функции и той информации о ней, которая может считаться доступной в процессе решения, а также которая известна априори (до опыта, заранее; здесь - до начала решения задачи).

Наиболее просты, с математической точки зрения, случаи, когда целевая функция задается явной формулой и является при этом дифференцируемой функцией. В этом случае для исследования свойств функции, определения направлений ее, возрастания и убывания, поиска точек локального экстремума может быть использована производная.

В условиях научно-технического прогресса круг задач оптимизации, поставленных практикой, резко расширился. Во многих из них целевая функция не задается формулой, ее значения могут получаться в результате сложных расчетов, браться из эксперимента и

т. д. Такие задачи являются более сложными, потому что для них нельзя провести исследование целевой функции с помощью производной. Пришлось уточнять их математическую постановку и разрабатывать специальные методы решения, рассчитанные на широкое применение ЭВМ. Следует также иметь в виду, что сложность задачи существенно зависит от ее размерности, т. е. от ***числа аргументов целевой функции***.

Начальные разделы данной работы посвящены одномерным задачам безусловной оптимизации, в последующих рассматриваются многомерные задачи. Выделение и подробный разбор одномерных задач имеет определенный смысл. Эти задачи наиболее просты, на них легче понять постановку вопроса, методы решения и возникающие трудности. В ряде случаев, хотя и очень редко, одномерные задачи имеют самостоятельный практический интерес. Однако самое главное заключается в том, что **алгоритмы решения многомерных задач оптимизации часто сводятся к последовательному многократному решению одномерных задач** и не могут быть поняты без умения решать такие задачи.

Для одномерных методов

Определения. Пусть задано множество $X \subset R^n$ и функция $f(x)=f(x_1,x_2,...,x_n)$, определенная на множестве X. Точка $x^* \in X$ называется *точкой локального минимума* функции $f(x)$ на множестве X, если существует шар $U_\varepsilon(x^*)=\{x: ||x-x^*||<=\varepsilon\}$ такой, что для любого $x \in U_\varepsilon(x^*)$ выполняется неравенство

$$f(x^*) \leq f(x). \qquad (1)$$

Если неравенство (1) выполняется как строгое (при $x \neq x^*$), то говорят, что x^* - точка *строгого локального минимума*.

Точка $x^* \in X$ называется *точкой глобального минимума* функции $f(x)$ на множестве X, если неравенство (1) выполняется для любого x из множества X.

Аналогично определяются точки локального и глобального максимума функции $f(x)$ на множестве X.

Точки локального максимума и минимума функции $f(x)$ называют *точками экстремума* этой функции.

Задача отыскания всех локальных минимумов (максимумов) функции $f(x)$, если множество X совпадает со всем n-мерным пространством, т.е. $X=R^n$, называется *задачей безусловной оптимизации*, а функция $f(x)$ - *целевой функцией*.

Задачу отыскания точек локального минимума целевой функции $f(x)$ символически записывают так:

$$f(x) \to \min,\ x \in R^n. \qquad (2)$$

Аналогично задачу отыскания точек локального максимума функции $f(x)$ символически записывают следующим образом:

$$f(x) \to \max,\ x \in R^n. \qquad (3)$$

Задача (3) эквивалентна задаче

$$-f(x) \to \min,\ x \in R^n$$

в том смысле, что множества локальных и глобальных решений этих задач соответственно совпадают.

Для многомерных методов

Определения. Пусть требуется решить задачу (2):

$$f(x) \to \min,\ x \in R^n. \qquad (4)$$

В двумерном пространтсве R^2 решению такой задачи можно дать геометрическую иллюстрацию. Пусть точка $x = (x_1,x_2)$ лежит на плоскости Ox_1x_2. Введем третью координату x_3 так, чтобы ось координат Ox_3 была перпендикулярна плоскости Ox_1x_2 (рис.1). Уравнению $x_3 = f(x_1,x_2)$ соответствует поверхность в трехмерном пространстве.

Если функция $f(x)$ достигает локального минимума в точке $x^* \in R^2$, то поверхность в некоторой окрестности точки x^* имеет форму чаши (рис.1).

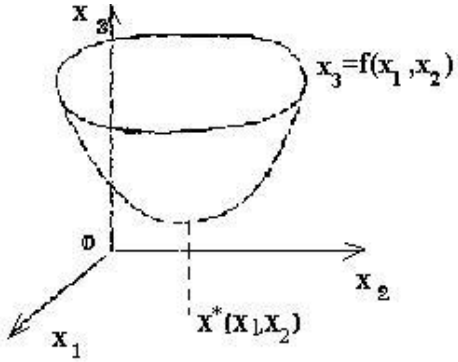


Рис.1

Напомним, что линиями уровня функции $f(x_1,x_2)$ называют семейство линий плоскости R^2, на которых функция принимает постоянное значение. Неявным уравнением линии уровня является

уравнение $f(x_1,x_2)=C$. Если функция $f(x)$ имеет в R^2 единственную точку локального минимума x^* (x^*_1, x^*_2), то такая функция называется *мономодальной*. Взаимное расположение ее линий уровня имеет вид, изображенный на рис. 2.

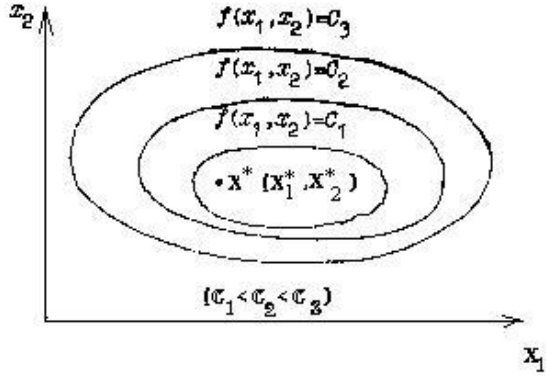


Рис. 2

Мультимодальными называются функции, которые имеют более одного экстремума. Такова, например, функция Химмельблау

$$F(x) = (x_1^2+x_2-11)^2+(x_1+x_2^2-7)^2,$$

имеющая четыре изолированные точки минимума.

Чтобы найти точку x^* локального минимума функции $f(x)$, составляют последовательность точек (приближений к решению) $\{x^{(k)}\}$ (k=0,1, ...) , сходящуюся к точке x^* (k=0,1,...).

Последовательность значений функции $f(x^{(k)})$ должна быть монотонно убывающей и ограниченной снизу:

$$f(x^{(0)})\geq f(x^{(1)}) \geq \ldots \geq f(x^{(k)}) \geq \ldots \geq f(x^{(*)}).$$

Геометрический образ решения задачи (2) для случая двух переменных напоминает спуск на дно чаши. Это мотивирует названия методов решения задачи (2) - «методы спуска». Для различных методов спуска сначала выбирают начальную точку последовательности $x^{(0)}$. Дальнейшие приближения $x(k)$ определяются соотношениями

$$x^{(k+1)}=x^{(k)}+t^{(k)}S^{(k)} \qquad (k= 0, 1, 2, \ldots), \qquad (5)$$

где $S^{(k)}$ - вектор направления спуска; скалярная величина $t^{(k)}$ является решением задачи одномерной минимизации

$$f(x^{(k)}+tS^{(k)} \rightarrow \min, \quad t \in R. \qquad (6)$$

Таким образом, задача поиска минимума функции нескольких переменных сводится к последовательности задач одномерной минимизации (6) по переменной t на отрезках n-мерного пространства, проходящих через точки $x^{(k)}$ в направлении векторов $S^{(k)}$.

Методы спуска различаются выбором вектора спуска и способом решения задачи одномерной минимизации. При решении последовательности задач (5) можно ограничиться методом сканирования для поиска минимума функции одной переменной. Выбрав произвольно начальную точку $x^{(0)}$ и размер начальюго шага h по переменной t, в методе сканирования можно получить различные точки минимума мультимодальной функции. Если функция $f(x)$ мономодальна, то независимо от выбора начальной точки траектория поиска должна привести к единственной точке локального минимума этой функции.

Пример. Задача о наилучшей консервной банке

Перед вами поставили задачу: указать наилучший вариант консервной банки фиксированного объема V, имеющей обычную форму прямого кругового цилиндра. Получив такое задание, вы неизбежно должны спросить: "По какому признаку следует сравнивать банки между собой, какая банка считается наилучшей?" Иными словами, вы попросите указать цель оптимизации. Рассмотрим два варианта этой задачи.

1. Наилучшая банка должна иметь наименьшую поверхность S. (На ее изготовление пойдет наименьшее количество жести.)

2. Наилучшая банка должна иметь наименьшую длину швов l. (Швы нужно сваривать, и мы хотим сделать эту работу минимальной.)

Для решения этой задачи запишем формулы для объема банки, площади ее поверхности и длины швов:

$$V=\pi r^2 h, \; S=2\pi r^2+2\pi rh, \; l=4\pi r+h. \qquad (7)$$

Объем банки задан, это устанавливает связь между радиусом r и высотой h. Выразим высоту через радиус: $h=V/(\pi r^2)$ и подставим полученное выражение в формулы для поверхности и длины швов. В результате получим

$$S(r)=2\pi r^2+2V/r, \qquad 0 < r < \infty, \qquad (8)$$

$$l(r)=4\pi r+V/(\pi r^2), \qquad 0 < r < \infty. \qquad (9)$$

Таким образом, с математической точки зрения, задача о наилучшей консервной банке сводится к определению такого значения r, при котором достигает своего наименьшего значения в одном случае функция $S(r)$, в другом - функция $l(r)$.

Рассмотрим первый вариант задачи. Вычислим производную функции $S(r)$:

$$S'(r)=4\pi r-2V/r^2=2/r^2(2\pi r^3-V) \qquad (10)$$

и исследуем ее знак. При $0<r<r_1=\sqrt[3]{V/(2\pi)}$ производная отрицательна и функция $S(r)$ убывает, при $r_1<r<\infty$ производная положительна и функция $S(r)$ возрастает. Следовательно, своего наименьшего значения эта функция достигает в точке $r=r_1$, в которой ее производная обращается в нуль. График функции $S(r)$, иллюстрирующий проведенный анализ, показан на рис.3.

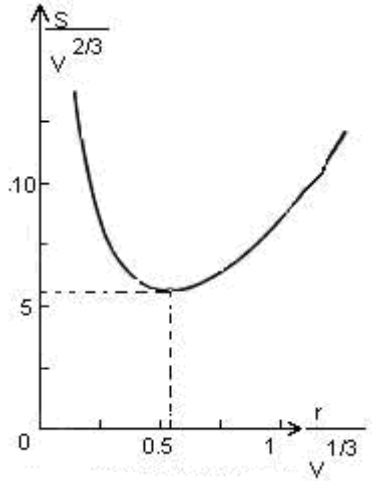


Рис. 3. График функции $S(r)$

Итак, радиус и высота банки, наилучшие с точки зрения условия минимальности $S(r)$, определяются формулами

$$r_1=\sqrt[3]{V/(2\pi)}, \quad h_1=2r_1, \qquad (11)$$

при этом

$$S(r_1)=3\sqrt[3]{2\pi V^2} <= S(r). \qquad (12)$$

Рассмотрим теперь задачу во второй постановке. Продифференцируем функцию $l(r)$:

$$l'(r)=4\pi-2V/\pi r^3=2/\pi r^3(2\pi^2 r^3-V). \qquad (13)$$

Как и в предыдущем случае, при $0<r<r_2=\sqrt[3]{V/(2\pi^2)}$ производная отрицательна и функция $l(r)$ убывает, при $r_2<r<\infty$ производная

положительна и функция l*(r)* возрастает. Следовательно, своего наименьшего значения эта функция достигает в точке $r=r_2$, в которой ее производная обращается в нуль. График функции показан па рис. 4.

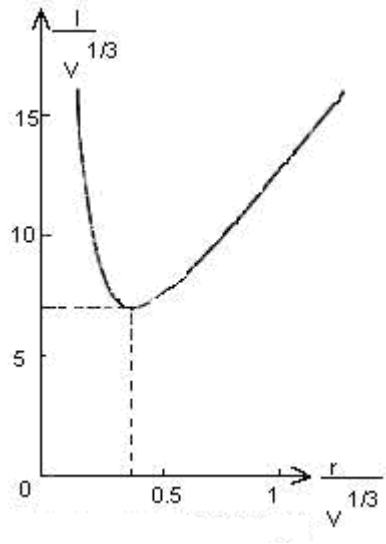


Рис. 4. График функции 1*(r)*

Итак, радиус и высота банки, наилучшие с точки зрения условия минимальности l(*r*), определяются формулами

$$r_2=\sqrt[3]{V/(2\pi^2)},\ h=2\pi r_2, \qquad (14)$$

при этом

$$1(r_2)=3\sqrt[3]{4\pi V} <= 1(r). \qquad (15)$$

Мы видим, что при разных критериях оптимизации получаются существенно разные ответы. В первом случае (11) высота "наилучшей" банки равна ее диаметру, во втором (14) она в π раз больше диаметра.

Свойства функции одной переменной

Напомним основные свойства функции одной переменной

Монотонность функции.

Функция *f(x)* является монотонной, если для любых x_1 и x_2 из области определения функции выполняется, таких, что $x_1 \le x_2$

выполняется неравенство $f(x_1) \le f(x_2)$, если функция монотонно возрастающая или $f(x_1) \ge f(x_2)$, если функция монотонно убывающая.

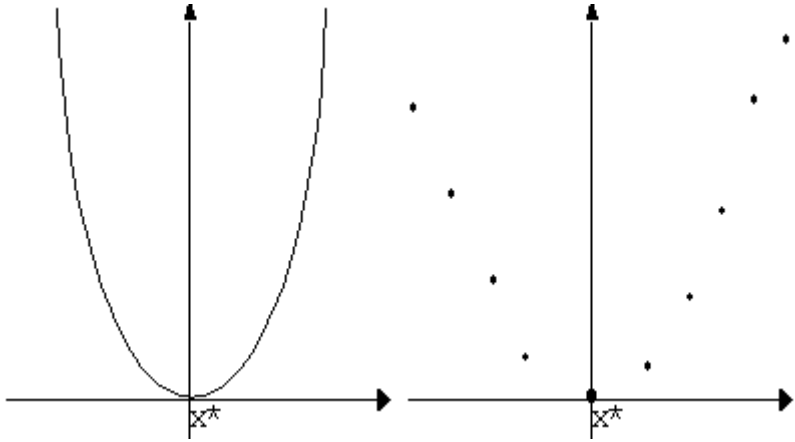
Унимодальность.

Функция *f(x)* является унимодальной на отрезке (*a, b*), если она монотонна по обе стороны от единственной на отрезке точки *x**, то есть

$$x^* \le x_1 \le x_2 \Rightarrow f(x^*) \le f(x_1) \le f(x_2)$$

или

$$x^* \ge x_1 \ge x_2 \Rightarrow f(x^*) \le f(x_1) \le f(x_2)$$



Критерии оптимальности для функций одной переменной.

Определение глобального минимума

Функция *f(x)*, определённая на множестве *S* достигает глобального минимума в точке $x^{**} \in S$, если $f(x^{**}) \le f(x)$ для всех $x \in S$.

Определение локального минимума.

Функция *f(x)*, определённая на множестве *S* имеет локальный минимум в точке $x^* \in S$, если существует такая ε-окрестность точки *x**, что для всех *x* из этой ε -окрестности $f(x^{**}) \le f(x)$.

$$\exists \varepsilon > 0, |x - x^*| \le \varepsilon, f(x^*) \le f(x)$$

Если функция *f(x)* не унимодальна, то наименьший из локальных минимумов будет глобальным (аналогично – наибольший из локальных максимумов будет глобальным максимумом).

Необходимые условия оптимальности

Чтобы точка *x** была точкой локального минимума (или максимума) дважды дифференцируемой функции *f(x)* на отрезке (*a, b*) необходимо, чтобы выполнялись следующие условия:

1. $$\left.\frac{df}{dx}\right|_{x=x^*} = 0$$

2. $$\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} \ge 0$$ (минимум)

или

$$\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} \le 0$$ (максимум)

Стационарной точкой называется x^*, в которой выполняется $\left.\frac{df}{dx}\right|_{x=x^*} = 0$.

Это точки максимума, минимума и перегиба.

Достаточные условия оптимальности.

Пусть в точке x^* первые $(n-1)$ производных функции обращаются в ноль, а (n) производная отлична от ноля, тогда если n- нечётное, то x^* - точка перегиба. Если n - чётное, то это точка оптимума. При этом, если n-я производная положительная, то точка локального минимума, отрицательна – точка локального максимума.

Алгоритм:

1. Найти 1-ю производную и станционарные точки.
2. Найти следующую производную, не равную нулю.
3. Анализировать найденную производную, как указано выше.

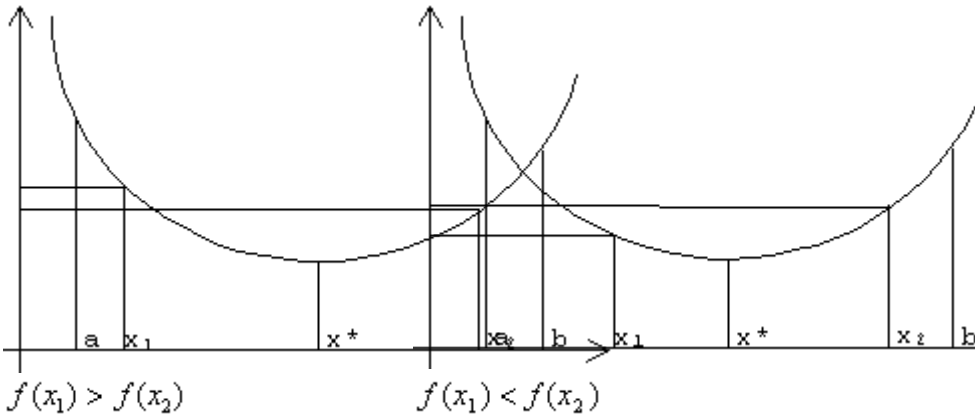
Методы одномерной оптимизации можно разделить на:

- методы исключения интервалов;
- методы точечного оценивания (полиномиальной аппроксимации);
- методы с использованием производных.

Методы интервалов

Методы ориентированы на нахождение точки оптимума внутри заданного интервала и основаны на свойстве унимодальности функции.

Правила исключения интервалов.



Пусть *f(x)* унимодальна на интервале (*a, b*) и достигает минимума в точке *x**. Рассмотрим точки x_1 и x_2 такие, что если $f(x_1) > f(x_2)$, то точка *x** принадлежит интервалу (x_1, *b*), а интервал (*a*, x_2) исключается.

Если $f(x_1) = f(x_2)$, то исключаются оба интервала (*a*, x_2) и (x_1, *b*), а точка оптимума находится принадлежит интервалу (x_1, x_2).

Достоинства метода.

- единственное ограничение на функцию – её унимодальность;
- требуется вычисления только значений функции.

В процессе применения этих методов можно выделить два этапа:

1. Этап установления границ интервалов.
2. Этап уменьшения интервалов.

Рассмотрим эти этапы.

Этап установления границ интервалов.

1. Выбирается исходная точка
2. С помощью эвристических приёмов строятся границы интервала.

Эвристический метод.

$x^{k+1} = x^k + 2^k \cdot \Delta$, где k=0,1,2,…

x^0 - произвольно выбранная точка

Δ - шаг, определяется путём сравнения значений $f(x^0)$,

$f(x^0 + |\Delta|)$, $f(x^0 - |\Delta|)$

Если $f(x^0 - |\Delta|) \geq f(x^0) \geq f(x^0 + |\Delta|)$ то x^0 правее, чем x^* и Δ>0.

Если $f(x^0 - |\Delta|) \leq f(x^0) \leq f(x^0 + |\Delta|)$ то x^0 левее, чем x^* и Δ<0.

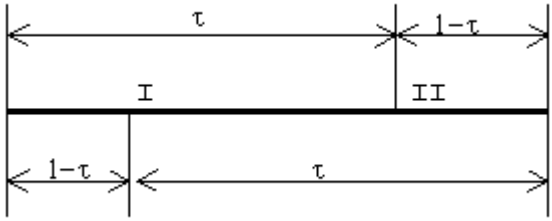
Если $f(x^0 - |\Delta|) \geq f(x^0) \leq f(x^0 + |\Delta|)$ то x^* лежит между точками $x^0 - |\Delta|$ и $x^0 + |\Delta|$ и поиск завершён.

Если при поиске минимума оказывается, что $f(x^0 - |\Delta|) \leq f(x^0) \geq f(x^0 + |\Delta|)$, то функция не унимодальна.

Этап установления интервала

Этап установления интервала основан на минимаксной стратегии поиска. Размещение пробных точек должно обеспечивать уменьшение интервала в одном и том же отношении, и это отношение должно быть максимальным.

Используется единичный интервал, поэтому найденный нужно привести к единичному. Пробные точки располагаются симметрично относительно концов интервала.

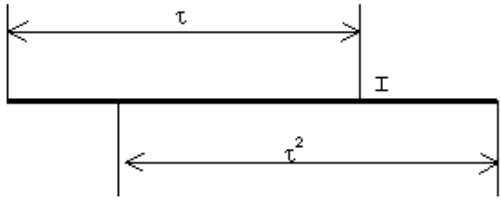


Длина остающегося после исключения интервала всегда равна τ. Пусть исключается правый интервал.

Для того, чтобы симметрия образца сохранилась расстояние 1- τ должно составлять τ часть от длинны интервала, который, в свою очередь составляет τ. 1-τ =τ^2 (Золотое сечение можно вычислить как $\tau = \frac{\sqrt{5}-1}{2} = 0{,}61803...$)

Если исходный интервал имеет единичную длину, длина интервала после *N* вычислений равна τ^{N-1}.

Если правая и левая границы интервала определены как x^R и x^L соответственно, то координаты всех последующих пробных точек вычисляются по формулам: $x= x^R- \tau^N$ или $x= x^L - \tau^N$ в зависимости от того, какой интервал был отброшен. *N* – количество вычислений.



2.2. Одномерная оптимизация

2.2.1. Минимизация функций одной переменной

Рассмотрим общие вопросы постановки и методов решения одномерных задач оптимизации. С математической точки зрения такую задачу можно сформулировать следующим образом.

Найти наименьшее (или наибольшее) значение целевой функции *f(x),* заданной на множестве *X*. Определить значение переменной $x \in X$, при котором она принимает свое экстремальное значение. В математическом анализе при изучении свойств функций, непрерывных на отрезке, доказывается следующая теорема.

Теорема Вейерштрасса. *Всякая функция f(x), непрерывная на отрезке* [*a, b*]*, принимает на этом отрезке свое наименьшее и наибольшее значения, т. е. на отрезке* [*a, b*] *существуют такие точки* x_1, x_2, *что для любого* $x \in [a, b]$*выполняются неравенства*

$$f(x_1) <= f(x) <= f(x_2). \qquad (1)$$

Не исключается, в частности, возможность того, что наименьшее или наибольшее значение достигается сразу в нескольких точках. Вы легко можете убедиться в этом, рассмотрев в качестве примера функцию $y=\sin x$ на отрезке $[0, 4\pi]$. Она достигает своего наименьшего значения, равного - 1, сразу в двух точках: $x=3\pi/2$, $x=7\pi/2$. Наибольшее значение, равное 1, достигается тоже в двух точках: $x=\pi/2$, $x=5\pi/2$. Теорема Вейерштрасса играет в данном случае роль теоремы существования: согласно этой теореме задача оптимизации, в которой целевая функция *f(x)* задана и непрерывна на отрезке, всегда имеет решение.

Теперь нам предстоит обсудить методы решения задач оптимизации. Рассмотрим наиболее простой класс задач. При их исследовании мы будем предполагать, что целевая функция *f(x)* дифференцируема на отрезке [*a, b*] и имеется возможность найти явное выражение для ее производной *f'(x)*. Точки, в которых производная обращается в нуль, называются *критическими* или *стационарными точками* функции *f(x)*. Если интерпретировать производную как скорость изменения функции, то в критических точках эта скорость равна нулю, изменение функции на мгновение "останавливается". Функция *f(x)* может достигать своего наименьшего (наибольшего) значения либо в одной из двух граничных точек отрезка [*a, b*], либо в какой-нибудь его внутренней точке. В последнем случае такая точка обязательно должна быть критической, это необходимое условие экстремума. Учитывая изложенное, можем сформулировать следующее правило решения задачи оптимизации для рассматриваемого класса функций.

Для того чтобы определить наименьшее и наибольшее значения дифференцируемой функции f (x) на отрезке [a, b], нужно найти все

ее критические точки на данном отрезке, присоединить к ним граничные точки a и b, и во всех этих точках сравнить значения функции. Наименьшее и наибольшее из них дадут наименьшее и наибольшее значения функции для всего отрезка.

Поскольку граничные точки a и b искать не нужно, то с технической точки зрения все сводится к определению критических точек, которые являются корнями уравнения

$$f'(x)=0. \qquad (2)$$

Для иллюстрации изложенного правила решения задачи оптимизации рассмотрим на отрезке [-2, 3] функцию

$$f(x)= 3x^4- 4x^3- 12x^2+2. \qquad (3)$$

Вычислим ее производную: $f'(x)=12x^3-12x^2-24x$.Таким образом, уравнение (2) для определения критических точек в данном случае принимает вид

$$x^3-x^2-2x=0. \qquad (4)$$

Все корни этого уравнения: $x_1= -1$, $x_2=0$, $x_3=2$ принадлежат исхному отрезку. Добавляя к ним граничные точки: $a=-2$, $b=3$, вычислим соответствующие значения функции (3):

$$f(-2) = 34,\ f(-1) = -3,\ f(0) = 2,\ f(2) = -30,\ f(3)= 29.$$

Из сравнения этих чисел следует, что наименьшее значение функции $f(x)$ достигается в одной из критических точек $x=2$, а наибольшее - в граничной точке $x= -2$, причем

$$f_{min}=f(2)= -30,$$
$$f_{max}=f(-2)= 34.$$

График функции (3), иллюстрирующий проведенное исследование, показан на рис.1.

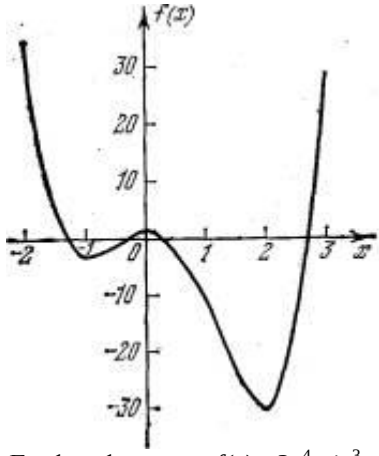


Рис. 1. График функции $f(x)= 3x^4- 4x^3- 12x^2+2$.

В простейших случаях, нули производной удается найти аналитически. На это в первую очередь и рассчитан данный метод, хотя не исключается возможность численного решения уравнения (2). Однако при этом важно найти все критические точки, иначе мы рискуем допустить ошибку, пропустив истинное наименьшее или наибольшее значение функции.

Рассмотрение функций как унимодальных во всей области определения в общем случае невозможно. Однако при включении в процесс минимизации предварительного этапа, на котором отрезок минимизации разделяют на несколько отрезков, на каждом из которых минимизируемая функция унимодальна, позволяет избежать ошибок, связанных с нахождением локальных минимумов.

В данном разделе будем рассматривать только методы *нулевого порядка,* т. е. методы, использующие информацию о функциях. Чаще они называются *прямыми методами.*

Из математического анализа известны следующие условия локального экстремума функции *f(x),* дифференцируемой достаточное число раз.

1. Если функция *f(x)* дифференцируема в точке $\tilde{x}$ и достигает в этой точке локального экстремума, то $f'(\tilde{x}) = 0$ (необходимое условие экстремума).

2. Пусть функция *f(x)* *n* раз дифференцируема в точке f и в этой точке все производные *f(x)* до *n*-1-го порядка включительно равны нулю, а $f^{(n)}(x) \neq 0$. Тогда, если *n* - нечетно, то $\tilde{x}$ не является точкой локального экстремума функции *f(x).* Если же *n* - четное число, то:

a) при $f^{(n)}(\tilde{x})>0$ $\tilde{x}$ - точка локального минимума *f(x)*;

b) при $f^{(n)}(\tilde{x})<0$ $\tilde{x}$ - точка локального максимума *f(x)* (достаточное условие экстремума).

Перечисленные условия позволяют предложить следующий путь решения задачи минимизации:

1) с помощью условия 1 находим все точки возможного экстремума функции *f(x)* на интервале *(a; b)*, т.е. корни уравнения

$$f'(x)=0 \qquad (5)$$

(стационарные точки), принадлежащие интервалу *(a; b)*;

2) найденные стационарные точки исследуем в соответствии с условием 2, выделяя из них только точки локальных минимумов *f(x)*;

3) значения *f(x)* в точках локальных минимумов и на концах отрезка *(a; b)* сравниваем между собой. Наименьшему из этих значений соответствует точка глобального минимума *f(x)* на (*f(x)*; *b)*.

Замечание. Применение условия 2 требует вычисления высших производных функций *f(x)*, поэтому в большинстве случаев бывает проще сравнить значения *f(x)* во всех стационарных точках, не интересуясь их характером. С учетом этого можно предложить следующий алгоритм минимизации *f(x)* на отрезке *(a; b) (классический метод)*.

Шаг 1. Решить уравнение (5) на интервале $x\in(a; b)$, т.е. найти все стационарные точки $x_1,\ldots,x_k \in (a;\ b)$. Положить $x_0=a$, $x_k=b$.

Шаг 2. Вычислить значения $f(x_i)$ функции *f(x)* в точках x_i, $i = 0,\ldots,k$.

Шаг 3. Найти $f^* = \min\limits_{0\le i\le k} f(x_i) = f(x_m)$. Положить $x^* = x_m$.

Пример 1. Классический метод минимизации.

Решить задачу $f(x) = x^3 - 3x + 1 \to \min,\ x \in [-2;\ 2]$.

Шаг 1. Находим корни уравнения $f'(x) = 3x^2 - 3 = 0$ из интервала (-2; 2): $x_1 = -1$, $x_2 = 1$. Полагаем $x_0 = -2$, $x_3 = 2$.

Шаг 2. Вычисляем значения *f*(*x*) в точках x_i, $i = 0,\ldots,3$:

$$f(x_0) = -17,\ f(x_1) = 3,\ f(x_2) = -1,\ f(x_3) = 1.$$

Шаг 3. Находим $f^* = \min(-17{,}3; -1{,}1) = -17 = f(x_0)$.

Поэтому $x^* = x_0 = -2$, $f^* = -17$.

В некоторых случаях нахождения экстремума задачи оптимизации, учитывая совокупность ограничений, позволяет через одну из управляемых переменных выразить остальные и исключить их из целевой функции. В результате задача будет сведена к поиску

экстремума скалярной функции одной переменной $f(x),\ \ x \in D(f) \subset R,$ выражающей критерий оптимальности. Как уже говорилось ранее, будем рассматривать задачу поиска наименьшего значения функции $f(x)$ в области допустимых решений $D(f)$.

Изучение методов *одномерной минимизации* имеет самостоятельное значение. Эти методы являются существенной составной частью методов многомерной оптимизации при помощи которых находят наименьшее значение действительных функций многих переменных.

Для существования минимума $f(x)$ в $D(f)$ необходимо и достаточно, чтобы $f(x)$ была непрерывна, а $D(f)$ - конечным отрезком. Однако при нарушении этих условий ($f(x)$ имеет в $D(f)$ точки разрыва или $D(f)$ - интервал или полуинтервал), наименьшее значение может и не достигаться в $D(f)$. В этом случае отыскивается $\inf\limits_{x \in D(f)} f(x)$, т. е. под решением задачи минимизации такой функции на $D(f)$ следует понимать построение последовательности $\{x_n\}$ точек из $D(f)$, для которых существует

$$\lim_{n \to \infty} f(x_n) = \inf_{x \in D(f)} f(x) = f^*.$$

Пример. Найти минимум $f(x) = \dfrac{1}{x},\ D(f) = [1;2)$. Функция не достигает наименьшего значения на $D(f)$, хотя и ограничена снизу

$$(f^* = \frac{1}{2}).$$

В качестве последовательности $\{x_n\}$ точек из полуинтервала [1;2) выберем $\left\{2 - \dfrac{1}{n}\right\}$. Тогда

$$f(x_n) = \frac{1}{x_n} = \frac{n}{2n-1}, \quad \lim_{n \to \infty} f(x_n) = \frac{1}{2} = \inf_{x \in [1\ 2)} f(x).$$

Функция может достигать наименьшего значения, как в единственной точке, так и на некотором множестве точек, конечном, счётном или несчётном. Фактически, количество значений точек

минимума зависит от того, является ли $f(x)$ сильно выпуклой, строго выпуклой или просто выпуклой.

Аналогом выпуклых функций в одномерном случае является *унимодальная функция.* Функция $f(x)$ называется *унимодальной* на отрезке $[a;b]$, если она непрерывна на $[a;b]$и существуют числа α и β, $a \le \alpha \le \beta \le b$, такие, что:

1) если $a < \alpha$, то $f(x)$ монотонно убывает при $x \in (a;\alpha]$;

2) если $x \in [\alpha;\beta]$, то $f(x) = f^* = \min\limits_{x \in [a,b]} \min f(x)$;

3) если $\beta < b$ то $f(x)$ монотонно возрастает при $x \in (\beta;b]$.

Множество унимодальных на отрезке $[a;b]$ функций мы будем обозначать через $Q[a;b]$.

Отметим, что возможно вырождение в точку одного или двух отрезков из $[a;\alpha]$, $[\alpha;\beta]$ и $[\beta;b]$. Некоторые варианты расположения и вырождения в точку отрезков монотонности и постоянства унимодальной функции показаны на рис. 2.

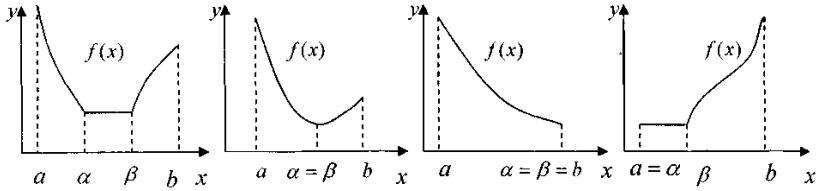


Рис. 2. Графики унимодальных функций

Известны следующие основные свойства унимодальных функций.

1. Любая из точек локального минимума унимодальной функции является и точкой её глобального минимума на отрезке $[a;b]$.

2. Функция унимодальная на отрезке $[a;b]$ является унимодальной на любом меньшем отрезке $[c;d] \subset [a;b]$.

3 Пусть $f(x) \in Q[a;b]$ и $a \le x_1 < x_2 \le b$. Тогда:

если $f(x_1) \le f(x_2)$, то $x^* \in [a;x_2]$;

если $f(x_1) > f(x_2)$, то $x^* \in [x_1;b]$.

где x^* - одна из точек минимума $f(x)$ на отрезке $[a;b]$.

Рассмотрение функций как унимодальных во всей области определения в общем случае невозможно. Однако при включении в процесс минимизации предварительного этапа, на котором отрезок минимизации разделяют на несколько отрезков, на каждом из которых

минимизируемая функция унимодальна, позволяет избежать ошибок, связанных с нахождением локальных минимумов.

При решении практических задач оптимизации классический метод имеет ограниченное применение. Это объясняется тем, что, во-первых, во многих случаях значения целевой функции $f(x)$ находятся из измерений или экспериментов, а измерение производной $f'(x)$ затруднительно или невозможно и, во-вторых, даже когда производная $f'(x)$ задана аналитически или поддается измерению, решение уравнения (5) зачастую вызывает затруднения.

2.2.2. Постановка задачи одномерной минимизации

Рассмотрим задачу безусловной минимизации функции одного переменного*:*

Требуется найти т. $x^* \in R$ *такую, что*

$$\Phi(x^*) = \min \Phi(x) \Leftrightarrow x^* \in \Phi(x) \; loc \min_{x \in R} \Phi(x). \qquad (6)$$

Если функция $\Phi(x) \in C^2(R)$ дважды непрерывно дифференцируема, то известны необходимое и достаточное условия минимума

необходимое условия экстремума	достаточное условия экстремума	
$\Phi'(x^*)=0$	$\Phi'= (x^*)$	(7)
$\Phi'' (x^*)\geq 0$	$\Phi'' (x^*)>0$	

(Взятые по отдельности – это соответствующие условия оптимальности точки x^* первого и второго порядков как необходимые, так и достаточные)

В таком стучае, при нахождении в достаточно малой окрестности точки x^*, разложение целевой функции в ряд Тейлора с центром в точке x^* имеет вид

$$\Phi(x^*+h)= \Phi(x^*)+ \Phi'(x^*)h+ \frac{1}{2!} \Phi'' (x^*)\, h^2+o(h^2).$$

В этом выражении $\Phi'(x^*)h \equiv 0$ в силу (7)

Мы говорим о *невырожденности минимума* в точке x^*, если $\Phi'' (x^*) \neq 0$, тем самым, согласно (7), $\Phi'' (x^*)>0$. В дальнейшем будем предполагать это условие выполненным.

Подчеркнем еше раз, что мы пытаемся рассмотреть способы минимизации задачи (6), а не решение задачи (7) из необходимого условия экстремума. Хотя, конечно, это тесно связанные проблемы.

3. Методы одномерной минимизации нулевого порядка (прямые методы)

Под методами минимизации *нулевого порядка* подразумевают группу методов не использующих явно производные целевой фvнкции.

Предположим что точки a и b определяют возможно и достаточно грубо, интервал, где расположено значение точки минимума x^* задачи

$$\Phi(x^*) = \min \Phi(x) \Leftrightarrow x^* \in \Phi(x) \; loc \min_{x \in R} \Phi(x).$$

Если считать, что внутри этого интервала функция $\Phi(x^*)$ *унимодальна,* т. е. имеет единственный минимум, то одна из возможностей построения последовательности стягавающихся отрезков $x^* \in [x_{k-1}, x_k]$, локализующих x^* возможна на основании прямых методов.

Для решения задачи минимизации функции $f(x)$ на отрезке $[a;b]$ на практике, как правило, применяют приближенные методы. Они позволяют найти решение этой задачи с необходимой точностью в результате определения конечного числа значений функции $f(x)$ и ее производных в некоторых точках отрезка. Методы, использующие только значения функции и не требующие вычисления ее производных, называются, как мы определили выше, прямыми методами минимизации.

Большим достоинством прямых методов является то, что от целевой функции не требуется дифференцируемости и, более того, она может быть не задана в аналитическом виде. Единственное, на чем основаны алгоритмы прямых методов минимизации, это возможность определения значений $f(x)$ в заданных точках.

Решение многих теоретических и практических задач сводится к отысканию экстремума (наибольшего или наименьшего значения) скалярной функции $f(x)$ n-мерного векторного аргументах. В дальнейшем под x будем понимать вектор-столбец (точку в n-мерном пространстве):

$$x = \begin{vmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{vmatrix}$$

Вектор-строка получается путем применения операции транспонирования:

$$x^T = (x_1, x_2, \ldots, x_n).$$

Вектор *x*,* определяющий минимум целевой функции, называют оптимальным.

Отметим, что задачу максимизации *f(x)* можно заменить эквивалентной ей задачей минимизации или наоборот. Рассмотрим это на примере функции одной переменной (рис. 1).

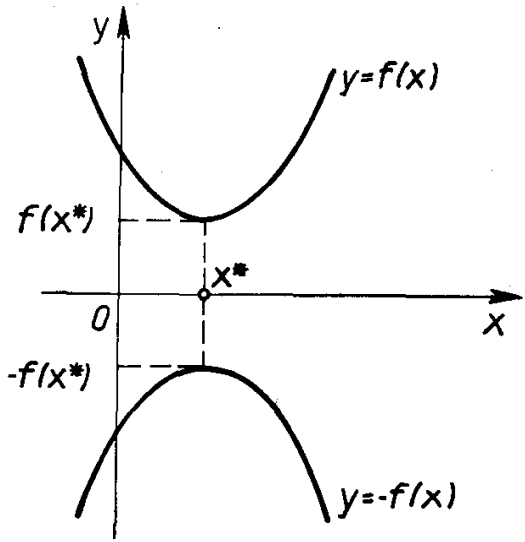


Рис. 1. Экстремум

Если *x** - точка минимума функции *y=f(x),* то для функции *y =- f(x)* она является точкой максимума, так как графики функций *f(x)* и - *f(x),* симметричны относительно оси абсцисс. Итак, минимум функции *f(x)* и максимум функции - *f(x)* достигаются при одном и том же значении переменной. Минимальное же значение функции *f(x)* равно максимальному значению функции - *f(x)*, взятому с противоположным знаком, т.е. min *f(x)* =-max*(f(x)).*

Рассуждая аналогично, этот вывод нетрудно распространить на случай функции многих переменных. Если требуется заменить задачу минимизации функции $f(x_1, ..., x_n)$ задачей максимизации, то достаточно вместо отыскания минимума этой функции найти максимум функции $f(x_1, ..., x_n)$. Экстремальные значения этих функций достигаются при одних и тех же значениях переменных. Минимальное значение функции $f(x_1, ..., x_n)$ равно максимальному значению функции $- f(x_1, ..., x_n)$, взятому с обратным знаком, т.е. $\min f(x_1, ..., x_n) = \max f(x_1, ..., x_n)$. Отмеченный факт позволяет в дальнейшем говорить только о задаче минимизации.

В реальных условиях на переменные x_i, $i=1, \ldots. n$, и некоторые функции $g_i(x)$, $h_i(x)$, характеризующие качественные свойства объекта, системы, процесса, могут быть наложены ограничения (условия) вида:

$$g_i(x) = 0, \; i=1, \ldots. n,$$

$$h_i(x) <= 0, \; i=1, \ldots. n,$$

$$a \le x \le b,$$

где

$$a = \begin{vmatrix} a_1 \\ a_2 \\ \ldots \\ a_n \end{vmatrix}; \quad b = \begin{vmatrix} b_1 \\ b_2 \\ \ldots \\ b_3 \end{vmatrix}$$

Такую задачу называют *задачей условной оптимизации.* При отсутствии ограничений имеет место *задача безусловной оптимизации.*

Каждая точка x в n-мерном пространстве переменных $x_1, ..., x_n$, в которой выполняются ограничения, называется *допустимой точкой задачи.* Множество всех допустимых точек называют *допустимой областью G. Решением задачи (оптимальной точкой)* называют допустимую точку x^*, в которой целевая функция $f(x)$ достигает своего минимального значения.

Точка x^* определяет *глобальный минимум* функции одной переменной $f(x)$, заданной на числовой прямой X, если $x^* \in X$ и $f(x^*) < f(x)$ для всех $x^* \in X$ (рис. 2, а). Точка x^* называется *точкой строгого глобального минимума,* если это неравенство выполняется как строгое. Если же в выражении $f(x^*) \le f(x)$ равенство возможно при x, *не равных* x^*, то реализуется *нестрогий минимум,* а под решением в этом случае понимают множество $x^* = [x^* \in X: f(x) = f(x^*)]$ (рис. 2, б).

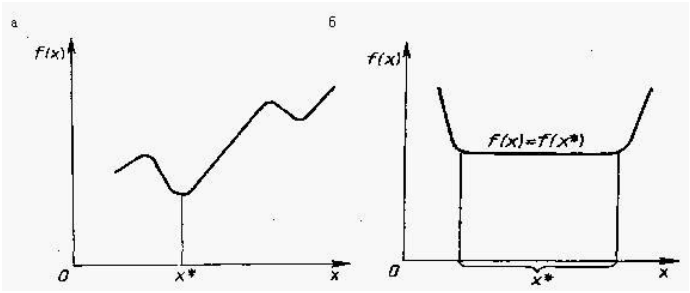


Рис. 2. Глобальный минимум: а - строгий, б - нестрогий

Точка $x^* \in X$ определяет *локальный минимум* функции $f(x)$ на множестве X, если при некотором достаточно малом $\varepsilon > 0$ для всех x, *не равных* x^*, $x \in X$, удовлетворяющих условию $|x - x^*| \le \varepsilon$, выполняется неравенство $f(x^*) < f(x)$. Если неравенство строгое, то x^* является точкой строгого локального минимума. Все определения для максимума функции получаются заменой знаков предыдущих неравенств на обратные. На рис. 3 показаны экстремумы функции одной переменной $f(x)$ на отрезке $[a, b]$. Здесь x_1, x_3, x_6 - точки локального максимума, а x_2, x_4 - локального минимума. В точке x_6 реализуется глобальный максимум, а в точке x_2 - глобальный минимум.

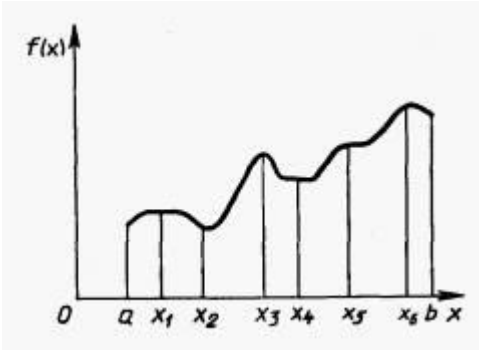


Рис. 3. Экстремумы функции

Возможны два подхода к решению задачи отыскания минимума функции многих переменных $f(x) = f(x_1, ..., x_n)$ при отсутствии ограничений на диапазон изменения неизвестных. Первый подход лежит в основе *косвенных методов оптимизации* и сводит решение задачи оптимизации к решению системы нелинейных уравнений, являющихся следствием условий экстремума функции многих переменных. Как известно, эти условия определяют, что в точке экстремума x^* все первые производные функции по независимым переменным равны нулю:

$$\frac{\partial f}{\partial x_i}\Big/_{x = x^*} = 0,$$

$i=1, ..., n.$

Эти условия образуют систему n нелинейных уравнений, среди решений которой находятся точки минимума. Вектор $f'(x)$, составленный из первых производных функции по каждой переменной, т.е.

$$f'(x) = (\partial f(x) / \partial x_1, \ldots, \partial f(x) / \partial x_n)^T,$$

называют градиентом скалярной функции *f(x).* Как видно, в точке минимума градиент равен нулю.

Решение систем нелинейных уравнений - задача весьма сложная и трудоемкая. Вследствие этого на практике используют второй подход к минимизации функций, составляющий основу *прямых методов.* Суть их состоит в построении последовательности векторов $x[0], x[1], \ldots, x[n]$, таких, что $f(x[0]) > f(x[1]) > f(x[n]) > \ldots$ В качестве начальной точки $x[0]$ может быть выбрана произвольная точка, однако стремятся использовать всю имеющуюся информацию о поведении функции *f(x),* чтобы точка $x[0]$ располагалась как можно ближе к точке минимума. Переход (итерация) от точки $x[k]$ к точке $x[k+1]$, $k = 0, 1, 2, \ldots$, состоит из двух этапов:

1. выбор направления движения из точки $x[k]$;
2. определение шага вдоль этого направления.

Методы построения таких последовательностей часто называют *методами спуска,* так как осуществляется переход от больших значений функций к меньшим.

Математически методы спуска описываются соотношением

$$x[k+1] = x[k] + a_k p[k], \; k = 0, 1, 2, \ldots,$$

где $p[k]$ - вектор, определяющий направление спуска; a_k - длина шага. В координатной форме:

$$\begin{vmatrix} x_1[k+1] = x_1[k] + a_k p_1[k] \\ x_2[k+1] = x_2[k] + a_k p_2[k] \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n[k+1] = x_n[k] + a_k p_n[k] \end{vmatrix}$$

Различные методы спуска отличаются друг от друга способами выбора **двух параметров - направления спуска и длины шага вдоль этого направления**. На практике применяются только методы, обладающие сходимостью. Они позволяют за конечное число шагов получить точку минимума или подойти к точке, достаточно близкой к

точке минимума. Качество сходящихся итерационных методов оценивают по скорости сходимости.

В методах спуска решение задачи теоретически получается за бесконечное число итераций. На практике вычисления прекращаются при выполнении некоторых критериев (условий) останова итерационного процесса. Например, это может быть условие малости приращения аргумента

$$|x[k] - x[k-1]| < \varepsilon$$

или функции

$$|f(x[x]) - f(x[k-1])| < \gamma.$$

Здесь k - номер итерации; ε, γ - заданные величины точности решения задачи.

Методы поиска точки минимума называются *детерминированными,* если оба элемента перехода от $x[k]$ к $x[k+1]$ (направление движения и величина шага) выбираются однозначно по доступной в точке $x[k]$ информации. Если же при переходе используется какой-либо случайный механизм, то алгоритм поиска называется *случайным поиском минимума.*

Детерминированные алгоритмы безусловной минимизации делят на классы в зависимости от вида используемой информации. Если на каждой итерации используются лишь значения минимизируемых функций, то метод называется *методом нулевого порядка.* Если, кроме того, требуется вычисление первых производных минимизируемой функции, то имеют место методы *первого порядка,* при необходимости дополнительного вычисления вторых производных - *методы второго порядка.*

В настоящее время разработано множество численных методов для задач как безусловной, так и условной оптимизации. Естественным является стремление выбрать для решения конкретной задачи наилучший метод, позволяющий за наименьшее время использования ЭВМ получить решение с заданной точностью.

Качество численного метода характеризуется многими факторами: скоростью сходимости, временем выполнения одной итерации, объемом памяти ЭВМ, необходимым для реализации метода, классом решаемых задач и т. д. Решаемые задачи также весьма разнообразны: они могут иметь высокую и малую размерность, быть унимодальными (обладающими одним экстремумом) и многоэкстремальными и т. д. Один и тот же метод, эффективный для решения задач одного типа, может оказаться совершенно неприемлемым для задач другого типа. Очевидно, что разумное сочетание разнообразных методов, учет их свойств позволят с наибольшей эффективностью решать поставленные задачи. Многометодный способ решения весьма удобен в диалоговом режиме работы с ЭВМ. Для успешной работы в таком режиме очень полезно знать основные свойства, специфику методов оптимизации. Это обеспечивает способность правильно ориентироваться в различных ситуациях, возникающих в процессе расчетов, и наилучшим образом решить задачу.

3.1. Общая характеристика методов нулевого порядка

В этих методах для определения направления спуска не требуется вычислять производные целевой функции. Направление минимизации в данном случае полностью определяется последовательными вычислениями значений функции. Следует отметить, что при решении задач безусловной минимизации методы первого и второго порядков обладают, как правило, более высокой скоростью сходимости, чем методы нулевого порядка. Однако на практике вычисление первых и вторых производных функции большого количества переменных весьма трудоемко. В ряде случаев они не могут быть получены в виде аналитических функций. Определение производных с помощью различных численных методов осуществляется с ошибками, которые могут ограничить применение таких методов. Кроме того, на практике встречаются задачи, решение которых возможно лишь с помощью методов нулевого порядка, например задачи минимизации функций с разрывными первыми производными. Критерий оптимальности может быть задан не в явном виде, а системой уравнений. В этом случае аналитическое или численное определение производных становится очень сложным, а иногда невозможным. Для решения таких практических задач оптимизации могут быть успешно применены методы нулевого порядка. Рассмотрим некоторые из них.

3.2. Нелокальная линейная аппроксимация.

а) *Метод конечных разностей.* Этот метод состоит в замене производных соответствующим образом выбранными разностями.

Одномерная задача. Рассмотрим задачу:

$$\begin{cases} -u'' + c(x)u = f(x), \\ u(0) = \alpha, \ u(L) = \beta, \end{cases} \tag{1}$$

где

$$c(x) \geqslant \gamma > 0.$$

Разложим $u(x + h)$ в ряд по степеням h:

$$u(x + h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + \frac{h^4}{24}u^{(4)}(x) + \ldots .$$

Отсюда следует

$$u''(x) = \frac{1}{h^2}[u(x + h) - 2u(x) + u(x - h)] - \frac{h^2}{12}u^{(4)}(\xi)$$

при $\xi \in [x - h, \ x + h]$.

Обозначим $u_i = u(ih)$, где $h = \dfrac{L}{n + 1}$. Тогда

$$\frac{1}{h^2}(-u_{i-1} + 2u_i - u_{i+1}) + c_i u_i = f_i. \tag{2}$$

Уравнения (2) образуют систему линейных уравнений относительно неизвестных $u_1, u_2, \ldots, u_n$, при этом $u_0 = \alpha$ и $u_{n+1} = \beta$. Эта система имеет ленточную симметричную трехдиагональную матрицу, что позволяет проводить вычисления быстро и точно.

В нашем примере легко показать сходимость приближенного решения к истинному при $n \to \infty$, т. е. $h \to 0$. Действительно, вычитая выражение (1) из (2), получим

$$\frac{1}{h^2}(-\delta_{i-1} + 2\delta_i - \delta_{i+1}) + c_i\delta_i = -\varepsilon_i,$$

где $\delta_i = u(x_i) - u_i$, $\varepsilon_i = \dfrac{h^2}{12}u^{(4)}(\xi_i)$.

Отсюда следует $\left(\dfrac{2}{h^2} + c_i\right)\delta_i = \dfrac{1}{h^2}(\delta_{i-1} + \delta_{i+1}) + \varepsilon_i,$

$$\left(\frac{2}{h^2} + \gamma\right)|\delta_i| \leqslant \left|\left(\frac{2}{h^2} + c_i\right)\delta_i\right| \leqslant \frac{1}{h^2}(|\delta_{i-1}| + |\delta_{i+1}|) + |\varepsilon_i|.$$

Пусть $\|\varepsilon\| = \max\limits_i |\varepsilon_i|$ и $\|\delta\| = \max\limits_i |\delta_i|$, тогда

$$\left(\frac{2}{h^2} + \gamma\right)\|\delta\| \leqslant \frac{1}{h^2}(\|\delta\| + \|\delta\|) + \|\varepsilon\|,$$

$$\|\delta\| \leqslant \frac{1}{\gamma}\|\varepsilon\| = O(h^2) \to 0,$$

что и доказывает сходимость.

Чтобы повысить порядок погрешности метода, разложим $u(x + 2h)$, $u(x + h)$, $u(x - h)$, $u(x - 2h)$ в ряд Тейлора в окрестности x:

$$\frac{1}{h^4}\{u_{k-2} - 4u_{k-1} + 6u_k - 4u_{k+1} + u_{k+2}\} = u_k^{(4)} + \frac{h^2}{6}u_k^{(6)} + 0(h^4).$$

Если учесть, что

$$\frac{1}{h^2}(-u_{k-1} + 2u_k - u_{k+1}) = -u_k'' - \frac{h^2}{12}u_k^{(4)} + 0(h^4),$$

и подставить $u^{(4)}_k$ в предыдущую формулу, получим систему линейных уравнений $Au = b$, где

$$A = \frac{1}{12h^2}D + \operatorname{diag}(c_1, c_2, \ldots, c_n),$$

$$D = \begin{vmatrix} 24 & -12 & 0 & & & \\ -16 & 30 & -16 & 1 & & \\ 1 & -16 & 30 & -16 & 1 & \\ & 1 & -16 & 30 & -16 & 1 \\ & & & & & \\ & \cdot & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \cdot & \end{vmatrix}, \quad b = \begin{vmatrix} f_1 + \alpha/h^2 \\ f_2 - \alpha/12h^2 \\ f_3 \\ \\ f_4 \\ \\ \\ \end{vmatrix}$$

В этом случае погрешность метода составляет

$$\varepsilon(x_k) = \frac{h^4}{90}u_k^{(6)} + 0(h^6).$$

б) *Симплексный метод.* В конечно-разностном градиентном методе *пробные* и *рабочие шаги* разделяют — точки $x^k + \alpha_i e_i$ служат только для оценки градиента в x^k, в x^{k+1} вся работа проводится заново. Можно поступить и иначе, и строить линейную аппроксимацию по набору точек, расположенных достаточно далеко.

Типичным примером служит так называемый *симплексный метод* (не путать с симплекс-методом в линейном программировании!). Пусть выбраны $n+1$ точек $x^0, x^1, \ldots, x^n$, образующие вершины правильного симплекса. Вычислим значения $f(x)$ в вершинах и найдем ту, для которой $f(x)$ максимальна:

$$j = \arg\max_{0 \le i \le n} f(x^i).$$

Построим новый симплекс, отличающийся от старого лишь одной вершиной; x^j заменяется на x^{n+1}:

$$x^{n+1} = 2n^{-1}(x^0 + \ldots + x^{j-1} + x^{j+1} \ldots + x^n) - x^j \quad (3)$$

(т. е. x^{n+1} симметрично с x^j относительно грани, противолежащей x^j). Если окажется, что в новом симплексе максимум достигается в x^{n+1}, то возвращаемся к исходному симплексу, заменив x^j на вершину, в которой значение $f(x)$ максимально среди оставшихся вершин и т. д. Если какая-либо точка сохраняется в n+1 последовательном симплексе, то последний симплекс сокращается вдвое подобным преобразованием с центром в этой вершине (рис. 1).

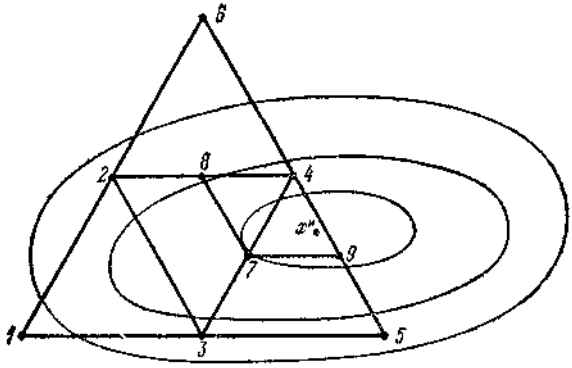


Рис. 1. Симплексный метод

Мы описали лишь простейший вариант метода. Существует много его модификаций, в которых симплекс не обязательно правильный, а величина шага и условия дробления могут быть иными. С теоретической точки зрения подобные методы слабо исследованы. Практика показывает их работоспособность для не слишком плохо обусловленных задач.

3.3. Квадратичная аппроксимация.

Вычислив значения $f(x)$ в достаточном числе точек, можно построить квадратичную аппроксимацию $f(x)$. Удобно это сделать, например, следующим образом *(метод барицентрических координат)*. Выбирается (как и в симплексном методе) n+1 базисных точек $x^0,\ldots,x^n$.

Вычисляются значения функции во всех этих точках и серединах соединяющих их отрезков (обозначим $f((x^i+x^j)/2)=f_{ij}$, $f(x^i)=f_{ii}$, $i, j=0, \ldots, n$). После этого решается система линейных (относительно $\lambda_1, \lambda_2, \ldots, \lambda_n$) уравнений

$$\begin{aligned} &1\sum_{j=0}^{n} f_{ij}\lambda_j + \lambda = f_{ii}, \quad i=0, \ldots, n, \\ &\sum_{j=0}^{n}\lambda_j = 1 \end{aligned} \tag{4}$$

и строится точка

$$x^{n+1}=\sum_{i=0}^{n}\lambda_i x^i. \tag{5}$$

Нетрудно проверить, что если f квадратична, то $x^{n+1} = x^* = A^{-1}b$ при любых $x^0, \ldots, x^n$ таких, что $x^n = x^0,\ldots,x^1 - x^0$ линейно независимы.

Далее (для неквадратичной $f(x)$) точка x^{n+1} включается в число базисных, а одна из прежних базисных точек (точка x^0 или та, в которой $f(x)$ максимальна) удаляется. На следующей итерации достаточно вычислить $f(x)$ в $n+1$ точках (в x^{n+1} и серединах отрезков, соединяющих x^{n+1} с остальными базисными точками). Новая система уравнений для λ_i будет отличаться от (4) лишь одной строкой, так что можно использовать результат известной леммы для построения решения. Аналогичным образом процесс продолжается дальше.

Удобство метода в том, чго сама квадратичная аппроксимация функции не выписывается явно, строится лишь точка минимума этой аппроксимации. По сравнению с конечно-разностным аналогом метода Ньютона здесь существенно меньше вычислений $f(x)$ на каждом шаге ($n+1$ вместо $n(n+1)/2$). Для придания устойчивости процессу в нем нужно ввести регулировку длины шага, принять меры для предотвращения вырождения системы базисных точек, проверять условие выпуклости $f_{ij} \le (f_{ii} + f_{jj})/2$ и т. п.

Другая группа методов прямого поиска использует идеи метода сопряженных направлений и сводит исходную задачу к *последовательности одномерных минимизаций.* В отличие от метода покоординатного спуска, где система направлений спуска жестко фиксируется (этой системой являются координатные орты), в данных методах направления спуска строятся в процессе минимизации. Принцип их построения — сделать их (для задачи минимизации квадратичной функции) сопряженными; тогда, как мы знаем процесс минимизации конечен в квадратичном случае. Основная идея методов

этой группы иллюстрируется рис. 2 — три последовательные одномерные минимизации приводят в точку минимума.

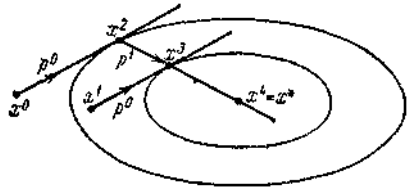


Рис. 2. Метод сопряженных направлений

В многомерном пространстве верен аналогичный результат.

Лемма 2. *Пусть* $f(x)=(Ax,\ x)/2-(b,\ x),\ A>0,\ x\in \mathbf{R}^n$, $p^1, ..., p^k$ *— сопряженные векторы:* $(Ap^i, p^j)=0,\ i\neq j,\ k<n$, *Тогда*

$$L^0=\left\{x:\ x=x^0+\sum_{i=1}^{n}\lambda_i p^i\right\},\ x^1\in L^0,\ L^1=\left\{x:\ x=x^1+\sum_{i=1}^{n}\lambda_i p^i\right\},$$

$$y^0=\operatorname*{argmin}_{x\in L^0} f(x),\ y^1=\operatorname*{argmin}_{x\in L^1} f(x).$$

Тогда вектор $p^{k+1}=y^1-y^0$ *является сопряженным с* $p^1, ..., p^k$.

Этот результат следует из условия минимума $f(x)$ на подпространстве.

На этой основе можно построить метод минимизации, например, следующим образом. Пусть x^k — полученное на k-й итерации приближение к решению, $p^0, ..., p^k$ — найденные направления (x^0 и p^0 произвольны). Построим $\bar{x}^k=x^k+h^k$, где h^k — произвольный вектор, не являющийся линейной комбинацией $p^0, ..., p^k$. Проведем цикл последовательных одномерных минимизаций по направлениям $p^0, ..., p^k$, начиная из точки $\bar{x}^k$; обозначим полученную в результате точку $\tilde{x}^{k+1}$. В качестве x^{k+1} возьмем минимум $f(x)$ на прямой, соединяющей $\tilde{x}^{k+1}$ с x^k, а в качестве p^{k+1} — вектор $\tilde{x}^{k+1}-x^k$. Для квадратичной функции в $\mathbf{R}^n$ такой *метод Пауэлла* приводит к минимуму не более чем за n шагов.

Существует и много других модификаций, основанных на подобной идее. Всего для отыскания минимума в квадратичном случае требуется $n(n+1)/2$ одномерных минимизаций. Если считать, что каждая из них включает три вычисления функции, то видно, что метод менее экономен, чем (4), (5) (где нужно $n(n+1)/2$ вычислений для той же цели). Однако в неквадратичном случае метод работоспособен даже

для плохого начального приближения (если принять меры против вырождения системы p^i), тогда как метод барицентрических координат подобно методу Ньютона требует хорошего начального приближения.

3.4. Метод перебора

Метод перебора или равномерного поиска является простейшим из прямых методов минимизации и состоит в следующем.

Разобьем отрезок $[a; b]$ на N равных частей точками деления $x_i = a + i(b-a)/n,\ \ i = 0,\ldots,n$. Вычислив значения $f(x)$ в точках x_i, путем сравнения найдем точку x_m, $0 \le m \le n$, для которой

$$f(x_m) = \min_{0 \le i \le n} f(x_i). \qquad (1)$$

Далее, положим $x^* \approx x_m,\ \ f^* \approx f(\mathrm{x_m})$.

Замечание:

1 Потребность определения точки минимума x^* функции $f(x)$ методом перебора не превосходит величины $\varepsilon_n = (b-a)/n$.

Предположим, что x_m из (1) является внутренней точкой разбиения отрезка $[a;b]$, т.е. $1 \le m \le n-1$ (случаи $m = 0$ и $m = n$ рассматриваются аналогично). Тогда из соотношения (1) с учетом свойства унимодальных функций следует, что:

а) $f(x_{m-1}) \ge f(x_m)$ т.е. $x^* \in [x_{m-1}; b]$;

б) $f(x_m) \le f(x_{m+1})$ т.е. $x^* \in [a; x_{m+1}]$.

Отсюда получаем, что

$$x^* \in [x_{m-1}; b] \cap [a; x_{m+1}] = [x_{m-1}; x_{m+1}].$$

Длина последнего отрезка равна $2(b-a/n)$, а точка x_m является его серединой. Поэтому $|x_n - x^*| \le (b-a)/n = \varepsilon_n$.

Таким образом, чтобы обеспечить требуемую точность ε определения точки x^*, число отрезков разбиения n необходимо выбрать из условия

$$\varepsilon_n = (b-a)/n \le \varepsilon, \text{ т.е. } n \ge (b-a)/\varepsilon.$$

2. Пусть реализация метода перебора потребовала N вычислений функции $f(x)$. Это означает, что отрезок $[a;b]$ был разбит на $n = N-1$ частей и достигнутая точность определения x^* составила

$$\varepsilon_n = \varepsilon_{N-1} = \frac{b-a}{N-1}.$$

Поэтому точность решения ε*(N),*которую обеспечивает метод перебора в результате *N* вычислений *f(x)* будет

$$\varepsilon(N) = \frac{b-a}{N-1}.$$

Пример. Метод перебора

Решить задачу $f(x) = x^4 + e^{-x} \to \min,\ x \in [0;1]$ с точностью до ε = 0.1

Функция *f*(*x*) унимодальна на отрезке [0;1]. Найдем число *n* отрезков разбиения $n \geq \frac{1-0}{0{,}1} = 10$ т.е., можно взять *n* = 10. Вычислим значения $f(x_i)$, где $x_i = 0{,}1 \cdot i$, $i = 0,...,10$ и запишем их в таблицу 1

Таблица 1

x_i	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$f(x_i)$	1.00	0.90	0.82	0.75	0.70	0.67	0.68	0.74	0.86	1.06	1.37

В этой таблице выделено минимальное из вычисленных значений *f(x).* Таким образом, $x^* = 0{,}5,\ f^* = 0{,}67$.

3.5. Метод поразрядного поиска

Рассмотрим возможности усовершенствования метода перебора с целью уменьшения количества значений *f(x),* которые необходимо находить в процессе минимизации.

Во-первых, если оказывается, что $f(x_i) \leq f(x_{i+1})$, то отпадает необходимость вычислять *f(x)* в точках x_{i+2}, x_{i+3} и т.д., так как $x^* \leq x_{i+1}$ (см. в п.3.4 (1)).

Во-вторых, разумно было бы сначала определить отрезок, содержащий *x**, грубо, т.е. найти точку *x** с небольшой точностью, а затем искать её на этом отрезке с меньшим шагом дискретизации, повышая точность.

Указанные возможности улучшения метода перебора реализованы в методе поразрядного поиска. В этом методе перебор точек отрезка происходит сначала с шагом $\Delta = x_{i+1} - x_i > \varepsilon$ до тех пор, пока не

выполнится условие $f(x_i) \le f(x_{i+1})$ или пока очередная из этих точек не совпадет с концом отрезка. После этого шаг уменьшается (обычно в 4 раза), и перебор точек с новым шагом производится в противоположном направлении до тех пор, пока значения *f(x)* снова не перестанут уменьшаться или очередная точка не совпадёт с другим концом отрезка и т.д. Описанный процесс завершается, когда перебор в данном направлении закончен, а использованный при этом шаг дискретизации не превосходит ε. Приведем описание алгоритма метода поразрядного поиска.

Шаг 1. Выбрать начальный шаг

$$\Delta = \frac{b-a}{4}.$$

Положить $x_0 = a$. Вычислить $f(x_0)$.

Шаг 2. Положить $x_1 = x_0 + \Delta$. Вычислить $f(x_1)$.

Шаг 3. Сравнить $f(x_0)$ и $f(x_1)$. Если $f(x_0) > f(x_1)$, то прейти к шагу 4, иначе - к шагу 5.

Шаг 4. Положить $x_0 = x_1$ и $f(x_0) = f(x_1)$. Проверить условие $x_0 \in (a;b)$. Если $a < x_0 < b$, то перейти к шагу 2, иначе - к шагу 5.

Шаг 5. Проверка на окончание поиска: если $|\Delta| \le \varepsilon$, то вычисление завершить, полагая $x^* = x_0, f^* = f(x_0)$, иначе - перейти к шагу 6.

Шаг 6. Изменение направления и шага поиска: продолжить $x_0 = x_1$, $f(x_0) = f(x_1)$, $\Delta = -\Delta/4$. Перейти к шагу 2.

Пример. Метод поразрядного поиска.

Решить задачу, приведенную в предыдущем примере.

$$f(x) = x^4 + e^{-x} \to \min, \quad x \in [0;\ 1],\ \varepsilon = 0{,}1.$$

Начальный шаг Δ = 1/4 = 0,25. Вычисляя последовательно значения *f(x)* в точках дискретизации с шагом 0,25, получим:

x	0→0 25→0 50→0 75
f(x)	1 000>0 783>0 669<0 789

Так как $f(0,50) < f(0,75)$, причем $|\Delta| > \varepsilon$, то поиск x^* продолжаем из начальной точки $x_0 = 0,75$, изменив его направление и уменьшив шаг в 4 раза:

x	0 4375←0 5000←0 5625←0 6250←0 6875←0 7500
f(x)	0 662>0 669<0 670<0 688<0 726<0789

Так как $|\Delta| = 0,0625 < \varepsilon$, то поиск завершен и $x^* \approx 0,5,\ f^* \approx 0,67$ (сравните с результатом решения предыдущего примера).

3.6. Методы исключения отрезков

В методе перебора, рассмотренном выше, точки x_i, в которых определяются значения $f(x)$, выбираются заранее. Если же для выбора очередной точки вычисления (измерения) $f(x)$ использовать информацию, содержащуюся в уже найденных значениях $f(x)$, то поиск точки минимума можно сделать более эффективным, т.е. сократить число определяемых для этого значений $f(x)$, как, например, в методе поразрядного поиска.

Один из путей такого более эффективного поиска точки x^* указывает свойство 3 унимодальных функций.

Пусть $a < x_1 < x_2 < b$. Сравнить значения $f(x)$ в точках x_1 и x_2 (пробных точках), можно сократить отрезок поиска точки x^*, перейдя к отрезку $[a;x_1]$, если $f(x_1) \le f(x_2)$, или к отрезку $[x_1;b]$, если $f(x_1) > f(x_2)$ (рис. 1).

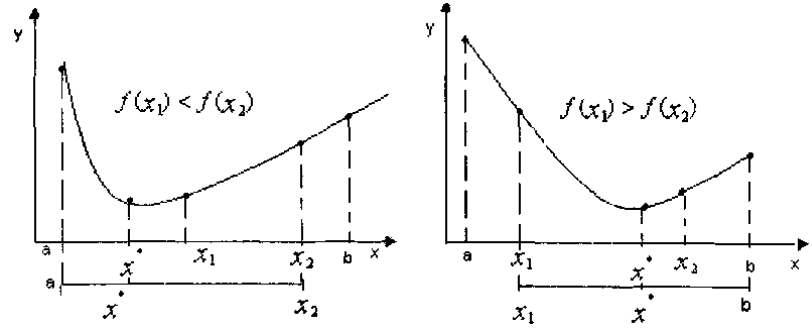


Рис. 1. Уменьшение отрезка поиска точки минимума методами

исключения отрезков

Описанную процедуру можно повторить необходимое число раз, последовательно уменьшая отрезок, содержащий точку минимума. Когда длина последнего из найденных отрезков станет достаточно малой, следует положить $x^* \approx \bar{x}$, где $\bar{x}$ - одна из точек этого отрезка, например, его середина. Методы минимизации, основанные на этом принципе, называются *методами исключения отрезков.*

Чтобы относительное уменьшение отрезка на каждой итерации не зависело от того, какая из его частей исключается из дальнейшего рассмотрения, пробные точки следует располагать симметрично относительно середины исходного отрезка. В зависимости от способа выбора пробных точек получаются различные методы исключения отрезков. На практике используются следующие:

1. Метод дихотомии (первый метод деления отрезка пополам)

Рассмотрим простейший однопараметрический метод безусловной оптимизации – метод дихотомии. Этот метод является методом прямого поиска. В нем при поиске экстремума целевой функции используются только вычисленные значения целевой функции.

В этом методе точки x_1 и x_2 располагаются близко к середине очередного отрезка $[a;b]$, т.е.

$$x_1 = \frac{b+a-\delta}{2},\ x_2 = \frac{b+a+\delta}{2}, \qquad (1)$$

где $\delta > 0$ - малое число. При этом отношение длин нового и исходного отрезков

$$\tau = \frac{b-x_1}{b-a} = \frac{x_2-a}{b-a}$$

близко к 1/2, этим и объясняется название метода.

Отметим, что для любых точек x_1 и x_2 величина $\tau > 1/2$, поэтому указанный выбор пробных точек объясняется стремлением обеспечить максимально возможное относительное уменьшение отрезка на каждой итерации поиска x^*.

В конце вычислений по методу дихотомии в качестве приближенного значения x^* берут середину последнего из

найденных отрезков $[a;b]$, убедившись предварительно, что достигнуто неравенство

$$\frac{b-a}{2} \le \varepsilon .$$

Опишем алгоритм метода деления отрезка пополам.

Шаг 1. Определить x_1 и x_2 по формулам (1). Вычислить $f(x_1)$ и $f(x_2)$.

Шаг 2. Сравнить $f(x_1)$ и $f(x_2)$. Если $f(x_1) \le f(x_2)$, то перейти к отрезку $[a; x_2]$, положив $b = x_2$, иначе - к отрезку $[x_1;b]$, положив $a = x_1$.

Шаг 3. Найти достигнутую точность $\varepsilon_n = \frac{b-a}{2}$. Если $\varepsilon_n > \varepsilon$, то перейти к следующей итерации, вернувшись к шагу 1. Если $\varepsilon_n \le \varepsilon$, то завершить поиск x^*, перейдя к шагу 4.

Шаг 4. Положить $x^* \approx \bar{x} = \frac{a+b}{2}$, $f^* \approx f(\bar{x})$.

Замечание:

1. Число δ из (1) выбирается на интервале $(0; 2\varepsilon)$ с учетом следующих соображений:

а) чем меньше δ, тем больше относительное уменьшение длины отрезка на каждой итерации, т.е. при уменьшении δ достигается более высокая скорость сходимости метода дихотомии;

б) при чрезмерно малом δ сравнение значений $f(x)$ в точках x_1 и x_2, отличающихся на величину δ, становиться затруднительным. Поэтому выбор δ должен быть согласован с точностью определения $f(x)$ и с количеством верных десятичных знаков при задании аргумента x.

2. Число n итераций метода дихотомии, необходимое для определения точки x^* с точностью ε, определяется неравенством

$$n \ge \log_2 \frac{b-a-\delta}{2\varepsilon-\delta}. \qquad (2)$$

Обозначим длину исходного отрезка $[a,b]$ через Δ_0. Длина отрезка, полученного после первой итерации, будет $\Delta_1 = \frac{\Delta_0}{2} + \frac{\delta}{2}$,

после второй итерации $\Delta_2 = \frac{\Delta_1}{2} + \frac{\delta}{2} = \frac{b-a}{4} + \delta(\frac{1}{4} + \frac{1}{2})$,

после третьей $\Delta_3 = \frac{\Delta_2}{2} + \frac{\delta}{2} = \frac{b-a}{8} + \delta(\frac{1}{8} + \frac{1}{4} + \frac{1}{2})$ и т.д.

Таким образом, в результате *n* итераций длина отрезка поиска точки *x** станет

$$\Delta_n = \frac{b-a}{2^n} + (\frac{1}{2^n} + \frac{1}{2^{n-1}} + ... + \frac{1}{2}) = \frac{b-a}{2^n} + (1 + \frac{1}{2^n})\delta.$$

При этом будет достигнута точность определения точки минимума $\varepsilon_n = \frac{\Delta_n}{2}$. Находя *n* из условия

$$\varepsilon_n = \frac{b-a}{2^{n+1}} + (1 - \frac{1}{2^n})\frac{\delta}{2} \le \varepsilon, \qquad (3)$$

получаем неравенство (2).

Величина δ может быть выбрана достаточно малой, поэтому, пренебрегая ею в (2), получаем $\varepsilon_n \approx \frac{b-a}{2^{n+1}}$. На каждой итерации метода дихотомии вычисляются два значения *f(x).* Поэтому после *N* вычислений *f(x)* производится *n = N/*2 итераций и достигается точность определения *x**.

$$\varepsilon(N) = \varepsilon_{\frac{N}{2}} \approx \frac{b-a}{2^{\frac{N}{2}+1}} \qquad (4)$$

Пример. Метод деления отрезка пополам.

Решить задачу, приведенную в двух предыдущих примерах:

$$f(x) = x^4 + e^{-x} \to \min,\ x \in [0,1],\ \varepsilon = 0,1.$$

Выберем δ=0,02.

Итерация 1

Шаг1 . $x_1 = 0,49, x_2 = 0,51, f(x_1) = 0,670, f(x_2) = 0,688.$

Шаг 2. $f(x_1) > f(x_2)$, поэтому полагаем $a = x_1 = 0,49$.

Шаг 3. $(b-a)/2 = 0,225 > 0,1$, т.е. переходим к следующей итерации.

Результат вычисления на остальных итерациях записаны в табл. 1

Таблица 1

Номер итерации	a	b	$\frac{b-a}{2}$	x_1	x_2	$f(x_1)$	$f(x_2)$	Сравнение $f(x_1)$ и $f(x_2)$
2	0.49	1	0.26	0.735	0.755	0.771	0.792	$f(x_1) > f(x_2)$
3	0.49	0.755	0.13	0.613	0.633	0.683	0.691	$f(x_1) > f(x_2)$
4	0.49	0.633	0.07	0.07 < 0.1 - точность достигнута				

Таким образом,

$$x^* \approx \frac{0{,}49 + 0{,}633}{2} \approx 0{,}56, \quad f^* \approx f(0{,}56) \approx 0{,}67$$

(сравнить с результатами решения двух предыдущих примеров).

Пусть дана функция $F(x)$. Необходимо найти $\overline{x}$, доставляющий минимум (или максимум) функции $F(x)$ на интервале $[a,b]$ с заданной точностью ε, т.е. найти

$$\overline{x} = \arg\min F(x),\ \overline{x} \in [a,b].$$

Запишем словесный алгоритм метода.

1) На каждом шаге процесса поиска делим отрезок $[a,b]$ пополам, $x=(a+b)/2$ - координата середины отрезка $[a,b]$.

2) Вычисляем значение функции $F(x)$ в окрестности ±ε вычисленной точки x, т.е.

$$F1 = F(x-\varepsilon),$$
$$F2 = F(x+\varepsilon).$$

3) Сравниваем *F*1 и *F*2 и отбрасываем одну из половинок отрезка [*a*,*b*] (рис. 2).

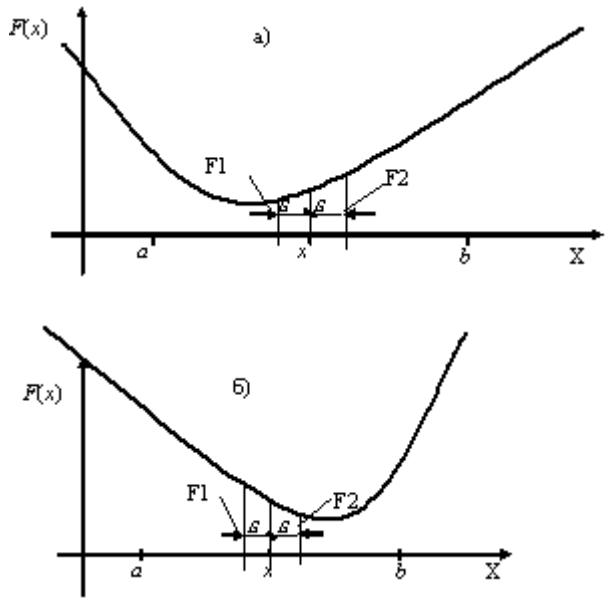


Рис. 2. Поиск экстремума функции *F(x)* методом дихотомии

При поиске минимума:

Если *F*1<*F*2, то отбрасываем отрезок [*x*,*b*], тогда *b*=*x*. (рис. 2,а)

Иначе отбрасываем отрезок [*a*,*x*], тогда *a*=*x*. (рис.2,б)

При поиске максимума:

Если *F*1<*F*2, то отбрасываем отрезок [*a*,*x*], тогда *a*=*x*.

Иначе отбрасываем отрезок [*x*,*b*], тогда *b*=*x*.

4) Деление отрезка [*a*,*b*] продолжается, пока его длина не станет меньше заданной точности ε, т.е. $|b-a| \le \varepsilon$

Схема алгоритма метода дихотомии представлена на рис 3.

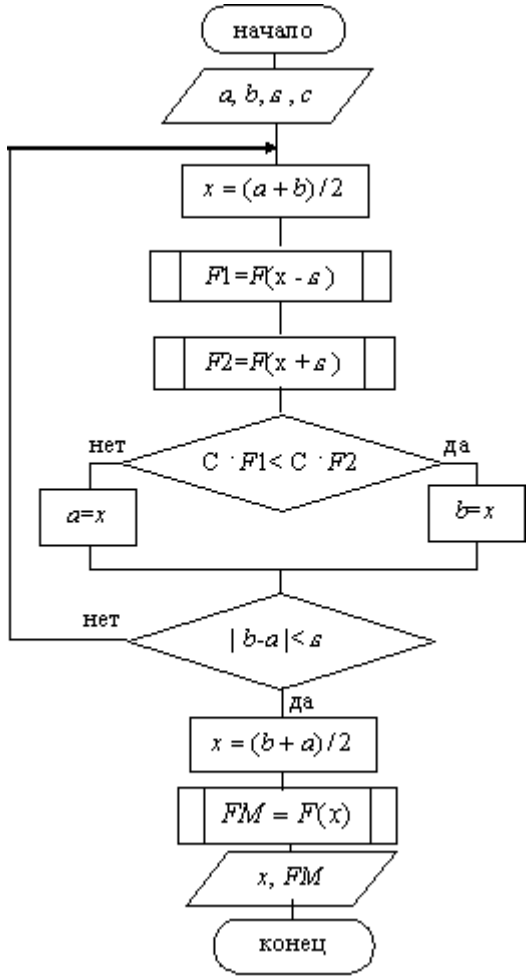


Рис. 3. Схема алгоритма метода дихотомии

На рис 3: *c*- константа,

$$c = \begin{cases} 1 & (\text{поиск минимума функции } F(x)), \\ -1 & (\text{поиск максимума функции } F(x)), \end{cases}$$

При выводе x – координата точки, в которой функция $F(x)$ имеет минимум (или максимум), FM – значение функции $F(x)$ в этой точке.

2. Метод "золотого сечения"

Следующий из методов одномерной оптимизаци называется методом "золотого сечения".

Рассмотрим геометрическую суть метода «золотое сечение».

Числа Фибоначчи и метод золотого сечения

1. Разделим отрезок AB единичной длины (рис. 13) на две части так, чтобы большая из его частей являлась средним пропорциональным между меньшей его частью и всем отрезком.

Рис. 4.

Обозначим для этого искомую длину большей части отрезка через x. Очевидно, длина его меньшей части при этом будет равна $1—x$, и условие задачи даст пропорцию

$$\frac{1}{x} = \frac{x}{1-x}, \qquad (5)$$

откуда

$$x^2 = 1 - x. \qquad (6)$$

Положительным корнем (6) является $\frac{-1+\sqrt{5}}{2}$, так что отношения в пропорции (5) равны

$$\frac{1}{x} = \frac{2}{-1+\sqrt{5}} = \frac{2(1+\sqrt{5})}{(-1+\sqrt{5})(1+\sqrt{5})} = \frac{1+\sqrt{5}}{2} = \alpha$$

каждое. Такое деление (точкой C_1) называется *делением в среднем и крайнем отношении.* Его часто называют также *золотым делением* или *золотым сечением.*

Если взять отрицательный корень уравнения (6), то делящая точка C_2 окажется вне отрезка AB (такого рода деление в геометрии называется

внешним делением), как это видно из рис. 4. Легко показать, что и здесь мы имеем дело с золотым сечением;

$$\frac{C_2B}{AB} = \frac{AB}{C_2A} = \alpha.$$

2. Фактическое построение точки, делящей отрезок золотым сечением, осуществляется без труда.

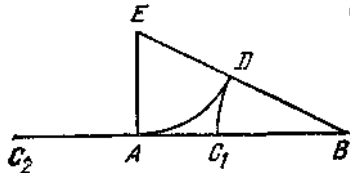


Рис. 5.

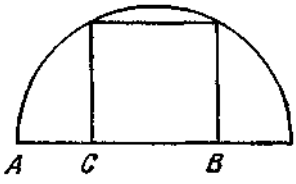


Рис. 6.

Пусть $AB = 1$; восставим из точки A перпендикуляр и возьмем точку E, для которой $AE = \frac{1}{2}$ (рис. 5). Тогда

$$EB = \sqrt{1 + \left(\frac{1}{2}\right)^2} = \frac{\sqrt{5}}{2}.$$

Проведя из E, как из центра, дугу через A до пересечения с EB в точке D, мы получаем

$$BD = \frac{\sqrt{5} - 1}{2}.$$

Наконец, проведя через D дугу с центром в B, мы находим искомую точку C_1. Точку внешнего деления C_2 можно найти из условия $AC_2 = BC_1$.

3. Золотое сечение довольно часто используется в оптимизации. Например, для квадрата, вписанного в полукруг (см. рис. 6), точка C делит золотым сечением отрезок AB.

Сторона a_{10} правильного десятиугольника (рис. 7), вписанного в круг радиуса R, как известно, равна

$$2R \sin \frac{360^\circ}{2 \cdot 10},$$

т. е. $2R$ sinl8°.

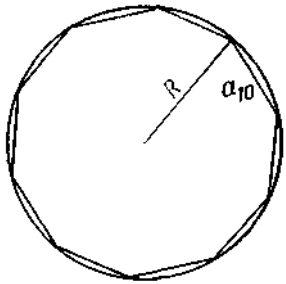


Рис. 7

Вычислим sin 18° На основании известных формул тригонометрии

$$\sin 36° = 2 \sin 18° \cos 18°,$$
$$\cos 36° = 1 - 2 \sin^2 18°,$$

так что

$$\sin 72° = 4 \sin 18° \cos 18° (1 - 2 \sin^2 18°) \qquad (7)$$

Так как $\sin 72° = \cos 18° \neq 0$. из (33) следует, что

$$1 = 4 \sin 18° (1 - 2 \sin^2 18°),$$

и потому sin 18° является одним из корней уравнения

$$1 = 4x (1 - 2x^2),$$

или

$$8x^3 - 4x + 1 = 0.$$

Разложив левую часть последнего уравнения на множители, мы получаем

$$(2x - 1)(4x^2 + 2x - 1) = 0,$$

от куда

$$x_1 = \frac{1}{2}, \quad x_2 = \frac{-1 + \sqrt{5}}{4}, \quad x_3 = \frac{-1 - \sqrt{5}}{4}.$$

Так как sin 18° есть положительное число, отличное от $\frac{1}{2}$

имеем

$$\sin 18° = \frac{\sqrt{5} - 1}{4} = \frac{1}{2\alpha}.$$

Заметим для дальнейшего, что

$$\cos 36° = 1 - 2 \sin^2 18° = 1 - 2 \frac{1}{4\alpha^4} = 1 - \frac{1}{2\alpha^2} = \frac{2\alpha^2 - 1}{2\alpha^2} =$$
$$= \frac{2 + 2\alpha - 1}{2\alpha^2} = \frac{2\alpha + 1}{2\alpha^2} = \frac{\alpha}{2\alpha^2} = \frac{\alpha}{2}.$$

Таким образом,

$$a_{10} = 2R\frac{\sqrt{5}-1}{4} = R\frac{\sqrt{5}-1}{2} = \frac{R}{\alpha}.$$

Иными словами, a_{10} равно большей части радиуса круга, разделенного золотым сечением.

Практически при вычислении a_{10} можно вместо α брать отношение соседних чисел Фибоначчи и считать приближенно, что a_{10} есть $\frac{8}{13}R$ или даже $\frac{5}{8}R$.

4. Рассмотрим правильный пятиугольник. Его диагонали образуют правильный звездчатый пятиугольник (рис. 8),

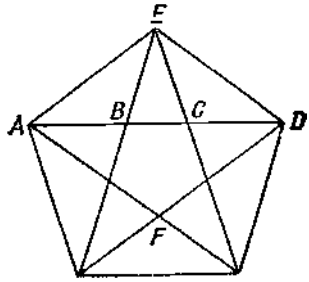


Рис. 8

Угол *AFD* равен 108°, а угол *ADF* равен 36°. Значит, по теореме синусов

$$\frac{AD}{AF} = \frac{\sin 108^\circ}{\sin 36^\circ} = \frac{\sin 72^\circ}{\sin 36^\circ} = 2\cos 36^\circ = 2\frac{1+\sqrt{5}}{4} = \alpha.$$

Так как очевидно, что $AF = AC$, должно быть

$$\frac{AD}{AF} = \frac{AD}{AC} = \alpha,$$

и точка *C* делит отрезок *AD* золотым сечением.

Но тогда, по определению золотого сечения,

$$\frac{AC}{CD} = \alpha.$$

Замечая, что $AB = CD$, мы получаем

$$\frac{AC}{AB} = \frac{AB}{BC} = \alpha.$$

Таким образом, среди отрезков

$$BC,\ AB,\ AC,\ AD$$

каждый последующий в α раз больше предыдущего. Пусть читатель попутно проверит, что и

$$\frac{AD}{AE} = \alpha.$$

5. Возьмем прямоугольник со сторонами a и b и будем вписывать в него наибольшие возможные квадраты, как это показано на рис. 9.

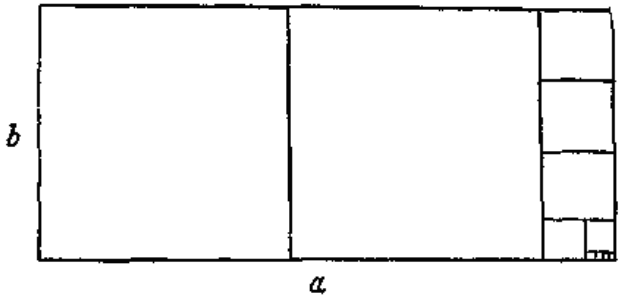


Рис. 9.

Рассуждения показывают, что такой процесс в случае целых a и b соответствует алгоритму Евклида, примененному к этим числам. Числа квадратов одинаковых размеров равны при этом соответствующим неполным частным разложения $\frac{a}{b}$ в непрерывную дробь.

Если разбивать так на квадраты прямоугольник, стороны которого относятся как соседние числа Фибоначчи (рис. 10), то, как известно, все квадраты, кроме двух самых маленьких, будут различными.

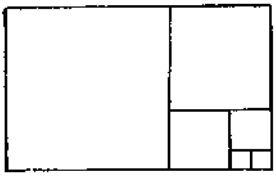


Рис. 10

Так как стороны всех этих квадратов равны соответственно $u_1, u_2, \ldots, u_n$, их суммарная площадь, очевидно, равна

$$u_1^2 + u_2^2 + \ldots + u_n^2.$$

Но это есть площадь разбиваемого нами прямоугольника, равная $u_n u_{n+1}$.

Таким образом, при любом n

$$u_1^2 + u_2^2 + \ldots + u_n^2 = u_n u_{n+1},$$

6. Пусть теперь отношение сторон прямоугольника равно α. (Такие прямоугольники мы будем для краткости называть *прямоугольниками золотого сечения.)* Докажем, что, вписав в прямоугольник золотого сечения наибольший возможный квадрат (рис. 11), мы снова получим прямоугольник золотого сечения.

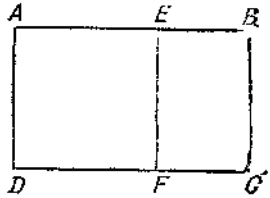


Рис. 11

В самом деле,

$$\frac{AB}{AD} = \alpha;$$

по условию $AD = AE = EF$, так как $AEFD$ — квадрат.

Значит

$$\frac{EF}{EB} = \frac{AB - EB}{EB} = \alpha^2 - 1.$$

Но

$$\alpha^2 - 1 = \alpha,$$

так что

$$\frac{EF}{EB} = \alpha.$$

На рис. 12 показано, как прямоугольник золотого сечения может быть «почти весь» исчерпан квадратами *I, II, III, ...*

Рис. 12

При этом каждый раз после вписывания очередного квадрата будет оставаться фигура, являющаяся прямоугольником золотого сечения.

Заметим, что если в квадрат вписать прямоугольник золотого сечения *I* и квадраты *II* и *III*, как это показано на рис. 13, то оставшийся прямоугольник тоже окажется прямоугольником золотого сечения.

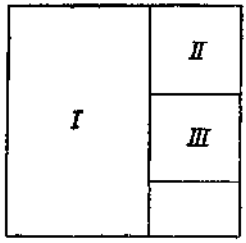


Рис. 13.

7. По аналогии с прямоугольниками золотого сечения можно говорить и о *треугольниках золотого сечения:* остроугольном — с углами 36°, 72° и 72° и тупоугольном — с углами 108°, 36° и 36°. На рис. 14 видно, как остроугольный треугольник золотого сечения разбивается на меньшие три треугольника золотого сечения, и обозначены величины углов и отрезков.

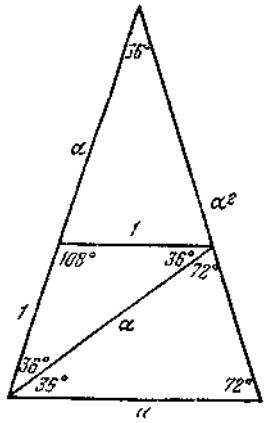


Рис. 14.

8. Природа дает нам многочисленные примеры расположений однородных предметов, описываемых числами Фибоначчи.

В разнообразных спиралевидных расположениях мелких частей растений обычно можно усмотреть два семейства спиралей. В

одном из этих семейств спирали завиваются по часовой стрелке, а в другом — против. Числа спиралей того и другого типов часто оказываются соседними числами Фибоначчи.

Так, взяв молодую сосновую веточку, легко заметить, что хвоинки образуют две спирали, идущих справа снизу налево вверх. Вместе с тем они же составляют три спирали, идущие слева снизу направо вверх.

На многих шишках семена (т. е. «чешуйки») расположены в трех спиралях, полого навивающихся на стержень шишки. Они же расположены в пяти спиралях, круто навивающихся в противоположном направлении. В крупных шишках удается наблюдать 5 и 8 и даже 8 и 13 спиралей. Хорошо заметны такие спирали и на ананасе: обычно их бывает 8 и 13.

У многих сложноцветных (например, у маргаритки или ромашки) заметно спиральное расположение отдельных цветков в соцветиях-корзинках. Число спиралей бывает здесь 13 в одном направлении и 21 в другом или даже соответственно 21 и 34. Особенно много спиралей можно наблюдать в расположении семечек крупного подсолнуха. Их число в каждом из направлений может достигать соответственно 55 и 89.

9. Прямоугольники золотого сечения выглядят «пропорционально» и приятны на вид. Вещами, имеющими такую форму, оказывается удобным пользоваться. Поэтому многим «прямоугольным» предметам нашего обихода (книгам, спичечным коробкам, чемоданам и т. п.) часто придается именно такая форма. Например, данная книга имеет форму прямоугольника с отношением сторон 1,62, а заполненная текстом часть ее страницы — форму прямоугольника с отношением сторон 1,64.

Различными философами-идеалистами древности и средневековья внешняя красота прямоугольников золотого сечения и других фигур, в которых наблюдается деление в среднем и крайнем отношении, возводилась в эстетический и даже философский принцип. Золотым сечением и еще некоторыми числовыми отношениями пытались не только описать, но и объяснить явления природы и даже общественной жизни, а с самим числом α и с его подходящими дробями производились разного рода мистические «операции». Разумеется, подобные «теории» ничего общего с наукой не имеют.

10. Числа Фибоначчи появляются также в вопросах, связанных с исследованием путей в различных геометрических конфигурациях. Рассмотрим, например, сеть путей, изображенную на рис. 15 (такие сети в математике принято называть *ориентированными графами),* и

подсчитаем число путей, которыми можно, двигаясь вдоль стрелок, перейти из вершины A или вершины B в вершину C_n.

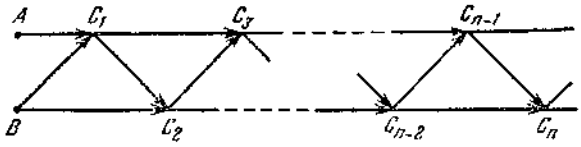


Рис. 15

Обозначим числа таких путей соответственно через a_n и b_n. Ясно, что при начале движения, как из точки A, так и из точки B, в вершину C_n можно попасть двумя способами: через вершину C_{n-1} с последующим шагом вдоль наклонного ребра и через вершину C_{n-2} с последующим шагом вдоль горизонтального ребра. Значит,

$$a_n = a_{n-1} + a_{n-2},$$
$$b_n = b_{n-1} + b_{n-2}.$$

Нам остается заметить, что $a_1 = a_2 = 1$, и $b_1 = 1$, $b_2 = 2$, откуда сразу следует, что $a_n = u_n$ и $b_n = u_{n+1}$.

11. Следующая задача будет касаться уже не подсчета числа путей в ориентированном графе, а выбора рациональных переходов по этим путям.

Рассмотрим следующую игру-состязание в ее традиционной постановке, называемой «цзяньшицзы». Пусть имеются две кучи предметов (например, спичек), и два игрока поочередно берут либо произвольное число предметов из одной кучи, либо поровну из каждой кучи. Выигравшим считается тот, кто забирает последние предметы.

В математизированной форме эту игру можно представить себе, как имеющийся перед игроками ориентированный граф, изображенный на рис. 16.

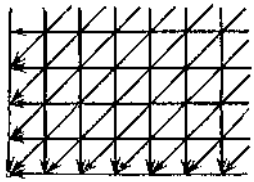


Рис. 16

Будем считать, что граф расположен на координатной плоскости, и две целочисленные координаты каждой из его вершин соответствуют числу предметов в первой и второй кучах (например, жирная точка на рис. 16 имеет координаты (5,3)). Начальное положение игры может

быть отмечено помещением фишки в соответствующую вершину графа. Процесс игры состоит в поочередном уменьшении игроками одной из координат вершины на целое число или обеих координат на одно и то же число, т. е. в прямолинейном передвижении фишки вдоль одного из указанных стрелками на рис. 16 направлений на любое расстояние. Ясно, что за конечное число ходов фишка окажется передвинутой в начало координат (отмеченное на графе кружком), и игрок, поставивший фишку в эту вершину, считается выигравшим. Эту игру будем далее для краткости называть игрой Г, а изображенный на рис. 16 граф — *графом игры* Г. Вершина графа, в которой находится (или может находиться) фишка, вместе с указанием, какой игрок имеет очередь хода, будет называться *позицией игры.*

Применительно к игре Г встает вопрос о тех вершинах графа (позициях игры), приходя в которые тот или иной игрок имеет возможность форсировать выигрыш (а его противник тем самым обречен на поражение). Примем следующую программу исследований этого вопроса.

Во-первых, сформулируем достаточно точно понятие выигрывающей позиции для игры Г (а фактически и для всех игр такого типа).

Во-вторых, сформулируем некоторую схему описания множества всех выигрывающих позиций.

В-третьих, опишем выигрывающие позиции в терминах фибоначчиевых представлений их координат.

Наконец, в-четвертых, мы перейдем от фибоначчиевых представлений координат выигрывающих позиций к их явным описаниям посредством формул.

12. Мы будем называть позицию *выигрывающей,* если игрок, приведший в нее фишку, гарантирует себе выигрыш, независимо от того, как будет вести себя его противник.

Тривиальным примером выигрывающей позиции в игре Г является вершина (0,0). Игрок, приведший фишку в эту позицию, уже выиграл, и никаких действий противника уже не последует.

Простым примером выигрывающей позиции является (1,2). Противник может перейти от нее к одной из позиций (0,2), (1,0) или (0, 1). Во всех трех случаях наш игрок от каждой из этих позиций может перейти к (0,0) и тем самым выиграть. В такой же мере выигрывающей позицией будет и позиция (2, 1).

Более обстоятельно, но не более трудно показывается, что (3, 5) и (5, 3) также суть выигрывающие позиции.

Формально говоря, в приведенное в начале этого пункта определение, а именно в оборот «независимо от того, как будет вести

себя его противник», используется новое понятие: «поведение» игрока. Чтобы дальнейшие рассуждения имели смысл, нам надлежит это понятие точно описать.

Представим себе для этого, что каждый из игроков, прежде чем сесть за игру, составил точный план игры, т. е. наметил ход, который он будет делать в этой позиции, как только он в эту позицию попадет. Такой план принято называть *стратегией* игрока. *Стратегия игрока в игре Г есть таким образом функция, определенная на множестве всех позиций, причем значением ее на данной позиции Р может быть любая позиция, в которую можно из Р перейти.* Как только оба игрока выбрали свои стратегии, все развитие игры уже можно считать предопределенным, в какой бы позиции фишка первоначально ни находилась: тот игрок, чья очередь хода, передвигает ее в соответствии со своей стратегией в некоторую вполне определенную позицию; но в новой позиции очередь хода будет принадлежать другому игроку, который согласно своей стратегии также должен будет сделать вполне определенный ход; после этого снова наступит очередь первого игрока и т. д. В результате фишка будет проходить по графу однозначно определенный путь.

Теперь мы можем уточнить понятие выигрывающей позиции в игре Г: позиция называется *выигрывающей,* если существует такая стратегия пришедшего в нее игрока А, что какова бы ни была стратегия его противника Б, игрок А приведет фишку в позицию (0, 0).

Важно отметить, что достижение игроком выигрывающей позиции еще ни в коей мере не дает ему оснований играть «спустя рукава». Напротив, это означает лишь то, что для него существует некоторая стратегия (ее естественно также назвать *выигрывающей),* которую ему еще предстоит точно установить и неукоснительно соблюдать.

Ясно, что выигрывающая стратегия должна после каждого хода противника снова приводить игру в одну из выигрывающих позиций. В противном случае, если на каком-то ходе игрок придет не в выигрывающую позицию, то у него не окажется выигрывающего продолжения, а это противоречит предположению о том, что выбранная им стратегия — выигрывающая. Таким образом, мы начали с введения «единичного» понятия выигрывающей позиции, но для точного его определения приходится рассматривать и все остальные выигрывающие позиции. Поэтому целесообразно с самого начала говорить одновременно о множестве всех выигрывающих позиций.

13. Рассмотрим некоторое множество позиций R игры на графе Г (или на любом другом ориентированном графе). Оно может обладать (или не обладать) следующими свойствами:

1°. Любой ход в позиции, принадлежащей *R,* выводит за пределы *R.* Это свойство множества позиций в игре (и в графе) называется его *внутренней устойчивостью.*

2°. В любой позиции, не принадлежащей *R,* существует ход, приводящий в позицию из *R.* Это свойство *R* называется его *внешней устойчивостью.*

Множества позиций в играх на ориентированных графах, которые являются одновременно внутренне и внешне устойчивыми, имеют большое значение в играх, связанных с поочередными перемещениями по вершинам графа. Такие множества называются *решениями* этих игр (а также решениями графов этих игр). Если фишка в ходе игры оказывается в принадлежащей решению позиции, то игрок, чья очередь хода, обречен все последующие ходы «пытаться уйти из решения»: какой бы ход он ни сделал, по свойству внутренней устойчивости он выведет фишку за пределы решения; но тогда по свойству внешней устойчивости его противник сумеет следующим ходом фишку в решение вернуть.

В рассматриваемой нами игре Г всякая партия заканчивается приведением фишки в начало координат, и игрок, приведший ее туда, выигрывает. Значит, если решение игры содержит начало координат, то игрок, имеющий очередь хода в одной из принадлежащих этому решению позиций, выигрывает. Следовательно, это решение состоит из выигрывающих позиций.

Все сказанное дает нам основание исследовать достаточно подробно вопросы, касающиеся решений игры, содержащих начало координат.

14. Прежде всего установим единственность такого решения.

Лемма. *Для игры* Г *существует не более одного решения, содержащего начало координат.*

Доказательство. Предположим, что, вопреки утверждению леммы, имеется два таких решения, *R* и *S,* причем некоторая позиция s_1 из S не принадлежит *R.* По внешней устойчивости *R* из позиции s_1 можно перейти в некоторую позицию r_1 из *R.* Но по внутренней устойчивости *S* позиция r_1 не может принадлежать *S*. Значит, по внешней устойчивости *S* мы можем из r_1 перейти в некоторую позицию s_2 из *S,* которая (в силу внутренней устойчивости *R)* не может принадлежать *R.* Повторяя этот процесс достаточно долго, мы получим последовательность позиций s_1, r_1, s_2, r_2, ..., которая заканчивается началом координат и в которой каждая позиция принадлежит лишь одному из решений *R* или S. Значит, и начало координат должно принадлежать только *R* или только *S,* и мы получили противоречие.

15. Определяющее, «характеристическое» свойство решения игры R, содержащего начало координат, описывается следующей теоремой.

Теорема. *Пусть R — множество позиций в игре* Г, *которое обладает следующими свойствами:*

1) *позиция* (0,0) *принадлежит R;*

2) *если (a, b) принадлежит R, то и (b, a) принадлежит R;*

3) *для всякого натурального a найдется ровно одно натуральное b, для которого (a, b) принадлежит R;*

4) *для всякого натурального d найдется ровно одна пара чисел (a, b) из R, для которой $a — b = d$;*

5) *если позиции (a, b) и (k, l) принадлежат R, причем $a<b$, $k<l$ и $b — a < l — k$, то $a < k$ и $b <l$.*

Тогда множество R является решением игры Г.

Доказательство. Заметим сначала, что, как следует из 3), каждое натуральное число является координатой ровно в одной симметричной паре (свойство 2)) позиций из R.

Перейдем к установлению свойств внутренней и внешней устойчивости множества R.

а) ***Внутренняя устойчивость***. Пусть (a, b) принадлежит R. Если уменьшить a или b, то возникает пара, сочетающая с b (соответственно с a) другое число, и потому по 3) не принадлежащая R. Если же уменьшить одновременно и одинаково a и b, то получится отличающаяся от (a, b) пара с той же разностью координат и не могущая поэтому в силу 4) принадлежать R.

б) ***Внешняя устойчивость***. Пусть (a, b) не принадлежит R. Если $a = b$, то от этой вершины можно перейти к вершине (0,0), которая по 1) принадлежит R.

Если $a \neq b$, то по 3) найдется такое c, что (a, c) принадлежит R, а по 4) найдутся такие k и l, что $l — k = b — a$ и (k, l) принадлежит R. Тогда при $c < b$ от (a, b) можно перейти к (a, c), уменьшив b, а при $c > b$ имеет место $c — a > b — a = l = k$, так что по 5) должно быть $c > l$ и $a > k$, и уменьшение каждой из координат позиции (a, b) на $a — k = b — l$ дает нам позицию (k, l).

Двойная устойчивость установлена, и R оказывается решением.

16. Теперь нетрудно построить некоторый развертывающийся (а по сути дела — рекуррентный) процесс, порождающий позиции из решения R игры Г, содержащего (0,0).

Начнем с позиции (0,0), а затем, уже выписав набор позиций

$$(a_1, b_1), \ldots, (a_n, b_n),$$
$$(0, 0), \qquad (8)$$
$$(b_1, a_1), \ldots, (b_n, a_n),$$

где $a_i<b_i$ для $i = 1.....n,$ положим a_{n+1} равным наименьшему из чисел, не участвовавших в наборах (8), и $b_{n+1} = a_{n+1} + (n + 1)$.

Фактически этот процесс приводит к системе позиций

(1,2), (3,5), (4,7), (6, 10), (8, 13), (9, 15), ...

(0, 0),

(2, 1), (5,3), (7,4), (10,6), (13,8), (15,9), ..

Позиции, составляющие это множество, расположены «почти» на двух лучах, как эго видно из рис. 17.

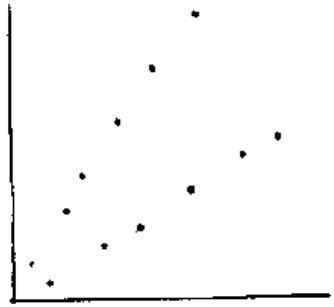


Рис. 17

Непосредственно из описанного построения видно, что полученная система позиций удовлетворяет условиям 1)—5) из доказанной в предыдущем пункте теоремы. Следовательно, она является решением игры, а в соответствии с п. 14 — и единственным решением. Заметим, что игра Г имеет еще решения, не содержащие позиции (0,0). Однако это обстоятельство нас интересовать не будет.

В принципе поставленная нами задача отыскания решения игры Г тем самым решена. Однако множество *R,* хотя и определено у нас однозначно, но имеет плохо обозримый вид. Изобразим его иначе.

17. Пусть Ф(*t*) обозначает фибоначчиево представление натурального числа *t.* Можно считать, что последними фибопаччиевыми цифрами представления каждого из чисел является некоторое количество нулей (если этих нулей нет, то их число, очевидно, равно нулю). Разделим все целые положительные числа на два класса: имеющие в конце своего фибоначчиевого представления четное или нечетное число нулей. Очевидно, каждое число из второго класса может быть получено ровно из одного числа первого класса приписыванием к его фибоначчиеву представлению одного нуля справа. Тем самым натуральные числа объединяются в пары. Покажем,

что множество всех таких пар *(a, b)* (и симметричных им пар *(b,a))* вместе с парой (0,0) удовлетворяют условиям теоремы из п. 15 и тем самым образуют решение игры Г.

Условия 1)—3) выполняются очевидным образом.

Рассмотрим разности сконструированных нами пар и покажем, что каждое значение разности *d* встречается ровно один раз. Далее, пользуясь фибопаччиевыми представлениями чисел, мы будем для удобства нумеровать фибоначчиевы цифры от низших разрядов к высшим, т. е. записывать фибоначчиево представление числа в виде $\varphi_n \varphi_{n-1} \ldots \varphi_2$ (где, естественно, φ_2 есть коэффициент при u_2).

Если фибоначчиеио представление

$$\Phi(d) = \varphi_{n-1} \ldots \varphi_2 \tag{9}$$

оканчивается нечетным числом нулей, то возьмем *a* и *b* с фибоначчиевыми представлениями

$$\begin{aligned} \Phi(a) &= \varphi_l \ldots \varphi_2 0, \\ \Phi(b) &= \varphi_l \ldots \varphi_2 00. \end{aligned} \tag{10}$$

Мы имеем

$$\begin{aligned} b - a &= (\varphi_l u_{l+2} + \ldots + \varphi_2 u_4) - (\varphi_l u_{l+1} + \ldots + \varphi_2 u_3) = \\ &= \varphi_l (u_{l+2} - u_{l+1}) + \ldots + \varphi_2 (u_4 - u_3) = \\ &= \varphi_l u_l + \ldots + \varphi_2 u_2 = d. \end{aligned}$$

Если же $\Phi(d)$ оканчивается четным числом нулей:

$$\varphi_2 = \varphi_3 = \ldots = \varphi_{2m+1} = 0 \quad \text{и} \quad \varphi_{2m+2} = 1 \ (m \geqq 1),$$

то возьмем

$$\begin{aligned} a &= \varphi_l \ldots \varphi_{2m+3} \underbrace{0101 \ldots 01}_{m+1 \text{ раз по } 01}, \\ b &= \varphi_l \ldots \varphi_{2m+3} \underbrace{0101 \ldots 010}_{m+1 \text{ раз по } 01}, \end{aligned} \tag{11}$$

и подсчитаем

$$\begin{aligned} b - a &= \varphi_l u_{l+2} + \ldots + \varphi_{2m+3} u_{2m+5} + u_{2m+3} + \ldots + u_1 - \\ &- (\varphi_l u_{l+1} + \ldots + \varphi_{2m+3} u_{2m+4} + u_{2m+2} + \ldots + u_2) = \\ &= \varphi_l u_l + \ldots + \varphi_{2m+3} u_{2m+3} + u_{2m+1} + \ldots + u_1, \end{aligned}$$

или,

$$b - a = \varphi_l u_l + \ldots + \varphi_{2m+3} u_{2m+3} + u_{2m+2} = d.$$

Проверим единственность пары с заданной разностью *d*.

Если $\Phi(d)$ имеет в конце нечетное число нулей, то при другой паре *(a, b)* и фибоначчиевы представления этих чисел (10) были бы иными; но тогда и $\Phi(d)$ было бы иным, а в силу единственности фибоначчиева представления иным было бы и *d*.

Случай, когда $\Phi(d)$ имеет в конце четное число нулей, по существу, столь же прост, хотя и требует для своего анализа некоторых подсчетов.

Пусть представление

$$\Phi(d) = \mu_k \ldots \mu_2$$

оканчивается ровно $2m$ нулями:

$$\mu_2 = \ldots = \mu_{2m+1} = 0, \quad \mu_{2m+2} = 1, \quad \mu_{2m+3} = 0,$$

и мы имеем

$$d = \mu_k u_k + \ldots + \mu_{2m+4} u_{2m+4} + u_{2m+2}. \qquad (12)$$

Представим d в виде разности $b - a$, где $\Phi(a)$ имеет в конце четное число нулей, а $\Phi(b)$ получается из $\Phi(a)$ путем приписывания к $\Phi(a)$ еще одного нуля справа.

Пусть

$$a = \varphi_l u_l + \varphi_{l-1} u_{l-1} + \ldots + \varphi_3 u_3 + \varphi_2 u_2,$$
$$b = \varphi_l u_{l+1} + \varphi_{l-1} u_l + \ldots + \varphi_3 u_4 + \varphi_2 u_3.$$

Тогда

$$d = b - a = \varphi_l u_{l-1} + \varphi_{l-1} u_{l-2} + \ldots + \varphi_3 u_2 + \varphi_2 u_1. \qquad (13)$$

Если при этом $\varphi_2 = 0$, то $\varphi_l \varphi_{l-1} \ldots \varphi_3$ является фибоначчиевым представлением d, и по единственности такого представления его цифры должны совпадать с цифрами из (12). В том числе должно быть

$$\varphi_2 = \varphi_3 = \ldots = \varphi_{2m+2} = 0, \quad \varphi_{2m+3} = 1,$$

т. е. фибоначчиево представление числа a оканчивается нечетным числом нулей, что противоречит выбору чисел a и b. Значит, $\varphi_2 = 1$. Но тогда

$$d - 1 = \varphi_l u_{l-1} + \varphi_{l-1} u_{l-2} + \ldots + \varphi_3 u_2$$

и

$$\Phi(d-1) = \varphi_l \varphi_{l-1} \ldots \varphi_3, \qquad (14)$$

а с другой стороны, из (12) следует, что

$$d - 1 = \mu_k u_k + \ldots + \mu_{2m+4} u_{2m+4} + u_{2m+2} - 1 =$$
$$= \mu_k u_k + \ldots + \mu_{2m+4} u_{2m+4} + u_{2m+1} + u_{2m-1} + \ldots + u_3,$$

так что

$$\Phi(d-1) = \mu_k \ldots \mu_{2m+4} 00101 \ldots 010.$$

В силу единственности фибоначчиева представления вместе с (14) это дает

$$k = l - 1, \quad \mu_k = \varphi_l, \ldots, \mu_{2m+4} = \varphi_{2m+5}, \quad \varphi_{2m+4} = 0,$$
$$\varphi_{2m+3} = \varphi_{2m+1} = \ldots = \varphi_3 = 0,$$
$$\varphi_{2m+2} = \varphi_{2m} = \ldots = \varphi_4 = 1.$$

Кроме того, как уже указывалось, $\varphi_2 = 1$. Следовательно, *a,* а потому и *b,* обязаны иметь вид из формулы (11).

Это значит, что соблюдается условие 4).

Наконец, ясно, что в условиях выполненного построения с ростом разности координат позиции должны возрастать и сами координаты. Это значит, что выполняется 5).

Таким образом, построенная система пар чисел является решением игры Г, содержащим (0,0). Ввиду доказанной единственности такого решения она должна совпадать с результатом построения из п. 16.

Фибоначчиевы представления чисел позволяют для каждого натурального числа непосредственно указывать «парное» ему. Найдем, например, «пару» к числу 31. Для этого числа мы имеем Ф(31)=1010010. Полученное представление оканчивается одним нулем. Значит, представление, парное к нему, получается в результате отбрасывания последнего нуля, т. е. будет 101001; оно является фибоначчиевым представлением числа 13 + 5+ 1 = 19.

18. Попытаемся изгнать из описания полученного решения игры Г последние заключенные в фибоначчиевых представлениях остатки рекуррентности. Как и следует ожидать, это будет связано с использованием числа $\alpha = \frac{1+\sqrt{5}}{2}$.

Предварительно докажем вспомогательную лемму.

Лемма. *Пусть* γ *и* δ *— положительные иррациональные числа, для которых*

$$\frac{1}{\gamma} + \frac{1}{\delta} = 1. \qquad (15)$$

Тогда среди чисел

$$a_n = [n\gamma], \quad n = 1, 2, \ldots,$$
$$b_n = [n\delta], \quad n = 1, 2, \ldots \qquad (16)$$

(где квадратные скобки означают целую часть стоящих внутри них чисел) *любое натуральное числа встретится ровно по разу.*

Доказательство. Заметим прежде всего, что $\gamma, \delta > 1$. Возьмем далее произвольное натуральное *N* и рассмотрим все натуральные значения *n,* для которых $[n\gamma] < N$, т. е. $n\gamma < N$, или $n < \frac{N}{\gamma}$, так что этому неравенству удовлетворяют все натуральные числа

$$n = 1, 2, \ldots, \left[\frac{N}{\gamma}\right].$$

Аналогично для всех

$$n = 1,\ 2,\ \ldots,\ \left[\frac{N}{\delta}\right]$$

будет $[n\delta] < N$. Значит, среди чисел 1, 2, ..., N будет всего

$$\left[\frac{N}{\gamma}\right] + \left[\frac{N}{\delta}\right]$$

чисел вида 16). Но числа

$$\frac{N}{\gamma} \quad \text{и} \quad \frac{N}{\delta}$$

неявляются целыми. Значит,

$$\frac{N}{\gamma} - 1 < \left[\frac{N}{\gamma}\right] < \frac{N}{\gamma}, \quad \frac{N}{\delta} - 1 < \left[\frac{N}{\delta}\right] < \frac{N}{\delta}.$$

Поэтому, складывая эти неравенства и учитывая (15), мы получаем

$$N\left(\frac{1}{\gamma} + \frac{1}{\delta}\right) - 2 =$$

$$= N - 2 < \left[\frac{N}{\gamma}\right] + \left[\frac{N}{\delta}\right] < N\left(\frac{1}{\gamma} + \frac{1}{\delta}\right) = N.$$

Значит, средняя часть написанного соотношения есть целое число, лежащее строго между N — 2 и N. Таким числом является N— 1:

$$\left[\frac{N}{\gamma}\right] + \left[\frac{N}{\delta}\right] = N - 1.$$

Таким образом, для любого натурального N среди чисел, меньших N, будет ровно N—1 чисел вида (16): ими будут все натуральные числа, меньше N. Нам остается сослаться на произвольность числа N.

19. Пары чисел ($[n\gamma]$, ($[n\gamma]$) удовлетворяют условиям 1)—3) теоремы п. 15. Чтобы они удовлетворяли также условиям 4) и 5) этой теоремы, нужно, чтобы при любом натуральном n было

$$b_n - a_n = [n\delta] - [n\gamma] = n.$$

Но так как

$$n = n + [n\gamma] - [n\gamma] = [n + n\gamma] - [n\gamma] = [n(1 + \gamma)] - [n\gamma],$$

это равносильно тому, чтобы при любом n было $[n\delta] = [n(1 + \gamma)]$. Но, как легко проверить, последнее возможно лишь при $\delta=1+\gamma$. т. е. ввиду (15) — при

$$\frac{1}{\gamma} + \frac{1}{\gamma + 1} = 1,$$

откуда

$$2\gamma + 1 = \gamma(\gamma + 1),$$

или

$$\gamma^2 - \gamma - 1 = 0,$$

так что ввиду положительности γ —

$$\gamma = \frac{1+\sqrt{5}}{2} = \alpha,$$

и

$$\delta = 1 + \alpha = \frac{3+\sqrt{5}}{2}.$$

Таким образом, координаты выигрывающих позиций в игре Г поддаются непосредственному вычислению как пары

$$\left(\left[n\frac{1+\sqrt{5}}{2}\right], \left[n\frac{3+\sqrt{5}}{2}\right]\right).$$

20. Закончим наше изложение небольшой геометрической шуткой. Сейчас мы наглядно «докажем», что 64=65. Возьмем для этого квадрат со стороной 8 и разрежем его на четыре части, как это показано на рис. 18. Эти части мы сложим в прямоугольник (рис. 19) со сторонами 13 и 5, т. е. с площадью, равной 65.

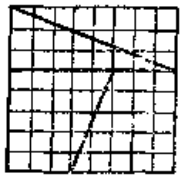


Рис. 18.

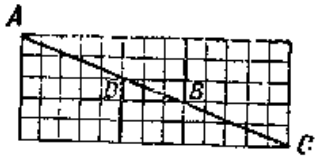


Рис19.

Объяснение этому, на первый взгляд загадочному, явлению найти нетрудно. Все дело в том, что точки *A, B, C* и *D* на рис. 19 на самом деле не лежат на одной прямой, а являются вершинами параллелограмма, площадь которого как раз и равна «лишней» единице.

Это правдоподобное, но неверное «доказательство» заведомо ложного высказывания (такие «доказательства» называются софизмами), можно проделать еще более наглядно и «убедительно», если вместо квадрата со стороной 8 взять квадрат со стороной, равной некоторому числу Фибоначчи с достаточно большим четным номером, u_{2n}. Разобьем этот квадрат на части (рис. 20) и сложим из этих частей прямоугольник (рис. 21). «Пустота» в виде параллелограмма, вытянутого вдоль диагонали прямоугольника имеет площадь, равную единице.

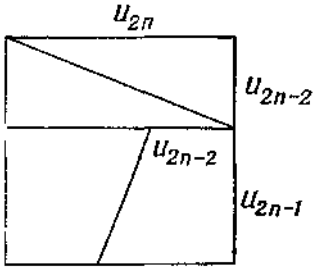


Рис. 20.

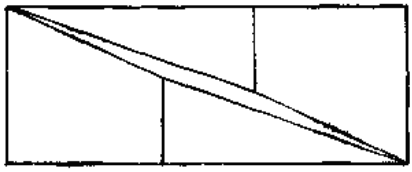


Рис. 21.

Наибольшая ширина этой щели, т. е. высота параллелограмма, равна, как легко вычислить,

$$\frac{1}{\sqrt{u_{2n}^2 + u_{2n-2}^2}}.$$

Поэтому если мы возьмем квадрат со стороной 21 см и «превратим» его в прямоугольник со сторонами 34 и 13 см, то наибольшая ширина щели получится

$$\frac{1}{\sqrt{21^2 + 8^2}} \text{ см},$$

т. е. около 0,4 мм, что почти незаметно для глаза.

Рассмотрим метод золотого сечения, который используется при решении оптимизационных задач нулевого порядка.

Рассмотрим такое симметричное расположение точек x_1 и x_2 на отрезке $[a;b]$, при котором одна из них становиться пробной точкой и на новом отрезке, полученном после исключения части исходного отрезка. Использование таких точек позволяет на каждой итерации метода исключения отрезков, кроме первой, ограничиться определением только одного значения $f(x)$, так как другое значение уже найдено из одной из предыдущих итераций.

Найдем точки x_1 и x_2 обладающие указанным свойством.

Рассмотрим сначала отрезок [0;1] и для определенности предположим, что при его уменьшении исключается правая часть этого отрезка. Пусть $x_2 = \tau$, тогда симметрично расположенная точка $x_1 = 1 - \tau$ (рис. 22)

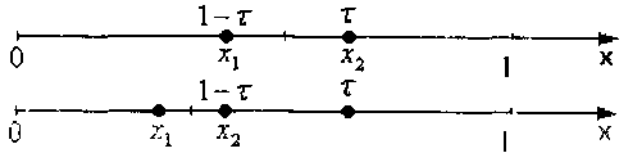


Рис. 22. К определению пробных точек в методе золотого сечения

Пробная точка x_1, отрезка [0; l] перейдет в пробную точку $x'_2 = 1 - \tau$ нового отрезка [0; г]. Чтобы точки $x_2 = \tau$ и $x'_2 = 1 - \tau$ делили отрезки и в одном и том же соотношении, должно выполнятся равенство $\frac{1}{\tau} = \frac{\tau}{1-\tau}$ или $\tau^2 = 1 - \tau$, откуда находим положительное значение $\tau = \frac{\sqrt{5}-1}{2} = 0{,}61803\ldots$

Таким образом, $x_1 = 1 - \tau = \frac{3-\sqrt{5}}{2}$, $x_1 = \tau = \frac{\sqrt{5}-1}{2}$.

Для произвольного отрезка выражения для пробных точек примут вид

$$x_1 = a + \frac{3-\sqrt{5}}{2}(b-a);\ x_2 = a + \frac{\sqrt{5}-1}{2}(b-a). \quad (17)$$

Замечания:

1 Точки x_1 и x_2 из (40) обладают следующим свойством: каждая из них делит отрезок [a; b] на две части так, что отношение длинны всего отрезка к длине его большей части равно отношению длин большей и меньшей частей отрезка. Точки с таким свойством называются *точками золотого сечения* отрезка [a ; b]. Это и объясняет название рассматриваемого метода.

2 На каждой итерации исключения отрезков с пробными точками (17) одна из них $\overline{x}$ переходит на следующий отрезок и значение $f(x)$ в этой точке вычислять не следует. Если новым отрезком становится [a; x_2], то на него переходит пробная точка $\overline{x} = x_1$ исходного отрезка, становясь его второй пробной точкой ($x'_2 = x_1$) (рис. 22). В случае перехода к отрезку [x_1;b] пробная точка $\overline{x} = x_2$ исходного отрезка становится первой пробной точкой отрезка [x_1;b].

3 Легко проверить, что $x_1 = a + b - x_2$ и $x_2 = a + b - x_1$. Поэтому на каждой итерации метода золотого сечения недостающую пробную

точку нового отрезка можно найти по перешедшей на него пробной точке с помощью сложения и вычитания, не используя формул (17).

4 В конце вычислений по методу золотого сечения в качестве приближенного значения x^* можно взять середину последнего из последних полученных отрезков $\bar{x} = \frac{a+b}{2}$.

На каждой итерации отрезок поиска точки минимума уменьшается в одном и том же отношении $\tau = \frac{\sqrt{5}-1}{2}$, поэтому в результате n итераций его длина становится $\Delta_n = \tau^n (b-a)$. Таким образом, точность ε_n определения точки x^* после n итераций находится из равенства

$$\varepsilon_n = \frac{\Delta_n}{2} = \frac{1}{2}\left(\frac{\sqrt{5}-1}{2}\right)^n (b-a), \qquad (18)$$

а условием окончания поиска точки x^* с точностью ε служит неравенство

$$\varepsilon_n \le \varepsilon .$$

Опишем алгоритм метода золотого сечения.

Шаг 1. Найти x_1 и x_2 по формулам (17). Вычислить $f(x_1)$ и $f(x_2)$. Положить $\tau = \frac{\sqrt{5}-1}{2}$, $\varepsilon_n = \frac{b-a}{2}$.

Шаг 2. Проверка на окончание поиска: если $\varepsilon_n > \varepsilon$, то перейти к шагу 3, иначе - к шагу 4.

Шаг 3. Переход к новому отрезку и новым пробным точкам. Если $f(x_1) \le f(x_2)$, то положить $b=x_2$, $x_2=x_1$, $f(x_2)=f(x_1)$, $x_1 = b-\tau(b-a)$ и вычислить $f(x_1)$, иначе - положить $a=x_1$, $x_1=x_2$, $f(x_1)=f(x_2)$, $x_2=a+\tau(b-a)$ и вычислить $f(x_2)$.

Положить $\varepsilon_n = \tau\varepsilon_n$ и перейти к шагу 2.

Шаг 4. Окончание поиска: положить

$$x^* \approx \bar{x} = \frac{a+b}{2}, \quad f^* \approx f(\bar{x}).$$

Пример. Метод золотого сечения

Решить задачу приведенную ранее

$$f(x) = x^4 + e^{-x} \to \min,$$

$x \in [0;1]$, $\varepsilon = 0{,}1$.

Итерация 1

Шаг 1. Находим:

$x_1 = 0{,}382$, $x_2 = 0{,}618$, $f(x_1) = 0{,}704$. $f(x_2) = 0{,}685$, $\varepsilon_n = 0{,}5$.

Шаг 2. $\varepsilon_n = 0{,}5 > \varepsilon = 0{,}1$, поэтому переходим к шагу 3.

Шаг 3. $f(x_1) > f(x_2)$, поэтому полагаем $a = 0{,}382$, $x_1 = -0{,}618$, $f(x_1) = 0{,}685$, $x_2 = 0{,}764$, $\varepsilon_n = 0{,}309$ и вычисляем $f(x_2) = 0.807$. Переходим к следующей итераиии, начиная с шага 2.

Результаты вычислений на остальных итерациях представлены в табл. 2.

Таблица 2

Номер итерации	a	b	ε_n	x_1	x_2	$f(x_1)$	$f(x_2)$	Сравнение $f(x_1)$ и $f(x_2)$
2	0,382	1,000	0,309	0.764	0,764	0,685	0,807	$f(x_1) < f(x_2)$
3	0,382	0,764	0,191	0.528	0,618	0,668	0,685	$f(x_1) < f(x_2)$
4	0382	0,618	0,118	0.472	0,528	0,673	0,668	$f(x_1) < f(x_2)$
5	0,472	0,618	0,073	0,073<0,1- точность достигнута				

Таким образом,

$$x^* \approx \frac{0{,}472 + 0{,}618}{2} \approx 0{,}55, \quad f^* \approx f(0{,}55) = 0{,}67$$

Замечание. Число итераций, необходимое для достижения заданной точности ε, можно найти из условия $\varepsilon_n \le \varepsilon$ с учетом соотношения (18)

$$n \ge \ln\left(\frac{2\varepsilon}{b-a}\right) \Big/ \ln\tau \approx -2{,}1 \ln\left(\frac{2\varepsilon}{b-a}\right).$$

Так как N вычислений $f(x)$ позволяют выполнить N-1 итераций

метода золотого сечения, то достигнутая в результате этих вычислений точность определения x^* составляет

$$\varepsilon(N) = \varepsilon_{N-1} = \frac{1}{2}\left(\frac{\sqrt{5}-1}{2}\right)^{N-1}(b-a). \qquad (19)$$

При решении задач оптимизации однопараметрических функций не всегда можно заранее определить, сколько раз придется вычислять функцию. Метод "золотого сечения" достаточно эффективен, так как при этом не требуется знать n - количество вычислений функции, определяемое вначале. После того как выполнено j вычислений, исходя из тех же соображений, что и ранее, записываем

$$L_{j-1} = L_j + L_{j+1} \qquad (20)$$

Однако если n не известно, то мы не можем использовать условие $L_{n-1}=L_n-e$. Если отношение последующих интервалов будет постоянным, т.е.

$$\frac{L_{j-1}}{L_j} = \frac{L_j}{L_{j+1}} = \frac{L_{j+1}}{L_{j+2}} = \ldots = \tau, \qquad (21)$$

то

$$\frac{L_{j-1}}{L_j} = 1 + \frac{L_{j+1}}{L_j},$$

т.е.

$$\tau = 1 + 1/\tau$$

Таким образом,

$$\tau^2 - \tau - 1 = 0,$$

откуда

$$\tau = (1+\sqrt{5})/2 \approx 1,618033989.$$

Тогда

$$\frac{L_{j-1}}{L_{j+1}} = \tau^2, \quad \frac{L_{j-2}}{L_{j+1}} = \tau^3 \text{ и т.д.}$$

Следовательно,

$$\frac{L_1}{L_n} = \tau^{n-1},$$

т.е.

$$L_n = \frac{L_1}{\tau^{n-1}} \tag{22}$$

В результате анализа двух рассмотренных значений функции будет определен тот интервал, который должен исследоваться в дальнейшем. Этот интервал будет содержать одну из предыдущих точек и следующую точку, помещаемую симметрично ей. Первая точка находится на расстоянии L_1/t от одного конца интервала, вторая - на таком же расстоянии от другого. Поскольку

$$\lim_{n\to\infty} F_{n-1}/F_n = 1/n,$$

то видно, что поиск методом "золотого сечения" является предельной формой поиска методом Фибоначчи.

Таким образом, если ищется интервал (x_0, x_3) и имеются два значения функции f_1 и f_2 в точках x_1 и x_2, то следует рассмотреть два случая (рис. 23).

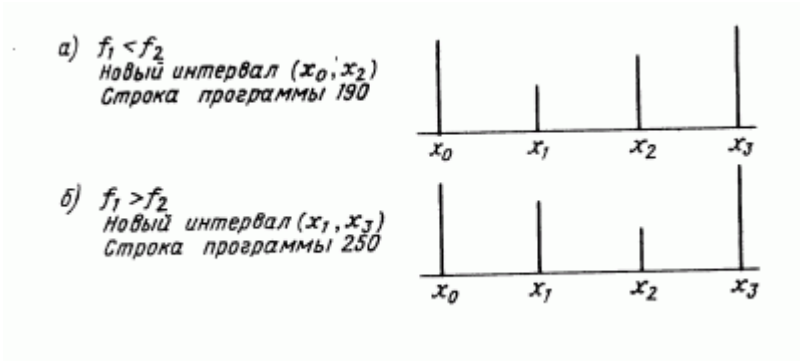


Рис. 23.

Метод гарантирует нахождение минимума в самых неблагоприятных условиях, однако он обладает медленной сходимостью.

Схема алгоритма метода "золотого сечения" представлена на рис. 24

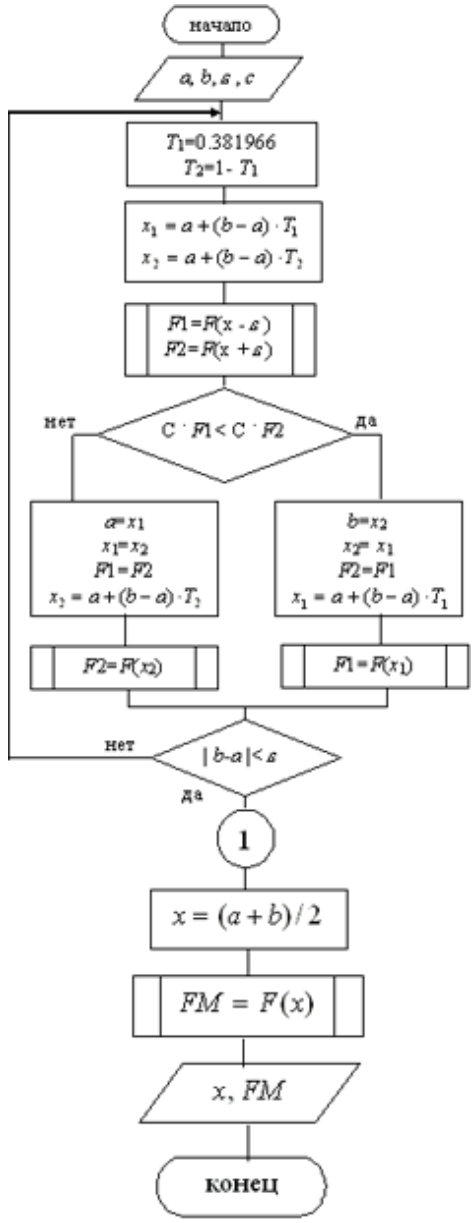


Рис. 24. Схема алгоритма метода "золотого сечения".

Здесь *c*- константа,

$$c = \begin{cases} 1 & (\text{поиск минимума функции } F(x)), \\ -1 & (\text{поиск максимума функции } F(x)), \end{cases}$$

При выводе *x* - координата точки, в которой функция *F(x)* имеет минимум (или максимум), *FM* – значение функции *F(x)* в этой точке.

Второй метод деления отрезка пополам. Этот метод, использующий на каждой итерации три пробные точки, обеспечивает последовательное уменьшение длины отрезка, содержащего *x*,* ровно вдвое.

Рассмотрим способ исключения отрезков, применяемый в рассматриваемом методе.

Разделим отрезок [*a;b*] на четыре равные части пробными точками $x_i = a + \frac{b-a}{4} i,\ i = 1, 2, 3.$ Сравним значения $f(x_1)$ и $f(x_2)$. Если $f(x_1) < f(x_2)$, то уменьшенный вдвое отрезок поиска точки *x** найден - это $[a;x_2]$. Если же $f(x_1) > f(x_2)$, то произведем еще одно сравнение значений $f(x)$ при $f(x_2) < f(x_3)$ перейдем к отрезку $[x_2:b]$.

Отметим, что каким бы ни оказался новый отрезок, одна из уже использованных пробных точек переходит на его середину, становясь новой точкой x_2. Таким образом, для проведения следующей итерации на вновь полученном отрезке потребуется вычисление не более двух новых значений $f(x)$ (либо только в точке x_1, либо еще и в точке x_3).

Перечислим основные шаги алгоритма второго метода деления отрезка пополам.

Шаг 1. Положить $x_2 = \frac{a+b}{2}$. Вычислить значение $f(x_2)$ и перейти к шагу 2.

Шаг 2. Положить $x_1 = \frac{a+x_2}{2}$. Вычислить значение $f(x_1)$ и перейти к шагу 3.

Шаг 3. Сравнить $f(x_1)$ и $f(x_2)$. Если $f(x_1) \le f(x_2)$, то продолжить поиск на отрезке $[a;x_2]$, положив

$$b = x_2,\ x_2 = x_1,\ f(x_2) = f(x_1),$$

и перейти к шагу 5, иначе — положить $x_3 = \frac{x_2 + b}{2}$, вычислить значение $f(x_3)$ и перейти к шагу 4.

Шаг 4. Сравнить $f(x_2)$ и $f(x_3)$. Если $f(x_2) \le f(x_3)$, то перейти к отрезку $[x_1;x_2]$, положив $a = x_1$, $b = x_3$, иначе - продолжить поиск на отрезке $[x_2;b]$, положив $a = x_2$, $x_2 = x_3$, $f(x_2) = f(x_3)$. Перейти к шагу 5.

Шаг 5. Проверка на окончание поиска. Вычислить $\varepsilon_n = \frac{b-a}{2}$ и сравнить с ε. Если $\varepsilon_n > \varepsilon$ то перейти к следующей итерации, вернувшись к шагу 2, иначе - завершить поиск, положив $x^* \approx x_2$, $f^* \approx f(x_2)$.

Пример. Второй метод деления отрезка пополам. Решить задачу

$$f(x) = x^4 + e^{-x} \to \min,\ x \in [0;1],\ \varepsilon = 0{,}1.$$

Итерация 1

Шаг 1. Находим $x_2 = 0{,}5,\ f(x_2) = 0{,}669$. Переходим к шагу 2.

Шаг 2. Определяем $x_1 = 0{,}25,\ f(x_1) = 0{,}783$. Переходим к шагу 3.

Шаг 3. $f(x_1) > f(x_2)$, поэтому полагаем x_3=0,75, вычисляем $f(x_3) = 0{,}789$ и переходим к шагу 4.

Шаг 4. $f(x_2) > f(x_3)$, поэтому полагаем $a = 0{,}25$, $b = 0{,}75$ и переходим к шагу 5.

Шаг 5. Находим $\varepsilon_n = 0{,}25 > 0{,}1$ т.е. переходим к следующей итерации, начиная с шага 2.

Результаты вычислений на остальных итерациях записаны в табл.3.

Таблица 3

Номер итерации	a	b	ε_n	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	*Сравнение* $f(x_1)$ *и* $f(x_3)$
2	0,250	0,750	0,25	0,375	0,500	0,625	0,707	0,669	0,688	$f(x_1) > f(x_2)$ $f(x_3) < f(x_1)$
3	0,375	0,625	0,13	0,438	0,500	0,563	0,669	0,669	0,670	$f(x_1) > f(x_2)$ $f(x_3) < f(x_1)$
4	0,438	0,563	0,06	0,06 < 0,1 – *точность достигнута*						

Таким образом, $x^* \approx x_2 = 0{,}5$, $f^* \approx f(x_2) = 0{,}67$. Сравните этот ответ с результатами решения предыдущих примеров.

Замечание. На первой итерации второго метода деления отрезка пополам вычисляется не более трех значений $f(x)$, а на остальных — не более двух. Поэтому N вычислений $f(x)$ гарантируют осуществление $(N-1)/2$ итераций и достигнутая точность определения x^* составляет

$$\varepsilon(N) = \frac{b-a}{2^{\frac{N-1}{2}+1}} \qquad (23)$$

Сравнение методов исключения отрезков и перебора. При сравнении прямых методов минимизации обычно учитывают количество N значений $f(x)$, гарантирующее заданную точность определения точки x^* тем или иным методом. Чем меньше N, тем эффективнее считается метод. При этом вспомогательные операции, такие, как выбор пробных точек, сравнение значений $f(x)$ и т.п. не учитываются. Во многих практических случаях определение значений целевой функции требует больших затрат (например, времени ЭВМ или средств для проведения экспериментов) и вспомогательными вычислениями можно пренебречь. А эффективность метода минимизации особенно важна именно в таких случаях, поскольку позволяет сократить указанные затраты.

Эффективность методов минимизации можно также сравнивать по гарантированной точности $\varepsilon(N)$ нахождения точки x^*, которую они обеспечивают в результате определения N значений $f(x)$. Из анализа формул (19), (23) следует, что наиболее эффективным из сравниваемых методов является метод золотого сечения, за ним идут методы деления отрезка пополам и наименее эффективен метод перебора. Этот вывод иллюстрирует табл. 4 значений достигнутой точности $\varepsilon(N)$ в зависимости от количества N найденных значений $f(x)$ на отрезке длины 1 для указанных методов.

Таблица 4

Методы минимизации	Количество найденных значений $f(x)$			
	$N=5$	$N=11$	$N=21$	$N=51$
Метод золотого сечения	0,073	$4{,}1\cdot10^{-3}$	$3{,}3\cdot10^{-5}$	$1{,}8\cdot10^{-11}$
Метод деления отрезка пополам	0,125	$1{,}6\cdot10^{-2}$	$4{,}9\cdot10^{-4}$	$1{,}5\cdot10^{-8}$
Метод перебора	0,250	0,100	0,050	0,020

3.7. Метод Фибоначчи

Одним из прямых методов однопараметрической оптимизации является метод Фибоначчи.

Прежде чем описать метод Фибоначчи, рассмотрим связь чисел Фибоначчи с теорией поиска.

1. Известно, что при очень малых скоростях автомобиль расходует на каждый километр пути сравнительно много бензина. Велик его расход и на больших скоростях. Какая-то промежуточная скорость является при этом «оптимальной»: при передвижении с этой скоростью автомобиль расходует на километр пути наименьшее количество горючего. Таким образом, мы можем предполагать, что примерный график зависимости расхода автомобилем горючего на километр пути от скорости автомобиля имеет вид, изображенный на рис. 1: сначала, по мере роста скорости, километровый расход горючего убывает до некоторой минимальной величины, а потом, с дальнейшим ростом скорости, начинает неуклонно (как принято говорить, «монотонно») возрастать.

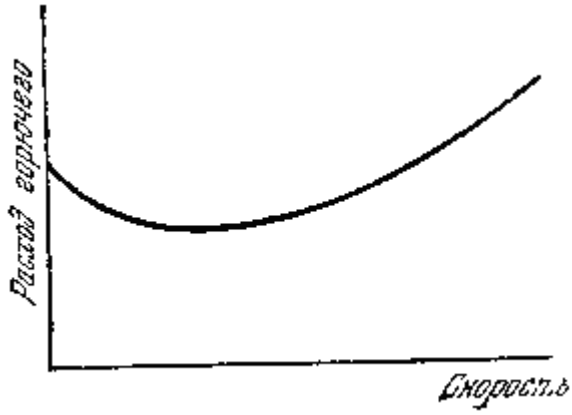


Рис. 1

Хотя общие очертания графика этой зависимости (сначала спуск, а потом подъем) одинаковы практически для всех автомобилей, его точная форма может несколько изменяться даже в пределах автомобилей одного типа, завися от индивидуальных особенностей машины, от степени износа тех или иных ее механизмов и устройств и т. д. В частности, и минимум на нашем графике также может располагаться в довольно широких пределах.

Предположим теперь, что мы получили в свое распоряжение автомашину и хотим предпринять путешествие по такой местности, где в пути не удастся заправиться топливом. Для того чтобы иметь возможность проехать наибольшее расстояние, мы должны достаточно точно определить скорость, соответствующую минимальному расходу горючего. Эта скорость называется *наиболее экономичной скоростью.*

Определять наиболее экономичную скорость автомобиля естественнее всего опытным путем, проезжая с различными скоростями километровые участки дороги, характер и качество которой типичны для условий предстоящего путешествия, и замеряя каждый раз расход бензина. Так как это занятие не из веселых, естественно задуматься над следующими вопросами: сколько опытов достаточно поставить для того, чтобы определить наиболее экономичную скорость автомобиля с заданной точностью? На каких скоростях следует определять в этих опытах расходы горючего? Близкими к этим вопросам являются следующие два: как организовать данное число опытов, чтобы найти экономичную скорость с наибольшей точностью? Какова эта наибольшая точность?

При этом под определением наиболее экономичной скорости «с *точностью до данного* ε» мы будем понимать указание такой скорости v, что истинное значение наиболее экономичной скорости лежит между $v - \varepsilon$ и $v + \varepsilon$ (т. е. что ошибка в определении этой скорости не может превосходить ε).

Для определенности будем считать заранее известным, что наиболее экономичная скорость нашего автомобиля лежит между некоторыми пределами v' и v''. В качестве v' следует взять скорость, которая заведомо не превосходит наиболее экономичной, а в качестве v'' — такую скорость, которая заведомо не меньше ее. (Например, в качестве v' можно взять наименьшую скорость, при которой еще возможна устойчивая работа двигателя, а в качестве v'' — максимальную скорость данного автомобиля.)

2. Отвлекаясь от описанного только что конкретного примера, рассмотрим следующую математическую задачу.

Пусть нам о функции $f(x)$ известно только то, что она от заданного x' до некоторого неизвестного $\overline{x}$ убывает, а от этого x до заданного x'' возрастает (рис. 2).

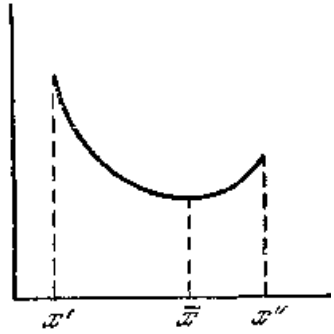


Рис. 2

В частности, мы допускаем, что неизвестная точка $\bar{x}$ в действительности совпадает с одним из концов отрезка, x' или x''. Очевидно, в этом случае функция будет все время возрастать (рис. 3) или все время убывать (рис. 4).

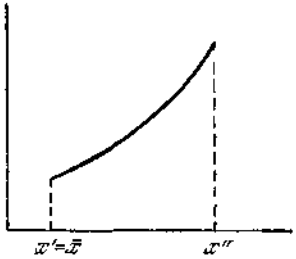


Рис. 3

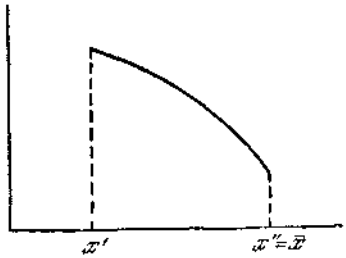


Рис. 4

Разумеется, если один из последних двух случаев и имеет место, то мы будем предполагать это обстоятельство заранее не известным. В точке $\bar{x}$ функция f принимает свое наименьшее значение $f(\bar{x})$, которое называется ее «минимальным» значением или, короче, минимумом. О точке $\bar{x}$ обычно в таких случаях говорят, что на ней функция *достигает минимума.* Ее также часто называют *«минамизирующей» точкой* функции.

Итак, мы далее будем рассматривать только такие функции, в которых убывание не может следовать за возрастанием. Такие функции как известно, называются *«функциями с одним минимумом».*

Нам предстоит проанализировать возможности точного определения положения минимизирующей точки функции f с одним

минимумом. То, что функция f является функцией с одним минимумом, мы далее будем все время предполагать, не оговаривая каждый раз. Совершенно ясно, что *с соответствующими изменениями все то, что мы далее будем говорить о минимумах (наименьших значениях) функций, справедливо и для их максимумов (наибольших значений).*

3. В поставленной проблеме, как и в широком круге аналогичных ей проблем, участвуют три фактора: *цели,* которые мы перед собой ставим, *возможности,* которыми мы располагаем для осуществления этих целей, и, наконец, те *условия,* в которых мы используем наши возможности для достижения целей.

В нашем случае цель состоит в повышении точности определения минимизирующей точки, т. е. в уменьшении ошибки, с которой указывается эта точка.

Возможности состоят в точном определении тем или иным путем (вычислением, измерением или простым угадыванием) некоторого числа значений функции f в произвольно выбираемых точках и в сравнениях между собой найденных в различных точках значений по их величине.

Наконец, условия определяются величиной области задания функции f, т. е. длиной L отрезка между x' и x''.

В соответствии со сказанным каждая конкретная задача поиска может иметь три аспекта.

1) Насколько осуществима поставленная цель при данных возможностях и в данных условиях? Применительно к интересующему нас вопросу это означает следующее.

Пусть мы имеем право совершить n последовательных определений значения f, выбирая каждый раз точку определения по своему усмотрению. В каких точках следует определять значения функции, чтобы точка $\overline{x}$ определилась с наибольшей точностью, и какова эта точность?

2) Какими возможностями необходимо располагать, чтобы осуществить поставленную цель в данных условиях?

В нашей задаче этот вопрос можно конкретизировать так. Пусть мы хотим определить минимизирующую функцию f точку $\overline{x}$ с заданной точностью ε, т. е. указать такое x, что $\overline{x}$ расположено между $x - \varepsilon$ и $x + \varepsilon$. Сколько определений значений функции f для этого необходимо произвести и как эти определения организовать?

3) В каких условиях данные возможности достаточны для достижения поставленной цели?

В данном случае речь идет о нахождении наибольшего интервала L изменения функции f (т. е. наибольшего значения разности $x'' - x'$),

для которого существует способ определения минимизирующей f точки с заданной точностью ε за n наблюдений.

4. Строго говоря, нам придется сейчас иметь дело не с одной задачей, а с двумя.

Во-первых, речь может идти о нахождении минимизирующей точки $\overline{x}$ *вместе* с тем значением *f(x),* которое функция в этой точке принимает.

Во-вторых, мы можем интересоваться *только* самой точкой $\overline{x}$, оставаясь безразличным к значению *f(x).*

Совершенно ясно, что в первой из этих задач (будем далее ее называть задачей *А)* наши цели шире, чем во второй (которую мы назовем задачей *Б).* Поэтому естественно ожидать:

- что при заданных возможностях и условиях цели задачи *А* удастся осуществить в меньшей степени, чем цели задачи *Б* (при данном числе n и длине L в задаче *Б* удается получить меньшее ε, чем в задаче *А);*

- что для осуществления в равной мере целей обеих задач при одинаковых условиях в задаче *А* необходимы большие возможности (при одной и той же погрешности ε и одинаковых длинах L интервалов изменения функции в задаче *А* необходимо большее *n);*

- что одинаковое осуществление целей при равных возможностях требует в задаче *А* более легких условий (данные ε и n в задаче *А* совместимы лишь с меньшими значениями L, чем в задаче *Б).*

5. Чтобы сделать сформулированные задачи математически вполне четкими, необходимо разъяснить следующее важное обстоятельство.

Допустим, что мы интересуемся возможностями определения минимизирующей точки $\overline{x}$ в отрезке длины L (очевидно, мы можем считать началом этого отрезка точку 0 на оси координат, а концом— точку L) с точностью ε. Будем считать, что мы решаем задачу *А*, т. е. что $\overline{x}$ интересует нас вместе со значением $f(\overline{x})$.

Предположим, что мы избрали следующий способ определения x.

Выберем совершенно произвольно некоторое $\overline{x}$ между 0 и L и определим значения функции f в точках $x - \varepsilon$, x и $x + \varepsilon$, т. е. вычислим величины

$$f(x-\varepsilon),\ f(x),\ f(x+\varepsilon)$$

(рис. 5).

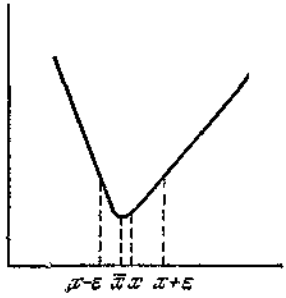


Рис. 5.

При всей произвольности выбора x мы считаем, что $x - \varepsilon \geqq 0$, так что значение функции $f(x - \varepsilon)$ можно фактически вычислить; точно так же мы принимаем, что $x + \varepsilon \leqq L$. Вполне может случиться, что

$$f(x - \varepsilon) > f(x) < f(x + \varepsilon).$$

Это значит, что функция f, убывающая в $x - \varepsilon$, при переходе к $x + \varepsilon$ начинает возрастать. Но переход от убывания функции к возрастанию неизбежно связан с ее прохождением через наименьшее значение. В данном случае это наименьшее значение функции f должно достигаться на некотором $\bar{x}$, лежащем между $x - \varepsilon$, x и $x + \varepsilon$.

Поэтому x будет отстоять от $\bar{x}$ не более, чем на ε, и x окажется как раз тем приближенным значением $\bar{x}$, которое мы ищем. В этом случае определение искомого $\bar{x}$ осуществляется в результате трех наблюдений. Такое *может случиться.* Однако никакой гарантии того, что это *действительно произойдет,* мы не имеем. Более того, если длина L отрезка велика, а ε мало́, то наступление этого явления может показаться довольно неожиданным. Наоборот, в этом случае вполне правдоподобно, что вблизи трех выбранных нами точек функция f будет принимать сравнительно большие значения, а минимума своего она будет достигать где-нибудь совсем в другом месте. Следовательно, трех наблюдений может хватить, а может и не хватить.

Нам же нужен план действий, *неизбежно* приводящий к определению $\bar{x}$ с точностью до ε, где бы в действительности эта точка x ни лежала. Такие планы существуют. Будем, например, систематически вычислять значения нашей функции

$$f(0),\ f(\varepsilon),\ f(2\varepsilon),\ \ldots \qquad (1)$$

до тех пор, пока не дойдем до такого $f(r\varepsilon)$, что $(r + 1)\varepsilon$ будет больше, чем L (рис. 6).

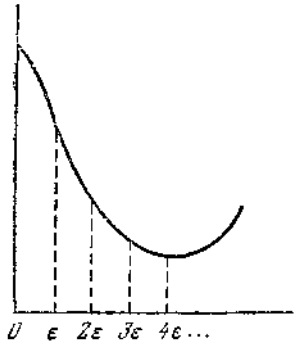


Рис. 6.

Ясно, что то $k\varepsilon$, для которого значение функции будет наименьшим в последовательности (1), и окажется искомым.

Смысл решаемых задач состоит в том, что мы хотим составить не просто план действий, дающий во всех и в том числе в наименее благоприятных случаях жизни значение $\overline{x}$ с предписанной точностью, а *наиболее экономичный* из таких планов, т. е. план, *«наилучший в наихудших условиях».* Но наихудшими условиями являются те, в которых число вычисляемых значений функции f максимально. Аналогично наиболее экономичным планом является такой план, который осуществляет поставленную цель с минимальным числом вычислений значений функции.

Поэтому наилучший в наихудших условиях план называют *минимаксимальным* планом, или планом *минимакса.* Мы будем этот план называть *оптимальным.*

Сущность действий, предписываемых оптимальным планом (как в этой, так и в любой аналогичной задаче), можно охарактеризовать как наиболее целесообразные поиски «прячущегося» от нас минимума функции, «стремящегося оказаться как раз не там, где мы его ищем». Сказанное является характеристикой тех наихудших обстоятельств, применительно к которым мы и квалифицируем наши действия как наилучшие.

6. Важно отметить, что не для всякой задачи поиска существуют оптимальные планы. Так, например, в задаче *Б* оптимального плана нет. В самом деле, пусть $L = 2$ и $n = 2$. Какую точное ε определения мы можем при этом гарантировать?

Будем считать концами нашего отрезка числа 0 и 2. Возьмем произвольно малое положительное ε и вычислим значения функции f в точках $1 - \varepsilon$ и $1 + \varepsilon$. Если при этом

$$f(1-\varepsilon) \leqq f(1+\varepsilon),$$

то искомый минимум $\bar{x}$ должен лежать между нулем и 1 + ε, а если

$$f(1-\varepsilon) \geqq f(1+\varepsilon),$$

то $\bar{x}$ расположено между 1—ε и 2.

Положим в первом случае

$$\bar{x} = \frac{1+\varepsilon}{2},$$

а во втором —

$$\bar{x} = \frac{(1-\varepsilon)+2}{2} = \frac{3-\varepsilon}{2}.$$

В наихудшем случае так определяемое $\bar{x}$ отличается от истинного минимума функции *f* на

$$\frac{1+\varepsilon}{2}.$$

Приближая ε к нулю, мы уменьшаем ошибку. Однако ε не может обратиться в нуль (ибо тогда точки 1-ε и 1+ε совпадут, и сравнение значения функции *f* (1—ε) с заведомо равным ему значением *f*(1+ε), вычисленным в той же точке, не даст нам никакой информации). Поэтому погрешность всегда остается большей, чем половина, хотя и может быть сделана сколь угодно близкой к этому числу.

Каждое положительное значение ε определяет здесь некоторый план. Чем ближе ε к нулю, тем этот план лучше. Так как для любого ε > 0 найдется еще меньшее положительное число, для любого плана найдется еще лучший. Следовательно, оптимального плана для задачи *Б* нет.

Однако для задачи *Б* существуют «почти оптимальные» планы, приводящие к таким результатам, которые можно улучшить лишь незначительно. Говоря точнее, каково бы ни было число γ> 0, существует такой план P_γ, что никакой другой план не сможет уменьшить даваемую планом P_γ ошибку больше, чем на γ.

7. План, описываемый последовательностью (1) при ε, достаточно малом по сравнению с длиной *L* рассматриваемого нами отрезка, оптимальным не является. Придерживаясь этого плана, нам придется в наихудших условиях выполнить все *r* вычислений.

Попробуем, однако, поступить несколько иначе. Будем вычислять члены последовательности (1) через один:

$$f(0), f(2\varepsilon), f(4\varepsilon), \ldots;$$

найдем в полученной последовательности наименьший из членов (пусть им будет $f(2k\varepsilon)$) и вычислим два значения функции $f((2k-1)\varepsilon)$ и

$f((\varepsilon k+1)\varepsilon)$. То из трех значений переменной $(2k—1)\varepsilon$, $2k\varepsilon$ и $(2k+1)\varepsilon$, при котором значение функции f будет наименьшим из трех

$$f((2k-1)\varepsilon), \quad f(2k\varepsilon), \quad f((2k+1)\varepsilon),$$

очевидно, и есть $\overline{x}$ с точностью до ε. Этот новый план приводит в наихудших условиях к цели после выполнения примерно $\frac{r}{2}+2$ вычислений. При больших r это существенно меньше, чем число вычислений, требуемых первым планом.

Итак, первый план не является оптимальным. По сходным причинам второй план также нельзя, вообще говоря, считать оптимальным.

Однако второй план отличается от первого одной весьма существенной чертой: предусматриваемые им точки определения значений функции планируются заранее лишь частично, а выбор оставшихся точек осуществляется на основе сравнений уже вычисленных значений функции. Интуитивно совершенно ясно, что пыбор наилучших действий всегда должен быть связан с использованием информации о результатах действий, которые мы уже произвели. Второй план является в этом отношении более совершенным, чем первый. Но и он, вообще говоря, поддается дальнейшим усовершенствованиям, которые в конце концов приведут нас к оптимальному плану.

Естественно в процессе определения местонахождения минимума функции сравнивать каждое вновь получаемое значение функции с теми или иными из ее значений, полученных при предыдущих наблюдениях. Выбор точки, в которой будет производиться следующее измерение (или решение о прекращении дальнейших измерений), будет поэтому как-то зависеть, во-первых, от тех точек, в которых значения функции нами вычислены, а во-вторых, от самих вычисленных значений функции.

Очевидно, такой процесс последовательного вычисления значений функции f вполне определяется некоторым законом, ставящим в соответствие для любого $k\geq 0$ произвольным наборам точек $x_1, x_2, \ldots x_k$ и значений функции f в этих точках ту или иную точку x_{k+1} или же решение закончить наблюдения над функцией f, приняв ту или иную точку в качестве $\overline{x}$. Этот закон соответствия принято называть *решающей функцией.*

Каждый план определяет некоторую решающую функцию. Точно так же и всякая решающая функция определяет некоторый план. В сущности, решающая функция — это и есть четкое и формализованное

описание плана. Например, решающая функция, определяющая первый из рассмотренных в предыдущем пункте планов, ставит в соответствие каждому числу $0 \leq k < r$ точку $(k + 1)\varepsilon$, а числу r — окончание процесса.

Понятие решающей функции принадлежит к числу важнейших понятий математики.

8. Пусть цель плана P состоит в определении с наименьшей погрешностью точки $\overline{x}$, минимизирующей функцию f на отрезке длины L на основе n наблюдений. Такой план мы далее будем называть *n-шаговым.*

Пусть в условиях некоторого n-шагового плана P удается определить $\overline{x}$ на отрезке длины L с точностью до ε. Эта точность зависит от самого плана P, а также от n и от L. Поэтому мы можем считать ее функцией от P, n и L и обозначать для задачи A через $\tau^A_P(n, L)$, а для задачи $Б$ — через $\tau^Б_P(n, L)$. Под $\tau_P(n,L)$ далее будет подниматься любое (но, конечно, в пределах одногорассуждения одно и то же) из выражений

$$\tau^A_P(n, L) \text{ и } \tau^Б_P(n, L).$$

n-шаювый план P_0 определения минимума f на отрезке длины L является оптимальным в задаче A, если $\tau^A_{P_0}(n, L)$ не более, чем

$$\tau^A_P(n, L)$$

для любого другого плана P, т. е.

$$\tau^A_{P_0}(n, L) \leqq \tau^A_P(n, L).$$

Это можно также записать как

$$\tau^A_{P_0}(n, L) = \min_P \tau^A_P(n, L) \qquad (2)$$

Число $\tau^A_{P_0}(n, L)$, таким образом, оказывается характеристикой уже не плана, а самой задачи (именно, задачи нахождения за n шагов минимизирующей точки функции f на отрезке длины L). Поэтому оно не зависит от какого-либо плана, а зависит только от n и L и может быть просто обозначено через $\tau^A(n, L)$.

В условиях задачи Б дело обстоит несколько сложнее. Здесь, как мы уже видели, *нет* оптимального плана, гарантирующего в наихудших условиях наименьшую погрешность. Однако существует такая погрешность, к которой можно подойти сколь угодно близко, если только выбрать подходящий план. Эта погрешность, которую называют *предельной погрешностью,* также зависит только от условий задачи. Поэтому ее можно обозначить через $\tau^Б(n, L)$. Любой план приводит к большей погрешности:

$$\tau^{Б}(n, L) < \tau_{P}^{Б}(n, L),$$

и поэтому мы не можем написать здесь равенство, аналогичное (2). Забегая несколько вперед, скажем, что итог всех наших рассуждений состоит в получении явных выражений для $\tau^{А}(n, L)$ и $\tau^{Б}(n, L)$. Как окажется, в эти выражения входят числа Фибоначчи:

$$\tau^{А}(n, L) = \frac{L}{u_{n+2}}; \quad (3)$$

$$\tau^{Б}(n, L) = \frac{L}{2u_{n+1}}. \quad (4)$$

Таким образом, отказ от нахождения минимального значения функции позволяет увеличить точность определения ее минимизирующей точки в

$$\frac{2u_{n+1}}{u_{n+2}}$$

раз.

Для достаточно больших *n* это отношение близко к $\frac{2}{\alpha} = 1{,}236$, что соответствует увеличению точности примерно на 23%.

9. Совершенно ясно, что для всех дальнейших рассуждений важны не каждое из чисел *L* и ε само по себе, а отношение *L* и ε. Это отношение является относительной ошибкой положения $\bar{x}$. Если это отношение нам дано, то мы можем, выбирая надлежащим образом единицу измерения величины *x* (т. е. единицу измерения длины нашего отрезка), взять одно из чисел *L* и ε совершенно произвольно.

Это соображение приводит к выводу.

Изменение масштаба вдоль оси *x* изменяет как численное выражение длины отрезка *L*, так и ошибку в определении положения искомой точки любым планом *P* в одно и то же число раз. Другими словами, для любого положительного λ должно быть

$$\tau_{P}(n, \lambda L) = \lambda \tau_{P}(n, L). \quad (5)$$

Точно так же, если мы будем в описании плана определения минимизирующей точки указывать положения тех или иных точек отрезка не в абсолютных мерах длины, а в относительных, то оптимальность плана не нарушится: в результате такого изменения в описании планов оптимальные планы останутся оптимальными, а неоптимальные — неоптимальными.

Отсюда непосредственно следует, что равномерное растяжение (или сжатие) интервала изменения функции *f* в любое число раз

осуществляет лишь «подобное преобразование» оптимального плана, не нарушая при этом его оптимальности.

Значит, ошибки $\tau_P\ (n, \lambda L)$ и $\tau_P\ (n, L)$, фигурирующие в равенстве (5), достигаются не просто в результате осуществления тех или иных планов, а могут быть достигнуты в результате применения одного и того же плана, различным образом «подобно преобразованного».

10. После всех этих предварительных рассмотрений перейдем к нахождению оптимального плана для задачи A и к доказательству формул (3) и (4).

Лемма. *Каковы бы ни были $n \geq 1$ и L, существует n-шаговый план поиска точки $\overline{x}$, минимизирующей значение функции f (с одним минимумом) на отрезке длины L за n шагов и обладающей следующими свойствами:*

1) *на каждом шаге рассматривается некоторый отрезок $x'x''$;*

2) *на первом шаге вычисляется значение функции f в одной из точек:*

$$\frac{u_n}{u_{n+2}} L \quad \text{или} \quad \frac{u_{n+1}}{u_{n+2}} L;$$

3) *к началу каждого из последующих шагов с номером k* (т. е. *при $1 < k \leq n$) известно значение f в одной из следующих точек:*

$$x_1 = x' + \frac{u_n}{u_{n+2}}(x'' - x') \quad \text{и} \quad x_2 = x' + \frac{u_{n+1}}{u_{n+2}}(x'' - x'); \qquad (6)$$

4) *на k-м* ($1 < k \leq n$) *шаге вычисляется значение в другой из точек* (6);

5) *на k-м ($1 < k \leq n$) шаге производится сравнение чисел $f(x_1)$ и $f(x_2)$; при этом, если окажется, что $f(x_1) \leq f(x_2)$, то на (k+1)-м шаге рассматривается отрезок $x'x_2$, а если $f(x\backslash) \geq f(x_2)$, то отрезок x_1x''.*

Доказательство ведется индукцией по n.

Если $n = 1$, то, очевидно, мы имеем дело с отрезком от 0 до L; значение функции f вычисляется в точке

$$\frac{u_1}{u_3} L = \frac{L}{2};$$

последующих же шагов в этом случае вооощe нет.

Предположим теперь, что существование некоторого n-шагового плана с требуемыми в условиях леммы свойствами нами уже установлено для любого отрезка. Займемся построением интересующего нас (n + 1) -шагового плана, проверяя параллельно соблюдение условий леммы. Будем на каждом шаге рассматривать некоторый отрезок $x'x''$.

Возьмем в качестве первого шага выбор точки

$$x_1 = \frac{u_{n+1}}{u_{n+3}} L,$$

а в качестве второго — выбор точки

$$x_2 = \frac{u_{n+2}}{u_{n+3}} L$$

и сравнение значений функции $f(x_1)$ и $f(x_2)$. В случае, когда $f(x_1) \leq f(x_2)$, мы приходим к рассмотрению отрезка между 0 и x_2 (здесь 0 играет роль x', а x_2 — роль x''), а в случае $f(x_1) > f(x_2)$ — к рассмотрению отрезка между x_1 и L (здесь x_1 выступает в роли x', а L в роли x''). Длина рассматриимасмого отрезка в обоих случаях равна

$$\frac{u_{n+2}}{u_{n+3}} L.$$

После выполнения этих двух шагов мы находимся применительно к рассматриваемому отрезку точно в таких же условиях, что и при осуществлении n-шагового процесса после выполнения его первого шага.

Именно, на отрезке длины $\frac{u_{n+2}}{u_{n+3}} L$ известно значение функции f в точке, отстоящей на

$$\frac{u_{n+1}}{u_{n+3}} L.$$

От одного из его концов. Поэтому мы можем «перейти» на этот n-шаговый процесс и довести его до конца. На основании индуктивного предположения мы можем считать, что для последних n шагов выполняются условия 3), 4) и 5). Следовательно, нам остается рассмотреть условия начала второго шага и его проведения. Но, очевидно, точка

$$\frac{u_{n+1}}{u_{n+2}} L$$

имеет вид первого из выражений (6) для случая k — 2, если вместо n в него подставить n+1, а роль второго выражения в соответствующей ситуации играет выбираемая нами точка $\frac{u_{n+2}}{u_{n+3}} L$.

Этим индуктивный переход обоснован и лемма доказана.

11. Будем называть n-шаговый план, существование которого было доказано в предыдущей лемме, *n-шаговым фибоиаччиевым планом,* или, короче, *планом* Φ_g.

Теорема. 1) *План* Φ_q *является единственным оптимальным n-шаговым планом.*

2) $\tau^A_{\Phi_n}(n, L) = \frac{L}{u_{n+2}}$.

Доказательство ведется индукцией по *n.*

Рассмотрим сначала одношаговый план, состоящий в выборе в качестве $\bar{x}$ некоторой точки $\tilde{x}$ из интервала от x' до x''. Очевидно, в наименее благоприятных условиях ошибка может достигнуть здесь наибольшего из чисел $x'' - \bar{x}$ и $\tilde{x} - x'$. Если эти числа различны, то эта максимальная ошибка превосходит $\frac{L}{2}$, если же они равны, то максимальная ошибка равна $\frac{L}{2}$.

Таким образом, план Φ_1 является оптимальным одношаговым планом, а

$$\tau^A_{\Phi_1}(1, L) = \frac{L}{2} = \frac{L}{u_3}.$$

При $n = 2$ мы имеем дело с планом Φ_2, состоящим в вычислении и сравнении значений функции

$$f\left(\frac{1}{3}L\right) \text{ и } f\left(\frac{2}{3}L\right)$$

и выбора в качестве x точки

$$\frac{1}{3}L, \quad \text{если} \quad f\left(\frac{1}{3}L\right) \leqq f\left(\frac{2}{3}L\right),$$
$$\frac{2}{3}L, \quad \text{если} \quad f\left(\frac{1}{3}L\right) > f\left(\frac{2}{3}L\right).$$

Максимальная ошибка в определении истинного значения $\bar{x}$ здесь, как легко видеть, достигает $\frac{L}{3} = \frac{L}{u_4}$:

$$\tau^A_{\Phi_2}(2, L) = \frac{L}{u_4}.$$

Любой иной выбор точки будет приводить к большим возможным ошибкам.

Основание индукции, таким образом, доказано.

Предположим теперь, что фибоначчиев план Φ_n обладает требуемым в условиях теоремы свойством, и рассмотрим $(n + 1)$-шаговые планы.

Произведя в плане Φ_n+1 первые два наблюдения над функцией *f,* мы в результате сравнения двух ее найденных значений сведем дело к

применению к отрезку длины $\frac{u_{n+2}}{u_{n+3}} L$, в котором известно значение f и одной из точек, плана Φ_n, что даст нам в наименее благоприятном случае ошибку

$$\tau^1_{\Phi_n}\left(n, \frac{u_{n+2}}{u_{n+3}} L\right) = \frac{u_{n+2}}{u_{n+3}} \tau^A_{\Phi_n}(n, L) = \frac{u_{n+2}}{u_{n+3}} \frac{L}{u_{n+2}} = \frac{L}{u_{n+3}}.$$

Следовательно,

$$\tau^A_{\Phi_{n+1}}(n+1, L) = \frac{L}{u_{n+3}}.$$

Нам остается показать, что план Φ_n+1 оптимален.

Возьмем с этой целью наблюдения над функцией f и двух произвольных точках, $\tilde{x}_1$ и $\tilde{x}_2$ (для определенности будем считать, что $\tilde{x}_1 < \tilde{x}_2$). Сопоставление значения $f(\tilde{x}_1)$ с $f(\tilde{x}_2)$ приводит к поискам точки $\bar{x}$ либо на отрезке от 0 до $\tilde{x}_2$, либо на отрезке от $\tilde{x}_1$ до L.

Если

$$\tilde{x}_1 < \frac{u_{n+1}}{u_{n+3}} L,$$

то и случае $f(\tilde{x}_1) > f(\tilde{x}_2)$ нам придется искать по не которому n-шаговому плану минимизирующую f точку на отрезке длины $L - \tilde{x}_1$, т. е. большей, чем

$$L - \frac{u_{n+1}}{u_{n+3}} L = \frac{u_{n+3} - u_{n+1}}{u_{n+3}} L = \frac{u_{n+2}}{u_{n+3}} L.$$

Даже если положение точки $\tilde{x}_2$ на этом отрезке наиболее благоприятно, то ошибка в определении окажется на основании индуктивного предположения большей, чем $\frac{L}{u_{n+3}}$.

Симметричные рассуждения показывают, что план, начинающийся выбором некоторой точки

$$\tilde{x}_2 > \frac{u_{n+2}}{u_{n+3}} L,$$

также может при соответствующих неблагоприятных условиях привести к большей ошибке в определении $\bar{x}$, чем план Φ_{n+1}. Пусть теперь

$$\tilde{x}_1 > \frac{u_{n+1}}{u_{n+3}} L.$$

Если в действительности x находится между 0 и $\tilde{x}_1$ то на поиски местоположения этой точки нам остается n — 1 наблюдение, а длина

отрезка, заключающего эту точку, больше, чем $\frac{u_{n+1}}{u_{n+3}} L$. Значит, даже план Φ_{n+1} (который, по предположению, в этих условиях оптимален) приведет нас к ошибке, большей, чем

$$\tau^A_{\Phi_{n-1}}\left(n-1, \frac{u_{n+1}}{u_{n+3}} L\right) = \\ = \frac{u_{n+1}}{u_{n+3}} \tau^A_{\Phi_{n-1}}(n-1, L) = \frac{u_{n+1}}{u_{n+3}} \frac{L}{u_{n+1}} = \frac{L}{u_{n+3}}.$$

Симметрично разбирается случай, когда

$$\tilde{x}_2 > \frac{u_{n+2}}{u_{n+3}} L.$$

Следовательно, план Φ_n+1 является оптимальным, и теорема доказана.

Итак, единственная минимизирующая функцию f точка может быть с помощью n наблюдений определена на отрезке длины L с ошибкой, не превосходящей $\frac{L}{u_{n+2}}$.

Поэтому n наблюдений позволяют определить точку, минимизирующую f, с ошибкой ε или меньше, на отрезке, длина которого не превосходит εu_{n+2}.

Наконец, чтобы быть уверенным в том, что точка, минимизирующая функцию f, определена на отрезке длины L с ошибкой, не превосходящей ε, необходимо произвести такое число n наблюдений, что

$$u_{n+1} < \frac{L}{\varepsilon} \leqq u_{n+2}.$$

Таким образом, мы ответили на все вопросы п. 3.

12. Решение задачи *Б* можно получить из описанного выше решения задачи *А* без особого труда.

Пусть нам дан отрезок длины L. Проделаем на этом отрезке первые n — 2 шага фибоначчиева плана Φ_{n-1}. В результате мы придем к отрезку длины $\frac{3L}{u_{n+1}}$ с концами x' и x'' и с известным значением f в одной из точек

$$x_1 = x' + \frac{L}{u_{n+1}} \text{ или } x_2 = x' + \frac{2L}{u_{n+1}}.$$

Ограничимся рассмотрением первого из этих случаев (второй рассматривается симметрично).

Итак, пусть $f(x_1)$ нам уже известно. Выберем произвольное число γ, по абсолютной величине меньшее, чем

$$\frac{L}{u_{n+1}};$$

вычислим $f(x_2 - \gamma)$ (это есть $(n-1)$-е вычисленное значение функции f) и (4>:шпим $f(x_1)$ и $f(x_2 - \gamma)$.

Если $f(x_1) \leqq f(x_2 - \gamma)$ (случай ○ на рис. 7), то, очевидно, $\bar{x}$ находится между x' и $x_2 - \gamma$.

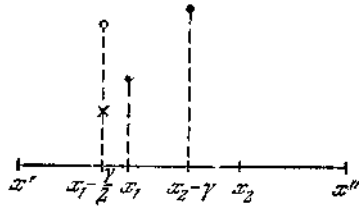


Рис. 7.

Вычислим

$$f\left(\frac{x' + (x_2 - \gamma)}{2}\right) = f\left(x_1 - \frac{\gamma}{2}\right)$$

(это — последнее, n-е вычисленное значение функции f).

Если при этом

$$f\left(x_1 - \frac{\gamma}{2}\right) \leqq f(x_1)$$

(случай × на рис. 7), то $\bar{x}$ расположено между x' и x_1. Положим

$$\bar{x} = \frac{x' + x_1}{2}.$$

Ошибка в определении $\bar{x}$ не превосходит половины длины отрезка от x' до x_1, т. е. $\frac{L}{2u_{n+1}}$. Если

$$f\left(x_1 - \frac{\gamma}{2}\right) > f(x_1)$$

(случай ○ на рис. 7), то $\bar{x}$ находится между $x_1 - \frac{\gamma}{2}$ и $x_2 - \gamma$. Положив $\bar{x} = \frac{1}{2}\left(\left(x_1 - \frac{\gamma}{2}\right) + (x_2 - \gamma)\right)$, мы совершим ошибку, не большую, чем

$$\frac{1}{2}\left((x_2 - \gamma) - \left(x_1 - \frac{\gamma}{2}\right)\right) = \frac{1}{2}\left(x_2 - x_1 - \frac{\gamma}{2}\right) =$$

$$= \frac{x_2 - x_1}{2} - \frac{\gamma}{4} = \frac{L}{2u_{n+1}} - \frac{\gamma}{4}.$$

Пусть теперь $f(x_1) > f(x_2 - \gamma)$ (рис. 8).

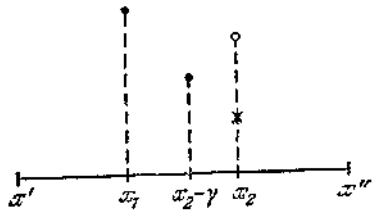


Рис. 8

Тогда $\bar{x}$ находится между x_1 и x''.

Вычислим $f(x_2)$ (последнее вычисленное значение f).

Если

$$f(x_2 - \gamma) \leqq f(x_2)$$

(случай ∘ на рис. 8), то $\bar{x}$ расположено между x_1 и x_2; беря $\bar{x} = \frac{1}{2}(x_1 + x_2)$, мы допустим ошибку, достигающую лишь

$$\frac{1}{2}(x_2 - x_1) = \frac{L}{2u_{n+1}}.$$

Если, наконец,

$$f(x_2 - \gamma) > f(x_2)$$

(случай × на рис. 8), то $\bar{x}$ расположено между $x_2 - \gamma$ и x''. положив $\bar{x} = \frac{1}{2}(x'' + (x_2 - \gamma))$, мы совершаем ошибку, не превосходящую

$$\frac{1}{2}(x'' - (x_2 - \gamma)) = \frac{1}{2}\left(\frac{L}{u_{n+1}} + \gamma\right) = \frac{L}{2u_{n+1}} + \frac{\gamma}{2}.$$

В наихудшем для нас случае при $\gamma > 0$ ошибка может таким образом достигнуть величины $\frac{L}{2u_{n+1}} + \frac{\gamma}{2}$, а при $\gamma < 0$ — величины $\frac{L}{2u_{n+1}} - \frac{\gamma}{4}$. Поскольку, однако, число γ находится в нашем распоряжении, мы можем сделать ошибку, сколь угодно близкой к $\frac{L}{2u_{n+1}}$.

Нам остается убедиться в том, что ошибку $\frac{L}{2u_{n+1}}$ уменьшить нельзя.

В самом деле, отклонения от описанного плана на каком-либо из первых n — 2 шагов могут привести, как видно из теоремы п. 11, только к увеличению длины отрезка, в котором местоположение

минимизирующей точки определяется последующими измерениями, и тем самым к заведомому увеличению максимальной ошибки. Остается проверить оптимальность действий, совершаемых на последних двух шагах.

Прежде всего, отклонение от описанных действий может означать окончательный выбор в качестве $\bar{x}$ не середины отрезка, где эта точка действительно расположена, а другой точки. Ясно, что это приведет к тому, что возможная ошибка окажется равной большей части отрезка, т. е. возрастет. Следовательно, должна быть выбрана именно середина отрезка.

Далее, мы могли бы выбрать для последнего определения f точку, не близкую к точке x_1 (или соответственно к x_2). Но тогда возможная ошибка увеличилась бы и притом пропорционально расстоянию между этими точками.

Наконец, к таким же последствиям привел бы выбор точки для предпоследнего определения f, далекой от x_2 (соответственно от x_1).

Итак, ни одно из отклонений от описанного плана не может повлечь за собой уменьшения возможной ошибки до числа, меньшего, чем

$$\frac{L}{2u_{n+1}}.$$

Это показывает, что задача *Б* нами решена.

Мы предоставляем читателю сформулировать в случае задачи *Б* ответы на остальные вопросы, перечисленные в п. 3.

13. В предыдущих пунктах описание самого плана поиска сопровождалось уточнениями постановки задачи, формулировками, связанными с понятием оптимальности и обоснованиями оптимальности конструируемою плана. Все эти отступления от прямого описания являются неотъемлемыми элементами всякого математического рассуждения, цель которого состоит не только в *указании* какого-то процесса, но и в *доказательстве* того, что этот процесс — именно тот, который нас интересует. Вместе с тем во многих случаях существенным является четкое описание действий как таковых, а вся аргументация этих действий становится совершенно неважной. Это бывает тогда, когда, например, после решения задачи имеется в виду фактическое осуществление этого решения. В таких случаях для реализации решения задачи на практике необходимо располагать не столько математическим обоснованием верности решения, сколько предельно четкими, не допускающими каких бы то ни было кривотолков, предписаниями по его претворению в жизнь.

План наиболее точных поисков на отрезке от x' до x'' точки $\bar{x}$, минимизирующей функцию f в условиях задачи A в изложении, преследующем только что описанные, так сказать, «практические» цели, напоминает план установления вида растения по ботаническому определителю (заметим, что определение растения есть тоже поиск!). Он принимает следующий вид (если в конце пункта не указывается, к какому пункту следует переходить, то нужно переходить к следующему пункту):

1°. Сравнить 1 и *n:*

а) если $n = 1$, то перейти к п. 2°;

б) если $n > 1$, то перейти к п. 4°.

2°. Вычислить $\tilde{x} = \frac{x' + x''}{2}$.

3°. Вычислить *f(x);* на этом процесс кончается.

4°. Вычислить

$$x_1 = x' + \frac{u_n}{u_{n+2}}(x'' - x') \quad \text{и} \quad x_2 = x' + \frac{u_{n+1}}{u_{n+2}}(x'' - x').$$

5°. Вычислить $f(x_1)$ и $f(x_2)$.

6°. Сравнить 2 и *n:*

а) если $n = 2$, то перейти к п. 7°;

б) если $n > 2$, то перейти к п. 10°.

7°. Сравнить f *(xi)* и $f(x_2)$*:*

а) если $f(x_1) \leqq f(x_2)$, то перейти к п. 8°;

б) если $f(x_1) > f(x_2)$, то перейти к п. 9°.

8°. Положить $\bar{x} = x_1$ и закончить процесс.

9°. Положить $\bar{x} = x_2$ и закончить процесс.

10°. Сравнить $f(x_1)$ и $f(x_2)$*:*

а) если $f(x_1) \leqq f(x_2)$, то перейти к п. 11°;

б) если $f(x_1) > f(x_2)$, то перейти к п. 14°.

11°. Переобозначить

x_2 через x'',
x_1 через x_2,
$n - 1$ через n.

12°. Вычислить

$$x_1 = x' + \frac{u_n}{u_{n+2}}(x'' - x').$$

13°. Вычислить $f(x_1)$ и перейти к п. 6°.

14°. Переобозначить

x_1 через x',
x_2 через x_1,
$n - 1$ через n.

15°. Вычислить

$$x_2 = x' + \frac{u_{n+1}}{u_{n+2}}(x'' - x').$$

16°. Вычислить $f(x_2)$ и перейти к п. 6°.

14. Хотя сформулированное описание оптимального плана поисков минимума функции f абсолютно четкое, не оставляющее места какому-либо произволу, и в применении к каждой конкретной функции f, отрезку от x' до x'' и числу n предписывает совершенно точную последовательность действий, оно является довольно запутанным и трудно обозримым.

Приведем поэтому для наглядности еще одно описание этого же плана в виде блок-схемы (рис. 9).

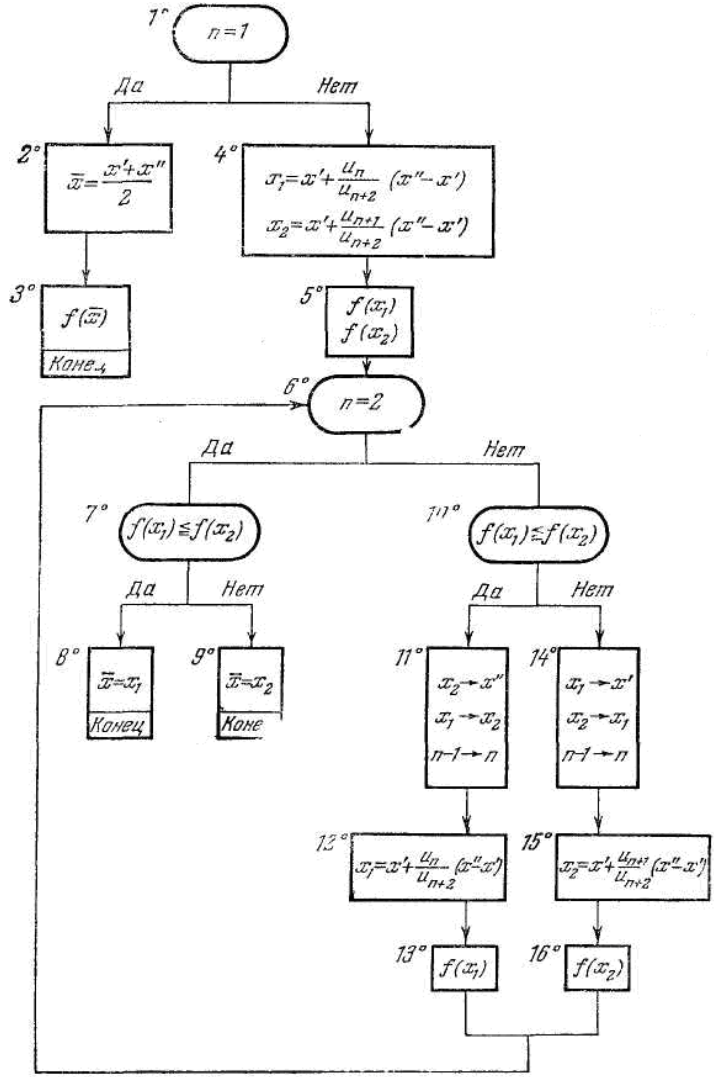


Рис. 9

15. Приведем в заключение пример использовании описанного в пп. 13 и 14 плана для нахождения с помощью пяти вычислений на отрезке от 1 до 2 точки $\bar{x}$, минимизирующей функцию

$$f(x) = \frac{1}{x} + \sqrt{x}.$$

Предварительно сделаем замечание.

Нахождение точки, минимизирующей (или максимизирующей) функцию, которая задана *аналитически*, обычно удобнее проводить не методами теории поиска, а другими, более приспособленными для этого приемами, которые относятся к дифференциальному исчислению. Поэтому следует иметь в виду, что приводимый далее пример носит чисто иллюстративный характер. Диференциальное исчисление позволяет без труда показать, что в этом случае $\bar{x} = \sqrt[3]{4} = 1{,}5874011\ldots$ Нам же удастся найти значительно более грубое приближение. Однако в тех случаях, когда заранее о функции нам неизвестно ничего (кроме того, что она не может переходить от возрастания к убыванию) или же выражения, которыми она задается, чересчур сложны, методика дифференциальною исчисления неприменима, и теория поиска оказывается полезным инструментом.

1°. Сравнение n=5 и 1 дает нам, что $n \neq 1$, поэтому переходим к п. 4°.

4°. Вычисляем:

$$x_1 = x' + \frac{u_n}{u_{n+2}}(x'' - x') = 1 + \frac{5}{13}(2 - 1) = 1{,}38461,$$

$$x_2 = x' + \frac{u_{n+1}}{u_{n+2}}(x'' - x') = 1 + \frac{8}{13}(2 - 1) = 1{,}61538.$$

5°. Вычисляем:

$$f(x_1) = \frac{1}{x_1} + \sqrt{x_1} = f(1{,}38461) = 0{,}72222 + 1{,}17670 = 1{,}89892,$$

$$f(x_2) = \frac{1}{x_2} + \sqrt{x_2} = f(1{,}61538) = 0{,}61905 + 1{,}27098 = 1{,}89003.$$

6°. Сравнение n = 5 и 2 дает, что $n \neq 2$; поэтому Переходим к п. 10°.

10°. Сравнение

$$f(x_1) = 1{,}89892 \quad \text{и} \quad f(x_2) = 1{,}89003$$

дает нам $f(x_1) > f(x_2)$; поэтому переходим к п. 14°.

14°. Переобозначаем:

$$x_1 \to x' = 1{,}38461,$$

$$x_2 \to x_1 = 1{,}61538,$$

$$n = 4.$$

15°. Вычисляем:

$$x_2 = x' + \frac{u_{n+1}}{u_{n+2}}(x'' - x') = 1{,}38461 + \frac{5}{8}(2 - 1{,}38461) = 1{,}76927.$$

16°. Вычисляем:

$$f(x_2) = \frac{1}{x_2} + \sqrt{x_2} = f(1{,}76927) = 0{,}56522 + 1{,}33012 = 1{,}89534$$

и переходим к п. 6°.

6°. Сравниваем $n = 4$ и 2; поскольку $n \neq 2$, переходим к п. 10°.

10°. Сравниваем $f(x_1) = 1{,}89003$ и $f(x_2) = 1{,}89534$; поскольку $f(x_1) \leqq f(x_2)$, переходим к п. 11°.

11°. Переобозначаем:

$$x_2 \to x'' = 1{,}76923,$$
$$x_1 \to x_2 = 1{,}61538,$$
$$n = 3.$$

12°. Вычисляем:

$$x_1 = x' + \frac{u_n}{u_{n+2}}(x'' - x') = 1{,}38461 + \frac{2}{5}(1{,}76923 - 1{,}38461) = 1{,}53846.$$

13°. Вычисляем:

$$f(x_1) = \frac{1}{x_1} + \sqrt{x_1} - f(1{,}53846) = 0{,}65000 + 1{,}24035 = 1{,}89035$$

и переходим к п. 6°.

6°. Сравниваем $n = 3$ и 2; поскольку $n \neq 2$, переходим к п. 10°.

10°. Сравниваем

$$f(x_1) = 1{,}89035 \quad \text{и} \quad f(x_2) = 1{,}89003;$$

поскольку $f(x_1) > f(x_2)$, переходим к п. 14°.

14°. Переобозначаем:

$$x_1 \to x' = 1{,}53846,$$
$$x_2 \to x_1 = 1{,}61538,$$
$$n = 2.$$

15°. Вычисляем:

$$x_2 = x' + \frac{u_{n+1}}{u_{n+2}}(x'' - x') = 1{,}53846 + \frac{2}{3}(1{,}76923 - 1{,}53846) = 1{,}69231.$$

16°. Вычисляем

$$f(x_2) = \frac{1}{x_2} + \sqrt{x_2} = f(1{,}69231) = 0{,}59091 + 1{,}30089 = 1{,}89170$$

и переходим к п. 6°.

6°. Сравнение g и 2 дает нам, что $n = 2$; переходим к п. 7°.

7°. Сравниваем

$$f(x_1) = 1{,}89003 \quad \text{и} \quad f(x_2) = 1{,}89170;$$

$f(x_1) \le f(x_2)$; поэтому переходим к п. 8.

8°. Полагаем $\bar{x}$ =1,61538.

На основании теоремы п. 11 найденное нами $\bar{x}$ может отличаться от истинного положения минимизирующей точки не более чем на

$$\frac{1}{u_{n+2}} = \frac{1}{u_7} = \frac{1}{13} = 0{,}077.$$

Фактически эта ошибка оказывается меньшей; она равна 0,028. Заметим, что принимаемое нами за наименьшее значение функции f, т. е. $f(x)$, равно 1,89003 и отличается от истинного наименьшего значения f, равного

$$f(\sqrt[3]{4}) = \frac{1}{\sqrt[3]{4}} + \sqrt[3]{2} = \frac{3}{2}\sqrt[3]{2} = 1{,}88988,$$

лишь, на 0,00015. Это показывает, что значения x можно было в ходе наших вычислений определять с меньшей точностью, чем значения f.

Сам по себе такой вывод не содержит ничего удивительного. В самом деле, значения x мы должны находить с той предельной точностью, с какой мы можем в наших условиях найти минимизирующую точку $\bar{x}$ (мы знаем, что эта точность равна $\frac{1}{u_{n+2}}$).

Значения же функции f должны вычисляться с точностью, обеспечивающей сравнение пар значений этой функции и выделение из каждой такой пары наименьшего и наибольшего значения. Поэтому если в действительности какие-нибудь $f(a)$ и $f(b)$ сильно отличаются друг от друга и это отличие заметно уже при грубом определении $f(a)$ и $f(b)$, то мы можем вычислять эти значения с малой точностью. Наоборот, если эти $f(a)$ и $f(b)$ в действительности близки, то для выяснения того, какое из них больше другого, приходится вести вычисление с большой точностью. Так как мы наперед (до фактического выполнения вычислений) не знаем, насколько отличаются друг ог друга сравниваемые значения функции, мы можем

«промахнуться» и вычислить их с недостаточной точностью, которая не даст возможности решить, какое из этих значений больше. В этом случае придется произвести повторные, более точные вычисления, затратив на это дополнительные усилия.

А теперь рассмотрим метод Фибоначчи как метод однопараметрической оптимизации.

Предположим, что нужно определить минимум как можно точнее, т.е. с наименьшим возможным интервалом неопределенности, но при этом можно выполнить только *n* вычислений функции. Как следует выбрать *n* точек, в которых вычисляется функция? С первого взгляда кажется ясным, что не следует искать решение для всех точек, получаемых в результате эксперимента. Напротив, надо попытаться сделать так, чтобы значения функции, полученные в предыдущих экспериментах, определяли положение последующих точек. Действительно, зная значения функции, мы тем самым имеем информацию о самой функции и положении ее минимума и используем эту информацию в дальнейшем поиске.

Предположим, что имеется интервал неопределенности (x_1,x_3) и известно значение функции $f(x_2)$ внутри этого интервала (см. рис. 10). Если можно вычислить функцию всего один раз в точке x_4, то где следует поместить точку x_4, для того чтобы получить наименьший возможный интервал неопределенности?

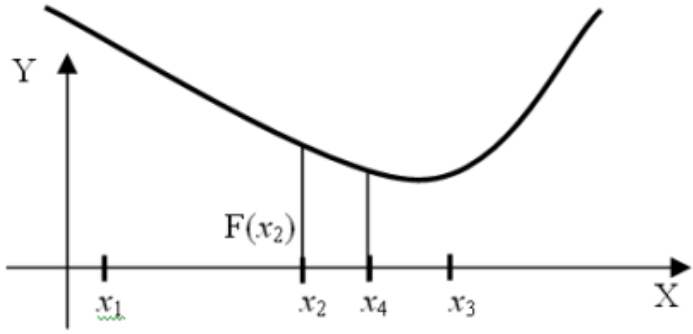


Рис. 10.

Положим $x_2–x_1=L$ и $x_3–x_2=R$, причем $L > R$, как показано на рис. 10, и эти значения будут фиксированы, если известны x_1, x_2 и x_3. Если x_4 находится в интервале $(x_1; x_2)$, то:

1. если $f(x_4) < f(x_2)$, то новым интервалом неопределенности будет (x_1,x_2) длиной $x_2–x_1=L$;
2. если $f(x_4)>f(x_2)$, то новым интервалом неопределенности будет (x_4,x_3) длиной $x_3–x_4$.

Поскольку не известно, какая из этих ситуаций будет иметь место, выберем x_4 таким образом, чтобы минимизировать наибольшую из длин x_3-x_4 и x_2-x_1. Достигнуть этого можно, сделав длины $x_3 – x_4$ и $x_2 – x_1$ равными т.е. поместив x_4 внутри интервала симметрично относительно точки x_2, уже лежащей внутри интервала. Любое другое положение точки x_4 может привести к тому, что полученный интервал будет больше L. Помещая x_4 симметрично относительно x_2, мы ничем не рискуем в любом случае. Если окажется, что можно выполнить еще одно вычисление функции, то следует применить описанную процедуру к интервалу (x_1, x_2), в котором уже есть значение функции, вычисленное в точке x_4, или к интервалу (x_4,x_3), в котором уже есть значение функции, вычисленное в точке x_2.

Следовательно, стратегия ясна с самого начала. Нужно поместить следующую точку внутри интервала неопределенности симметрично относительно уже находящейся там точке. Парадоксально, но, чтобы понять, как следует начинать вычисления, необходимо разобраться в том, как его следует кончать.

На n-м вычислении n-ю точку следует поместить симметрично по отношению к $(n — 1)$-й точке. Положение этой последней точки в принципе зависит от нас. Для того чтобы получить наибольшее уменьшение интервала на данном этапе, следует разделить пополам предыдущий интервал. Тогда точка x будет совпадать с точкой x_{n-1}. Однако при этом мы не получаем никакой новой информации. Обычно точки x_{n-1} и x_n отстоят друг от друга на достаточном расстоянии, чтобы определить, в какой половине, левой или правой, находится интервал неопределенности. Они помещаются на расстоянии $\varepsilon/2$ по обе стороны от середины отрезка L_{n-1}; можно самим задать величину ε или выбрать эту величину равной минимально возможному расстоянию между двумя точками.

Интервал неопределенности будет иметь длину L_n, следовательно, $L_{n-1}=2L_n - \varepsilon$ (рис. 11, нижняя часть). На предыдущем этапе точки x_{n-1} и x_{n-2} должны быть помещены симметрично внутри интервала L_{n-2} на расстоянии L_{n-2} от концов этого интервала. Следовательно, $L_{n-2} = L_{n-1}+L_n$ (рис. 11, средняя часть).

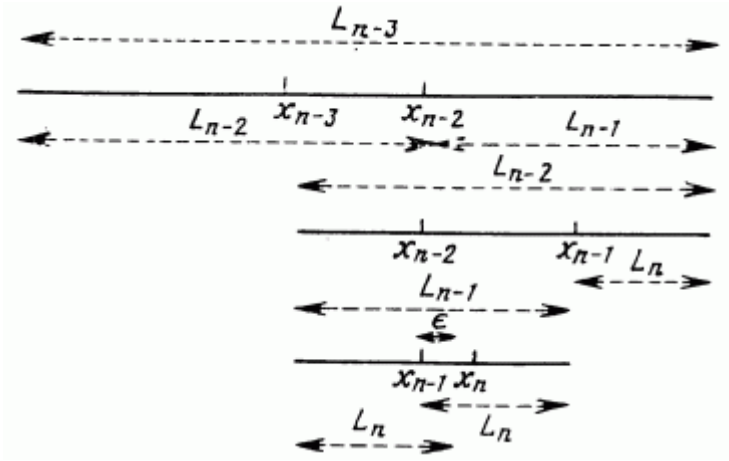


Рис. 11.

Замечание. Из рисунка ясно, что на предпоследнем этапе x_{n-2} остается в качестве внутренней точки.

Аналогично $L_{n-3}=L_{n-2}+L_{n-1}$ (рис. 11, верхняя часть)

В общем случае $L_{j-1}=L_j + L_{j+1}$ при $1<j<n$.

Таким образом,

$$L_{n-1} = 2L_n - \varepsilon,$$
$$L_{n-2} = L_{n-1} + L_n = 3L_n - \varepsilon,$$
$$L_{n-3} = L_{n-2} + L_{n-1} = 5L_n - 2\varepsilon,$$
$$L_{n-4} = L_{n-3} + L_{n-2} = 8L_n - 3\varepsilon \quad \text{и т.д.}$$

Если определить последовательность чисел Фибоначчи следующим образом: F_0=1, F_1=l, и F_k=F_{k-1}+F_{k-2} для k = 2, 3,..., то

$$L_{n-j} = F_{j+1}L_n - F_{j-1}\varepsilon, \quad j = 1, 2, \ldots, n-1. \tag{7}$$

Если начальный интервал (*a;b*) имеет длину L = (*b-a*), то

$$\begin{aligned} &L_1 = F_n L_n - \varepsilon F_{n-2}, \text{ т.е.} \\ &L_n = \frac{L_1}{F_n} + \varepsilon\frac{F_{n-2}}{F_n}. \end{aligned} \tag{8}$$

Следовательно, произведя n вычислений функции, мы уменьшим начальный интервал неопределенности в l/F_n раз по сравнению с его начальной длиной (пренебрегая ε), и это - наилучший результат.

Если поиск начат, то его несложно продолжить, используя описанное выше правило симметрии. Следовательно, необходимо найти положение первой точки, которая помещается на расстоянии L_2 от одного из концов начального интервала, причем не важно, от какого конца, поскольку вторая точка помещается согласно правилу симметрии на расстоянии L_2 от второго конца интервала:

$$\begin{aligned} L_2 &= F_{n-1}L_n - \varepsilon F_{n-3} = \\ &= F_{n-1}\frac{L_1}{F_n} + \varepsilon\frac{\left(F_{n-1}F_{n-2} - F_n F_{n-3}\right)}{F_n} = \\ &= \frac{F_{n-1}}{F_n}L_1 + \frac{(-1)^n\varepsilon}{F_n}. \end{aligned} \tag{9}$$

После того как найдено положение первой точки, числа Фибоначчи больше не нужны. Используемое значение ε может определяться из практических соображений. Оно должно быть меньше $L_1\backslash F_{n+x}$, в противном случае мы будем напрасно тратить время на вычисление функции.

Таким образом, поиск методом Фибоначчи, названный так ввиду появления при поиске чисел Фибоначчи, является итерационной процедурой. В процессе поиска интервала (x_1; x_2) с точкой x_2, уже лежащей в этом интервале, следующая точка x_2 всегда выбирается такой, что x_3–x_4 = x_2–x_1 или x_4-x_1 = x_3-x_2, т.е. x_4=x_1-x_2+x_3.

Если $f(x_2)$ = f_2 и $f(x_4)$ = f_4, то можно рассмотреть четыре случая (рис. 12).

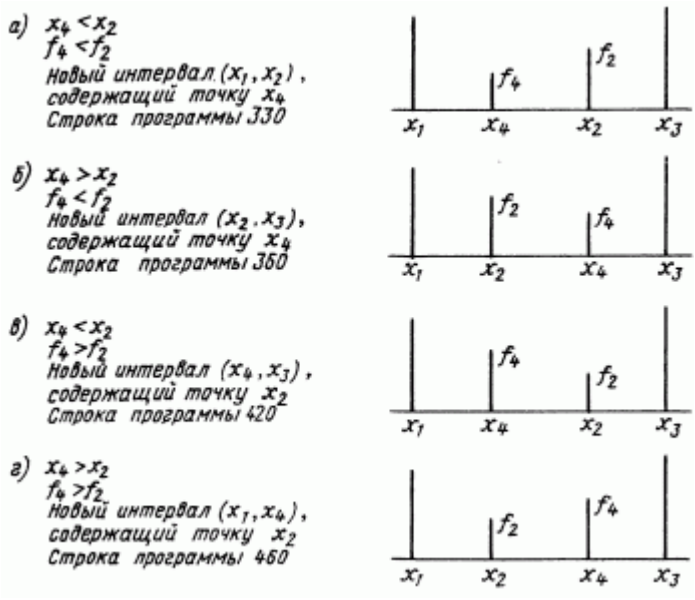


Рис. 12.

3.8. Метод конфигураций

При решении вопроса о выборе численного метода рекомендуется оценить поведение линий уровня целевой функции в окрестностях предполагаемой точки экстремума. Число $m=L/l$, где L и l - максимальное и минимальное собственные значения гессиана функции f в предполагаемой точке экстремума x^0 (характеризующее разброс собственных значений оператора $f(x)$), называется *числом обусловленности* гессиана функции f в точке x^0. Если $m >> 1$, то

функция f называется *плохо обусловленной* или *овражной*. *Овражность*, то есть вытянутость линий уровня вдоль одного направления, приводит к тому, что градиентные методы поиска экстремума функции сходятся медленно.

В зависимости от наивысшего порядка частных производных функции $f(x)$, используемых для формирования d^k и t_k, численные методы используют методы нулевого порядка, использующие информацию только о значениях функции $f(x)$ **(методы деформируемого многогранника, конфигураций)**. Эти методы могут применяться в тех случаях, когда функция задана неявно или не задана аналитически, но известен ряд значений функции или эти значения вычисляются непосредственно в ходе реализации алгоритма. Они также могут быть полезны в случаях, когда производные функции могут быть заданы аналитически, но их выражения очень громоздки.

Следует выделить два этапа метода конфигураций:

1) исследование с циклическим изменением переменных и 2) ускорение поиска по образцам.

Исследующий поиск начинается в точке x^0, называемой старым базисом. Направления поиска - координатные направления. По каждому направлению поочередно с шагом $+t_0$ ($-t_0$) проверяется выполнение условия и в качестве нового базиса берется точка с координатами, полученными в результате удачных шагов из начальной точки по каждому направлению. Направление от старого базиса к новому задает направление ускорения поиска: в качестве следующей точки минимизирующей последовательности проверяется точка $y^1=x^0+(x^1-x^0)$. Здесь - ускоряющий множитель, задаваемый пользователем. Если полученная точка является удачной, то она берется в качестве следующей точки для исследования. В противном случае исследование ведется из точки x^1.

При решении задачи оптимального проектирования часто приходится иметь дело с математическими моделями, в которых не имеется аналитических выражений для первых производных минимизируемой функции $Q(x)$. В связи с чем поиск оптимального решения x^* приходится вести по результатам вычислений функции $Q(x)$. Методы, которые используют для выбора точки очередного испытания x^r информацию только о значениях функции $Q(x)$, как мы

уже говорили, называются методами прямого поиска (методами нулевого порядка, методами минимизации без вычисления производных).

Наиболее простыми из алгоритмов данного класса методов являются алгоритмы, реализующие метод покоординатного спуска. Основная идея этого метода заключается в том, что поиск точки минимума x^* сводится к поочередному изменению переменных вдоль одной из координатных осей:

$$x_i^{r+1} = x_i^r + \lambda_i^r I_i,\ i = 1,2,\ \ldots,\ n. \qquad (1)$$

где I_i — i-й координатный n-мерный вектор с компонентами:

$l_{ij} = 1$, если $i = j$;

$l_{ij} = 0$ – в противном случае.

Длина шага λ_i^r вдоль направления поиска I_i может выбираться равной некоторой постоянной величине Δ_i по следующему правилу:

$$\lambda_i^r = \Delta_i, \text{ если } Q(x^r + \Delta_i I_i) < Q(x^r);$$

$$\lambda_i^r = -\Delta_i, \text{ если } Q(x^r - \Delta_i I_i) < Q(x^r) < Q(x^r + \Delta_i I_i). \qquad (2)$$

Если окажется, что $\lambda_i^r = 0$ для всех $i = 1, 2, \ldots, n$, то длина пробных шагов Δ_i должна быть уменьшена ($\Delta_i = \Delta_i/\beta$, где $\beta > 1$). Поиск считается законченным при выполнении условия:

$$\max \Delta_i < \varepsilon. \qquad (3)$$

Алгоритм F^{29}, реализующий описанную стратегию поиска точки минимума x^*, называется методом **покоординатного спуска с постоянным шагом**.

Когда длина шага λ_i^r на каждой итерации определяется с помощью одномерной задачи оптимизации

$$Q(x^r + \lambda_i^r I_i) = \min Q(x^r + \sum \lambda_k^r I_k + \lambda_i I_i) \qquad (4)$$

приходим к алгоритму F^{30}, реализующему **релаксационный метод Гаусса — Зейделя**, процедура поиска точки минимума x^* в котором сводится к следующей последовательности действий.

1. Задается начальное приближение $x^r = x^\circ$.
2. Осуществляется циклический покоординатный спуск из точки x^r по формуле (1) с выбором длины шага λ_k^r, из условия (4) для всех i от 1 до n. Эта процедура образует внутренний цикл, в процессе которого осуществляется одномерная минимизация функции $Q(x)$ по каждой переменной:

$$\min Q(x_1^r, \ldots, x_{i-1}^r, x_i, x_{i+1}^r, \ldots, x_n^r),\ i = 1, 2, \ldots, n.$$

3. После окончания внутреннего цикла в качестве начального приближения x° принимается точка x^n и все вычисления повторяются с п. 2.

4. Поиск точки минимума x^* заканчивается, если после очередного внутреннего цикла выполняется условие $\|x^r - x^n\| < \varepsilon$.

Геометрической интерпретацией траектории поиска, которая получается по алгоритмам F^{29} и F^{30} является ломаная, состоящая из отрезков прямых, параллельных осям координат.

Недостатком методов покоординатного спуска (алгоритмы F^{29} и F^{30}) является то, что при минимизации функций, имеющих овраг, дно которого не ориентировано вдоль какой-то из координатных осей, процесс поиска сильно замедляется и может остановиться далеко от точки истинного минимума x^*.

В связи с этим рассмотрим алгоритм F^{31}, реализующий **метод конфигураций**, который позволяет осуществлять поиск вдоль произвольно ориентированного относительно координатных осей дна оврага.

Процесс поиска начинается из начального приближения x°, которое принимается за базовую точку x^r, характеризующуюся тем, что она является исходной точкой очередной итерации. Каждая итерация состоит из двух процедур: «пробного движения» в Δ-окрестности текущей точки испытания и «движения в допустимом направлении», т.

е. в направлении вдоль которого гарантируется уменьшение функции $Q(x)$. Процедура «пробного движения» заключается в обследовании Δ-окрестности базовой точки x^r с целью определения допустимого (удачного в смысле уменьшения функции $Q(x)$) направления S^r. Для этого в циклическом порядке, начиная с $i = 1$, по формуле (1) изменяется каждая переменная x_i, $i = 1,2, \ldots, n$, где размер шага вдоль координатного направления I_i выбирается из условия (2). При этом начальный размер шага Δ_i для каждой из переменных может иметь различные значения. Если полученное значение λ_i^r не равно нулю, то при выполнении пробного движения вдоль $(i+1)$-й координаты в качестве значения $Q(x^r)$ рассматривается либо $Q(x^r+\Delta_i I_i)$ (если $\lambda_i^r=\Delta_i$), либо $Q(x^r - \Delta_i I_i)$ (если $\lambda_i^r = -\Delta_i$). После просмотра всех координатных направлений I_i получается точка x_n^r, в которой значение функции $Q(x_n^r)$ меньше или равно значению функции в базовой точке $Q(x^r)$. Если окажется, что $x_n^r = x^r$ т. е. величина принятого пробного шага Δ настолько велика, что не позволяет определить допустимого направления, то необходимо его уменьшить ($\Delta_i = \Delta_i/\beta$, $\beta > 1$) и повторить пробные движения снова. Таким образом, по мере приближения к точке минимума x^* длина пробного шага Δ уменьшается. Поиск считается законченным, если размер всех пробных шагов Δ_i, $i=1, 2, \ldots, n$, станет меньше заданной точности ε. В случае выполнения неравенства

$$Q(x_n^r) < Q(x^r)$$

в качестве допустимого направления S^r выбирается вектор $(x_n^r - x^r)$, который указывает направление поиска вдоль дна оврага минимизируемой функции. Периодическое повторение пробных движений позволяет подстраивать траекторию поиска вдоль дна оврага в тех случаях, когда (вследствие криволинейности оврага) установленное на предыдущей r-й итерации допустимое направление S^r оказывается неудачным для $(r + 1)$-й итерации.

Процедура «движения в заданном направлении» сводится к следующей последовательности действий. Вдоль направления определяется по формуле

$$x_i^{r+1} = x^r + h(x_n^r - x^r), \qquad (5)$$

где $h > 1$ шаг вдоль допустимого направления.

После каждого шага $i = 1, 2,\ldots$, вдоль допустимого направления относительно точки x_i^{r+1} проводится процедура «пробного движения», целью которой является определение, не нуждается ли направление S в коррекции. Если полученная после проведения n пробных движений точка x_{in}^{r+1} не совпадает с точкой x_i^{r+1}, то в качестве скорректированного допустимого направления выбирается вектор $(x_{in}^{r+1} - x_i^{r+1})$, вдоль которого делается шаг $h > 1$:

$$x_{i+1}^{r+1} = x_i^r + h(x_{in}^{r+1} - x_i^{r+1}), \qquad (6)$$

где x_i^{r+1} — «удачная точка» вдоль допустимого направления S^r. Если точка x_{in}^r лежит на одной прямой с точками x^r и x_n^r, то направление S^r сохраняется (не корректируется). В обоих случаях вычисление функции $Q(x)$ вдоль допустимого направления продолжается до тех пор, пока в очередных точках испытания x_{i+1}^{r+1} получаются уменьшающиеся значения функции $Q(x)$. Когда в допустимом направлении не удается найти точку испытания x_{i+1}^{r+1} с меньшим значением функции $Q(x)$, то поиск в направлении S^r считается законченным. В этом случае точка предыдущего удачного испытания x_i^{r+1} выбирается в качестве базовой точки для $(r+1)$-й итерации, из которой делается пробное движение с целью определения нового допустимого направления S^{r+1}.

На рис. 1 показана траектория поиска, реализующая пробные движения и движения в допустимом направлении для функции $Q(x_1, x_2)$ «овражного» типа.

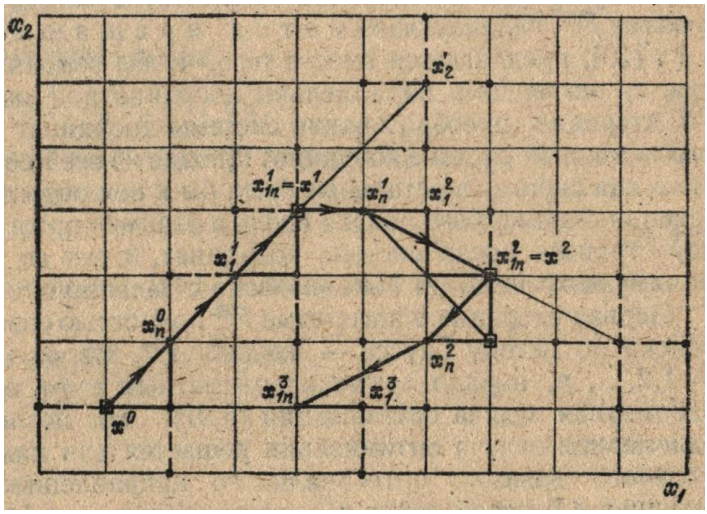


Рис. 1. Траектория поиска по методу конфигураций минимума функции Q(x) с «криволинейным» оврагом

Применение алгоритма F^{31} оказывается эффективным при минимизации функций Q (x) с «прямолинейными оврагами». В этом случае экспериментально показано, что число испытаний, необходимое для локализации точки минимума x^* с заданной точностью ε, прямо пропорционально числу переменных n.

Недостатком алгоритма является то, что в процессе проведения пробных движений направление дна оврага может быть пропущено, так как пробные шаги делаются только параллельно координатным осям. По этой же причине поиск может «остановиться» на дне оврага вдали от точки истинного минимума x^*, если в базовой точке линии уровня минимизируемой функции (Q (x) = const) очень изогнуты.

Алгоритм метода конфигураций (метод Хука-Дживса)

Алгоритм метода включает в себя два основных этапа поиска. В начале обследуется окрестность выбранной точки (базисной точки), в результате чего определяется приемлемое направление спуска. Затем в этом направлении определяется точка с наименьшим значением целевой функции. Таким образом находится новая базисная точка.

Эта процедура продолжается до тех пор, пока в окрестностях базисных точек удается находить приемлемые направления спуска.

Алгоритм

Шаг 1. Задаются начальное приближение (первая базисная точка)

$$\boldsymbol{x}_0 = \{\boldsymbol{x}_0^{(1)}, \boldsymbol{x}_0^{(2)}, \ldots, \boldsymbol{x}_0^{(n)}\},$$

начальный шаг h для поиска направления спуска, точность решения δ (предельное значение для шага h). Присваивается k=0.

Шаг 2. (Первый этап). Определяется направление минимизации целевой функции $f(x)=f(x^{(1)}, x^{(2)},\ldots,x^{(n)})$ в базисной точке

$$\boldsymbol{x}_0 = \{\boldsymbol{x}_0^{(1)}, \boldsymbol{x}_0^{(2)}, \ldots, \boldsymbol{x}_0^{(n)}\}.$$

Для этого последовательно дают приращение переменным $x^{(j)}$ в точке x_k. Присвоим $z=x_k$. Циклически даем приращение переменным $x^{(j)}$ и формируем $z^{(j)}=x_k^{(j)}+h$, если $f(z)<f(x_k)$, если же нет, то $z^{(j)}=x_k^{(j)}-h$, если $f(z)<f(x_k)$, иначе $z^{(j)}=x_k^{(j)}$. Так для всех j (j=1,2,…,n).

Шаг 3. Если $z=x_k$, то есть не определилось подходящее направление, то обследование окрестности базисной точки $x_к$ повторяется, но с меньшим шагом h (например, $h=h/2$).

Если $h>\delta$, то перейти к шагу 2, то есть повторить обследование точки x_k.

$$\tilde{\mathbf{x}} = \mathbf{x}_k, \tilde{\mathbf{y}} = \mathbf{f}(\mathbf{x}_k)$$

Если $h\le\delta$, то поиск заканчивается, то есть достигнуто предельное значение для шага h и найти приемлемое направление спуска не удается. В этом случае полагается .

Шаг 4. (Второй этап). Если $z\neq x_k$, то требуется найти новую базисную точку в направлении вектора z-x_k: $x_{k+1}=x_k + \lambda(z-x_k)$, где λ - коэффициент «ускорения поиска». Определяется такое значение $\lambda=\lambda_k$, при котором достигается наименьшее значение целевой функции в выбранном направлении, то есть функции $f(x_k +\lambda(z-x_k) = \varphi(\lambda)$.

В зависимости от способа выбора λ_k возможны варианты метода:

а) $\lambda_k=\lambda$=const постоянная для всех итераций;

б) задается начальное $\lambda_0=\lambda$, а далее $\lambda_k=\lambda_{k-1}$, если $f(x_{k+1})<f(x_k)$, иначе дробим λ_k, пока не выполнится это условие;

в) λ_k определяется решением задачи одномерной минимизации функции $\varphi(\lambda)$.

Таким образом определяется новая базисная точка $x_{k+1}=x_k + \lambda(z-x_k)$. Полагаем $k=k+1$ и поиск оптимального решения повторяется с шага 2.

Для устранения отмеченного выше недостатка метода конфигураций в алгоритме F[32], реализующем **метод вращающихся координат**, предлагается вместо того, чтобы изменять каждую переменную x_i независимо параллельно координатной оси, осуществлять на r-й итерации преобразование системы координат (x) таким образом, чтобы в новой системе координат (ξ) одна из осей совпадала с направлением дна оврага, а остальные были бы к ней ортогональны. После проведения одномерного поиска вдоль n взаимно ортогональных направлений строится новая система координат, и так до тех пор, пока точка минимума x^* не будет локализована с заданной точностью ε.

Первая итерация в алгоритме F[32] полностью совпадает с процедурой поиска по методу Гаусса — Зейделя F[30]. Вдоль направлений I_i, i = 1,2, …, n, параллельных координатным осям, поочередно решается одномерная задача оптимизации (4). На последующих итерациях одномерная задача оптимизации решается для каждого линейно-независимого взаимно ортогонального направления ξ_i, i = 1, 2, …, n. Начиная с базовой точки x^r, определяется шаг $\lambda_1{}^r$ вдоль направления $\xi_1{}^r$, при котором достигается min Q ($x^r + \lambda_1\xi_1{}^r$).

3.9. Метод деформируемого многогранника

Впервые метод деформируемого многогранника был предложен Нелдером и Мидом. Они предложили метод поиска, оказавшийся весьма эффективным и легко осуществляемым на ЭВМ. Чтобы можно было оценить стратегию Нелдера и Мида, кратко опишем симплексный поиск Спендли, Хекста и Химсворта, разработанный в связи со статистическим планированием эксперимента. Вспомним, что регулярные многогранники в E^n являются симплексами. Например, как видно из рисунка 1, для случая двух переменных регулярный симплекс представляет собой равносторонний треугольник (три точки); в случае трёх переменных

регулярный симплекс представляет собой тетраэдр (четыре точки) и т.д.

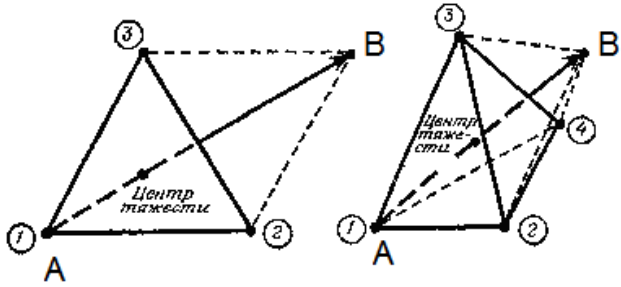


Рис. 1.
Регулярные симплексы для случая двух (а) и трёх (б) независимых переменных.

① обозначает наибольшее значение *f(x)*. Стрелка указывает направление наискорейшего улучшения.

При поиске минимума целевой функции *f(x)* пробные векторы *x* могут быть выбраны в точках E^n, находящихся в вершинах симплекса, как было первоначально предложено Спендли, Хекстом и Химсвортом. Из аналитической геометрии известно, что координаты вершин регулярного симплекса определяются следующей матрицей D, в которой столбцы представляют собой вершины, пронумерованные от 1 до (*n*+1), а строчки – координаты, *i* принимает значения от 1 до *n*:

$$D = \begin{bmatrix} 0 & d_1 & d_2 & \dots & d_2 \\ 0 & d_2 & d_1 & \dots & d_2 \\ 0 & d_2 & d_2 & \dots & d_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & d_2 & d_2 & \dots & d_1 \end{bmatrix} \text{ – матрица } n \times (n+1),$$

где

$$d_1 = \frac{t}{n\sqrt{2}}\left(\sqrt{n+1}+n-1\right),$$

$$d_2 = \frac{t}{n\sqrt{2}}\left(\sqrt{n+1}-1\right),$$

t – расстояние между двумя вершинами. Например, для n=2 и t=1 треугольник, приведённый на рисунке 1, имеет следующие координаты:

Вершина	$x_{1,i}$	$x_{2,i}$
1	0	0
2	0.965	0.259
3	0.259	0.965

Целевая функция может быть вычислена в каждой из вершин симплекса; из вершины, где целевая функция максимальна (точка А на рисунке 1), проводится проектирующая прямая через центр тяжести симплекса. Затем точка А исключается и строится новый симплекс, называемый *отражённым*, из оставшихся прежних точек и одной новой точки В, расположенной на проектирующей прямой на надлежащем расстоянии от центра тяжести. Продолжение этой процедуры, в которой каждый раз вычёркивается вершина, где целевая функция максимальна, а также использование правил уменьшения размера симплекса и предотвращения циклического движения в окрестности экстремума позволяют осуществить поиск, не использующий производные и в котором величина шага на любом этапе *k* фиксирована, а направление поиска можно изменять. На рисунке 2 приведены последовательные симплексы, построенные в двумерном пространстве с «хорошей» целевой функцией.

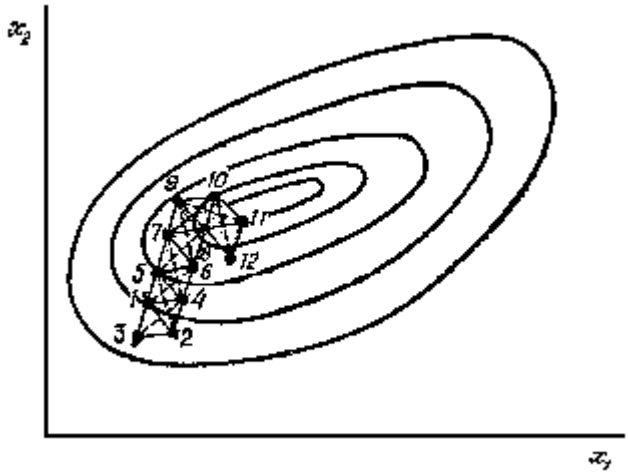


Рис. 2.
Последовательность регулярных симплексов, полученных при минимизации *f(x).*
----- проекция

Определённые практические трудности, встречающиеся при использовании регулярных симплексов, а именно отсутствие ускорения поиска и трудности при проведении поиска на искривлённых «оврагах» и «хребтах», привели к необходимости некоторых улучшений методов. Далее будет изложен метод Нелдера и Мида, в котором симплекс может изменять свою форму и таким образом уже не будет оставаться симплексом. Именно поэтому здесь использовано более подходящее название «деформируемый многогранник».

В методе Нелдера и Мида минимизируется функция n независимых переменных с использованием $n+1$ вершин деформируемого многогранника в E^n. Каждая вершина может быть идентифицирована вектором x. Вершина (точка) в E^n, в которой значение $f(x)$ максимально, проектируется через центр тяжести (центроид) *оставшихся* вершин. Улучшенные (более низкие) значения целевой функции находятся последовательной заменой точки с максимальным значением $f(x)$ на более «хорошие точки», пока не будет найден минимум $f(x)$.

Более подробно этот алгоритм может быть описан следующим образом.

Пусть

$$x_i^{(k)} = \left[x_{i1}^{(k)}, \ldots, x_{ij}^{(k)}, \ldots, x_{in}^{(k)}\right]^T, i = 1, \ldots, n+1,$$

является *i*-й вершиной (точкой) в E^n на *k*-м этапе поиска, *k*=0, 1, …, и пусть значение целевой функции в $x^{(k)}_i$ равно $f(x^{(k)}_i)$. Кроме того, отметим те векторы *x* многогранника, которые дают максимальное и минимальное значения *f(x).*

Определим

$$f\left(x_h^{(k)}\right) = \max\left\{f\left(x_1^{(k)}\right), \ldots, f\left(x_{n+1}^{(k)}\right)\right\},$$

где $x_h^{(k)} = x_i^{(k)}$, и

$$f\left(x_l^{(k)}\right) = \min\left\{f\left(x_1^{(k)}\right), \ldots, f\left(x_{n+1}^{(k)}\right)\right\},$$

где $x_l^{(k)} = x_i^{(k)}$.

Поскольку многогранник в E^n состоит из (n+1) вершин $x_1, \ldots, x_{n+1}$, пусть x_{n+2} будет центром тяжести всех вершин, исключая x_h.

Тогда координаты этого центра определяются формулой

$$x_{n+2,j}^{(k)} = \frac{1}{n}\left[\left(\sum_{i=1}^{n+1} x_{ij}^{(k)}\right) - x_{hj}^{(k)}\right], \quad j = 1, \ldots, n, \qquad (1)$$

где индекс j обозначает координатное направление.

Начальный многогранник обычно выбирается в виде регулярного симплекса (но это не обязательно) с точкой 1 в качестве начала координат; можно начало координат поместить в центр тяжести. Процедура отыскания вершины в E^n, в которой $f(x)$ имеет лучшее значение, состоит из следующих операций:

1. *Отражение* – проектирование $x^{(k)}_h$ через центр тяжести в соответствии с соотношением

$$x_{n+3}^{(k)} = x_{n+2}^{(k)} + \alpha\left(x_{n+2}^{(k)} - x_h^{(k)}\right), \qquad (2)$$

где $\alpha>0$ является коэффициентом отражения; $x_{n+2}^{(k)}$ – центр тяжести, вычисляемый по формуле $\max\Delta_i < \varepsilon$.; $x_h^{(k)}$ – вершина, в которой функция *f(x)* принимает наибольшее из *n*+1 значений на *k*-м этапе.

2. *Растяжение*. Эта операция заключается в следующем: если $f\left(x_{n+3}^{(k)}\right) \le f\left(x_l^{(k)}\right)$, то вектор $\left(x_{n+3}^{(k)} - x_{n+2}^{(k)}\right)$ растягивается в соответствии с соотношением

$$x_{n+4}^{(k)} = x_{n+2}^{(k)} + \gamma\left(x_{n+3}^{(k)} - x_{n+2}^{(k)}\right) \qquad (3)$$

где $\gamma>1$ представляет собой коэффициент растяжения. Если $f\left(x_{n+4}^{(k)}\right) \le f\left(x_l^{(k)}\right)$, то $x_h^{(k)}$ заменяется на $x_{n+4}^{(k)}$ и процедура продолжается снова с операции 1 при $k=k+1$. В противном случае $x_h^{(k)}$ заменяется на $x_{n+3}^{(k)}$ и также осуществляется переход к операции 1 при $k=k+1$.

3. *Сжатие*. Если $f\left(x_{n+3}^{(k)}\right) > f\left(x_i^{(k)}\right)$ для всех $i \ne h$, то вектор $\left(x_h^{(k)} - x_{n+2}^{(k)}\right)$ сжимается в соответствии с формулой

$$x_{n+5}^{(k)} = x_{n+2}^{(k)} + \beta\left(x_h^{(k)} - x_{n+2}^{(k)}\right) \qquad (4)$$

где $0<\beta<1$ представляет собой коэффициент сжатия. Затем $x_h^{(k)}$ заменяем на $x_{n+5}^{(k)}$ и возвращаемся к операции 1 для продолжения поиска на $(k+1)$-м шаге.

4. *Редукция*. Если $f\left(x_{n+3}^{(k)}\right) > f\left(x_h^{(k)}\right)$, все векторы $\left(x_i^{(k)} - x_l^{(k)}\right)$, $i = 1, \ldots, n+1$, уменьшаются в 2 раза с отсчётом от $x_l^{(k)}$ в соответствии с формулой

$$x_i^{(k)} = x_l^{(k)} + 0{,}5\left(x_i^{(k)} - x_l^{(k)}\right), \quad i = 1, \ldots, n+1. \qquad (5)$$

Затем возвращаемся к операции 1 для продолжения поиска на $(k+1)$-м шаге.

Критерий окончания поиска, использованный Нелдером и Мидом, состоял в проверке условия

$$\left\{\frac{1}{n+1}\sum_{i=1}^{n+1}\left[f\left(x_i^{(k)}\right) - f\left(x_{n+2}^{(k)}\right)\right]^2\right\}^{1/2} \le \varepsilon, \qquad (6)$$

где ε – произвольное малое число, а $f\left(x_{n+2}^{(k)}\right)$ – значение целевой функции в центре тяжести $x_{n+2}^{(k)}$.

На схеме 1 приведена блок-схема поиска методом деформируемого многогранника, а на рисунке 3 показана последовательность поиска для функции Розенброка, начиная их $x^{(0)}=[-1{,}2\ \ 1{,}0]^T$. Деформируемый многогранник в противоположность жёсткому симплексу адаптируется к топографии целевой функции, вытягиваясь вдоль длинных наклонных плоскостей, изменяя направление в изогнутых впадинах и сжимаясь в окрестности минимума.

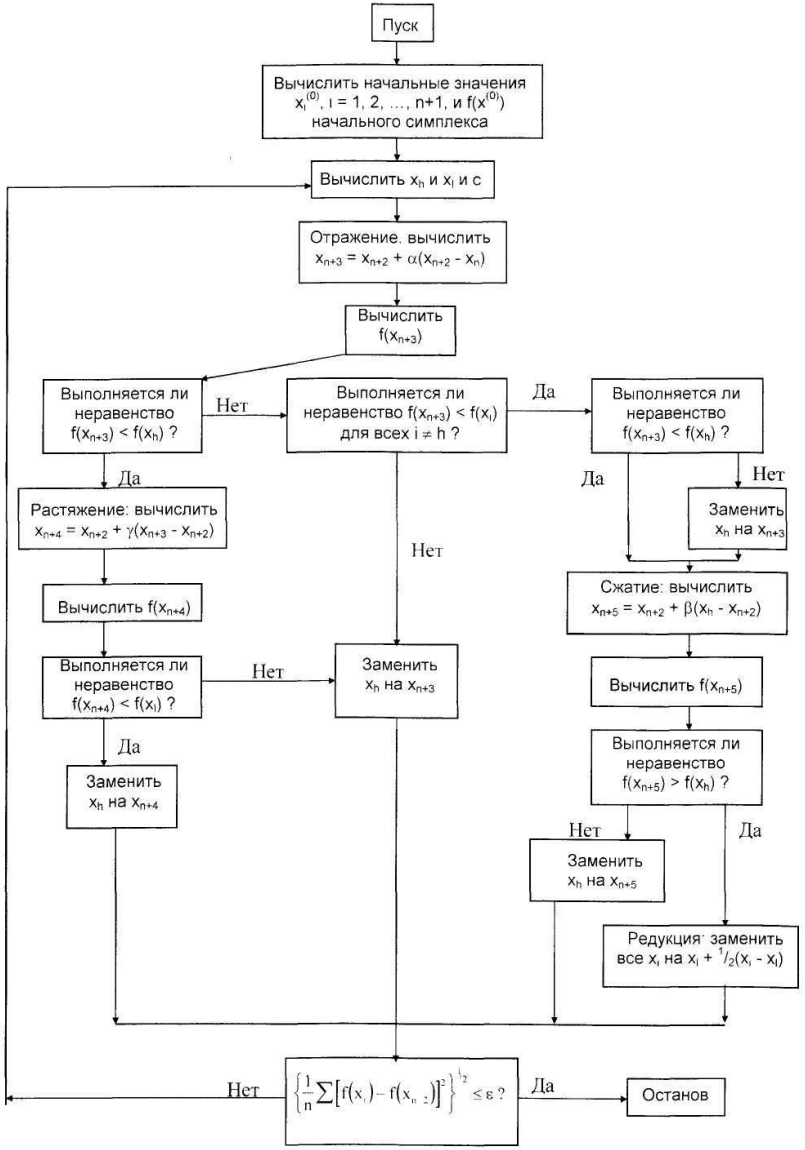


Схема 1. Блок-схема поиска методом деформируемого многогранника

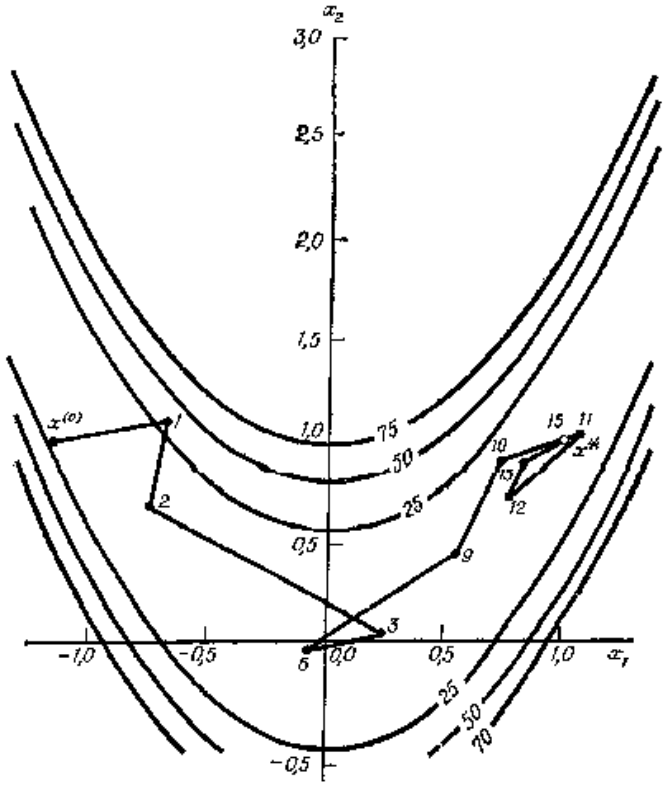


Рис. 3.
Поиск минимума функции Розенброка методом деформируемого многогранника, начиная с точки $x(0)=[-1,2\ 1,0]T$ (числа указывают номер шага).

Коэффициент отражения α используется для проектирования вершины с наибольшим значением $f(x)$ через центр тяжести деформируемого многогранника. Коэффициент γ вводится для растяжения вектора поиска в случае, если отражение даёт вершину со значением $f(x)$, меньшим, чем наименьшее значение $f(x)$, полученное до отражения. Коэффициент сжатия β используется для уменьшения вектора поиска, если операция отражения не привела к вершине со значением $f(x)$, меньшим, чем второе по величине (после наибольшего) значение $f(x)$, полученное до отражения. Таким образом, с помощью операций растяжений или сжатия размеры и форма деформируемого

многогранника масштабируются так, чтобы они удовлетворяли топологии решаемой задачи.

Естественно возникает вопрос, какие значения параметров α, β и γ должны быть выбраны. После того как деформируемый многогранник подходящим образом промасштабирован, его размеры должны поддерживаться неизменными, пока изменения в топологии задачи не потребуют применения многогранника другой формы. Это возможно реализовать только при $\alpha=1$. Кроме того, Нелдер и Мид показали, что при решении задачи с $\alpha=1$ требуется меньшее количество вычислений функции, чем при $\alpha<1$. С другой стороны, α не должно быть много больше единицы, поскольку

1) деформируемый многогранник легче адаптируется к топологии задачи при меньших значениях α, особенно когда необходимо изменять направление поиска, столкнувшись с изогнутой впадиной, и
2) в области локального минимума размеры многогранника должны уменьшаться и большое α в этих условиях замедлит сходимость.

Таким образом, значение $\alpha=1$ выбирается как компромисс.

Чтобы выяснить, какое влияние на процедуру поиска имеет выбор β и γ, Нелдер и Мид (а также Павиани) провели решение нескольких тестовых задач, используя большое число различных комбинаций значений β и γ. В качестве удовлетворительных значений этих параметров при оптимизации без ограничений Нелдер и Мид рекомендовали $\alpha=1$, $\beta=0,5$ и $\gamma=2$. Размеры и ориентация исходного многогранника в некоторой степени влияли на время решения, а значения α, β и γ оказывали значительно большее влияние. Павиани отмечает, что нельзя чётко решить вопрос относительно выбора β и γ и что влияние выбора β на эффективность поиска несколько более заметно, чем влияние γ. Павиани рекомендует следующие диапазоны значений для этих параметров:

$$0,4 \le \beta \le 0,6,$$
$$2,8 \le \gamma \le 3,0.$$

При $0<\beta<0,4$ существует вероятность того, что из-за уплощения многогранника будет иметь место преждевременное окончание процесса. При $\beta>0,6$ может потребоваться избыточное число шагов и больше машинного времени для достижения окончательного решения.

Пример

Поиск методом деформируемого многогранника.

Для иллюстрации метода Нелдера и Мида рассмотрим задачу минимизации функции $f(x)=4(x_1-5)^2+(x_2-6)^2$, имеющей минимум в точке $x^*=[5\ 6]^T$. Поскольку $f(x)$ зависит от двух переменных, в начале поиска используется многоугольник с тремя вершинами. В этом примере в качестве начального многогранника взят треугольник с вершинами $x_1^{(0)}=[8\ 9]^T$, $x_2^{(0)}=[10\ 11]^T$ и $x_3^{(0)}=[8\ 11]^T$, хотя можно было бы использовать любую другую конфигурацию из трёх точек.

На нулевом этапе поиска, $k=0$, вычисляя значения функции, получаем $f(8,9)=45$, $f(10,11)=125$ и $f(8,11)=65$. Затем отражаем $x_2^{(0)}=[10\ 11]^T$ через центр тяжести точек $x_1^{(0)}$ и $x_3^{(0)}$ [по формуле (64)], который обозначим через $x_4^{(0)}$:

$$x_{4,1}^{(0)}=\frac{1}{2}[8+10+8-10]=8,$$

$$x_{4,2}^{(0)}=\frac{1}{2}[9+11+11-11]=10$$

с тем, чтобы получить $x_5^{(0)}$.

$$x_{5,1}^{(0)}=8+1(8-10)=6,$$

$$x_{5,2}^{(0)}=10+1(10-11)=9,$$

$$f(6,9)=13.$$

Поскольку $f(6,9)=13<f(8,9)=45$, переходим к операции растяжения:

$$x_{6,1}^{(0)}=8+2(6-8)=4,$$

$$x_{6,2}^{(0)}=10+2(9-10)=8,$$

$$f(4,8)=8.$$

Поскольку $f(4,8)=8<f(8,9)=45$, заменяем $x_2^{(0)}$ на $x_6^{(0)}$ и полагаем $x_6^{(0)}=x_2^{(1)}$ на следующем этапе поиска.

Наконец, поскольку

$$\frac{1}{3}[7^2+13^2+44^2]^{1/2}=26{,}8>10^{-6},$$

начинаем этап поиска $k=1$. На рисунке 4 приведена траектория поиска на начальных этапах. На рисунке 5 изображена полная траектория поиска до его окончания. Для уменьшения $f(x)$ до значения $1\cdot10^{-6}$ потребовалось 32 этапа.

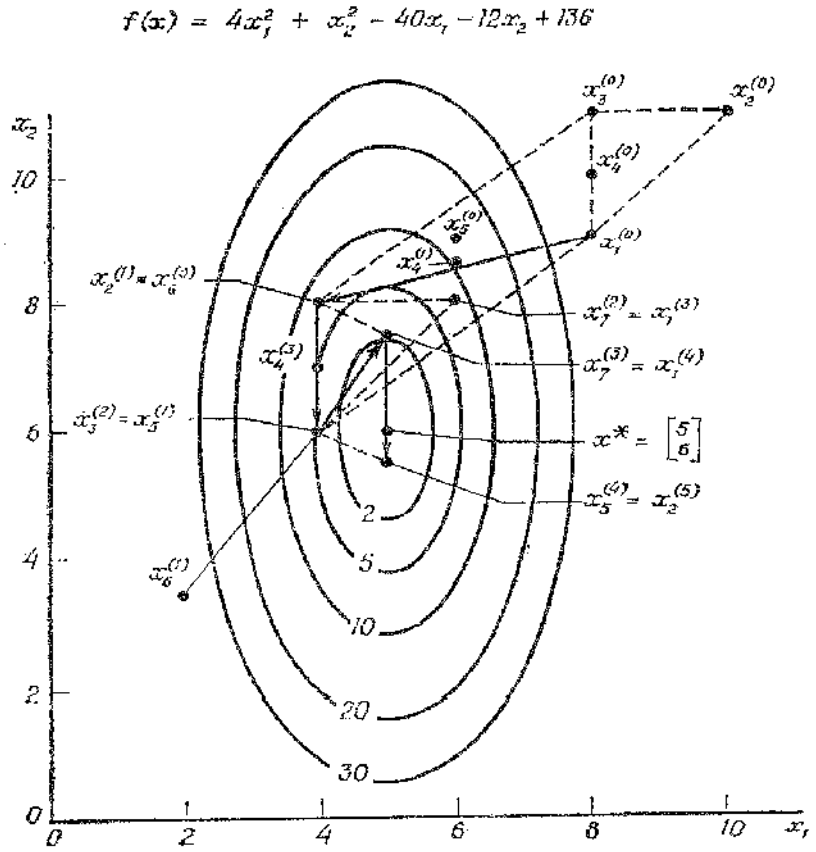


Рис.4. Метод Нелдера и Мида при отсутствии ограничений.

Рис. 5.
Траектория поиска с помощью алгоритма Нелдера и Мида.

С помощью операции растяжения и сжатия размеры и форма деформируемого многогранника адаптируются к топографии целевой функции. В результате многогранник вытягивается вдоль длинных наклонных поверхностей, изменяет направление в изогнутых впадинах,

сжимается в окрестности минимума, что определяет эффективность рассмотренного метода.

Алгоритм метода симплекса

Напомним, что под симплексом понимается n-мерный выпуклый многогранник n-мерного пространства, имеющий $n+1$ вершину. Для $n=2$ это треугольник, а при $n=3$ это тетраэдр.

Идея метода состоит в сравнении значений функции в $n+1$ вершинах симплекса и перемещении симплекса в направлении лучшей точки. В рассматриваемом методе симплекс перемещается с помощью операций отражения. Далее принято следующее: $x_0(k), x_1(k), \ldots, x_n(k)$ – вершины симплекса, где k - номер итерации.

Алгоритм

Шаг 1. Построение начального симплекса. Задаются начальная точка $x_0(0)$ и длина ребра симплекса l. Формируются остальные вершины симплекса:

$x_i(0) = x_0(0) + l*e_i$ (i=1,2,…,n), где e_i – единичные векторы.

Шаг 2. Определение направления улучшения решения. Для этого на k-й итерации вычисляются значения целевой функции в каждой точке симплекса. Пусть для всех i:

$$f(x_{\min}(k)) \le f(x_i(k)) \le f(x_{\max}(k)),$$

где min, max, i – номера соответствующих вершин симплекса. Определим центр тяжести всех точек, исключая точку $x_{\max}(k)$,

$$C_k=(\Sigma x_i(k))/n .$$

Тогда направление улучшения решения определяется вектором C_k-$x_{\max}(k)$.

Шаг 3. Построение отраженной точки. Замена вершины $x_{\max}(k)$ с максимальным значением целевой функции на новую точку с помощью операции отражения, результатом которой является новая точка:

$$u_k=c_k+(c_k-x_{\max}(k))=2c_k-x_{\max}(k)$$

Шаг 4. Построение нового симплекса. Вычисляем $f(u_k)$. При этом возможен один из двух случаев:

а) $f(u_k)<f(x_{\max}(k))$;

б) $f(u_k)\ge f(x_{\max}(k))$.

а) Вершина $x_{\max}$ заменяется на u_k, чем определяется набор вершин $k+1$-й итерации и k-я итерация заканчивается.

б) В результате отражения получается новая точка u_k, значение функции в которой еще хуже, чем в точке $x_{\max}$, то есть отражать

симплекс некуда. Поэтому в этом случае производится пропорциональное уменьшение симплекса (например, в 2 раза) в сторону вершины $x_{\min}(k)$:

$$x_i(k+1)=x_i=(x_i(k)+x_{\min}(k))/2, \text{ где } i=0,1,\ldots,n.$$

На этом k-я итерация заканчивается.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(f\left(x_i(k+1)\right)-f\left(x_0(k+1)\right)\right)^2}\le\varepsilon,$$

Шаг 5. Проверка сходимости. Если

$$\tilde{x}=x_0(k+1),\ \tilde{y}=f(x_0(k+1))$$

то поиск минимума заканчивается и полагается
В противном случае $k=k+1$ и происходит переход к шагу 2.

Алгоритм метода деформируемого симплекса (метод Нелдера – Мида)

Метод деформируемого симплекса обладает большей общностью и позволяет учитывать локальные свойства поверхности целевой функции. Симплексы вытягиваются в направлении наклона поверхности, их оси поворачиваются при встрече с оврагом на поверхности целевой функции, вблизи минимума они сжимаются.

В рассматриваемом методе симплекс перемещается с помощью трех основных операций над симплексом: отражение, растяжение и сжатие.

Алгоритм

Шаг 1. Построение начального симплекса. Задаются начальная точка $x_0(0)$ и длина ребра l. Формируются остальные вершины симплекса: $x_i(0)=x_0(0)+le_i$ ($i=1,2,\ldots,n$), где e_i – единичные векторы.

Шаг 2. Определение направления улучшения решения. Для этого на каждой итерации вычисляются значения целевой функции в каждой вершине симплекса. Пусть для всех i

$$f(x_{\min}(k)) \le f(x_i(k)) \le f(x_m(k)) \le f(x_{\max}(k)),$$

где min, m, max, i-номера соответствующих вершин симплекса. Определим центр тяжести всех точек, исключая точку $x_{\max}(k)$,

$$C_k = \frac{1}{n} \sum_{i \neq max} X_i(k)$$

Тогда направление улучшения решения определяется векторов $C_k - x_{max}(k)$.

Шаг 3. Построение нового симплекса. Замена вершины $x_{max}(k)$ с максимальным значением целевой функции на новую точку с помощью операции отражения, результат которой является новая точка $u_k=C_k+\alpha*(C_k-x_{max}(k))$, где α-коэффициент отражения.

Шаг 4. Построение нового симплекса. Вычисляем $f(u_k)$, при этом возможно один из трех случаев:

а) $f(u_k)< f(x_{min}(k))$;

б) $f(u_k)>f(x_m(k))$;

в) $f(x_{min}(k)) \leq f(u_k) \leq f(x_m(k))$;

а) Отражённая точка является точкой с наилучшим значением целевой функции. Поэтому направление отражение является перспективным и можно попытаться растянуть симплекс в этом направлении. Для этого строиться точка

$V_k= C_k+\beta*(u_k-C_k)$, где $\beta>1$ –коэффициент расширения.

Если $f(v_k)<f(u_k)$, то вершина $x_{max}(k)$ заменяется на v_k, в противном случае на u_k и k-ая итерация заканчивается.

б) В результате отражения получается новая точка u_k, которая, если заменить $x_{max}(k)$, сама станет наихудшей. Поэтому в этом случае производится сжатие симплекса. Для этого строится точка v_k:

$C_k+\gamma*(x_{max}(k)-C_k)$, если $f(x_{max}(k))\ f(u_k)$,

$v_k= \quad C_k+\gamma*(u_k-C_k)$, если $f(x_{max}(k))> f(u_k)$,\

$$v_k = \begin{cases} C_k+\gamma*(x_{max}(k)-C_k), & \text{если } f(x_{max}(k)) \leq f(u_k), \\ C_k+\gamma*(u_k-C_k), & \text{если } f(x_{max}(k))> f(u_k), \end{cases}$$

где $0<\gamma<1$ – коэффициент сжатия.

Если $f(v_k)<\min\{f(x_{max}(k)),f(u_k)\}$, то вершина $x_{max}(k)$ заменяется на v_k .

В противном случае вершинам $x_i(k+1)$ ($i=0,1,2,..,n$) присваивается значение:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}_i(\mathbf{k}) + \mathbf{x}_{\min}(\mathbf{k})}{2}$$

и на этом *k*-ая итерация заканчивается.

в) Вершина $x_{max}(k)$ заменяется на u_k, чем определяется набор вершин k+1-й итерации и k–ая итерация заканчивается.

Шаг 5. Проверка сходимости. Если

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(f(x_i(k+1)) - f(x_0(k+1))\right)^2} \le \varepsilon,$$

то поиск минимума заканчивается и полагается

$$\tilde{x} = x_0(k+1),\ \tilde{y} = f(x_0(k+1))$$

В противном случае $k=k$+1 и происходит переход к шагу 2.

Опыт использования описанного алгоритма показывает, что целесообразно брать следующие значения параметров: α=1, β=2, γ=0.5.

3.10. Метод прямого поиска (метод Хука-Дживса)

Суть этого метода состоит в следующем. Задаются некоторой начальной точкой $x[0]$. Изменяя компоненты вектора $x[0]$, обследуют окрестность данной точки, в результате чего находят направление, в котором происходит уменьшение минимизируемой функции $f(x)$. В выбранном направлении осуществляют спуск до тех пор, пока значение функции уменьшается. После того как в данном направлении не удается найти точку с меньшим значением функции, уменьшают величину шага спуска. Если последовательные дробления шага не приводят к уменьшению функции, от выбранного направления спуска отказываются и осуществляют новое обследование окрестности и т. д.

Алгоритм метода прямого поиска состоит в следующем.

1. Задаются значениями координат $x_i[0]$, i = 1, ..., n , начальной точки $x[0]$, вектором изменения координат Dx в процессе обследования окрестности, наименьшим допустимым значением ε компонентов Dx.

2. Полагают, что $x[0]$ является *базисной точкой* $x^б$, и вычисляют значение $f(x^б)$.

3. Циклически изменяют каждую координату $x^{б}_i$, $i = 1, ..., n$, базисной точки $x^{б}$ на величину εx_i, $i = 1, ..., n$, т. е. $x_i[k] = x^{б} + \Delta x$; $x_i[k] = x^{б}_i - \varepsilon x_i$. При этом вычисляют значения $f(x[k])$ и сравнивают их со значением $f(x^{б})$. Если $f(x[k]) < f(x^{б})$, то соответствующая координата x_i, $i = 1, ..., n$, приобретает новое значение, вычисленное по одному из приведенных выражений. В противном случае значение этой координаты остается неизменным. Если после изменения последней n-й координаты $f(x[k]) < f(x^{б})$, то переходят к п. 4. В противном случае - к п. 7.

4. Полагают, что $x[k]$ является новой базисной точкой $x^{б}$, и вычисляют значение $f(x^{б})$.

5. Осуществляют спуск из точки $x[k] > x_i[k+1] = 2x_i[k] - x^{б}$, $i =1, ..., n$, где $x^{б}$ - координаты предыдущей базисной точки. Вычисляют значение $f(x[k+1])$.

6. Как и в п. 3, циклически изменяют каждую координату точки $x[k+1]$, осуществляя сравнение соответствующих значений функции $f(x)$ со значением $f(x[k+1])$, полученным в п. 5. После изменения последней координаты сравнивают соответствующее значение функции $f(x[k])$ со значением $f(x^{б})$, полученным в п.4. Если $f(x[k])<f(x^{б})$, то переходят к п. 4, в противном случае - к п. 3. При этом в качестве базисной используют последнюю из полученных базисных точек.

7. Сравнивают значения Δx и ε. Если $\Delta x<\varepsilon$, то вычисления прекращаются. В противном случае уменьшают значения Δx и переходят к п. 3.

Достоинством метода прямого поиска является простота его программирования на компьютере. Он не требует знания целевой функции в явном виде, а также легко учитывает ограничения на отдельные переменные, а также сложные ограничения на область поиска.

Недостаток метода прямого поиска состоит в том, что в случае сильно вытянутых, изогнутых или обладающих острыми углами линий уровня целевой функции он может оказаться неспособным обеспечить продвижение к точке минимума. Действительно, в случаях, изображенных на рис. 1, *а* и *б,* каким бы малым ни брать шаг в

направлении x_1 или x_2 из точки x' нельзя получить уменьшения значения целевой функции.

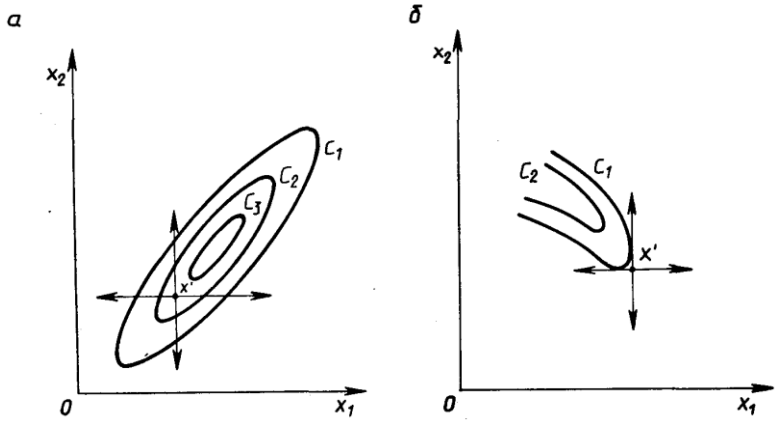


Рис. 1. Прямой поиск: невозможность продвижения к минимуму: а – $C_1 > C_2 > C_3$; б - $C_1 > C_2$

Напомним, что *поверхностью уровня* (на плоскости - *линией уровня)* является поверхность, получаемая приравниванием выражения функции *f(x)* некоторой постоянной величине *C,* т. е. $f(x) = C$. Во всех точках этой поверхности функция имеет одно и то же значение *C.* Давая величине *C* различные значения C_1, ..., C_n, получают ряд поверхностей, геометрически иллюстрирующих характер функции.

3.11. Метод вращающихся координат (метод Розенброка)

Суть метода состоит во вращении системы координат в соответствии с изменением скорости убывания целевой функции. Новые направления координатных осей определяются таким образом, чтобы одна из них соответствовала направлению наиболее быстрого убывания целевой функции, а остальные находятся из условия ортогональности. Идея метода состоит в следующем (рис. 1).

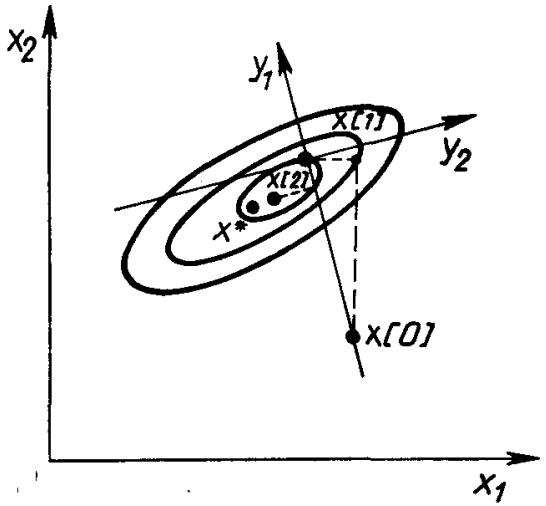


Рис. 1. Геометрическая интерпретация метода Розенброка

Из начальной точки $x[0]$ осуществляют спуск в точку $x[1]$ по направлениям, параллельным координатным осям. На следующей итерации одна из осей должна проходить в направлении $y_1 = x[1] - x[0]$, а другая - в направлении, перпендикулярном к y_1. Спуск вдоль этих осей приводит в точку $x[2]$, что дает возможность построить новый вектор $x[2] - x[1]$ и на его базе новую систему направлений поиска. В общем случае данный метод эффективен при минимизации овражных функций, так как результирующее направление поиска стремится расположиться вдоль оси оврага.

Алгоритм метода вращающихся координат состоит в следующем.

1. Обозначают через $p_1[k], ..., p_n[k]$ направления координатных осей в некоторой точке $x[k]$ (на k-й итерации). Выполняют пробный шаг h_1 вдоль оси $p_1[k]$, т. е.

$$x[k1] = x[k] + h_1 p_1[k].$$

Если при этом $f(x[kl]) < f(x[k])$, то шаг h умножают на величину $b > 1$;

Если $f(x[kl]) > f(x[k])$, - то на величину $(-b)$, $0 < |b| < 1$;

$$x[kl] = x[k] + b\, h_1 p_1[k].$$

Полагая $bh_1 = a_1$ получают

$$x[kl] = x[k] + a_1 p_1[k].$$

2. Из точки $x[k1]$ выполняют шаг h_2 вдоль оси $p_2[k]$:

$$x[k2] = x[k] + a_1 p_1[k] + h_2 p_2[k].$$

Повторяют операцию п. 1, т. е.

$$x[k2] = x[k] + a_1 p_1[k] + a_2 p_2[k].$$

Эту процедуру выполняют для всех остальных координатных осей. На последнем шаге получают точку

$$x[kn] = x[k+1] = x[k] + \sum_{i=1}^{n} a_i\, p_i[k].$$

3. Выбирают новые оси координат $p_1[k+1]$, …, $p_n[k+1]$. В качестве первой оси принимается вектор

$$p_1[k+1] = x[k+l] - x[k].$$

Остальные оси строят ортогональными к первой оси с помощью процедуры ортогонализации Грама - Шмидта. Повторяют вычисления с п. 1 до удовлетворения условий сходимости.

Коэффициенты b подбираются эмпирически. Хорошие результаты дают значения b=-0,5 при неудачных пробах $(f(x[ki]) > f(x[k]))$ и $b = 3$ при удачных пробах $(f(x[ki]) < f(x[k]))$.

В отличие от других методов нулевого порядка алгоритм Розенброка ориентирован на отыскание оптимальной точки в каждом направлении, а не просто на фиксированный сдвиг по всем направлениям. Величина шага в процессе поиска непрерывно изменяется в зависимости от рельефа поверхности уровня. Сочетание вращения координат с регулированием шага делает метод Розенброка эффективным при решении сложных задач оптимизации.

3.12. Метод параллельных касательных (метод Пауэлла)

Этот метод использует свойство квадратичной функции, заключающееся в том, что любая прямая, которая проходит через точку минимума функции x^*, пересекает под равными углами касательные к поверхностям равного уровня функции в точках пересечения (рис. 1).

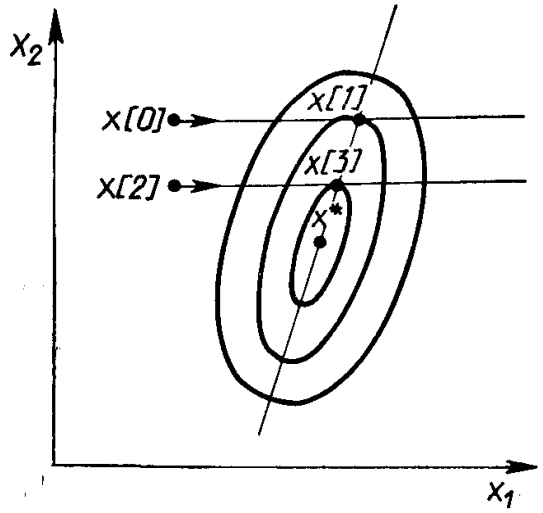


Рис. 1. Геометрическая интерпретация метода Пауэлла

Суть метода такова. Выбирается некоторая начальная точка $x[0]$ и выполняется одномерный поиск вдоль произвольного направления, приводящий в точку $x[1]$. Затем выбирается точка $x[2]$, не лежащая на прямой $x[0]$ - $x[1]$, и осуществляется одномерный поиск вдоль прямой,

параллельной $x[0]$ - $x[1]$. Полученная в результате точка $x[3]$ вместе с точкой $x[1]$ определяет направление $x[1]$ - $x[3]$ одномерного поиска, дающее точку минимума x^*. В случае квадратичной функции n переменных оптимальное значение находится за n итераций. Поиск минимума при этом в конечном счете осуществляется во взаимно сопряженных направлениях. В случае неквадратичной целевой функции направления поиска оказываются сопряженными относительно матрицы Гессе. Алгоритм метода параллельных касательных состоит в следующем.

1. Задаются начальной точкой $x[0]$. За начальные направления поиска $p[1], ..., p[0]$ принимают направления осей координат, т. е. $p[i] = e[i]$, $i = 1, ..., n$ (здесь $e[i]= (0, ..., 0, 1, 0, \dots 0)^T$).

2. Выполняют n одномерных поисков вдоль ортогональных направлений $p[i]$, $i = 1, ..., n$. При этом каждый следующий поиск производится из точки минимума, полученной на предыдущем шаге. Величина шага a_k находится из условия

$$f(x[k] + a_k p[k]) = \min_a f(x[k] + ap[k]).$$

Полученный шаг определяет точку

$$x[k+1] = x[k] + a_k p[k] .$$

3. Выбирают новое направление $p =-x[n] - x[0]$ и заменяют направления $p[1], ..., p[n]$ на $p[2], ..., p[n], p$. Последним присваивают обозначения $p[1], ..., p[n]$

4. Осуществляют одномерный поиск вдоль направления $p=p[n]=x[n] - x[0]$. Заменяют $x[0]$ на $x[n+1]=x[n]+a_n p[n]$ и принимают эту точку за начальную точку $x[0]$ для следующей итерации. Переходят к п. 1.

Таким образом, в результате выполнения рассмотренной процедуры осуществляется поочередная замена принятых вначале направлений поиска. В итоге после n шагов они окажутся взаимно сопряженными.

3.13. Краткий обзор других методов

Метод дробления шага.

В данном методе строится релаксационная последовательность точек, т.е. таких точек $\{x^k\}$, k=0,1,…, что $f(x^k) < f(x^{k-1})$, k=0,1,…. Точки последовательности $\{x^k\}$ вычисляются по следующему правилу:

$$x^{k+1}=x^k-t_k \text{grad } f(x^k),\ k=0,1,\ldots \qquad (1)$$

Начальная точка x^0 и начальный шаг t_0 задаются пользователем. Величина шага t_0 не изменяется до тех пор, пока функция убывает в точках последовательности. Это контролируется путем проверки выполнения условия $f(x^{k+1}) - f(x^k) < 0$ (или <-e). Если условие убывания не выполняется, то величина шага уменьшается, как правило, вдвое, т.е. $t_k=t_k/2$.

Метод наискорейшего градиентного спуска

Как и в предыдущем методе, точки релаксационной последовательности $\{x^k\}$, k=0,1,… вычисляются по правилу (1). Точка x^0 задается пользователем; величина шага t_k определяется из условия минимума одномерной функции $f(t_k)=f(x^k-t_k \text{grad } f(x^k))$. Задача минимизации функции $f(t_k)$ может быть решена с использованием необходимого условия минимума =0 с последующей проверкой достаточного условия минимума >0 или с использованием численных методов.

Метод сопряженных направлений (Флетчера - Ривса).

В данном методе используются свойства векторов, сопряженных относительно некоторой матрицы.

Определение. Векторы p и q называются сопряженными относительно матрицы Q, если выполняется равенство pQq=0.

Точки релаксационной последовательности $\{x^k\}$, k=0,1,… вычисляются по правилу

$$x^{k+1}=x^k-t_k d^k,\ k=0,1,\ldots;$$

$$d^k = - \text{grad } f(x^k) + в_{k-1}\, d^{k-1}; \qquad (2)$$

$$d^0 = -\operatorname{grad} f(x^0);$$

$$в_{k-1} = |\operatorname{grad} f(x^k)|^2 ? |\operatorname{grad} f(x^{k-1})|^2.$$

Точка x^0 задается пользователем; величина шага t_k определяется из условия минимума функции $f(t)=f(x^k-td^k)$. Задача минимизации одномерной функции $f(t_k)$ может быть решена с использованием необходимого условия минимума =0 с последующей проверкой достаточного условия минимума >0 или с использованием численных методов. Коэффициент $в_{k-1}$ вычисляется из условия сопряженности направлений d^k и d^{k-1}.

Метод Ньютона.

Строится последовательность точек $\{x^k\}$, k=0,1,…, таких, что, k=0,1,… Точки последовательности $\{x^k\}$ вычисляются по правилу $x^{k+1}=x^k+d^k$, k=0,1,… Точка x^0 задается пользователем с учетом знакопостоянства и невырожденности матрицы Гессе в задаваемой начальной точке и близости выбранной точки к предполагаемой точке минимума. Направление спуска определяется для каждого значения k по формуле $d^k = -H^{-1}(x^k) \operatorname{grad} f(x^k)$, где H - матрица Гессе.

4. Методы минимизации первого порядка

4.1. Минимизация функций. Основные положения

Градиентом дифференцируемой функции f(x) в точке x[0] называется n-мерный вектор $f(x$[0]$)$, компоненты которого являются частными производными функции $f(x)$, вычисленными в точке x[0], т. е.

$$f'(x[0]) = (\partial f(x[0])/\partial x_1, \ldots, \partial f(x[0])/\partial x_n)^T.$$

Этот вектор перпендикулярен к плоскости, проведенной через точку x[0], и касательной к поверхности уровня функции $f(x)$, проходящей через точку x[0]. В каждой точке такой поверхности функция $f(x)$ принимает одинаковое значение. Приравнивая функцию различным постоянным величинам C_0, C_1, ... , получим серию поверхностей, характеризующих ее топологию (рис. 1).

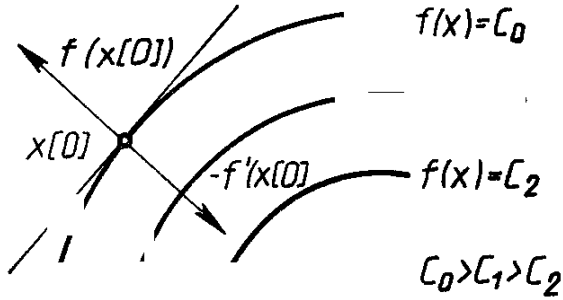


Рис. 1. Градиент

Вектор-градиент направлен в сторону наискорейшего возрастания функции в данной точке. Вектор, противоположный градиенту *(-f'(x*[0]*))*, называется *антиградиентом* и направлен в сторону наискорейшего убывания функции. В точке минимума градиент функции равен нулю. На свойствах градиента основаны методы **первого порядка, называемые также градиентными методами минимизации.** Использование этих методов в общем случае позволяет определить точку локального минимума функции.

Очевидно, что если нет дополнительной информации, то из начальной точки $x[0]$ разумно перейти в точку $x[1]$, лежащую в направлении антиградиента - наискорейшего убывания функции. Выбирая в качестве направления спуска $p[k]$ антиградиент $-f'(x[k])$ в точке $x[k]$, получаем итерационный процесс вида

$$x[k+1] = x[k]-a_k f'(x[k]),\ a_k > 0;\ k=0, 1, 2, \ldots$$

В координатной форме этот процесс записывается следующим образом:

$$x_i[k+1]=x_i[k] - a_k \partial f(x[k])/\partial x_i$$

$i = 1, \ldots, n;\ k= 0, 1, 2,\ldots$

В качестве критерия останова итерационного процесса используют либо выполнение условия малости приращения аргумента $\| x[k+1] - x[k] \| \le \varepsilon$, либо выполнение условия малости градиента

$$\| f'(x[k+1]) \| \le \gamma,$$

Здесь ε и γ - заданные малые величины.

Возможен и комбинированный критерий, состоящий в одновременном выполнении указанных условий. Градиентные методы отличаются друг от друга способами выбора величины шага a_k.

При методе с постоянным шагом для всех итераций выбирается некоторая постоянная величина шага. Достаточно малый шаг a_k обеспечит убывание функции, т. е. выполнение неравенства

$$f(x[k+1]) = f(x[k] - a_k f'(x[k])) < f(x[k]).$$

Однако это может привести к необходимости проводить неприемлемо большое количество итераций для достижения точки минимума. С другой стороны, слишком большой шаг может вызвать неожиданный рост функции либо привести к колебаниям около точки минимума (зацикливанию). Из-за сложности получения необходимой информации для выбора величины шага методы с постоянным шагом применяются на практике редко.

Более экономичны в смысле количества итераций и надежности градиентные *методы с переменным шагом,* когда в зависимости от результатов вычислений величина шага некоторым образом меняется. Рассмотрим применяемые на практике варианты таких методов.

Градиентные методы. Общие соображения и определения.

Наиболее распространенные и эффективные методы приближенного решения задачи безусловной оптимизации

$$f(x) \to \min, \qquad (1)$$

где f: $\mathbf{R}^m \to \mathbf{R}$, укладываются в следующую грубую схему. Начиная с некоторого $x^0 \in \mathbf{R}^m$, строится последовательность $\{x^n\} \subset \mathbf{R}^m$ такая, что

$$f(x^{n+1}) < f(x^n) \tag{2}$$

при всех $n \in \mathbf{N}$. Такие последовательности иногда называют *релаксационными*, а методы построения релаксационных последовательностей — *итерационными методами* или *методами спуска*. Последовательность, удовлетворяющую (2), строят в надежде, что уменьшая на каждом шаге (переходе от x^n к x^{n+1}) значение функции, мы приближаемся к минимуму (по крайней мере, локальному).

Мы будем говорить, что метод, начиная с данного $x^0 \subset \mathbf{R}^m$,

а) *условно сходится*, если последовательность $\{x^n\}$ релаксационна и

$f'(x^n) \to \Theta$ при $n \to \infty$;

б) *сходится*, если

$x^n \to x^* = \operatorname{argmin} f(x)$ при $n \to \infty$;

в) *линейно сходится* (или *сходится со скоростью геометрической прогрессии*, или *имеет первый порядок сходимости*), если при некоторых $C > 0$ и $q \in [0, 1)$

$$||x^n - x^*|| \le Cq^n; \tag{3}$$

г) *сверхлинейно сходится*, если для любого $q \in (0, 1)$ и некоторого (зависящего от q) C выполнено неравенство (3);

д) *квадратично сходится* (или *имеет второй порядок сходимости*), если при некоторых $C > 0$ и $q \in [0, 1)$ и всех $n \in \mathbf{N}$

$||x^n - x^*|| \le Cq^{2n}$.

Если эти свойства выполняются только для x^0 достаточно близких к x^*, то как всегда добавляется эпитет *"локально"*.

Будем говорить, что на данной последовательности метод *сходится с порядком p* (или *имеет p-ый порядок сходимости*), если при некотором C

$||x^{n+1} - x^*|| \le C||x^n - x^*||^p$.

Эвристические соображения, приводящие к градиентным методам.

Выше уже отмечалось, что если x не является точкой локального минимума функции f, то двигаясь из x в направлении, противоположном градиенту (еще говорят, в *направлении антиградиента*), мы можем локально уменьшить значение функции. Этот факт позволяет надеяться, что последовательность $\{x^n\}$, рекуррентно определяемая формулой

$$x^{n+1} = x^n - \alpha f'(x^n), \qquad (4)$$

где α - некоторое положительное число, будет релаксационной.

К этой же формуле приводит и следующее рассуждение. Пусть у нас есть некоторое приближение x^n. Заменим в шаре $B(x^n, \varepsilon)$ с центром в точке x^n функцию f ее линейным (вернее, афинным) приближением:

$f(x) \approx \varphi(x) \stackrel{\text{def}}{=} f(x^n) + (f'(x^n), x - x^n)$

(функция φ аппроксимирует f в окрестности точки x^n с точностью $o(x - x^n)$). Разумеется, (линейная) безусловная задача $\varphi(x) \to \min$ неразрешима, если $f'(x^n) \ne \Theta$. В окрестности же $B(x^n, \varepsilon)$ функция φ имеет точку минимума. Эту точку естественно взять за следующее приближение x^{n+1}.

4.2. Метод парабол

Поиск точки минимума методами исключения отрезков основан на сравнении значений функции в двух точках. При таком сравнении разности значений *f(x)* в этих точках не учитываются, важны только их знаки.

Учесть информацию, содержащуюся в относительных изменениях значений *f(x)* в пробных точках, позволяют методы полиномиальной аппроксимации, основная идея которых состоит в том, что для функции *f(x)* строится аппроксимирующий многочлен и его точка минимума служит приближением к *x**. Для эффективного использования этих методов на функцию *f(x),* кроме унимодальности, налагается дополнительное требование достаточной гладкости (по крайней мере, непрерывности).

Обоснованием указанных методов является известная из математического анализа теорема Вейерштрасса об аппроксимации, согласно которой непрерывную на отрезке функцию можно с любой точностью приблизить на этом отрезке некоторым полиномом.

Для повышения точности аппроксимации можно, во-первых, увеличивать порядок полинома и, во-вторых, уменьшать длину отрезка аппроксимации. Первый путь приводит к быстрому усложнению вычислительных процедур, поэтому на практике используют аппроксимирующие полиномы не выше третьего порядка. В то же время уменьшение отрезка, содержащего точку минимума унимодальной функции, не представляет особого труда.

В простейшем методе полиномиальной аппроксимации - методе парабол используются полиномы второго порядка. На каждой итерации этого метода строится квадратный трехчлен, график которого (парабола) проходит через три выбранные точки графика функции *f(x)* (рис. 1).

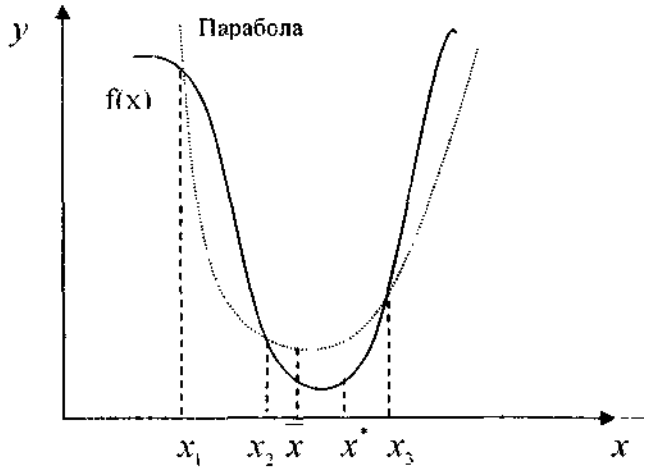


Рис. 1. Иллюстрация к методу парабол

Опишем метод парабол. Рассмотрим унимодальную на отрезке [*a; b*] функцию *f(x),* достигающую минимума во внутренней точке этого отрезка. Выберем три точки x_1, x_2 и x_3 отрезка [*a; b*], для которых выполняются неравенства:

$$x_1 < x_2 < x_3,\ f(x_1) \geq f(x_2) \leq f(x_3). \qquad (1)$$

Из унимодальности *f(x)* следует, что $x^* \in [x_1;x_3]$.
Построим квадратный трехчлен

$$q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2),$$

график которого проходит через точки $(x_1, f(x_1))$, $(x_2, f(x_2))$, $(x_3, f(x_3))$ графика функции *f(x).* Будем считать, что хотя бы одно из неравенств (1) для *f(x)* является строгим (если $f(x_1) = f(x_2) = f(x_3)$, то поиск точки *x* на этом закончен, так как из унимодальности функции *f(x)* следует, что она достигает минимума в каждой точке отрезка $[x_1;x_3]$). Тогда из (1) следует, что ветви параболы направлены вверх, а точка минимума трехчлена *q(x)* принадлежит отрезку $[x_1;x_3]$.

Определяя коэффициенты a_0, a_1 и a_2 из системы уравнений

$$q(x_1) = f(x_1) = f_1,$$
$$q(x_2) = f(x_2) = f_2,$$
$$q(x_3) = f(x_3) = f_3,$$

находим:

$$a_0 = f_1 \quad , a_1 = \frac{f_2 - f_1}{x_2 - x_1}, \; a_2 = \frac{1}{x_3 - x_2}\left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}\right).$$

Точку минимума $\bar{x}$ квадратного трехчлена $q(x)$ вычислим, приравняв его производную к нулю. Получим

$$\bar{x} = \frac{1}{2}\left(x_1 + x_2 - \frac{a_1}{a_2}\right) = \frac{1}{2}\left[x_1 + x_2 - \frac{(f_2 - f_1)(x_3 - x_2)}{x_2 - x_1} / \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}\right)\right] \quad (2)$$

Число $\bar{x}$ из (2) служит очередным приближением метода парабол к x^*. Далее описанная процедура повторяется для новых точек x_1, x_2, x_3, удовлетворяющих неравенства (1).

Выбрать эти точки среди x_1, x_2, x_3 и $\bar{x}$ можно с помощью перехода от исходного к новому отрезку $[x_1;x_3]$, содержащему точку x^*, методом исключения отрезков. Для этого перехода используются пробные точки x_2 и $\bar{x}$ и сравниваются значения $f(x)$ в этих точках. Начало и конец нового отрезка, а также пробная точка, попавшая на него, образуют тройку точек, обладающих свойством (1).

Заметим, что на каждой итерации метода парабол, кроме первой, определяется только одно новое значение $f(x)$.

Условием окончания поиска служит близость к нулю разности Δ чисел $\bar{x}$, найденных на данной и предыдущей итерациях, т.е. неравенство $|\Delta| \le \varepsilon$, где ε — заданное число, характеризующее точность определения x^*.

Перечислим основные шаги алгоритма метода парабол

Шаг 1. Выбрать точки x_1, x_2, x_3 удовлетворяющие условиям (1). Перейти к шагу 2.

Шаг 2. Найти $\bar{x}$ по формуле (2). На первой итерации перейти к шагу 4, на остальных - к шагу 3.

Шаг 3. Проверка на окончание поиска. Сравнить модуль разности значений $\bar{x}$ на данной и предыдущей итерациях Δ с числом ε. Если $|\Delta| \le \varepsilon$, то поиск завершить, полагая $x^* \approx \bar{x}$, $f^* \approx f(x)$, иначе - перейти к шагу 4.

Шаг 4. Вычислить значение $f(\bar{x})$. Перейти к шагу 5.

Шаг 5. Определить новую тройку чисел x_1, x_2, x_3 . Присвоить $f(x_1)$, $f(x_2)$ и $f(x_3)$ соответствующие значения $f(x)$, найденные ранее. Перейти к шагу 2.

Пример. Метод парабол

Решить задачу $f(x) = x^4 + e^{-x} \to \min,\ x \in [0;1]$ с точностью $|\Delta| \le \varepsilon = 0{,}0025.$

Итерация 1

Шаг 1. Выберем точки: x_1=0,25, x_2 =0,5, x_3 =0,75. Функция принимает в этих точках значения, соответственно f_1 = 0,7817, f_2 = 0,6690, f_3 = 0,7888, удовлетворяющие неравенствам (1). Переходим к шагу 2.

Шаг 2. По формуле (56) находим $\bar{x}$ =0,4968. Переходим к шагу 4.

Шаг 4. Вычисляем: $f(\bar{x})$ = 0,6694 . Переходим к шагу 5.

Шаг 5. На данной итерации имеем

$$x_1 < \bar{x} < x_2 < x_3,\ f(\bar{x}) > f(x_2),$$

следовательно, $x^* \in [\bar{x};\ x_3]$. Поэтому полагаем $x_2 = \bar{x}$ = 0,4968, $f(x_1) = f(\bar{x}) = 0.6694,$ а точки x_2, x_3 и значения $f(x)$ в них не изменяются. Переходим к следующей итерации, начиная с шага 2.

Итерация 2

Шаг 2. Находим: $\bar{x}$ = 0,5224 . Переходим к шагу 3.

Шаг 3. $\Delta = |0{,}4968 - 0{,}5224| = 0{,}026 > 0{,}0025,$ поэтому переходим к шагу 4.

Шаг 4. Вычисляем: $f(\bar{x})$= 0,6676 . Переходим к шагу 5.

Шаг 5. На этой итерации

$$x_1 < x_2 < \bar{x} < x_3,\ f(x_2) > f(\bar{x}),$$ поэтому $x^* \in [x_2; x_3]$,

полагаем:

$$x_1 = x_2 = 0{,}5,\ f(x_1) = f(x_2) = 0{,}6690,\ x_2 = \bar{x} = 0{,}5524.\ f(x_2) = f(\bar{x}) = 0{,}6676,$$

а точка x_3 и значение $f(x_3)$ остаются прежними. Переходим к следующей итерации.

Итерация 3

Шаг 2. Находим x = 0,5248. Переходим к шагу 3.

Шаг 3.

Определяем $\Delta = |0{,}5224 - 0{,}5248| = 0{,}0024 < 0{,}0025,$ т.е. требуемая точность достигнута. Поэтому полагаем $x^* \approx \bar{x} \approx 0{,}525.$

Отметим, что в результате пяти вычислений $f(x)$ в точке x^* была найдена с весьма высокой точностью (сравните с точным до четвертого знака значением $x^* = 0{,}5283$).

Численное решение задачи минимизации, как правило, связано с построением минимизирующей последовательности точек $x^0, x^1, x^2, \ldots, x^n, \ldots,$ обладающих свойством

$$f(x^k) < f(x^{k-1}),\ k=0,1,\ldots \qquad (3)$$

Общее правило построения минимизирующей последовательности имеет вид

$$x^{k+1} = x^k + t_k d^k,\ k=0,1,\ldots,$$

где x^0 - начальная точка поиска; d^k - приемлемое направление перехода из точки x^k в точку x^{k+1}, которое обеспечивает выполнение условий (3) и называется направлением спуска; t_k - величина шага. Начальная точка поиска задается исходя из физического содержания решаемой задачи и априорных данных о существовании и положении точек экстремума.

4.3. Градиентный метод как классический метод оптимизации

1. Эвристические соображения. Проанализируем один из наиболее важных в идейном отношении метод безусловной оптимизации – градиентный. Это метод, редко применяемый на практике в «чистом виде», служит моделью для построения более реалистических алгоритмов. На примере данного метода будет подробно разобран вопрос о сходимости — будут даны различные доказательства сходимости, описана общая техника построения доказательств, обсуждены соотношения между теоретическими результатами о сходимости и практическим использованием метода.

Предположим, что в любой точке x можно вычислить градиент функции $\nabla f(x)$. В такой ситуации наиболее простым методом

минимизации $f(x)$ является *градиентный,* в котором, начиная с некоторого начального приближения x^0, строится итерационная последовательность

$$x^{k+1} = x^k - \gamma_k \nabla f(x^k), \qquad (1)$$

где параметр $\gamma_k \geq 0$ задает длину шага. К методу (1) можно прийти из разных соображений.

Во-первых, при доказательстве необходимых условий экстремума можно использовать то обстоятельство, что если в точке x условие экстремума не выполняется ($\nabla(x) \neq 0$), то значение функции можно уменьшить, перейдя к точке $x - \tau \nabla(x)$ при достаточно малом $\tau > 0$. Итеративно применяя этот прием, приходим к методу (1).

Во-вторых, в точке x^k дифференцируемая функция $f(x)$ приближается линейной $f_k(x) = f(x^k) + (\nabla f(x^k), x - x^k)$ с точностью до членов порядка $o(x\text{-}x^k)$. Поэтому можно искать минимум аппроксимации $f_k(x)$ в окрестности x^k. Например, можно задаться некоторым ε_k и решить вспомогательную задачу

$$\min_{\|x - x^k\| \leq \varepsilon_k} f_k(x). \qquad (2)$$

Ее решение естественно принять за новое приближение x^{k+1}. Можно остаться в окрестности x^k и иначе, добавив к $f_k(x)$ «штраф» за отклонение от x^k. Например, можно решить вспомогательную задачу

$$\min [f_k(x) + \alpha_k \| x - x^k \|^2] \qquad (3)$$

и ее решение взять в качестве x^{k+1}. Читателю предоставляется убедиться в том, что решение задач (2), (3) задается формулой (1).

В-третьих, можно в точке x^k выбрать *направление* локального *наискорейшего спуска,* т. е. то направление y^k, $\|y^k\| = 1$, для которого достигается минимум $f'(x^k; y)$. Используя формулу

$$f'(x; y) = \varphi'(0) = (\nabla f(x), y)$$

для производной по направлению, получаем

$$y^k = \arg\min_{\|y\|=1} (\nabla f(x^k), y) = -\nabla f(x^k) / \|\nabla f(x^k)\|. \qquad (4)$$

Таким образом, ***направление наискорейшего спуска противоположно направлению градиента***.

Мы привели здесь столь подробно эти соображения, поскольку они же будут использоваться при построении методов оптимизации в более сложных ситуациях (например, при наличии ограничений). Однако в этих ситуациях они могут привести к различным методам.

2. Сходимость. Рассмотрим простейший вариант градиентного метода, в котором $\gamma_k \equiv \gamma$:

$$x^{k+1} = x^k - \gamma \nabla f(x^k). \qquad (5)$$

Нас будет интересовать поведение этого метода при различных предположениях относительно $f(x)$ и γ.

Теорема 1. *Пусть $f(x)$ дифференцируема на $\mathbf{R}^n$, градиент $f(x)$ удовлетворяет условию Липшица:*

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|, \tag{6}$$

$f(x)$ ограничена снизу:

$$f(x) \geqslant f^* > -\infty \tag{7}$$

и γ удовлетворяет условию

$$0 < \gamma < 2/L. \tag{8}$$

Тогда в методе (5) *градиент стремится к* 0:

$$\lim_{k\to\infty} \nabla f(x^k) = 0,$$

а функция $f(x)$ монотонно убывает: $f(x^{k+1}) \leqslant f(x^k)$.

Доказательство. Подставим в формулу градиента функции

$$f(x+y) = f(x) + \int_0^1 (\nabla f(x+\tau y),\ y)\,d\tau =$$

$$= f(x) + (\nabla f(x),\ y) + \int_0^1 (\nabla f(y+\tau y) - \nabla f(x),\ y)\,d\tau.$$

$x = x^k$, $y = -\gamma\nabla f(x^k)$ и воспользуемся (6):

$$f(x^{k+1}) = f(x^k) - \gamma\|\nabla f(x^k)\|^2 - \gamma\int_0^1 (\nabla f(x^k - \tau\gamma\nabla f(x^k)) -$$

$$- \nabla f(x^k),\ \nabla f(x^k))\,d\tau \leqslant f(x^k) - \gamma\|\nabla f(x^k)\|^2 +$$

$$+ L\gamma^2\|\nabla f(x^k)\|^2\int_0^1 \tau\,d\tau = f(x^k) - \gamma\left(1 - \frac{1}{2}L\gamma\right)\|\nabla f(x^k)\|^2.$$

Суммируя неравенства

$$f(x^{k+1}) \leqslant f(x^k) - \alpha\|\nabla f(x^k)\|^2, \quad \alpha = \gamma(1 - L\gamma/2) \tag{9}$$

по k от 0 до s, получаем

$$f(x^{s+1}) \leqslant f(x^0) - \alpha\sum_{k=0}^{s}\|\nabla f(x^k)\|^2.$$

Поскольку $\alpha > 0$ в силу (8), то

$$\sum_{k=0}^{s}\|\nabla f(x^k)\|^2 \leqslant \alpha^{-1}(f(x^0) - f(x^{s+1})) \leqslant \alpha^{-1}(f(x^0) - f^*)$$

при всех s, т. е. $\sum_{k=0}^{\infty} \|\nabla f(x^k)\|^2 < \infty$. Отсюда $\|\nabla f(x^k)\| \to 0$.

Покажем, что все условия этой теоремы существенны. Нарушения условия (6) могут быть двух типов. Во-первых, функция $f(x)$ может быть недостаточно гладкой в какой-либо точке. Пусть, например, $f(x) = \|x\|^{1+\alpha}$, $0 < \alpha < 1$. Эта функция дифференцируема, но ее градиент не удовлетворяет условию Липшица, так как $\|\nabla f(x) - \nabla f(0)\| / \|x - 0\| = (\alpha + 1)\|x\|^{\alpha-1} \to \infty$ при $||x|| \to 0$. В этом случае будет $\gamma\|\nabla f(x^k)\| \gg \|x^k - x^*\| = \|x^k\|$ при малых $||x^k||$, т. е. шаг в методе (5) получается большим и монотонность убывания $f(x)$ нарушается. Во-вторых, (6) не выполняется для функций, растущих быстрее квадратичной. Пусть, например,

$$f(x) = \|x\|^{2+\alpha}, \ \alpha > 0,$$

тогда

$\|\nabla f(x) - \nabla f(0)\| / \|x - 0\| = (2 + \alpha)\|x\|^{\alpha} \to \infty$ при $\|x\| \to \infty$. При этом для всякого $\gamma > 0$ можно указать такое x^0, что метод (5), примененный к функции $\|x\|^{2+\alpha}$, $\alpha > 0$, с начальным приближением x^0, расходится, поскольку будет $\|x^{k+1}\| > \|x^k\|$, $k = 0, 1, \ldots$

Если не выполнено условие (7), то функция $f(x)$ не достигает минимума и градиент в методе (5) не обязан стремиться к 0 (например, если $f(x)$ линейна: $f(x) = (c, x)$, то $\|\nabla f(x)\| \equiv \|c\| > 0$).

Наконец, выбирать γ, нарушая условие (8), вообще говоря, также нельзя, что видно на примере функции $f(x) = Lx^2/2$, $x \in \mathbf{R}^1$. Действительно, если $\gamma \geq 2/L$, то в методе (5) для этой функции будет

$$f(x^{k+1}) \geqslant f(x^k), \quad k = 0, 1, \ldots,$$

при любом x^0.

С другой стороны, при сделанных в теореме 1 предположениях нельзя доказать ничего большего, например, сходимость последовательности x^k. Примером может служить $f(x) = 1/(1 + \|x\|^2)$. Эта функция удовлетворяет условиям теоремы и при любом $x^0 \neq 0$ будет $\|x^k\| \to \infty$.

Если потребовать, чтобы множество $\{x: f(x) \leqslant f(x^0)\}$ было ограничено, то из x^k можно выбрать подпоследовательность, сходящуюся к некоторой стационарной точке x^*. Однако точка x^* не обязана быть точкой локального или глобального минимума. В частности, градиентный метод (5) (или даже (1) с произвольным выбором γ_k), начатый из некоторой стационарной точки x^0, останется в этой точке: $x^k = x^0$ для всех k. Иными словами, градиентный метод

«застревает» в любой стационарной точке — ***точке максимума, минимума или седловой***. Что же касается поиска глобального минимума, то градиентный метод «не отличает» точек локального минимума от глобального и никакой гарантии сходимости к глобальному минимуму он не дает.

Наконец, в условиях теоремы 1 скорость сходимости $\nabla f(x^k)$ к 0 может быть очень медленной. Например, для $f(x)= 1/x$ при $x \geq 1$ (вид $f(x)$ при $x < 1$ безразличен) метод (5) при $\gamma=1$, $x^0 = 1$ принимает вид $x^{k+1} = x^k + (x^k)^{-2}$, при этом можно показать, что

$$|f'(x^k)| = O(k^{-2/3}).$$

Рассмотрим поведение градиентного метода для более узкого класса функций — сильно выпуклых. Естественно, здесь удается доказать более сильные результаты, чем в теореме 1 - именно, сходимость итераций x^k к точке глобального минимума со скоростью геометрической прогрессии.

Нам понадобится несколько неравенств, относящихся к дифференцируемым, выпуклым и сильно выпуклым функциям.

Лемма 1. *Пусть f(x) дифференцируема, $\nabla f(x)$ удовлетворяет условию Липшица с константой Y и f(x) ≥f* для всех x.*
Тогда

$$\|\nabla f(x)\|^2 \leqslant 2L(f(x) - f^*). \qquad (10)$$

Доказательство. Сделаем из точки x шаг градиентного метода с $\gamma = 1/L$. Тогда (см. (9))

$$f^* \leqslant f(x - L^{-1}\nabla f(x)) \leqslant f(x) - (2L)^{-1}\|\nabla f(x)\|^2.$$

Лемма 2. *Пусть f(x) выпукла и дифференцируема, а $\nabla f(x)$ удовлетворяет условию Липшица с константой L. Тогда*

$$(\nabla f(x) - \nabla f(y),\ x - y) \geqslant L^{-1}\|\nabla f(x) - \nabla f(y)\|^2. \qquad (11)$$

Доказательство. Докажем (11) лишь для дважды дифференцируемых функций. Тогда

$$\nabla f(y) = \nabla f(x) + \int_0^1 \nabla^2 f(x + \tau(y - x))(y - x)\,d\tau = \nabla f(x) + A(y - x),$$

где матрица $A = \int_0^1 \nabla^2 f(x + \tau(y - x))\,d\tau$ симметрична и неотрицательно определена, т. е. $A \geq 0$. Кроме того, $\|A\| \leq L$, так как $\|\nabla^2 f(x)\| \leqslant L$, для всех x в силу условия Липшица на градиент. Поэтому

$$(\nabla f(x) - \nabla f(y),\ x - y) =$$

$$= (A(x - y),\ x - y) \geqslant \|A\|^{-1} \|A(x - y)\|^2 \geqslant L^{-1} \|\nabla f(x) - \nabla f(y)\|^2.$$

Лемма 3. *Пусть f(x)— дифференцируемая сильно выпуклая (с константой l) функция, x* — ее точка минимума (она существует). Тогда*

$$\|\nabla f(x)\|^2 \geqslant 2l(f(x) - f(x^*)).$$

Теорема 2. *Пусть f(x) дифференцируема на* $\mathbf{R}^n$, *ее градиент удовлетворяет условию Липшица с константой L и f(x) является сильно выпуклой функцией с константой l. Тогда при* $0 < \gamma < 2/L$ *метод* (5) *сходится к единственной точке глобального минимума x* со скоростью геометрической прогрессии:*

$$\|x^k - x^*\| \leqslant cq^k, \quad 0 \leqslant q < 1. \qquad (12)$$

Доказательство. Выполнены все условия теоремы 1, поэтому справедливо неравенство (9):

$$f(x^{k+1}) \leqslant f(x^k) - \gamma(1 - L\gamma/2)\|\nabla f(x^k)\|^2.$$

Используем лемму 3:

$$f(x^{k+1}) \leqslant f(x^k) - l\gamma(2 - L\gamma)(f(x^k) - f(x^*)).$$

Отсюда

$$f(x^{k+1}) - f(x^*) \leqslant (1 - l\gamma(2 - L\gamma))(f(x^k) - f(x^*)) = q_1(f(x^k) - f(x^*)),$$

$$f(x^k) - f(x^*) \leqslant q_1^k(f(x^0) - f(x^*)), \quad q_1 = 1 - 2l\gamma + Ll\gamma^2.$$

Поскольку $0 < \gamma < 2/L$, то $0 < q_1 < 1$, и следовательно, $f(x^k) \to f(x^*)$. Из неравенства

$$f(x) \geq f(x^*) + l\|x - x^*\|^2/2$$

следует

$$\|x^k - x^*\|^2 \leqslant (2/l)\, q_1^k (f(x^0) - f(x^*)).$$

Рассмотрим еще более узкий класс функций — сильно выпуклых дважды дифференцируемых.

Теорема 3. *Пусть f(x) дважды дифференцируема и*

$$lI \leqslant \nabla^2 f(x) \leqslant LI, \quad l > 0, \qquad (13)$$

для всех x. Тогда при $0 < \gamma < 2/L$

$$\|x^k - x^*\| \leqslant \|x^0 - x^*\| q^k, \quad q = \max\{|1 - \gamma l|, |1 - \gamma L|\} < 1. \qquad (14)$$

Величина q минимальна и равна

$$q^* = (L - l)/(L + l) \quad при \quad \gamma = \gamma^* = 2/(L + l). \qquad (15)$$

Доказательство. По формуле

$$g(x+y)=g(x)+\int_0^1 g'(x+\tau y)\,y\,d\tau=$$

$$=g(x)+g'(x)\,y+\int_0^1 (g'(x+\tau y)-g'(x))\,y\,d\tau.$$

определяем

$$\nabla f(x^k)=\nabla f(x^*)+\int \nabla^2 f(x^*+\tau(x^k-x^*))(x^k-x^*)\,d\tau=A_k(x^k-x^*),$$

где в силу (13) $lI\leqslant A_k\leqslant LI$. Поэтому

$$\|x^{k+1}-x^*\|=\|x^k-x^*-\gamma\nabla f(x^k)\|=$$

$$=\|(I-\gamma A_k)(x^k-x^*)\|\leqslant\|I-\gamma A_k\|\,\|x^k-x^*\|.$$

Для всякой симметричной матрицы A имеем

$$\|I-A\|=\max\{|1-\lambda_1|,\ |1-\lambda_n|\},$$

где λ_1 и λ_n — наименьшее и наибольшее собственные значения A. Поэтому

$$\|x^{k+1}-x^*\|\leqslant q\|x^{k+1}-x^*\|,\ q=\max\{|1-\gamma l|,|1-\gamma L|\}.$$

Поскольку

$0<\gamma<2/L$, $0<l\leqslant L$, то $|1-\gamma l|<1$, $|1-\gamma L|<1$, т. е. $q<1$.

Минимизируя q по γ получаем (15).

Покажем, что оценка скорости сходимости, даваемая теоремой 3, точная, она достигается для любой квадратичной функцией. Пусть

$$f(x)=(Ax,\ x)/2-(b,\ x),\ A>0,\ 0<l=\lambda_1\leqslant\lambda_2\ \ldots\ \leqslant\lambda_n=L,$$

где λ_i — сооственные числа матрицы A. Возьмем произвольное $0<\gamma<2/L$. Предположим, что $|1-\gamma l|\geqslant|1-\gamma L|$. Выберем $x^0=x^*+e^1$, где e^1 — собственный вектор, отвечающий λ_1, $\|e^1\|=1$. Тогда $x^k-x^*=(I-\gamma A)^k(x^0-x^*)=$

$=(1-\gamma\lambda_1)^k e^1$, $\|x^k-x^*\|=|(1-\gamma l)|^k=q^k\|x^0-x^*\|$.

Аналогичным образом, если $|1-\gamma L|\geqslant|1-\gamma l|$, то выберем $x^0=x^*+e^n$, e^n — собственный вектор, отвечающий λ_n, $\|e^n\|=1$, и получим так же

$$\|x^k-x^*\|=|(1-\gamma L)|^k=q_k\|x^0-x^*\|.$$

Таким образом, для всякого $0<\gamma\ 2/L$ найдется x^0 такое, что Оценку

$$\|x^k - x^*\| \leqslant (q^*)^k \|x^0 - x^*\|,\ q^* = (L-l)/(L+l)$$

нельзя улучшить, даже если выбирать γ оптимальным образом для каждого x^0. Действительно, возьмем $x^0=x^*+e^1+e^n$ (обозначения те же, что и выше). Тогда пои любом $0 < \gamma < 2/L$

$$x^k - x^* = (I - \gamma A)^k (x^0 - x^*) = (1-\gamma l)^k e^1 + (1-\gamma L)^k e^n,$$
$$\|x^k - x^*\| = [(1-\gamma l)^{2k} + (1-\gamma L)^{2k}]^{1/2} \|x^0 - x^*\| / \sqrt{2}.$$

Поэтому, если либо $|1-\gamma l| > q^*$, либо $|1-\gamma L| > q^*$, то $||x^k - x^*||$ убывает медленнее, чем $(q^*)^k$. Но $q=\max\{|1-\gamma l|, |1-\gamma L|\} \leq q^*$ лишь при $\gamma = \gamma^*$, при этом

$$|1-\gamma^* l| = |1-\gamma^* L| = q^* \text{ и } \|x^k - x^*\| = (q^*)^k \|x^0 - x^*\|.$$

Аналогичное расуждение справедливо для любой точки x^0 такой, что

$$(x^0 - x^*, e^1) \neq 0,\ (x^0 - x^*, e^n) \neq 0.$$

Локальный аналог теоремы 3 справедлив и для невыпуклых функций.

Теорема 4. *Пусть x^* — невырожденная точка локального минимума $f(x)$. Тогда при $0 < \gamma < 2/\|\nabla^2 f(x^*)\|$ метод* (5) *локально сходится к x^* со скоростью геометрической прогрессии, т. е. для всякого $\delta > 0$ найдется $\varepsilon > 0$ такое, что при $\|x^0 - x^*\| \leqslant \varepsilon$ будет*

$$\|x^k - x^*\| \leqslant \|x^0 - x^*\| (q+\delta)^k,$$
$$q = \max\{|1-\gamma l|, |1-\gamma L|\} < 1, \quad 0 < l I \leqslant \nabla^2 f(x^*) \leqslant L I. \qquad (16)$$

Величина q минимальна и равна $q^* = (L-l)/(L+l)$ при $\gamma^* = 2/(L+l)$.

4.4. Метод наискорейшего спуска

При использовании метода наискорейшего спуска на каждой итерации величина шага a_k выбирается из условия минимума функции $f(x)$ в направлении спуска, т. е.

$$f(x[k] - a_k f'(x[k])) = \min_{a>0} f(x[k] - af'(x[k])).$$

Это условие означает, что движение вдоль антиградиента происходит до тех пор, пока значение функции $f(x)$ убывает. С математической точки зрения на каждой итерации необходимо решать задачу одномерной минимизации по a функции $\varphi(a) = f(x[k] - af'(x[k]))$.

Алгоритм метода наискорейшего спуска состоит в следующем.

1. Задаются координаты начальной точки $x[0]$.

2. В точке $x[k]$, $k = 0, 1, 2, \ldots$ вычисляется значение градиента $f'(x[k])$.

3. Определяется величина шага a_k, путем одномерной минимизации по a функции $\varphi(a) = f(x[k] - af'(x[k]))$.

4. Определяются координаты точки $x[k+1]$:

$$x_i[k+1] = x_i[k] - a_k f'_i(x[k]),\ i = 1, \ldots, n.$$

5. Проверяются условия останова итерационного процесса. Если они выполняются, то вычисления прекращаются. В противном случае осуществляется переход к п. 1.

В рассматриваемом методе направление движения из точки $x[k]$ касается линии уровня в точке $x[k+1]$ (рис. 1). Траектория спуска зигзагообразная, причем соседние звенья зигзага ортогональны друг другу. Действительно, шаг a_k выбирается путем минимизации по a функции $f(a) = f(x[k] - af'(x[k]))$. Необходимое условие минимума функции $d\varphi(a)/da=0$. Вычислив производную сложной функции, получим условие ортогональности векторов направлений спуска в соседних точках:

$$d\varphi(a)/da = -f'(x[k+1]f'(x[k]) = 0.$$

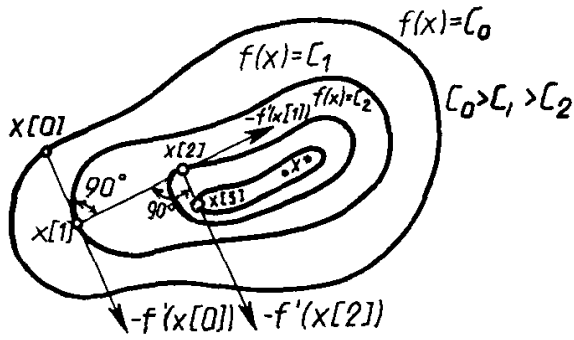


Рис. 1. Геометрическая интерпретация метода наискорейшего спуска

Градиентные методы сходятся к минимуму с высокой скоростью (со скоростью геометрической прогрессии) для гладких выпуклых функций. У таких функций наибольшее M и наименьшее m собственные значения матрицы вторых производных (матрицы Гессе)

$$H(x) = \begin{vmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{vmatrix}$$

мало отличаются друг от друга, т. е. матрица $H(x)$ хорошо обусловлена. Напомним, что собственными значениями λ_i, i =1, …, n, матрицы являются корни характеристического уравнения

$$\begin{Vmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{Vmatrix} = 0.$$

Однако на практике, как правило, минимизируемые функции имеют плохо обусловленные матрицы вторых производных *(m/M<<1)*. Значения таких функций вдоль некоторых направлений изменяются гораздо быстрее (иногда на несколько порядков), чем в других направлениях. Их поверхности уровня в простейшем случае сильно вытягиваются (рис. 2), а в более сложных случаях изгибаются и представляют собой овраги. Функции, обладающие такими свойствами, называют *овражными.* Направление антиградиента этих функций (см. рис. 2) существенно отклоняется от направления в точку минимума, что приводит к замедлению скорости сходимости.

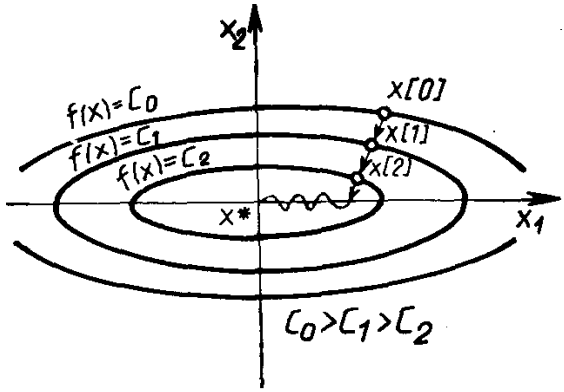


Рис. 2. Овражная функция

Скорость сходимости градиентных методов существенно зависит также от точности вычислений градиента. Потеря точности, а это обычно происходит в окрестности точек минимума или в овражной ситуации, может вообще нарушить сходимость процесса градиентного

спуска. Вследствие перечисленных причин градиентные методы зачастую используются в комбинации с другими, более эффективными методами на начальной стадии решения задачи. В этом случае точка $x[0]$ находится далеко от точки минимума, и шаги в направлении антиградиента позволяют достичь существенного убывания функции.

4.5. Метод градиентного спуска

Введение

В разделе рассматривается задача поиска минимума функции $f(x):\mathbb{R}^n \to \mathbb{R}$, записываемая в виде:

$$f(x) \to \min_{x \in \mathbb{R}^n} \tag{1}$$

Пусть функция *f(x)* такова, что можно вычислить ее градиент. Тогда можно применить метод градиентного спуска, описанный ниже.

В разделе приведены теоремы сходимости метода градиентного спуска, а также рассмотрена его варианты:

Градиентный метод с постоянным шагом.

Идея метода

Основная идея метода заключается в том, чтобы осуществлять оптимизацию в направлении наискорейшего спуска, а это направление задаётся антиградиентом - ∇f:

$$x^{[k+1]} = x^{[k]} - \lambda^{[k]} \nabla f(x^{[k]})$$

где $\lambda^{[k]}$ выбирается

- постоянной, в этом случае метод может расходиться;

- дробным шагом, т.е. длина шага в процессе спуска делится на некое число;
- наискорейшим спуском:

$$\lambda^{[k]} = \arg\min_{\lambda} f(x^{[k]} - \lambda \nabla f(x^{[k]}))$$

Алгоритм

Вход: функция $f: \mathbb{R}^n \to \mathbb{R}$

Выход: найденная точка оптимума *x*

1. Повторять:
2. $x^{[k+1]} = x^{[k]} - \lambda^{[k]} \nabla f(x^{[k]})$, где $\lambda^{[k]}$ выбирается одним из описанных выше способов
3. если выполен критерий останова, то возвращаем текущее значение $x^{[k+1]}$

Критерий останова

Критерии остановки процесса приближенного нахождения минимума могут быть основаны на различных соображениях. Некоторые из них:

1. $\|x^{[k+1]} - x^{[k]}\| \leq \epsilon$
2. $\|f(x^{[k+1]}) - f(x^{[k]})\| \leq \epsilon$

Здесь $x^{[k]} \in \mathbb{R}^n$ - значение, полученное после *k*-го шага оптимизации. ε - наперед заданное положительное число.

В общем случае число α может на каждом шаге (т. е. для каждого *n*) выбираться заново:

$x^{n+1} = x^n - \alpha^n f'(x^n)$. (2)

Именно методы, задаваемые формулой (2), называются *градентными*. Если $\alpha^n=\alpha$ при всех n, то получающийся метод называется *градиентным методом с постоянным шагом* (*с шагом* α.)

Поясним геометрическую суть градиентного метода. Для этого мы выберем способ изображения функции с помощью линий уровня. *Линией уровня функции* f (*изолинией*) называется любое множество вида $\{x \in \mathbf{R}^m: f(x) = c\}$. Каждому значению c отвечает своя линия уровня (см. рис. 1).

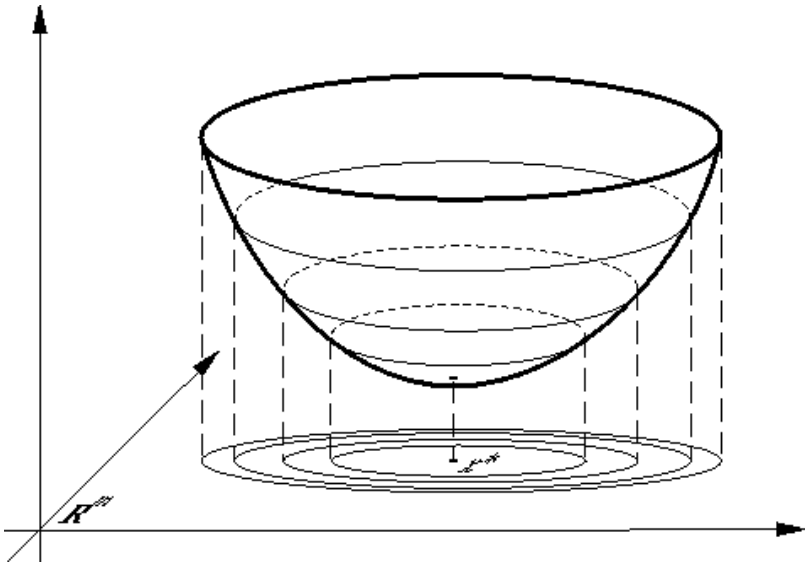


Рис. 1.

Геометрическая интерпретация градиентного метода с постоянным шагом изображена на рис. 2. На каждом шаге мы сдвигаемся по вектору антиградиента, "уменьшенному в α раз".

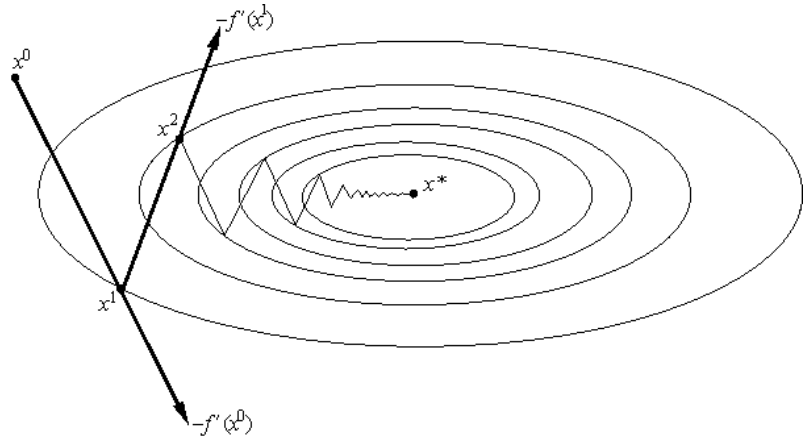


Рис. 2.

Пример исследования сходимости.

Изучим сходимость градиентного метода с постоянным шагом на примере функции

$$f(x) = |x|^p,$$

где $p > 1$ (случай $p \le 1$ мы не рассматриваем, поскольку тогда функция f не будет гладкой, а мы такой случай не исследуем). Очевидно, задача (1) с такой функцией f имеет единственное решение $x^* = 0$. Для этой функции приближения x^n градиентного метода имеют вид:

$$x^{n+1} = x^n - \alpha p|x^n|^{p-1}\operatorname{sign} x^n. \qquad (3)$$

Пределом этой последовательности может быть только 0. Действительно, если $x^{**} = \lim_{n\to\infty} x^n \neq 0$, то, переходя к пределу в (3) при $n \to \infty$, получаем противоречащее предположению $x^{**} \neq 0$ равенство

$$x^{**} = x^{**} - \alpha p|x^{**}|^{p-1}\operatorname{sign}\ x^{**},$$

откуда $x^{**} = 0$. Очевидно также, что если $x^0 = 0$, то и $x^n = 0$ при всех n.

Покажем, что если $p < 2$, то при любом шаге $\alpha > 0$ и любом начальном приближении x^0 (за исключением не более чем счетного числа точек) приближения (3) не являются сходящимися. Для этого заметим, что если $0 < |x^n| < (2/\alpha p)^{1/2(2-p)}$, то

$$|x^{n+1}| > |x^n|. \tag{4}$$

Поэтому, если x^n не обращается в нуль, то она не может сходиться к нулю и, следовательно, не может сходиться вообще.

Таким образом, осталось доказать (4). В силу (3)

$$|x^{n+1}| = |x^n - \alpha p|x^n|^{p-1} \cdot \operatorname{sign} x^n| = |x^n| \cdot |1 - \alpha p|x^n|^{p-2} \cdot \operatorname{sign} x^n|.$$

Остается заметить, что если $0 < |x^n| < (2/\alpha p)^{1/(2-p)}$, то, как нетрудно видеть, $|1 - \alpha p|x^n|^{p-2} \cdot \operatorname{sign} x^n| > 1$, что и требовалось.

Замечание. Число начальных точек x^0, для которых x^n обращается в нуль при некотором n (и следовательно, при всех бóльших), не более чем счетно.

Если $p = 2$, т. е. $f(x) = x^2$, то (3) переписывается в виде

$$|x^{n+1}| = |x^n| \cdot |1 - 2\alpha|.$$

Поэтому, если $\alpha \in (0, 1)$, то $|1 - 2\alpha| < 1$, а следовательно,

$$|x^{n+1}| = |1 - 2\alpha|^{n+1} \cdot |x^0| \to 0 \text{ при } n \to \infty.$$

Если же $\alpha \geq 1$, то

$$|x^{n+1}| \geq |x^n|,$$

и последовательность $\{x^n\}$, начинающаяся из ненулевой начальной точки, расходится.

Замечание. Если $p > 2$, то градиентный метод (3) сходится при $\alpha p|x^0|^{p-2} < 2$ и расходится при $\alpha p|x^0|^{p-2} \geq 2$ для любых начальных точек, за исключением может быть счетного множества.

Таким образом, есть функции, для которых градиентный метод не сходится даже при сколь угодно малом шаге α и есть функции, для которых он сходится только при достаточно малых шагах. В следующих пунктах мы приведем ряд теорем о сходимости градиентного метода.

Теорема об условной сходимости градиентного метода с постоянным шагом.

Пусть в задаче (1) *функция f ограничена снизу, непрерывно дифференцируема и, более того, f' удовлетворяет условию Липшица:*

$$\|f'(x) - f'(y)\| \leq \Lambda \|x - y\| \text{ при всех } x, y \in \mathbf{R}^m.$$

Тогда при $\alpha \in (0, 2/\Lambda)$ *градиентный метод с постоянным шагом условно сходится.*

Доказательство. Положим $z^n = -\alpha f'(x^n)$ и обозначим $f(x^n + tz^n)$ через $\varphi(t)$. Тогда, как легко видеть,

$$\varphi'(t) = (f'(x^n + tz^n), z^n)$$

и поэтому по формуле Ньютона — Лейбница для функции φ

$$f(x^{n+1}) - f(x^n) = f(x^n + z^n) - f(x^n) = \varphi(1) - \varphi(0) =$$

$$= \int_0^1 \varphi'(s)\, ds = \int_0^1 (f'(x^n + sz^n), z^n)\, ds.$$

Добавив и отняв $(f'(x^n),\ z^n) = \int_0^1 (f'(x^n),\ z^n)\,ds$ и воспользовавшись неравенством $(x, y) \le \|x\| \cdot \|y\|$, получим

$$f(x^{n+1}) - f(x^n) = (f'(x^n), z^n) + \int_0^1 (f'(x^n + sz^n) - f'(x^n), z^n)\,ds \le$$

$$\le (f'(x^n), -\alpha f'(x^n)) + \int_0^1 \|f'(x^n + sz^n) - f'(x^n)\| \cdot \|z^n\|\,ds.$$

Учитывая условие Липшица для f', эту цепочку можно продолжить:

$$\begin{aligned} f(x^{n+1}) - f(x^n) &\le -\alpha\|f'(x^n)\|^2 + \Lambda\,\|z^n\|^2 \int_0^1 s\,ds = \\ &= -\alpha\|f'(x^n)\|^2 + \frac{\Lambda\alpha^2}{2}\|f'(x^n)\|^2 = -\alpha\|f'(x^n)\|^2\left(1 - \frac{\Lambda\alpha}{2}\right). \end{aligned} \qquad (5)$$

Поскольку $1 - \Lambda\alpha/2 > 0$, последовательность $\{f(x^n)\}$ не возрастает и, следовательно, релаксационность $\{x^n\}$ доказана. А так как в силу условий теоремы f еще и ограничена снизу, последовательность $\{f(x^n)\}$ сходится. Поэтому, в частности, $f(x^{n+1}) - f(x^n) \to 0$ при $n \to \infty$. Отсюда и из (5) получаем

$$\|f'(x^n)\|^2 \le \alpha^{-1}\left(1 - \frac{\Lambda\alpha}{2}\right)^{-1} [f(x^n) - f(x^{n+1})] \to 0 \text{ при } n \to \infty.$$

Замечания о сходимости.

Подчеркнем, что приведенная теорема не гарантирует сходимости метода, но лишь его условную сходимость, причем, локальную. Например, для функции $f(x) = (1 + x^2)^{-1}$ на $\mathbf{R}$ последовательность $\{x^n\}$ градиентного метода с постоянным шагом, начинающаяся с произвольного x^0 стремится к ∞.

Поскольку в приведенной теореме градиент непрерывен, любая предельная точка последовательности $\{x^n\}$ является стационарной. Однако эта точка вовсе не обязана быть точкой минимума, даже локального. Например, рассмотрим для функции $f(x) = x^2 \operatorname{sign} x$ градиентный метод с шагом $\alpha \in (0, 1/2)$. Тогда, как легко видеть, если $x^0 > 0$, то $x^n \to 0$ при $n \to \infty$. Точка же $x = 0$ не является локальным минимумом функции f.

Заметим также, что описанный метод не различает точек локального и глобального минимумов. Поэтому для того, чтобы сделать заключение о сходимости x^n к точке $x^* = \operatorname{argmin} f(x)$ приходится налагать дополнительные ограничения, гарантирующие, в частности, существование и единственность решения задачи (1). Один вариант таких ограничений описывается ниже.

Теорема о линейной сходимости градиентного метода с постоянным шагом.

Пусть выполнены условия предыдущей теоремы и, кроме того, f дважды непрерывно дифференцируема и сильно выпукла с константой λ. Тогда при $\alpha \in (0, 2/\Lambda)$ градиентный метод с шагом α сходится со скоростью геометрической прогрессии со знаменателем $q = \max\{|1 - \alpha\lambda|, |1 - \alpha\Lambda|\}$:

$$\|x^n - x^*\| \le q^n \|x^0 - x^*\|.$$

Доказательство. Решение $x^* = \operatorname{argmin} f(x)$ существует и единственно в силу известных теорем. Для функции $F(x) = f'(x)$ воспользуемся аналогом формулы Ньютона — Лейбница

$$F(y) = F(x) + \int_0^1 F'[x + s(y - x)](y - x)\, ds,$$

или, для $x = x^*$ и $y = x^n$, учитывая, что $f'(x^*) = \Theta$,

$$\int 1 \qquad (6)$$

$$f'(x^n) = \int_0^1 f''[x^* + s(x^n - x^*)](x^n - x^*)\, ds$$

Далее, в силу известного утверждения $f''(x) \le \Lambda$ при всех $x \in \mathbf{R}^m$. Кроме того, по условию $f''(x) \ge \lambda$ при тех же x. Поэтому, так как

$$\lambda\|h\|^2 \le (f''[x^* + s(x^n - x^*)]h, h) \le \Lambda\, \|h\|^2,$$

выполнено неравенство

$$\lambda\|h\|^2 \le \left(\left(\int_0^1 f''[x^* + s(x^n - x^*)]\, ds \right) h, h \right) \le \Lambda\, \|h\|^2. \qquad (7)$$

Интеграл, стоящий в этом неравенстве, определяет линейный (симметричный в силу симметричности f) оператор на $\mathbf{R}^m$, обозначим его L^n. Неравенство (7) означает, что $\lambda \le \mathrm{L}^n \le \Lambda$. В силу (6) градиентный метод записывается в виде

$$x^{n+1} = x^n - \alpha \mathrm{L}^n (x^n - x^*).$$

Но тогда

$$\|x^{n+1} - x^n\| = \|x^n - x^* - \alpha \mathrm{L}^n (x^n - x^*)\| =$$

$$= \|(I - \alpha \mathrm{L}^n)(x^n - x^*)\| \le \|I - \alpha \mathrm{L}^n\| \cdot \|x^n - x^*\|.$$

Спектр $\sigma(I - \alpha \mathrm{L}^n)$ оператора $I - \alpha \mathrm{L}^n$ состоит из чисел вида $\sigma_i = 1 - \alpha\lambda_i$, где $\lambda_i \in \sigma(\mathrm{L}^n)$. В силу (7) и известного неравенства,

$$1 - \alpha\lambda \ge \sigma_i \ge 1 - \alpha\Lambda,$$

и следовательно

$$\|I - \alpha L^n\| \le \max\{|1 - \alpha\lambda|, |1 - \alpha\Lambda|\} = q.$$

Таким образом,

$$\|x^{n+1} - x^n\| \le q\|x^n - x^*\|.$$

Из этого неравенства вытекает утверждение теоремы.

Об оптимальном выборе шага.

Константа q, фигурирующая в предыдущей теореме и характеризующая скорость сходимости метода, зависит от шага α. Нетрудно видеть, что величина

$$q = q(\alpha) = \max\{|1 - \alpha\lambda|, |1 - \alpha\Lambda|\}$$

минимальна, если шаг α выбирается из условия $|1 - \alpha\lambda| = |1 - \alpha\Lambda|$ (см. рис. 3), т. е. если $\alpha = \alpha^* = 2/(\lambda + \Lambda)$. При таком выборе шага оценка сходимости будет наилучшей и будет характеризоваться величиной

$$q = q^* = \frac{\Lambda - \lambda}{\Lambda + \lambda}.$$

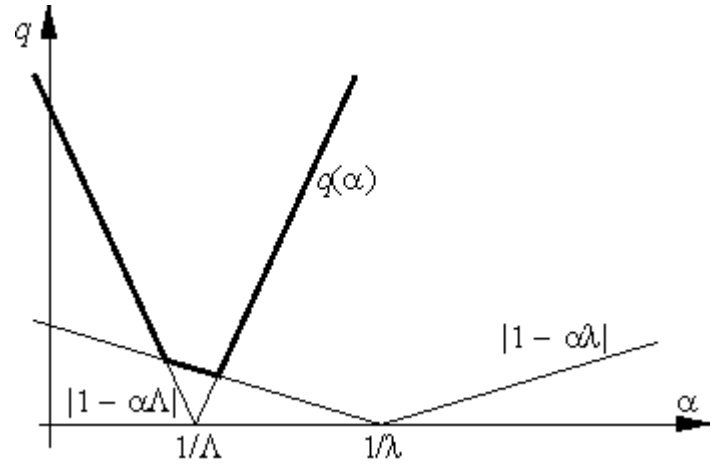


Рис. 3.

Напомним, что в качестве λ и Λ могут выступать равномерные по x оценки сверху и снизу собственных значений оператора $f''(x)$. Если $\lambda << \Lambda$, то $q^* \approx 1$ и метод сходится очень медленно. Геометрически случай $\lambda << \Lambda$ соответствует функциям с сильно вытянутыми линиями уровня (см. рис. 4). Простейшим примером такой функции может служить функция на $\mathbf{R}^2$, задаваемая формулой

$f(x_1, x_2) = \lambda x^2_1 + \Lambda x^2_2 \text{c } \lambda << \Lambda.$

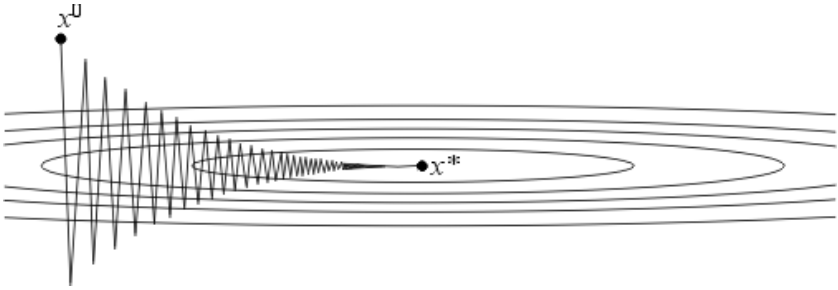


Рис. 4.

Поведение итераций градиентного метода для этой функции изображено на рис. 4 — они, быстро спустившись на "дно оврага", затем медленно "зигзагообразно" приближаются к точке минимума.

Число $\mu = \Lambda/\lambda$ (характеризующее, грубо говоря, разброс собственных значений оператора $f''(x)$) называют *числом обусловленности функции* f. Если $\mu >> 1$, то функции называют *плохо обусловленными* или *овражными*. Для таких функций градиентный метод сходится медленно.

Но даже для хорошо обусловленных функций проблема выбора шага нетривиальна в силу отсутствия априорной информации о минимизируемой функции. Если шаг выбирается малым (чтобы гарантировать сходимость), то метод сходится медленно. Увеличение же шага (с целью ускорения сходимости) может привести к расходимости метода. Мы опишем сейчас два алгоритма автоматического выбора шага, позволяющие частично обойти указанные трудности.

4.6. Градиентный метод с дроблением шага.

В этом варианте градиентного метода величина шага α^n на каждой итерации выбирается из условия выполнения неравенства

$$f(x^{n+1}) = f(x^n - \alpha^n f'(x^n)) \le f(x^n) - \varepsilon\alpha^n \|f'(x^n)\|^2, \qquad (8)$$

где $\varepsilon \in (0, 1)$ — некоторая заранее выбранная константа. Условие (8) гарантирует (если, конечно, такие α^n удастся найти), что получающаяся последовательность будет релаксационной. Процедуру нахождения такого α^n обычно оформляют так. Выбирается число $\delta \in (0, 1)$ и некоторый начальный шаг α^0. Теперь для каждого n полагают $\alpha^n = \alpha^0$ и делают шаг градиентного метода. Если с таким α^n условие (8) выполняется, то переходят к следующему n. Если же (8) не выполняется, то умножают α^n на δ ("дробят шаг") и повторяют эту процедуру до тех пор пока неравенство (6) не будет выполняться. В условиях вышеприведенной теоремы эта процедура для каждого n за конечное число шагов приводит к нужному α^n.

Можно показать, что в условиях известной теоремы градиентный метод с дроблением шага линейно сходится. Описанный алгоритм избавляет нас от проблемы выбора α на каждом шаге, заменяя ее на проблему выбора параметров ε, δ и α^0, к которым градиентный метод менее чувствителен. При этом, разумеется, объем вычислений возрастает (в связи с необходимостью процедуры дробления шага),

впрочем, не очень сильно, поскольку в большинстве задач основные вычислительные затраты ложатся на вычисление градиента.

Числовые примеры

Метод градиентного спуска с постоянным шагом

Для исследования сходимости метода градиентного спуска с постоянным шагом была выбрана функция:

$$f(x_1,x_2)=10*x_1^2+x_2^2.$$

Начальное приближение - точка (10,10). Использован критерий останова:

$$\|f(x^{[k+1]})-f(x^{[k]})\|\le 10^{-5}$$

Результаты эксперимента отражены в таблице:

Значение шага λ	**Достигнутая точность**	**Количество итераций**
0.1	метод расходится	
0.01	2e-4	320
0.001	2e-3	2648
0.0001	1e-2	20734

Из полученных результатов можно сделать вывод, что при слишком большом шаге метод расходится, при слишком малом сходится медленно и точность хуже. Надо выбирать шаг наибольшим из тех, при которых метод сходится.

Градиентный метод с дроблением шага

Для исследования сходимости метода градиентного спуска с дроблением шага была выбрана функция:

$$f(x_1,x_2)=10*x_1^2+x_2^2.$$

Начальное приближение - точка (10,10). Использован критерий останова:

$$\|f(x^{[k+1]})-f(x^{[k]})\| \le 10^{-5}$$

Результаты эксперимента отражены в таблице:

Значение параметра ε	**Значение параметра** δ	**Значение параметра** $\lambda^{[k]}$	**Достигнутая точность**	**Количество итераций**
0.95	0.95	1	5e-4	629
0.1	0.95	1	1e-5	41
0.1	0.1	1	2e-4	320
0.1	0.95	0.01	2e-4	320

Из полученных результатов можно сделать вывод об оптимальном выборе параметров: ε=0.1, δ=0.95, $\lambda^{[0]}$=1, хотя метод не сильно чувствителен к выбору параметров.

Метод наискорейшего спуска

Для исследования сходимости метода наискорейшего спуска была выбрана функция:

$$f(x_1,x_2) = 10^* x_1^2 + x_2^2.$$

Начальное приближение - точка (10,10). Использован критерий останова:

$$\|f(x^{[k+1]})-f(x^{[k]})\| \le 10^{-5}$$

Для решения одномерных задач оптимизации использован метод золотого сечения.

Метод получил точность 6e-8 за 9 итераций.

Отсюда можно сделать вывод, что метод наискорейшего спуска сходится быстрее, чем градиентный метод с дроблением шага и метод градиентного спуска с постоянным шагом.

Недостатком методом наискорейшего спуска являлется необходимость решать одномерную задачу оптимизации.

Рекомендации программисту

При программировании методов градиентного спуска следует аккуратно относится к выбору параметров, а именно

- Метод градиентного спуска с постоянным шагом: шаг λ следует выбирать меньше 0.01, иначе метод расходится (метод может расходится и при таком шаге в зависимости от исследуемой функции).
- Градиентный метод с дроблением шага не очень чувствителен к выбору параметров. Один из вариантов выбора параметров:

$$\varepsilon=0.1,\ \delta=0.95,\ \lambda^{[0]}=1$$

- Метод наискорейшего спуска: в качестве метода одномерной оптимизации можно использовать метод золотого сечения (когда он применим).

Заключение

Методы градиентного спуска являются достаточно мощным инструментом решения задач оптимизации. Главным недостатком методов является ограниченная область применимости.

4.7. Метод сопряженных градиентов

Рассмотренные выше градиентные методы отыскивают точку минимума функции в общем случае лишь за бесконечное число итераций. Метод сопряженных градиентов формирует направления поиска, в большей мере соответствующие геометрии минимизируемой функции. Это существенно увеличивает скорость их сходимости и позволяет, например, минимизировать квадратичную функцию

$$f(x) = (x, Hx) + (b, x) + a$$

с симметрической положительно определенной матрицей H за конечное число шагов n, равное числу переменных функции. Любая гладкая функция в окрестности точки минимума хорошо аппроксимируется квадратичной, поэтому методы сопряженных градиентов успешно применяют для минимизации и неквадратичных функций. В таком случае они перестают быть конечными и становятся итеративными.

По определению, два n-мерных вектора x и y называют *сопряженными* по отношению к матрице H (или H-сопряженными), если скалярное произведение $(x, Hy) = 0$. Здесь H - симметрическая положительно определенная матрица размером $n \times n$.

Одной из наиболее существенных проблем в методах сопряженных градиентов является проблема эффективного построения направлений. Метод Флетчера-Ривса решает эту проблему путем преобразования на каждом шаге антиградиента $-f(x[k])$ в направление $p[k]$, H-сопряженное с ранее найденными направлениями $p[0]$, $p[1]$, ..., $p[k\text{-}1]$. Рассмотрим сначала этот метод применительно к задаче минимизации квадратичной функции.

Направления $p[k]$ вычисляют по формулам:

$$p[k] = -f'(x[k]) + \beta_{k-1} p[k\text{-}1], \; k >= 1;$$

$$p[0] = -f'(x[0]).$$

Величины β_{k-1} выбираются так, чтобы направления $p[k]$, $p[k\text{-}1]$ были H-сопряженными**:**

$$(p[k], Hp[k\text{-}1]) = 0.$$

В результате для квадратичной функции

$$\beta_{k-1} = \frac{(f'(x[k], f'(x[k]))}{(f'(x[k-1], f'(x[k-1]))},$$

итерационный процесс минимизации имеет вид

$$x[k+1] = x[k] + a_k p[k],$$

где $p[k]$ - направление спуска на k-м шаге; a_k - величина шага. Последняя выбирается из условия минимума функции $f(x)$ по a в направлении движения, т. е. в результате решения задачи одномерной минимизации:

$$f(x[k] + a_k p[k]) = \min_{a \geq 0} f(x[k] + ap\,[k]).$$

Для квадратичной функции

$$a_k = -\frac{(f'(x[k]), p[k])}{(p[k], Hp[k])}$$

Алгоритм метода сопряженных градиентов Флетчера-Ривса состоит в следующем.

1. В точке $x[0]$ вычисляется $p[0] = -f'(x[0])$.

2. На k-м шаге по приведенным выше формулам определяются шаг a_k. и точка $x[k+1]$.

3. Вычисляются величины $f(x[k+1])$ и $f'(x[k+1])$.

4. Если $f'(x[k+1]) = 0$, то точка $x[k+1]$ является точкой минимума функции $f(x)$. В противном случае определяется новое направление $p[k+1]$ из соотношения

$$p[k+1] = -f'(x[k+1]) + \frac{(f'(x[k+1]), f'(x[k+1]))}{(f'(x[k]), f'(x[k]))} p[k]$$

и осуществляется переход к следующей итерации. Эта процедура найдет минимум квадратичной функции не более чем за n шагов. При минимизации неквадратичных функций метод Флетчера-Ривса из конечного становится итеративным. В таком случае после $(n+1)$-й

итерации процедуры 1-4 циклически повторяются с заменой $x[0]$ на $x[n+1]$, а вычисления заканчиваются при $\|f'(x[k])\| < \varepsilon$, где ε - заданное число. При этом применяют следующую модификацию метода:

$x[k+1] = x[k] + a_k p[k]$,

$p[k] = -f'(x[k]) + \beta_{k-1} p[k-1]$, $k >= 1$;

$p[0] = -f'(x[0])$;

$f(x[k] + a_k p[k]) = \min\limits_{a \geq 0} f(x[k] + ap[k])$;

$$\beta_{k-1} = \begin{cases} \dfrac{(f'(x[k]), f'(x[k]) - f'(x[k-1]))}{(f'(x[k]), f'(x[k]))}, k \notin I \\ 0, k \in I \end{cases}.$$

Здесь I - множество индексов: $I = \{0, n, 2n, 3n, ...\}$, т. е. обновление метода происходит через каждые n шагов.

Геометрический смысл метода сопряженных градиентов состоит в следующем (рис. 1). Из заданной начальной точки $x[0]$ осуществляется спуск в направлении $p[0] = -f'(x[0])$. В точке $x[1]$ определяется вектор-градиент $f'(x[1])$. Поскольку $x[1]$ является точкой минимума функции в направлении $p[0]$, то $f'(x[1])$ ортогонален вектору $p[0]$. Затем отыскивается вектор $p[1]$, H-сопряженный к $p[0]$. Далее отыскивается минимум функции вдоль направления $p[1]$ и т. д.

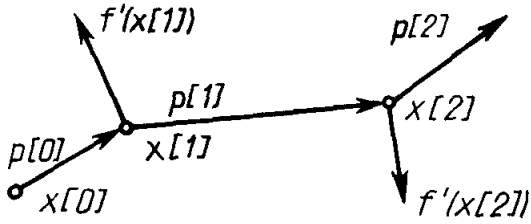


Рис. 1. Траектория спуска в методе сопряженных градиентов

Методы сопряженных направлений являются одними из наиболее эффективных для решения задач минимизации. Однако следует отметить, что они чувствительны к ошибкам, возникающим в процессе счета. При большом числе переменных погрешность может настолько возрасти, что процесс придется повторять даже для квадратичной функции, т. е. процесс для нее не всегда укладывается в n шагов.

Метод сопряженных градиентов - математический аппарат

Некоторые авторы говорят, что термин "метод сопряженных градиентов" – один из примеров того, как бессмысленные словосочетания, став привычными, воспринимаются сами собой разумеющимися и не вызывают никакого недоумения. Дело в том, что, за исключением частного и не представляющего практического интереса случая, градиенты не являются сопряженными, а сопряженные направления не имеют ничего общего с градиентами. Название метода отражает тот факт, что данный метод отыскания безусловного экстремума сочетает в себе понятия градиента целевой функции и сопряженных направлений.

Несколько слов об обозначениях, используемых далее.

Скалярное произведение двух векторов записывается $x^T y$ и представляет сумму скаляров: $\sum_{i=1}^{n} x_i y_i$. Заметим, что $x^T y = y^T x$. Если x и y ортогональны, то $x^T y = 0$. В общем, выражения, которые преобразуются к матрице 1×1, такие как $x^T y$ и $x^T Ax$, рассматриваются как скалярные величины.

Первоначально метод сопряженных градиентов был разработан для решения систем линейных алгебраических уравнений вида:

$$Ax = b \qquad (1)$$

где x – неизвестный вектор, b – известный вектор, а A – известная, квадратная, симметричная, положительно–определенная матрица. Решение этой системы эквивалентно нахождению минимума соответствующей квадратичной формы.

Квадратичная форма – это просто скаляр, квадратичная функция некого вектора *x* следующего вида:

$$f(x) = (1/2)x^T Ax - b^T x + c \qquad (2)$$

Наличие такой связи между матрицей линейного преобразования *A* и скалярной функцией *f(x)* дает возможность проиллюстрировать некоторые формулы линейной алгебры интуитивно понятными рисунками. Например, матрица *A* называется положительно-определенной, если для любого ненулевого вектора *x* справедливо следующее:

$$x^T Ax > 0 \qquad (3)$$

На рисунке 2 изображено как выглядят квадратичные формы соответственно для положительно-определенной матрицы (a), отрицательно-определенной матрицы (b), положительно-неопределенной матрицы (c), неопределенной матрицы (d).

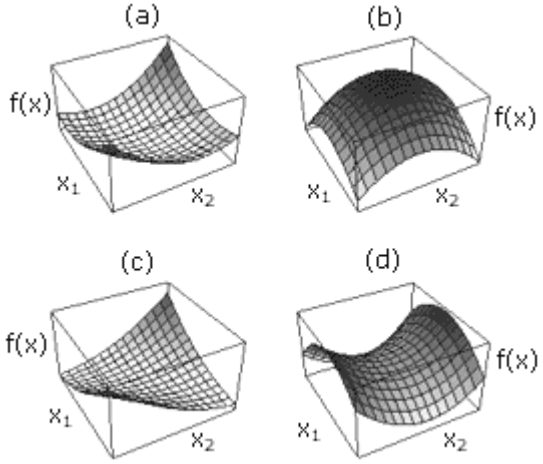


Рис. 2. Квадратичные формы для положительно-определенной матрицы, отрицательно-определенной матрицы, положительно-неопределенной матрицы, неопределенной матрицы.

То есть, если матрица *A* – положительно-определенная, то вместо того, чтобы решать систему уравнений 1, можно найти минимум ее квадратичной функции. Причем, метод сопряженных градиентов

сделает это за n или менее шагов, где n – размерность неизвестного вектора x. Так как любая гладкая функция в окрестностях точки своего минимума хорошо аппроксимируется квадратичной, этот же метод можно применить для минимизации и неквадратичных функций. При этом метод перестает быть конечным, а становится итеративным.

Рассмотрение метода сопряженных градиентов целесообразно начать с рассмотрения более простого метода поиска экстремума функции – метода наискорейшего спуска. На рисунке 3 изображена траектория движения в точку минимума методом наискорейшего спуска.

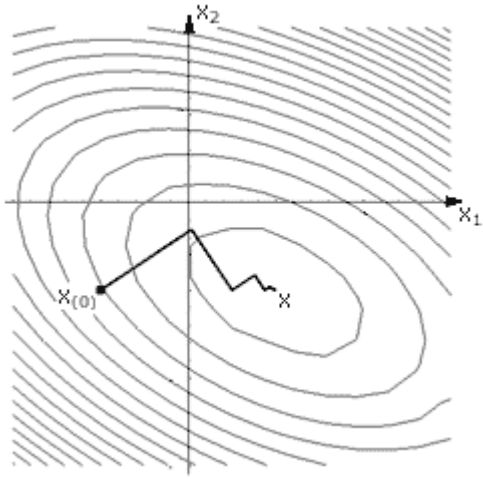


Рис. 3. Траектория движения в точку минимума методом наискорейшего спуска.

Суть этого метода:

- в начальной точке $x(0)$ вычисляется градиент, и движение осуществляется в направлении антиградиента до тех пор, пока уменьшается целевая функция;
- в точке, где функция перестает уменьшаться, опять вычисляется градиент, и спуск продолжается в новом направлении;
- процесс повторяется до достижения точки минимума.

В данном случае каждое новое направление движения ортогонально предыдущему. Не существует ли более разумного способа выбора нового направления движения? Существует, и он называется метод сопряженных направлений. А метод сопряженных градиентов как раз относится к группе методов сопряженных направлений. На рисунке 4 изображена траектория движения в точку минимума при использовании метода сопряженных градиентов.

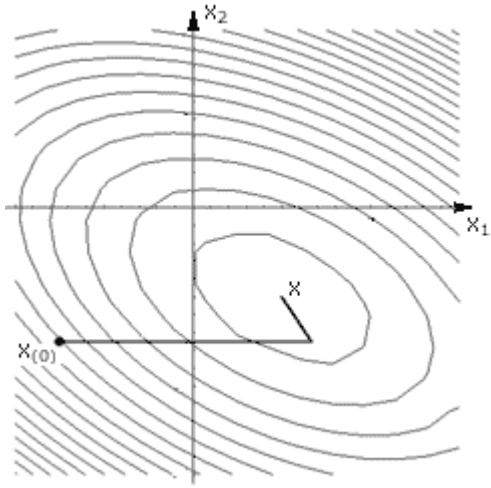


Рис. 4. Траектория движения в точку минимума при использовании метода сопряженных градиентов

Определение сопряженности формулируется следующим образом: два вектора x и y называют A-сопряженными (или сопряженными по отношению к матрице A) или A–ортогональными, если скалярное произведение x и Ay равно нулю, то есть:

$$x^T Ay = 0 \qquad (4)$$

Сопряженность можно считать обобщением понятия ортогональности. Действительно, когда матрица A – единичная матрица, в соответствии с равенством 4, векторы x и y – ортогональны. Можно и иначе продемонстрировать взаимосвязь понятий ортогональности и сопряженности: мысленно растяните рисунок 4 таким образом, чтобы линии равного уровня из эллипсов превратились в окружности, при этом сопряженные направления станут просто ортогональными.

Остается выяснить, каким образом вычислять сопряженные направления. Один из возможных способов – использовать методы линейной алгебры, в частности, процесс ортогонализации Грамма–Шмидта. Но для этого необходимо знать матрицу *A*, поэтому для большинства задач (например, обучение многослойных нейросетей) этот метод не годится. Существуют другие, итеративные способы вычисления сопряженного направления, самый известный – формула Флетчера-Ривса:

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)} \tag{5}$$

где:

$$\beta_{(i+1)} = \frac{r^T_{(i+1)}}{r^T_{(i)}} \frac{r_{(i+1)}}{r_{(i)}} \tag{6}$$

Формула 5 означает, что новое сопряженное направление получается сложением антиградиента в точке поворота и предыдущего направления движения, умноженного на коэффициент, вычисленный по формуле 6. Направления, вычисленные по формуле 5, оказываются сопряженными, если минимизируемая функция задана в форме 2. То есть для квадратичных функций метод сопряженных градиентов находит минимум за *n* шагов (*n* – размерность пространства поиска). Для функций общего вида алгоритм перестает быть конечным и становится итеративным. При этом, Флетчер и Ривс предлагают возобновлять алгоритмическую процедуру через каждые *n* + 1 шагов.

Можно привести еще одну формулу для определения сопряженного направления, формула Полака–Райбера (Polak-Ribiere):

$$\beta_{(i+1)} = \frac{r^T_{(i+1)} (r_{(i+1)} - r_{(i)})}{r^T_{(i)} r_{(i)}} \tag{7}$$

Метод Флетчера-Ривса сходится, если начальная точка достаточно близка к требуемому минимуму, тогда как метод Полака-Райбера

может в редких случаях бесконечно циклиться. Однако последний часто сходится быстрее первого метода. Сходимость метода Полака-Райбера может быть гарантирована выбором $\beta = \max\{\beta, 0\}$. Это эквивалентно рестарту алгорима по условию $\beta \leq 0$. Рестарт алгоритмической процедуры необходим, чтобы забыть последнее направление поиска и стартовать алгоритм заново в направлении скорейшего спуска.

Ниже приведен алгоритм сопряженных градиентов для минимизации функций общего вида (неквадратичных).

1. Вычисляется антиградиент в произвольной точке $x_{(0)}$. $d_{(0)} = r_{(0)} = -f'(x_{(0)})$

2. Осуществляется спуск в вычисленном направлении пока функция уменьшается, иными словами, поиск $a_{(i)}$, который минимизирует $f(x_{(i)} + a_{(i)} d_{(i)})$

3. Переход в точку, найденную в предыдущем пункте $x_{(i+1)} = x_{(i)} + a_{(i)} d_{(i)}$

4. Вычисление антиградиента в этой точке $r_{(i+1)} = -f'(x_{(i+1)})$

5. Вычисления по формуле 6 или 7. Чтобы осуществить рестарт алгоритма, то есть забыть последнее направление поиска и стартовать алгоритм заново в направлении скорейшего спуска, для формулы Флетчера–Ривса присваивается 0 через каждые n+1 шагов, для формулы Полака-Райбера – $\beta_{(i+1)} = \max\{\beta_{(i+1)}, 0\}$

6. Вычисление нового сопряженного направления $d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)}$

7. Переход на пункт 2.

Из приведенного алгоритма следует, что на шаге 2 осуществляется одномерная минимизация функции. Для этого, в частности, можно

воспользоваться методом Фибоначчи, методом золотого сечения или методом бисекций. Более быструю сходимость обеспечивает метод Ньютона–Рафсона, но для этого необходимо иметь возможность вычисления матрицы Гессе. В последнем случае, переменная, по которой осуществляется оптимизация, вычисляется на каждом шаге итерации по формуле:

$$a = -\frac{f'^{T}(x)d}{d^{T}f''(x)d}$$

где

$$f''(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Матрица Гессе

Это дает основания некоторым авторам относить метод сопряженных градиентов к методам второго порядка, хотя суть метода вовсе не предполагает необходимым вычисление вторых производных.

Несколько слов об использовании метода сопряженных направлений при обучении нейронных сетей. В этом случае используется обучение по эпохам, то есть при вычислении целевой функции предъявляются все шаблоны обучающего множества и вычисляется средний квадрат функции ошибки (или некая ее модификация). То же самое – при вычислении градиента, то есть используется суммарный градиент по всему обучающему набору. Градиент для каждого примера вычисляется с использованием алгоритма обратного распространения (BackProp).

В заключение приведем один из возможных алгоритмов программной реализации метода сопряженных градиентов.

Сопряженность в данном случае вычисляется по формуле Флетчера–Ривса, а для одномерной оптимизации используется один из вышеперечисленных методов. По мнению некоторых специалистов скорость сходимости алгоритма мало зависит от оптимизационной формулы, применяемой на шаге 2 приведенного выше алгоритма, поэтому можно рекомендовать, например, метод золотого сечения, который не требует вычисления производных. Вариант метода сопряженных направлений, использующий формулу Флетчера-Ривса для расчета сопряженных направлений.

```
i:=0

k:=0
r:=-f'(x)// антиградиент целевой функции
d := r // начальное направление спуска совпадает с антиградиентом
Sigma_new:=r^T*r// квадрат модуля антиградиента
Sigma_0:=Sigma_new // Цикл поиска (выход по счетчику или ошибке)
while i<i_max and Sigma_new>Eps^2*Sigma_0
begin
   j:=0
   Sigma_d:=d^T*d    // Цикл одномерной минимизации (спуск по
направлению d)
   repeat
      a : =
      x : = x + a
      j : = j + 1
   until (j >= j_max) or (a^2 * Sigma_d <= Eps^2)

   r : = -f'(x) // антиградиент целевой функции в новой точке
   Sigma_old : = Sigma_new
   Sigma_new : = r^T * r
   beta : = Sigma_new / Sigma_old
   d : = r + beta * d // Вычисление сопряженного направления
   k : = k + 1

   if (k = n) or (r^T * d <= 0) then // Рестарт алгоритма
   begin
      d : = r
      k : = 0
   end
```

```
   i : = i + 1
end
```

Метод сопряженных градиентов является методом первого порядка, в то же время скорость его сходимости квадратична. Этим он выгодно отличается от обычных градиентных методов. Например, метод наискорейшего спуска и метод координатного спуска для квадратичной функции сходятся лишь в пределе, в то время как метод сопряженных градиентов оптимизирует квадратичную функцию за конечное число итераций. При оптимизации функций общего вида, метод сопряженных направлений сходится в 4-5 раз быстрее метода наискорейшего спуска. При этом, в отличие от методов второго порядка, не требуется трудоемких вычислений вторых частных производных.

4.8. Методы оврагов

Градиентные методы медленно сходятся в тех случаях, когда поверхности уровня целевой функции $f(x)$ сильно вытянуты. Этот факт известен в литературе как «эффект оврагов». Суть эффекта в том, что небольшие изменения одних переменных приводят к резкому изменению значений функции – эта группа переменных характеризует «склон оврага», а по остальным переменным, задающим направление «дно оврага», функция меняется незначительно. На рис. 1 зображены линии уровня «овражной» функции. Траектория градиентного метода характеризуется довольно быстрым спуском на «дно оврага», и затем медленным зигзагообразным движением в точку минимума.

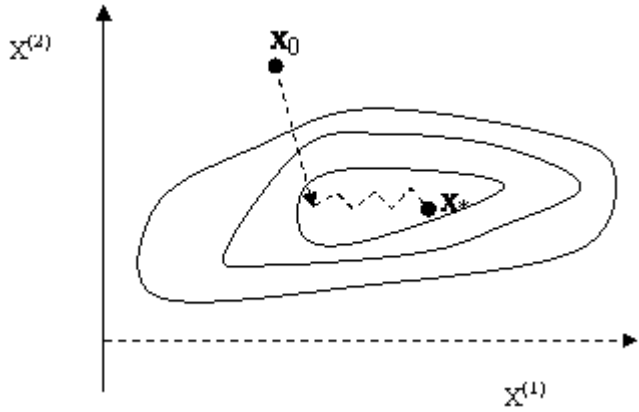


Рис. 1. Линии уровня овражной функции.

Существуют различные подходы для определения точки минимума функции $f(x)$ в овражной ситуации. Большинство из них основаны на эвристических (то есть интуитивных, не обоснованных строго) соображениях. Их можно применять, когда более совершенные методы нецелесообразны, например, когда значение целевой функции вычисляется со значительными погрешностями, информации о ее свойствах недостаточно и т. д. Эти методы просты в реализации и довольно часто применяются на практике, позволяя в ряде случаев получить удовлетворительное решение задачи.

Эвристический алгоритм

Иногда, используя градиентный спуск для минимизации функций со сложной топографической структурой, применяют эвристические схемы, которые идейно близки к методам спуска. Мы рассмотрим такую схему.

Первая эвристическая схема содержит два основных этапа. Оба этапа представляют собой аналоги градиентного спуска с постоянным шагом. Только вместо градиента $f'(x_k)$ используется вектор $g(x)$, формируемый из координат $f'(x_k)$, но на каждом из этапов по разным правилам.

На первом этапе задается малое число $\delta_1<<1$, и используется градиентный спуск, где вместо градиента $f'(x_k)$ берется вектор $g(x)=\{g^{(1)}(x),...,g^{(n)}(x)\}$, который определяется следующим образом:

$$\begin{cases} g^{(i)}(x) = \dfrac{\partial f}{\partial x^i}(x) & \text{если} \quad \left|\dfrac{\partial f}{\partial x^i(x)}\right| > \delta_1, \\ g^{(i)}(x) = 0 & \text{если} \quad \left|\dfrac{\partial f}{\partial x^i(x)}\right| \le \delta_1, \\ i = 1,2,\ldots n. \end{cases}$$

Таким образом, спуск производится лишь по тем переменным, в направлении которых производная целевой функции достаточно велика. Это позволяет быстро спуститься на «дно оврага». Мы спускаемся до тех пор, пока метод не зациклится, то есть до тех пор, пока каждая следующая итерация позволяет найти точку, в которой значение функции меньше, чем значение, найденное в предыдущей итерации. После этого переходим к следующему этапу.

На втором этапе задается некоторое большое число $\delta_2 >> 1$ и используется процедура спуска, где вместо градиента $f'(x_k)$ берется вектор $g(x)=\{g^{(1)}(x),\ldots,g^{(n)}(x)\}$, который определяется следующим образом:

$$\begin{cases} g^{(i)}(x) = \dfrac{\partial f}{\partial x^i}(x) & \text{если} \quad \left|\dfrac{\partial f}{\partial x^i(x)}\right| < \delta_2, \\ g^{(i)}(x) = 0 & \text{если} \quad \left|\dfrac{\partial f}{\partial x^i(x)}\right| \ge \delta_2, \\ i = 1,2,\ldots n. \end{cases}$$

В этом случае перемещение происходит по «берегу» оврага вдоль его «дна». Как и на первом этапе, спуск продолжается до тех пор, пока метод не зациклится.

После выполнения первого и второго этапов принимается решение о завершении работы или продолжении. Для этого сравнивается норма разности предыдущей точки, то есть точки, которую мы имели до применения первого и второго этапов, с текущей точкой, то есть полученной после применения с точностью решения задачи ε_1. Если эта норма меньше ε_1 и норма градиента в текущей точке меньше ε_3, то поиск заканчивается и последняя вычисленная точка принимается за приближенное решение задачи. Иначе для текущей точки вновь повторяем первый и второй этапы и т. д.

Алгоритм

Шаг 1. Задаются x_0, ε_1, ε_3,δ_1,δ_2,α_1 – постоянный шаг пункта 1 и α_2 – постоянный шаг пункта 2 ($\alpha_1<\alpha_2$). Присваивается k=0.

Шаг 2. (Первый этап). Из точки x_k осуществляется спуск на «дно оврага» с постоянным шагом α_1. При спуске вычисление очередной точки осуществляется с использованием формул:

$x_{j+1} = x_j - \alpha_1 g(x_j)$, *где* $g(x)=\{g^{(1)}(x),\ldots,g^{(n)}(x)\}$,

$$\begin{cases} g^{(i)}(x) = \dfrac{\partial f}{\partial x^i}(x) & \text{если } \left|\dfrac{\partial f}{\partial x^i(x)}\right| > \delta_1, \\ g^{(i)}(x) = 0 & \text{если } \left|\dfrac{\partial f}{\partial x^i(x)}\right| \le \delta_1, \\ i = 1,2,\ldots n. \end{cases}$$

Пусть этот процесс остановится в точке x_l.

Шаг 3. (Второй этап). Из точки x_l осуществляется спуск вдоль «дна оврага» с постоянным шагом α_2. При спуске используются формулы: $x_{j+1} = x_j - \alpha_2 g(x_j)$, где $g(x)=\{g^{(1)}(x),\ldots,g^{(n)}(x)\}$,

$$\begin{cases} g^{(i)}(x) = \dfrac{\partial f}{\partial x^i}(x) & \text{если } \left|\dfrac{\partial f}{\partial x^i(x)}\right| < \delta_2, \\ g^{(i)}(x) = 0 & \text{если } \left|\dfrac{\partial f}{\partial x^i(x)}\right| \ge \delta_2, \\ i = 1,2,\ldots n. \end{cases}$$

Пусть этот процесс остановился в точке x_m.

Шаг 4. Если $\|x_k - x_m\| \le \varepsilon_1$ и $\| f'(x_m) \| \le \varepsilon_3$, то полагаем:

$$\tilde{x} = x_m, \tilde{y} = f(x_m)$$

и поиск минимума заканчивается. Иначе $k=m$ и переходим к шагу 2.

4.9. Метод Флетчера-Ривса

Метод Флетчера-Ривса основан на том, что для квадратичной функции n переменных n одномерных поисков вдоль взаимно сопряженных направлений позволяют найти минимум.

Рассмотрим функцию

$$f(x) = a + b^T x + \frac{1}{2} x^T G x.$$

Одномерный поиск будем вести вдоль направлений, взаимно сопряженных по отношению к матрице G.

В качестве первого направления поиска из первой точки x_1 возьмем направление наискорейшего спуска

$$d_1 = -g_1 \qquad (1)$$

и найдем значение λ_1, минимизирующее функцию

$$f(x_1 + \lambda d_1).$$

Положим

$$x_2 = x_1 + \lambda_1 d_1 \qquad (2)$$

и произведем поиск в направлении d_2, сопряженном направлению d_1 (выберем вектор d_2 как линейную комбинацию векторов d_1 и $-g_2$), и найдем

$$x_3 = x_2 + \lambda_2 d_2 \qquad (3)$$

минимизацией функции $f(x_2 + \lambda d_2)$. Направление поиска d_2 из точки x_3 выбирается сопряженным направлениям d_1 и d_2. На $(k + 1)$ - м шаге выбираем d_{k+1} в виде линейной комбинации $-g_{k+1}$, $d_1, d_2, \ldots, d_k$, сопряженной всем направлениям $d_1, d_2, \ldots, d_k$.

$\sum_{r=1}^{k}$ Таким образом, $d_{k+1} = -g_{k+1} + \sum_{r=1}^{k} \alpha_r d_r$, $k = 1, 2 \ldots$ Оказывается, все α_r равны нулю, за исключением α_k, так что

$$d_{k+1} = -g_{k+1} + \alpha_k d_k \qquad (4)$$

и

$$\alpha_k = g^2_{k+1}/g^2_k. \qquad (5)$$

Прежде чем перейти к индуктивным рассуждениям, докажем справедливость соотношений (4) и (5) при k=1. Поскольку $f(x_2)=f(x_1+\lambda_1 d_1)$ является минимумом функции $f(x_1+\lambda_1 d_1)$ на прямой, то

$$g^T_2 d_1 = -g^T_2 g_1 = 0. \qquad (6)$$

Много раз мы уже получали этот результат раньше. Он, конечно, справедлив и для квадратичных функций

$$g_2 = b + Gx_2,\ g_1 = b + Gx_1.$$

Тогда, если d_1 и $d_2 = -g_2 + \alpha_1 d_1$ сопряжены, то

$$d^T_2 G d_1 = 0,$$

т.е.

$$-g^T_2 G d_1 + \alpha_1 d^T_1 G d_1 = 0,$$

следовательно,

$$(-g^T_2 - \alpha_1 g^T_1) G (x_2 - x_1)/\lambda_1 = 0,$$

откуда

$$(-g^T_2 - \alpha_1 g^T_1)(g_2 - g_1)/\lambda_1 = 0.$$

Таким образом,

$$-g^2_2 + \alpha_1 g^2_1 = 0.$$

Остальные члены исчезают из соотношения (6), и, следовательно

$$\alpha_1 = g^2_2/g^2_1,$$

что и требовалось доказать. Это как раз и есть соотношение (5) при k=1.

Теперь перейдем к доказательству соотношений (4) и (5) по индукции, полагая, что векторы $d_1,d_2,..,d_k$ получены описанным выше способом и являются взаимно сопряженными.

Точка $x_{k+1}=x_k+\lambda_k d_k$ является минимумом функции $f(x_k+\lambda_k d_k)$ на прямой $x_k+\lambda_k d_k$.

Тогда

$$g^T_{k+1} d_k=0. \qquad (7)$$

Имеем

$$x_{k+1}=x_k+\lambda_k d_k= x_{k-1}+\lambda_{k-1} d_{k-1}+\lambda_k d_k \text{ и т.д.}$$

Таким образом,

$$x_{k+1}=x_{j+1}+e^{k}_{i=j+1}\lambda_i d_i; \text{ при } 1\le j\le k\text{-}1, \qquad (8)$$

следовательно,

$$Gx_{k+1}=Gx_{j+1}+e^{k}_{i=j+1}\lambda_i Gd_i,$$

тогда

$$g^T_{k+1}=g^T_{j+1}+e^{k}_{i=j+1}\lambda_i d^T_i G \text{ при при } 1\le j\le k\text{-}1,$$

откуда

$$g^T_{k+1} d_j=g^T_{j+1} d_j+e^{k}_{i=j+1}\lambda_i d^T_i G d_j.$$

В результате преобразований имеем $g^T_{j+1} d_j=0$ (в соответствии с соотношениями (6) и (7)) и из-за взаимной сопряженности $d^T_i G d_j=0$ при $j<i$. Таким образом, каждое слагаемое в правой части равно нулю.

Следовательно

$$g^T_{k+1}d_j=0 \text{ при } j=1,2,...,k\text{-}1 \qquad (9)$$

и из соотношения (7) окончательно имеем

$$g^T_{k+1}d_j=0 \text{ при } j=1,2,...,k. \qquad (10)$$

Таким образом, было доказано, что вектор g_{k+1} ортогонален каждому из векторов $d_1,d_2,...,d_k$.

Можно также показать, что вектор g_{k+1} ортогонален векторам $g_1,g_2,...,g_k$.

Из соотношения (10) имеем

$$g^T_{k+1}d_j=0 \text{ при } j=1,2,...,k.$$

Так как из предположения в начале доказательства по индукции

$$d_j=-g_j+\alpha_{j-1}d_{j-1},$$

то приведенное выше соотношение принимает вид

$$-g^T_{k+1}g_j+\alpha_{j-1}g^T_{k+1}d_{j-1}=0,$$

следовательно, $-g_{k+1}g_j=0$, поскольку $g^T_{k+1}d_{j-1}=0$ из соотношения (10).

Таким образом

$$g^T_{k+1}g_j=0 \text{ при } j=1,2,...,k. \qquad (11)$$

Доказательство по индукции будет закончено, если показать, что вектор d_{k+1}, определенный в соотношении (4), сопряжен с векторами $d_1,d_2,...,d_k$.

Для $j=1,2,...,k\text{-}1$ имеем

$$d^T_{k+1}Gd_j=-g^T_{k+1}Gd_j+\alpha_k d^T_k Gd_j=-g^T_{k+1}Gd_j$$

в силу взаимной сопряженности.

Тогда

$$-g^T_{k+1}Gd_j=-g^T_{k+1}G(x_{j+1}-x_j)/\lambda_j= g^T_{k+1}G(g_{j+1}-g_j)/\lambda_j=0$$

с учетом соотношения (11).

Таким образом, $d^T_kGd_j=0$ при $j=1,2,...,k-1$, и это справедливо для любого α_k. Для завершения доказательства необходимо определить α_k так, чтобы выполнялось равенство

$$d^T_k Gd_k=0:$$

Следовательно,

$$d^T_k Gd_k=(-g^2_{k+1}+\alpha_k g^2_k)/\lambda_k,$$

поскольку все другие члены из правой части исчезают в силу соотношений (10) и (11).

Следовательно, направление d_{k+1} будет сопряжено с направлением d_k, если $\alpha_k=g^2_{k+1}/g^2_k$, что и требовалось доказать.

Таким образом, направления поиска в методе Флетчера-Ривса являются взаимно сопряженными и в данном методе минимум квадратичной функции n переменных можно найти не более чем за n шагов. Это означает, что одномерный поиск производится с нужной точностью и устраняются любые ошибки округления, которые могут возникнуть.

Вышеописанный метод будет применим и к неквадратичным функциям, так как если поиск осуществляется вблизи минимума, то можно надеяться на достижение квадратичной сходимости, когда имеет место квадратичная аппроксимация. Флетчер и Ривс полагают, что в этой ситуации каждое n-е направление поиска должно быть

направлением наискорейшего спуска и при построении сопряженных направлений должен быть произведен *рестарт*.

Алгоритм метода Флетчера-Ривса.

Множество X называется выпуклым, если оно содержит всякий отрезок, концы которого принадлежат X , т.е.

$\lambda * x_1 + (1 - \lambda) * x_2 \in X,$

$x_1, x_2 \in X, \ \lambda \in [0,1].$

Функция $f(x)$, определенная на выпуклом множестве X, называется выпуклой, если

$$f(\lambda * x_1 + (1 - \lambda) * x_2) \le f(\lambda * x_1) + f((1 - \lambda) * x_2), \ x_1, x_2 \in X, \lambda \in [0, 1].$$

Алгоритм

1. Задаются: x^0— начальное приближение, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, M – предельное число итераций;
2. Количество итераций $n = 0$;
3. Вычисляется: $\text{grad} f(x^n)$;
4. Вычисляется $\| \text{grad} f(x^n) \|$;

   4.1) если $\| \text{grad} f(x^n) \| < \varepsilon_1$, то

   $x^* = x^n$;

   4.2) если $\| \text{grad} f(x^n) \| > \varepsilon_1$, то к 5);

5. $n \ge M$

   5.1) если выполняется, то $x^* = x^n$;

5.2), если не выполняется, то при $n = 0$ к 6)

при $n \geq 1$ к 7)

6. $p^0 = -\text{grad}f(x^0)$;
7. $\beta = (||\text{grad}f(x^n)|| / ||\text{grad}f(x^{n-1})||)^2$;
8. $p^n = -\text{grad}f(x^n) + \beta_{n-1} * p^{n-1}$
9. найти минимум функции $\varphi(t_n) = f(x^n - t_n * p^n)$;
10. $x^{n-1} = x^n - t_n * p^n$;
11. Проверяется условие $||x^{n-1} - x^n|| < \varepsilon_2$ и $f(x^{n-1}) - f(x^n) < \varepsilon_2$

11.1) если оба условия выполняются, то $x^* = x^{n-1}$;

11.2) если хотя бы одно не выполняется , то $n = n+1$ и к 3).

4.10. Минимизация неквадратичной целевой функции

Метод Флетчера-Ривса может применятся для минимизации и неквадратичных функций. Он является методом первого порядка и в тоже время скорость его сходимости квадратична. Разумеется, если целевая функция не квадратична, метод уже не будет конечным. Поэтому после $(n+1)$-й итерации процедура повторяется с заменой x_0 на x_{n+1}, а счет заканчивается при $||f'(x_{k+1})|| \leq \varepsilon$, где ε – заданное число. При минимизации неквадратичных функций обычно применяется следующая модификация метода Флетчера-Ривса.

Алгоритм метода Флетчера-Ривса для неквадратичных целевых функций

Шаг 1. При $k = 0$ ввод начального приближения x_0 и условия останова ε_3. Вычисление антиградиента $S_0 = -f'(x_0)$.

Шаг 2. Решение задачи одномерной минимизации по α функции $f(x_k + \alpha \cdot S_k)$, в результате чего определяется величина шага α_k и точка $x_{k+1}=x_k+\alpha_k \cdot S_k$.

Шаг 3. Вычисление величин $f(x_{k+1})$ и $f'(x_{k+1})$.

Шаг 4. Если $\|f'(x_{k+1})\| \le \varepsilon_3$, то точка x_{k+1} – решение задачи и на этом поиск заканчивается. Иначе определяется коэффициент β_k по формуле:

$$\beta_k = \begin{cases} \dfrac{(f'(x), f'(x_{k+1}) - f'(x_k))}{(f'(x_k), f'(x_k))}, при \dots k+1 \notin I \\ 0, при \dots k+1 \in I \end{cases}$$

Шаг 5. Вычисление S_{k+1} по формуле $S_{k+1}= - f'(x_{k+1})+\beta_k \cdot S_k$; $k = k + 1$, переход к шагу 2.

Здесь I – множество индексов, $I = \{0, n, 2n, 3n, \dots\}$. Значения k, для которых $\beta_k = 0$, называют *моментами обновления метода*. Таким образом, обновление метода происходит через каждые n шагов.

4.11. Метод Дэвидона — Флетчера — Пауэлла (ДФП)

Первоначально метод был предложен Дэвидоном и затем развит Флетчером и Пауэллом. Метод Дэвидона-Флетчера-Пауэлла называют также и методом переменной метрики. Он попадает в общий класс квазиньютоновских процедур, в которых направления поиска задаются в виде $-D_j$*grad($f(y)$). Направление градиента является, таким образом, отклоненным в результате умножения на $-D_j$, где D_j - положительно определенная симметрическая матрица порядка $n \times n$, аппроксимирующая обратную матрицу Гессе. На следующем шаге матрица D_{j+1} представляется в виде суммы D_j и двух симметрических матриц ранга один каждая. В связи с этим схема иногда называется схемой коррекции ранга два.

Алгорим метода Дэвидона - Флетчера - Пауэлла

Начальный этап. Пусть eps >0 - константа для остановки. Выбрать точку x_1 и начальную симметрическую положительно определенную матрицу D_1 . Положить $y_1 = x_1$, k=j=1 и перейти к основному этапу.

Основной этап.

Шаг 1. Если ||grad($f(x)$)|| < eps , то остановиться; в противном случае положить $d_j = -D_j$*grad($f(y_j)$) и взять в качестве lym$j$ - оптимальное решение задачи минимизации $f(y_j$ + lym*d_j) при lym ≥ 0. Положить $y_{[j+1]} = y_j$ + lym$j$*d_j. Если $j < n$, то перейти к шагу 2. Если j=n, то положить y_1=$x_{[k+1]}$=$y_{[n+1]}$, заменить k на k+1, положить j=1 и повторить шаг 1.

Шаг 2. Построить D_{j+1} следующим образом:

$$D_{j+1} = D_j + \frac{p_j p_j(t)}{p_j(t) q_j} - \frac{D_j q_j q_j(t) D_j}{q_j(t) D_j q_j},$$

где

p_j = lym$j$*d_j,

q_j = grad($f(y_{[j+1]})$) - grad($f(y_j)$).

Заменить j на j+1 и перети к шагу 1.

4.12. Некоторые методы первого порядка в иной интерпретации

В основе всех методов, описываемых в этом разделе, лежит идея восстановления квадратичной аппроксимации функции по значениям ее градиентов в ряде точек. Тем самым методы объединяют достоинства градиентного метода (не требуется вычисление матрицы вторых производных) и метода Ньютона (быстрая сходимость вследствие использования квадратичной аппроксимации).

1. Квазиньютоновские методы. Эти методы имеют общую структуру:

$$x^{k+1} = x^k - \gamma_k H_k \nabla f(x^k), \qquad (1)$$

где матрица H_k пересчитывается рекуррентным способом на основе информации, полученной на k-й итерации, так что

$$H_k - [\nabla^2 f(x^k)]^{-1} \to 0.$$

Таким образом, методы в пределе переходят в ньютоновский, что и объясняет их название. Отметим некоторые общие свойства методов такого типа. Доказательство приводимых ниже лемм может быть без труда получено с использованием описанной ранее техники.

Лемма 1. *Пусть* $f(x) \geqslant f^*$, $f(x)$ *дифференцируема,* $\nabla f(x)$ *удовлетворяет условию Липшица и*

$$mI \leqslant H_k \leqslant MI, \quad m > 0. \qquad (2)$$

Тогда в методе (1) *с* $\gamma_k \equiv \gamma$, *где* $\gamma > 0$ *достаточно мало, будет* $\nabla f(x^k) \to 0$.

Лемма 2. *Пусть* x^**— невырожденная точка минимума* $f(x)$, $f(x)$ *дважды непрерывно дифференцируема в окрестности* x^* *и*

$$\| H_k - [\nabla^2 f(x^*)]^{-1} \| \to 0. \qquad (3)$$

Тогда метод (1) *с* $\gamma_k = 1$ *локально сходится к* x^* *быстрее любой геометрической прогрессии.*

Таким образом, при любых равномерно положительно определенных H_k метод (1) обладает глобальной сходимостью, а при условии (3) в окрестности минимума метод сходится со сверхлинейной скоростью.

Перейдем к вопросу о способах построения матриц H_k, аппроксимирующих $[\nabla^2 f(x^k)]^{-1}$. В принципе их можно формировать с помощью конечно-разностной аппроксимации. Именно, из точки x^k можно сделать n «пробных шагов» длины α_k по координатным осям и вычислить в этих точках градиенты. Соответствующая разностная аппроксимация будет искомой, если $\alpha_k \to 0$.

Однако такой прямолинейный способ аппроксимации неэкономен — в нем делается n пробных вычислений градиента на каждой итерации и никак не используются градиенты, найденные на предыдущих итерациях. Кроме того, в нем требуется обращать матрицу. Основная идея квазиньютоновских методов заключается, во-первых, в том, чтобы не делать специальных пробных шагов, а использовать найденные градиенты в предыдущих точках (поскольку они близки к x^k), а во-вторых, в том, чтобы строить аппроксимацию непосредственно для обратной матрицы $[\nabla^2 f(x^k)]^{-1}$. Обозначим

$$p^k = -H_k \nabla f(x^k), \quad y^k = \nabla f(x^{k+1}) - \nabla f(x^k). \tag{4}$$

Тогда для квадратичной фукции

$$f(x) = (Ax, \ x)/2 - (b, \ x),$$

имеем $y^k = A(x^{k+1} - x^k) = \gamma_k A p^k$, т. е.

$$\gamma_k p_k = A^{-1} y^k. \tag{5}$$

Поэтому для нового приближения H_{k+1} к $[\nabla^2 f(x^{k+1})]^{-1}$ естественно потребовать выполнения так называемого *квазиньютоновского условия*

$$H_{k+1} y^k = \gamma_k p^k. \tag{6}$$

Кроме того, удобно получать H_{k+1} как поправку к H_k с помощью матриц первого или второго ранга. Наконец, эти поправки должны быть такими, чтобы для квадратичного случая оказалось $H_n = A^{-1}$.

Основным техническим инструментом анализа подобных методов является следующая лемма об обращении матриц.

Лемма 3. *Пусть B — матрица $n \times n$, для которой B^{-1} существует, a, b — векторы из* $\mathbf{R}^n$,

$$(B^{-1}a, b) \neq -1, \ A = B + ab^T.$$

Тогда

$$A^{-1} = B^{-1} - (1 + (B^{-1}a, \ b))^{-1} B^{-1} a (B^{-1} b)^T. \tag{7}$$

Лемма доказывается прямой проверкой.

Таким образом, если известна матрица, обратная к B, а матрица A получена из B добавлением матрицы ранга 1, то обратная к A находится без труда.

Приведем примеры формул пересчета матриц H_k:

а) *метод Давидона — Флетчера — Пауэлла (ДФП):*

$$H_{k+1} = H_k - \frac{H_k y^k (y^k)^T H_k}{(H_k y^k, \ y^k)} + \gamma_k \frac{p^k (p^k)^T}{(p^k, \ y^k)}, \quad H_0 > 0; \tag{8}$$

б) *метод Бройдена:*

$$H_{k+1} = H_k - \frac{(\gamma_k p^k - H_k y^k)(\gamma_k p^k - H_k y^k)^T}{(\gamma_k p^k - H_k y^k, \ y^k)}, \quad H_0 > 0; \tag{9}$$

в) *метод Бройдена — Флетчера — Шенно (БФШ):*

$$H_{k+1} = H_k + \frac{\rho_k p^k (p^k)^T - p^k (y^k)^T H_k - H_k y^k (p^k)^T}{(y^k, p^k)},$$
$$\rho_k = \gamma_k + \frac{(H_k y^k, y^k)}{(y^k, p^k)}, \quad H_0 > 0. \tag{10}$$

Оказывается, для всех формул (8) — (10) выполнено квазиньютоновское условие (6). А если $\gamma_k > 0$ — произвольные числа, p^k — произвольные, линейно независимые векторы, y^k удовлетворяют соотношению (5) с $A^{-1} > 0$, то при любом $H_0 > 0$ будет $H_n = A^{-1}$. Отсюда следует

Теорема 1. *При любых* x^0, $H_0 > 0$ *метод* (1), (4) *с любой из формул пересчета* (8), (9), (10) *и* $\gamma_k = \underset{\gamma}{\operatorname{argmin}} f(x^k + \gamma p^k)$

для $f(x) = (Ax, x)/2 - (b, x)$, $A > 0$, *будет конечным:* $x^n = x^* = A^{-1}b$.

Более того, можно показать, что, несмотря на различие формул пересчета, последовательности x^k, генерируемые каждым вариантом метода, для квадратичной функции $f(x)$ совпадают.

Для неквадратичных функций квазиньютоновские методы в записанной выше форме применимы, но они перестают быть конечными. В связи с этим при $k > n$ можно либо продолжать счет по этим же формулам, либо ввести процедуру обновления (заменять матрицу H_k на H_0 через каждые n итераций).

Доказана сверхлинейная (или квадратичная) скорость сходимости многих вариантов квазиньютоновских методов в окрестности невырожденной точки минимума.

Эти результаты выглядят естественными в свете утверждений лемм 1 и 2 и теоремы 1, однако их полное доказательство весьма громоздко.

Квазиньютоновские методы чрезвычайно популярны, им посвящен огромное количество работ. Такое внимание объясняется упоминавшимися выше достоинствами методов — они требуют лишь одного вычисления градиента на каждом шаге, в них не нужно обращать матрицу или решать систему линейных уравнений, они обладают глобальной сходимостью, в окрестности решения скорость сходимости высока (часто квадратична) и т. п. Однако они имеют и дефекты по сравнению, например, с методом сопряженных градиентов. Главный из них заключается в необходимости хранить и пересчитывать матрицу H_k размерности $n \times n$, что для больших n требует значительного объема памяти ЭВМ.

При численной проверке методов обычно наилучшие резульаты дает вариант (10).

2. Методы переменной метрики и методы сопряженных направлений. Выше квазиньютоновскне методы были получены как приближения к методу Ньютона. Однако на них можно посмотреть и с другой точки зрения.

Выясним прежде всего, как влияет *выбор метрики* на вид и свойства градиентного метода. Пусть в пространстве $\mathbf{R}^n$ наряду с исходным скалярным произведением (x, y) задано с помощью матрицы $A > 0$ другое скалярное произведение

$$(x, y)_1 = (Ax, y). \tag{11}$$

В этом случае A задает новую метрику в $\mathbf{R}^n$:

$$\| x - y \|_1^2 = (A(x - y), x - y). \tag{12}$$

Выпишем градиент дифференцируемой функции $f(x)$ в новой метрике:

$$f(x + y) = f(x) + (\nabla f(x), y) + o(\| y \|) =$$
$$= f(x) + (AA^{-1}\nabla f(x), y) + o(\| y \|) = f(x) + (a, y)_1 + o(\| y \|_1),$$
$$a = A^{-1}\nabla f(x).$$

В соответствии с определением вектор a есть градиент $f(x)$ в пространстве со скалярным произведением (11). Итак,

$$\nabla_1 f(x) = A^{-1}\nabla f(x). \tag{13}$$

В новой метрике градиентный метод приобретает вид

$$x^{k+1} = x^k - \gamma_k \nabla_1 f(x^k) = x^k - \gamma_k A^{-1}\nabla f(x^k) \tag{14}$$

и отличается от исходного градиентного метода наличием матрицы A^{-1} Иными словами, градиентный метод не инвариантен к выбору метрики пространства. Естественно попытаться выбрать метрику так, чтобы ускорить сходимость метода. Для квадратичной функции

$$f(x) = (Bx, x)/2 - (b, x) = (1/2)(A^{-1}Bx, x)_1 - (A^{-1}b, x)_1 \tag{15}$$

скорость сходимости (14) определяется знаменателем прогрессии q — $(L— l)/(L+l)$, где L, I — наибольшее и наименьшее собственные значения матрицы $A^{-1}B$. Чем ближе эта матрица к единичной, тем меньше q. Наилучший способ — выбрать $A = B$, тогда $A^{-1}B = I$, $q = 0$, т. е. если задать метрику с помощью матрицы B, то градиентный метод (с $\gamma_k \equiv 1$) даст точное решение за 1 шаг. Это не удивительно, так как в этой метрике $f(x) = (1/2)(x, x)_1 - (A^{-1}b, x)_1$, т. е. линии уровня $f(x)$ — сферы, а обусловленность μ равна единице.

Для неквадратичной функции метод

$$x^{k+1} = x^k - \gamma_k H_k \nabla f(x^k), \qquad H_k > 0, \tag{16}$$

может рассматриваться как градиентный в метрике

$$(x,\ y)_k = \left(H_k^{-1}x,\ y\right), \tag{17}$$

и «оптимальным» выбором метрики является $H_k = [\nabla^2 f(x^k)]^{-1}$. Иными словами, квазиньютоновские методы могут трактоваться как градиентные, в которых на каждом шаге выбирается новая метрика, по возможности близкая к наилучшей. В связи с этим часто употребляют термин *методы переменной метрики* как синоним квазиньютоновских методов.

Такая интерпретация полезна и как эвристический способ построения новых вариантов квазиньютоновских методов. Например, можно получить новую метрику путем «растяжения» пространства в направлении последнего градиента или в направлении разности двух последовательных градиентов и т. п. Мы остановимся на таких методах подробнее в последующих разделах.

Другой подход к построению эффективных методов первого порядка связан с использованием понятия *сопряженных направлений.* Мы уже отмечали, что, зная набор сопряженных направлений $p^1, \ldots, p^n$:

$$(Ap^i,\ p^j) = 0, \quad i \neq j, \tag{18}$$

можно найти минимум квадратичной функции $f(x) = (Ax,\ x)/2 - (b,\ x)$ за n одномерных минимизаций:

$$x^{k+1} = x^k - \alpha_k p^k, \quad \alpha_k = \arg\min_{\alpha} f(x^k - \alpha p^k). \tag{19}$$

Тогда при любом x^0 будет $x^n = x^* = A^{-1}b$. Один способ построения сопряженных направлений использовался в методе сопряженных градиентов — в нем процессу A-ортогонализации подвергались последовательно вычисляемые градиенты. Однако возможны и другие способы.

Пусть $p^1, \ldots, p^k,\ k < n,$ —уже построенные сопряженные векторы,

$$(Ap^i,\ p^j) = 0, \quad 1 \leqslant i,\ j \leqslant k, \quad i \neq j, \tag{20}$$

а x^k — соответствующие им точки в методе (19). Следующий вектор p^{k+1} должен удовлетворять соотношению

$$(p^{k+1},\ Ap^i) = 0, \quad i = 1, \ldots, k.$$

Поскольку

$$p^i = \alpha_i^{-1}(x^{i+1} - x^i),\ Ap^i = \alpha_i^{-1}(\nabla f(x^{i+1}) - \nabla f(x^i)) = \alpha_i^{-1} y^i,$$

то это эквивалентно условию

$$(p^{k+1}, y^i) = 0, \quad i = 1, \ldots, k. \tag{21}$$

Итак, новое сопряженное направление p^{k+1} должно удовлетворять условиям ортогональности (21). Подвергая такому процессу ортогонализации любой набор линейно независимых векторов,

получим различные наборы сопряженных направлений. Этот же процесс может быть применен к неквадратичной функции:

$$x^{k+1} = x^k - \alpha_k p^k, \quad \alpha_k = \arg\min_{\alpha \geq 0} f(x^k - \alpha p^k),$$

$$(p^{k+1}, y^i) = 0, \quad i = 1, \ldots, k, \quad y^i = \nabla f(x^{i+1}) - \nabla f(x^i). \quad (22)$$

Обычно при этом ищут p^{k+1} в виде

$$p^{k+1} = H_{k+1} \nabla f(x^{k+1}), \quad H_{k+1} = H_k + \Delta H_k \quad (23)$$

и вместо непосредственного запоминания векторов y^i, $i = 1, \ldots, k$, запоминают матрицу H_k. Таким образом, методы принимают ту же форму (1), что и квазиньютоновские. Разница лишь в том, что при этом не обязательно $H_k \to [\nabla^2 f(x^k)]^{-1}$; в некоторых вариантах метода оказывается (для квадратичной функции) $H_n = 0$. Поэтому в таких методах обязательно должно осуществляться обновление.

Выпишем алгоритм одного из простейших методов данного класса:

$$x^{k+1} = x^k + \alpha_k p^k, \qquad \alpha_k = \arg\min_{\alpha \geq 0} f(x^k + \alpha p^k),$$

$$p^k = -H_k \nabla f(x^k), \quad y^k = \nabla f(x^{k+1}) - \nabla f(x^k),$$

$$H_{k+1} = H_k - \frac{H_k y^k (y^k)^T H_k}{(H_k y^k, y^k)}, \qquad k+1 \neq n, 2n, \ldots$$

$$H_0 = H_n = H_{2n} = \ldots = I. \quad (24)$$

Оказывается, что для квадратичной функции в методе (24) p^k являются сопряженными направлениями, $H_k \geq 0$ для всех $k \leq n$, $H_n = 0$. Для неквадратичных функций доказана квадратичная локальная сходимость методов данного класса в окрестности невырожденного минимума.

3. Метод секущих. Одним из простейших и наиболее распространенных методов решения одномерного уравнения

$$g(x) = 0 \quad (25)$$

является *метод секущих,* сущность которого видна из рис. 1.

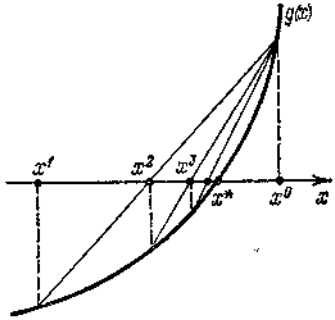


Рис. 1. Метод секущих

Его можно обобщить на многомерный случай — если g: $\mathbf{R}^n \to \mathbf{R}^n$, то можно вычислить g в n+1 точках, построить линейную аппроксимацию и найти ее корень, который является очередным приближением к решению (25).

Применительно к задаче минимизации $f(x)$ в $\mathbf{R}^n$, т. е. к задаче решения уравнения $\nabla f(x) = 0$, метод принимает следующий вид. Пусть x^k, x^{k-1x}.....x^{k-n} — n+ 1 точек в $\mathbf{R}^n$, $\nabla f(x^k)$,....., $\nabla f(x^{k-n})$ — вычисленные в них градиенты. Решим систему n+1 линейных уравнений с n+1 переменными $\lambda_0, \lambda_1, \ldots, \lambda_n$:

$$\sum_{i=0}^{n} \lambda_i \nabla f(x^{k-i}) = 0, \quad \sum_{i=0}^{n} \lambda_i = 1 \tag{26}$$

и построим точку

$$x^{k+1} = \sum_{i=0}^{n} \lambda_i x^{k-i}. \tag{27}$$

Далее процесс повторяется для n + 1 последних точек x^{k+1}, x^k, ..., x^{k-n+1} и т. д. Нетрудно проверить, что для n— 1 такой метод совпадает с методом секущих для решения уравнения $\nabla f(x) = 0$.

Теорема 2. *Если векторы $x^1 - x^0$, $x^2 - x^0$, ..., $x^n - x^0$ линейно независимы, а $f(x)$ квадратична с $\nabla^2 f(x) \equiv A > 0$, то x^{n+1} — точка минимума $f(x)$.*

В системе линейных уравнений (26) на каждой итерации меняется лишь один столбец, поэтому нет необходимости решать ее каждый раз заново, а можно воспользоваться следующим результатом.

Лемма 4. *Пусть B — квадратная матрица $n \times n$ со столбцами b^1,..., b^n, а B отличается от нее первым столбцом (b^1 заменено на $\tilde{b}^1$). Тогда*

$$\tilde{c}^i = c^i - \frac{(\tilde{b}^1 - b^1, c^i)}{1 + (\tilde{b}^1 - b^1, c^1)} c^1, \qquad (28)$$

где c^i *— строки* B^{-1}, $\tilde{c}^{i\,1}$ *— строки* $\tilde{B}^{-1}$.

Для доказательства достаточно представить $\tilde{B}$ в виде $\tilde{B} = B + (\tilde{b}^1 - b^1)e^{\mathrm{T}}$, где $e = (1, 0, ..., 0)$, и воспользоваться леммой 3.

Однако в описанной выше форме метод секущих не является удовлетворительным. Так, он не обладает свойством глобальной сходимости. Для устранения этого недостатка можно применять стандартные средства, например регулировку длины шага (из x^k делается шаг по направлению $\sum_i \lambda_i x^{k-i}$). Вторым дефектом метода является его склонность к вырождению — в процессе счета последовательные приближения оказываются лежащими (приближенно) в подпространстве пространства $\mathbf{R}^n$. Соответствующая система линейных уравнений (26) плохо обусловлена и ее решение неустойчиво. Для преодоления этого недостатка можно модифицировать метод с тем, чтобы система базисных точек была заведомо невырожденной. Например, можно добавлять на каждой итерации точку, делая шаг по координатным осям (в циклическом порядке). Для модифицированных подобным образом методов можно доказать сверхлинейную сходимость.

4. Другие идеи построения методов первого порядка. При всем разнообразии описанных выше алгоритмов первого порядка идея их оставалась одинаковой — использовать квадратичную аппроксимацию функции вблизи минимума. Как правило, эти алгоритмы конечны для квадратичных функций, а в общем случае их эффективность тем выше, чем ближе функция к квадратичной. Однако квадратичная модель может считаться естественной лишь в окрестности экстремума; вдали от него поведение минимизируемой функции может быть совсем иным. Поэтому для всех описанных выше методов отнюдь не гарантируется даже разумность стратегии оптимизации на начальных этапах поиска.

В связи с этим целесообразно использовать другие модели функции, отличные от квадратичной. На первый взгляд естественно попытаться строить полиномиальные модели на основе старших производных — следующих членов ряда Тейлора. Такие попытки делались, однако они вряд ли перспективны. Во-первых, прямое вычисление старших производных в многомерных задачах обычно требует слишком громоздких вычислений и большого объема памяти, а их восстановление по младшим производным предполагает вычисление

последних в огромном числе точек. Во-вторых, решение вспомогательных задач минимизации полиномиальных функций, за редкими исключениями, не может быть осуществлено в аналитической форме.

Простой и важный класс представляют модели, основанные на аппроксимации функции однородной. Функция $f(x)$, $x \in \mathbf{R}^n$, называется *однородной относительно точки* x^* *с показателем* $\gamma > 0$, если

$$f(x^* + \lambda(x - x^*)) - f(x^*) = \lambda^\gamma (f(x) - f(x^*)) \qquad (29)$$

для всех $x \in \mathbf{R}^n$ и $\lambda \geq 0$.

Ниже приведены примеры однородных функций.

1. Аффинная функция $f(x) = (a, x)$— β однородна с $\gamma = 1$ для любого x^*.
2. Квадратичная функция $f(x) = (Ax, x)/2 - (b, x)$, где A^{-1} существует, является однородна относительно $x^* = A^{-1}b$ с $\gamma = 2$.
3. Пусть существует решение x^* системы $(a^i, x) = \beta_i$, $i = 1, \ldots, m$, $x \in \in \mathbf{R}^n$. Функция

$$f(x) = \sum_{i=1}^{m} |(a^i, x) - \beta_i|^\gamma, \ \gamma > 0,$$

однородна относительно x^* с показателем γ.

4. Если $\varphi^* = 0$, $F(z) = |z|^\alpha$, $\alpha > 0$, то $f(x)$ вида (36) — однородная относительно x^* с показателем 2α.

Дифференцируемая однородная функция удовлетворяет важному соотношению

$$f(x) - f(x^*) = \gamma^{-1} (\nabla f(x), x - x^*). \qquad (30)$$

Чтобы доказать (30), возьмем в (29) $\lambda = 1+\varepsilon$

$$f(x + \varepsilon(x - x^*)) - f(x^*) = (1 + \varepsilon)^\gamma (f(x) - f(x^*)),$$
$$\varepsilon\gamma (f(x) - f(x^*)) = \varepsilon (\nabla f(x), x - x^*) + o(\varepsilon).$$

Устремляя ε к 0, получаем (30),

Точка x^* не обязательно является минимумом $f(x)$ (см. примеры 2 и 3). Однако если $f(x)$ достигает минимума, то x^* — точка глобального минимума $f(x)$. Действительно, пусть $f(\tilde{x}) = f^* = \min f(x)$, тогда $\nabla f(x)=0$. Подставляя $\tilde{x}$ вместо x в (30), получаем, что $f(x^*) = f(\tilde{x}) = f^*$, т. е. x^* — точка глобального минимума. Именно этот случай и будет рассматриваться далее.

С помощью (30) можно найти точку минимума x^*, вычислив $f(x)$ и $\nabla f(x)$ в конечном числе точек. Действительно, если γ известно, то, взяв $n + 1$ точек $x^0, \ldots, x^n$, мы получаем систему

$$\gamma f(x^i) - \alpha + (\nabla f(x^i), x^*) = (\nabla f(x^i), x^i), \quad i = 0, \ldots, n, \qquad (31)$$

линейную относительно $n + 1$ переменных x^*, α $(\alpha = \gamma f(x^*))$.

Исключая переменную α, получаем *n* линейных уравнений для определения $x^* \in \mathbb{R}^n$:

$$(\nabla f(x^i) - \nabla f(x^0),\ x^*) =$$
$$= (\nabla f(x^i),\ x^i) - (\nabla f(x^0),\ x^0) - \gamma\,(f(x^i) - f(x^0)), \quad i = 1, \ldots, n. \quad (32)$$

Если же γ неизвестно, то можно взять *n*+2 точек $x^0, \ldots, x^{n+1}$ и определить *n* + 1 переменных γ, *x** из линейной системы (32), в которой следует взять *n*+ 1 уравнений.

Аналогичный подход можно применить для минимизации функций общего вида подобно тому, как это делалось в методе секущих. В самом деле, пусть уже построены приближения $x^0 \ldots\ldots x^k$, $k > n$. Взяв последние *n* + 1 из них (или *n* + 2, если γ неизвестно), решим систему (относительно *x,* α, γ, либо *x,* α*)*

$$(\nabla f(x^i),\ x) - \alpha + \gamma f(x^i) = (\nabla f(x^i),\ x^i), \qquad i = k,\ k-1, \ldots, \quad (33)$$

а решение *x* выберем в качестве x^{k+1}. Для γ = 2 получаем метод, близкий к методу секущих, но отличающийся от него (в нем, в отличие от метода секущих, используются не только $\nabla f(x^i)$, но и значения функции $f(x^i)$).

Такой процесс следует модифицировать с помощью тех же приемов, что и метод секущих (бороться с вырождением точек x^k путем добавления новых точек, линейно независимых от предыдущих; регулировать длину шага и т. д.). Полезно также сравнивать фактическое значение $f(x^{k+1})$ с «предсказанным» (равным α/γ). Это может служить проверкой предположения о близости функции к однородной. При решении систем линейных уравнений целесообразно использовать близость этих уравнений на соседних итерациях (см. лемму 4).

Для минимизации однородных и близких к ним функций можно применять и другие методы. Так, в *градиентном методе* можно применять специальные способы выбора длины шага. Пусть функция $f(x)$ удовлетворяет условию (30), причем величины $f^* = f(x^*)$ и γ известны. Рассмотрим градиентный метод

$$x^{k+1} = x^k - \frac{\gamma\,(f(x^k) - f^*)}{\|\nabla f(x^k)\|^2}\,\nabla f(x^k). \quad (34)$$

Выбор шага $\gamma_k = \frac{\gamma\,(f(x^k) - f^*)}{\|\nabla f(x^k)\|^2}$ здесь сделан так, чтобы для $x^{k+1} = x^k - \gamma_k \nabla f(x^k)$ удовлетворялось равенство

$$f(x^k) - f^* = \gamma^{-1}(\nabla f(x^k),\ x^k - x^{k+1})$$

ср. с (30). Тогда

$$\|x^{k+1}-x^*\|^2=\|x^k-x^*\|^2-\frac{2\gamma(f(x^k)-f^*)}{\|\nabla f(x^k)\|^2}(\nabla f(x^k),\ x^k-x^*)+$$

$$+\frac{\gamma^2(f(x^k)-f^*)^2}{\|\nabla f(x^k)\|^2}=\|x^k-x^*\|^2-\frac{\gamma^2(f(x^k)-f^*)^2}{\|\nabla f(x^k)\|^2}.$$

Отсюда следует, что если $\|\nabla f(x)\|$ ограничена на множестве $\{x:\ \|x-x^*\|\leqslant\|x^0-x^*\|\}$, то $f(x^k)\to f^*$. Нетрудно видеть, что этот же результат остается справедливым, если в (30) равенство заменить на неравенство

$$f(x)-f^*\leqslant\gamma^{-1}(\nabla f(x),\ x-x^*). \tag{35}$$

Несколько иной класс (по сравнению с однородными) задается формулой

$$f(x)=F(\varphi(x)),\quad \varphi(x)=(Ax,\ x)/2-(b,\ x),\quad A>0, \tag{36}$$

где F: $\mathbf{R}^1\to\mathbf{R}^1$ — монотонная на $[\varphi^*,\infty)$ функция, $\varphi^*=\varphi(x^*)=\min\varphi(x)$. Очевидно, что x^* является точкой минимума $f(x)$.

Если задан явный вид F и φ, то в соответствии с последним замечанием вместо минимизации $f(x)$ можно решать более простую задачу минимизации $\varphi(x)$. Однако часто доступна меньшая информация о задаче. Тогда можно применить следующий *вариант метода сопряженных градиентов:*

$$\begin{aligned}&x^{k+1}=x^k+\alpha_k p^k,\quad \alpha_k=\operatorname*{argmin}_{\alpha\geqslant 0} f(x^k+\alpha p^k),\\ &p^k=-\nabla f(x^k)+\beta_k p^{k-1},\\ &\beta_k=\frac{F'(\varphi(x^{k-1}))\,\|\nabla f(x^k)\|^2}{F'(\varphi(x^k))\,\|\nabla f(x^{k-1})\|^2},\quad \beta_0=0.\end{aligned} \tag{37}$$

Нетрудно проверить, что метод (37) порождает ту же последовательность точек, что и метод сопряженных градиентов для минимизации $\varphi(x)$, а потому является конечным.

Величину $\rho_k=F'(\varphi(x^k))/F'(\varphi(x^{k-1}))$, входящую в формулу для β_k, можно оценивать приближенно, аппроксимируя $F(z)$ квадратичной или степенной функцией. При этом метод (37) можно применять и для минимизации функций, не обязательно имеющих вид (36).

Пример. Для дважды дифференцируемой однородной функции справедливо соотношение $\nabla^2 f(x)(x-x^*)=(\gamma-1)\nabla f(x)$

В целом методы, основанные на однородных и близких к ним аппроксимациях функций, пока мало исследованы.

5. Методы минимизации второго порядка

5.1. Особенности методов второго порядка

Методы безусловной оптимизации второго порядка используют вторые частные производные минимизируемой функции *f(x).* Суть этих методов состоит в следующем.

Необходимым условием экстремума функции многих переменных *f(x)* в точке *x** является равенство нулю ее градиента в этой точке:

$$f'(x^*) = 0.$$

Разложение *f'(x)* в окрестности точки *x*[*k*] в ряд Тейлора с точностью до членов первого порядка позволяет переписать предыдущее уравнение в виде

$$f'(x) = f'(x[k]) + f''(x[k])\,(x - x[k]) = 0.$$

Здесь $f''(x[k]) = H(x[k])$ - матрица вторых производных (матрица Гессе) минимизируемой функции. Следовательно, итерационный процесс для построения последовательных приближений к решению задачи минимизации функции *f(x)* описывается выражением

$$x[k+1] = x[k] - H^{-1}(x[k])\,f'(x[k]),$$

где $H^{-1}(x[k])$ - обратная матрица для матрицы Гессе, а $H^{-1}(x[k])f'(x[k]) = p[k]$ - направление спуска.

Полученный метод минимизации называют *методом Ньютона.* Очевидно, что в данном методе величина шага вдоль направления *p*[*k*] полагается равной единице. Последовательность точек {*x*[*k*]}, получаемая в результате применения итерационного процесса, при определенных предположениях сходится к некоторой стационарной точке *x** функции *f(x).* Если матрица Гессе *H(x*)* положительно определена, точка *x** будет точкой строгого локального минимума функции *f(x).* Последовательность *x*[*k*] сходится к точке *x** только в том случае, когда матрица Гессе целевой функции положительно определена на каждой итерации.

Если функция $f(x)$ является квадратичной, то, независимо от начального приближения $x[0]$ и степени овражности, с помощью метода Ньютона ее минимум находится за один шаг. Это объясняется тем, что направление спуска $p[k] = H^{-1}(x[k])f'(x[k])$ в любых точках $x[0]$ всегда совпадает с направлением в точку минимума x^*. Если же функция $f(x)$ не квадратичная, но выпуклая, метод Ньютона гарантирует ее монотонное убывание от итерации к итерации. При минимизации овражных функций скорость сходимости метода Ньютона более высока по сравнению с градиентными методами. В таком случае вектор $p[k]$ не указывает направление в точку минимума функции $f(x)$, однако имеет большую составляющую вдоль оси оврага и значительно ближе к направлению на минимум, чем антиградиент.

Существенным недостатком метода Ньютона является зависимость сходимости для невыпуклых функций от начального приближения $x[0]$. Если $x[0]$ находится достаточно далеко от точки минимума, то метод может расходиться, т. е. при проведении итерации каждая следующая точка будет более удаленной от точки минимума, чем предыдущая. Сходимость метода, независимо от начального приближения, обеспечивается выбором не только направления спуска $p[k] = H^{-1}(x[k])f'(x[k])$, но и величины шага a вдоль этого направления. Соответствующий алгоритм называют *методом Ньютона с регулировкой шага.* Итерационный процесс в таком случае определяется выражением

$$x[k+1] = x[k] - a_k H^{-1}(x[k])f'(x[k]).$$

Величина шага a_k выбирается из условия минимума функции $f(x)$ по a в направлении движения, т. е. в результате решения задачи одномерной минимизации:

$$f(x[k] - a_k H^{-1}(x[k])f'(x[k]) = \min_{a \geq 0} (f(x[k] - aH^{-1}(x[k])f'(x[k])).$$

Вследствие накопления ошибок в процессе счета матрица Гессе на некоторой итерации может оказаться отрицательно определенной или ее нельзя будет обратить. В таких случаях в подпрограммах оптимизации полагается $H^{-1}(x[k]) = E$, где E — единичная матрица. Очевидно, что итерация при этом осуществляется по методу наискорейшего спуска.

5.2. Методы линейной аппроксимации.

Для оценки градиента функции $f\colon \mathbf{R}^n \to \mathbf{R}^1$ в точке x составим конечно-разностные отношения

$$\Delta_1 = \alpha^{-1}[f(x+\alpha y) - f(x)], \quad \Delta_2 = (2\alpha)^{-1}[f(x+\alpha y) - f(x-\alpha y)], \quad (1)$$

где $y \in \mathbf{R}^n$ — произвольный вектор.

Лемма 1. а) *Если f дифференцируема в x, то*

$$|\Delta_1 - (\nabla f(x), y)| \to 0 \ \textit{при} \ \alpha \to 0. \quad (2)$$

б) *Если* ∇f *удовлетворяет условию Липшица с константой L в окрестности x, то при достаточно малых* α

$$|\Delta_1 - (\nabla f(x), y)| \leqslant L\alpha \|y\|^2/2. \quad (3)$$

в) *Если f дважды дифференцируема и* $\nabla^2 f$ *удовлетворяет условию Липшица в окрестности x, то при достаточно малых* α

$$|\Delta_2 - (\nabla f(x), y)| \leqslant L\alpha^2 \|y\|^3/6. \quad (4)$$

г) *Если f(x) квадратична, то при любом* α

$$\Delta_2 = (\nabla f(x), y). \quad (5)$$

Таким образом, разностные отношения ∇_1 и ∇_2 могут служить приближением для линейной аппроксимации *f(x).* Рассмотрим методы вида

$$x^{k+1} = x^k - \gamma_k s^k, \quad (6)$$

где $\gamma_k \geq 0$ - длина шага, а s^k вычисляется по одной из двух формул

$$s^k = \sum_{i=1}^{m} \alpha_k^{-1}\left[f\left(x^k + \alpha_k h^i\right) - f\left(x^k\right)\right] h^i, \quad (7)$$

$$s^k = \sum_{i=1}^{m} (2\alpha_k)^{-1}\left[f\left(x^k + \alpha_k h^i\right) - f\left(x^k - \alpha_k h^i\right)\right] h^i. \quad (8)$$

Здесь h^i, $i = 1, \ldots, m$, — векторы, задающие направления пробных шагов, α_k — длина пробного шага. Выбирая различные h^i и m, получим те или иные алгоритмы.

а) *Градиентный спуск* — метод нахождения локального минимума (максимума) функции с помощью движения вдоль градиента. Для минимизации функции в направлении градиента используются методы одномерной оптимизации, например, метод золотого сечения. Также можно искать не наилучшую точку в направлении градиента, а какую-

либо лучше текущей. Сходимость метода градиентного спуска зависит от отношения максимального и минимального собственных чисел матрицы Гессе в окрестности минимума (максимума). Чем больше это отношение, тем хуже сходимость метода.

Пусть целевая функция имеет вид:

$$F(\vec{x}) : \mathbb{X} \to \mathbb{R}.$$

И задача оптимизации задана следующим образом:

$$F(\vec{x}) \to \min_{\vec{x} \in \mathbb{X}}$$

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla F$:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]})$$

где $\lambda^{[j]}$ выбирается

- постоянной, в этом случае метод может расходиться;
- дробным шагом, т.е. длина шага в процессе спуска делится на некоторое число;
- наискорейшим спуском:
  $\lambda^{[j]} = \operatorname{argmin}_\lambda F(\vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]}))$

Алгоритм

1. Задают начальное приближение и точность расчёта x^{-0}, ε
2. Рассчитывают $\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]})$, где $\lambda^{[j]} = \operatorname{argmin}_\lambda F(\vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]}))$
3. Проверяют условие остановки:

- Если $\left|\vec{x}^{[j+1]} - \vec{x}^{[j]}\right| > \varepsilon$, то $j = j + 1$ и переход к шагу 2.
- Иначе $\vec{x} = \vec{x}^{[j+1]}$ и останов.

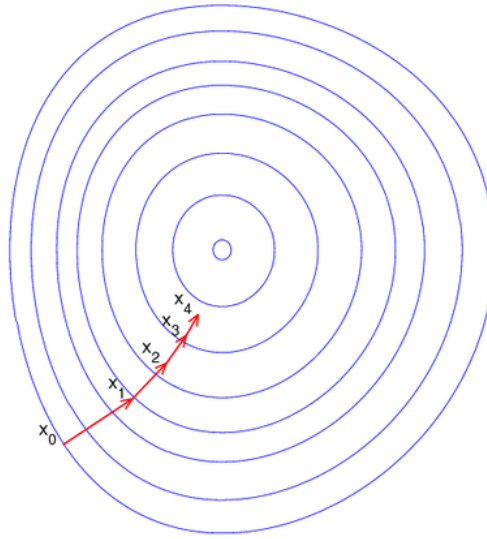


Рис. 1

На рис. 1 приведена иллюстрация последовательных приближений к точке экстремума в направлении наискорейшего спуска (в направлении стрелок) в случае дробного шага. Линии уровня изображены неправильной формы овалами.

Пример

Применим градиентный метод к функции

$$F(x,y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$$

Тогда последовательные приближения будут выглядеть так (рис. 2):

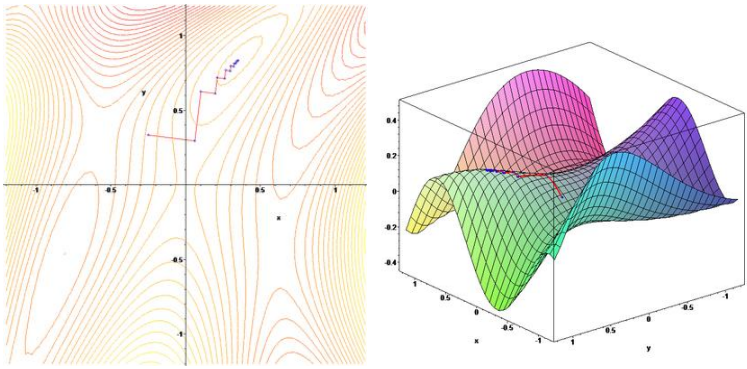


Рис. 2

Упомянем, что метод наискорейшего спуска может иметь трудности в патологических случаях овражных функций, так, к примеру, в случае функции Розенброка.

б) *Метод наискорейшего спуска (метод градиента)*

Выбирают $v_i^{[j]} = -\dfrac{\partial F}{\partial x_i}$, где все производные вычисляются при $x_i = x_i^{[j]}$, и уменьшают длину шага $\lambda^{[j]}$ по мере приближения к минимуму функции *F*.

Для аналитических функций F и малых значений f_i тейлоровское разложение $F(\lambda^{[j]})$ позволяет выбрать оптимальную величину шага

$$\lambda^{[j]} = \frac{\sum_{k=1}^{n}\left(\frac{\partial F}{\partial x_k}\right)^2}{\sum_{k=1}^{n}\sum_{h=1}^{n}\frac{\partial^2 F}{\partial x_k dx_h}\frac{\partial F}{\partial x_k}\frac{\partial F}{\partial x_h}}$$

где все производные вычисляются при $x_i = x_i^{[j]}$. Параболическая интерполяция функции $F(\lambda^{[j]})$ может оказаться более удобной.

Алгоритм

1. Задаются начальное приближение и точность расчёта x^{-0}, ε
2. Рассчитывают $\overrightarrow{x}^{[j+1]} = \overrightarrow{x}^{[j]} - \lambda^{[j]} \nabla F(\overrightarrow{x}^{[j]})$, где $\lambda^{[j]} = \operatorname{argmin}_\lambda F(\vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]}))$
3. Проверяют условие останова:
   - Если $|\vec{x}^{[j+1]} - \vec{x}^{[j]}| > \epsilon$, то $j = j + 1$ и переход к шагу 2.
   - Иначе $\vec{x} = \vec{x}^{[j+1]}$ и останов.

а) *Разностный аналог градиентного метода:* $m = n$, $h^i = e_i$, $i = 1, ..., n$, где e_i — координатные орты. Иначе говоря, пробные шаги делаются по координатным осям, так что метод (6), (7) в координатной записи имеет вид

$$x_i^{k+1} = x_i^k - (\gamma_k/\alpha_k)\left[f\left(x^k + \alpha_k e_i\right) - f\left(x^k\right)\right]. \qquad (9)$$

В соответствии с леммой 1

$$s^k = \sum_{i=1}^{n} \left(\nabla f\left(x^k\right),\ e_i\right) e_i + \varepsilon^k = \nabla f\left(x^k\right) + \varepsilon^k, \qquad (10)$$

где остаточный член ε^k может быть оценен для каждой из формул (7), (8) в зависимости от гладкости $f(x)$.

г) *Метод покоординатного спуска:* $m=1$, $h = e_i$, $j=k \pmod n$. Шаги делаются по координатным осям, выбираемым в циклическом порядке:

$$x_i^{k+1} = \begin{cases} x_i^k - (\gamma_k/\alpha_k)\left[f\left(x^k + \alpha_k e_i\right) - f\left(x^k\right)\right], & i = k \pmod n, \\ x_i^k \quad \text{в противном случае.} \end{cases} \qquad (11)$$

При этом $s^k = \nabla f\left(x^k\right)_l e_l + \varepsilon^k$.

Рассмотрим задачу поиска минимума функции $f(x):\mathbb{R}^n \to \mathbb{R}$, записываемую в виде:

$$f(x) \to \min_{x \in \mathbb{R}^n}$$

В этой постановке описан метод покоординатного спуска, решающий поставленную задачу. Также приведена теорема сходимости метода покоординатного спуска.

Алгоритм

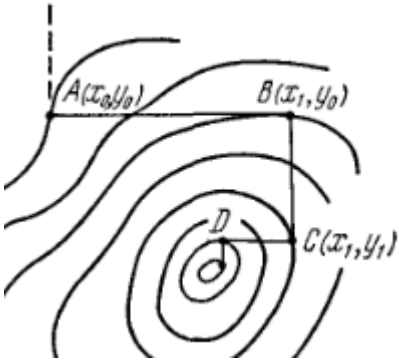


Рис. 3. Иллюстрация метода

Вход: функция f: $R^n \rightarrow R$

Выход: найденная точка оптимума x

1. Инициализация некоторым значением $x_0 \in R^n$
2. Повторять:
   - для $i=1,..,n$
     1. фиксируем значения всех переменных кроме x_i, получая одномерную функцию $f(x_i)$
     2. проводим одномерную оптимизацию по переменной x_i, любым методом одномерной оптимизации
     3. если выполен критерий останова (варианты описаны ниже), то возвращаем текущее значение $x=(x_1,\ldots,x_n)$.

Критерий останова

Критерии остановки процесса приближенного нахождения минимума могут быть основаны на различных соображениях. Некоторые из них:

1. $\|x^{[k+1]} - x^{[k]}\| \leq \epsilon$
2. $\|f(x^{[k+1]}) - f(x^{[k]})\| \leq \epsilon$

Здесь $x^{[k]} \in \mathbb{R}^n$ - значение, полученное после *k*-го шага оптимизации. ε - наперед заданное положительное число.

Сходимость метода

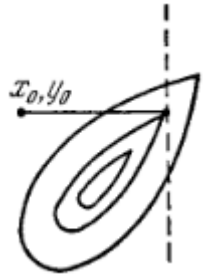


Рис. 4

Легко убедится, что существуют функции, когда метод координатного спуска не приводит даже в локальный оптимум.

Пусть линии уровня образуют истинный овраг (рис. 4), когда спуск по любой координате приводит на <<дно>> оврага, а любое движение по следующей координате (пунктирная линия) ведет на подъем. Никакой дальнейший спуск по координатам в данном случае невозможен, хотя минимум еще не достигнут.

Теорема о сходимости метода покоординатного спуска.

Для простоты рассмотрим функцию двух переменных *f(x,y)*. Выберем некоторое начальное приближение (x_0,y_0) и проведем линию уровня через эту точку. Пусть в области *G*, ограниченной этой линией уровня, выполняются неравенства, означающий положительную определенность квадратичной формы:

$$f''_{xx} \geq a > 0,\ f''_{yy} \geq b > 0,\ |f''_{xy}| \leq c,\ ab > c^2.$$

Тогда спуск по координатам сходится к минимуму из данного начального приближения, причем линейно.

Пример. Для исследования сходимости метода покоординатного спуска была выбрана функция:

$$f(x_1,x_2)=(x_1-1)^2+(x_2-1)^2-x_0{}^*x_1.$$

Начальное приближение - точка (10,10). Использован критерий останова:

$$\|f(x^{[k+1]})-f(x^{[k]})\|\le 10^{-5}$$

Для решения одномерных задач оптимизации использован метод золотого сечения.

Метод получил точность 1e-8 за 7 итераций. Отсюда можно сделать вывод, что метод координатного спуска сходится неплохо на примерах, для которых он применим.

Возникающую одномерную задачу оптимизации можно решать любым методом одномерной оптимизации, например методом золотого сечения.

Метод координатного спуска является простым в реализации методом оптимизации. Главным недостатком метода является его ограниченная применимость.

д) *Метод случайного покоординатного спуска:* $m=1$, $h=e_j$, где j принимает значения 1, ..., n с равной вероятностью. Шаг делается, как и выше, по координатным осям, но они выбираются в случайном порядке.

е) *Метод случайного поиска:* $m=1$, h — случайный вектор, равномерно распределенный на единичной сфере. Здесь движение производится по случайному направлению, а знак и величина шага определяются разностным отношением:

$$x^{k+1} = x^k - (\gamma_k/\alpha_k)\,[f(x^k + \alpha_k h) - f(x^k)]\,h. \qquad (12)$$

Сходимость всех методов гарантируется условием $\alpha_k \to 0$.

Скорость сходимости зависит от гладкости $f(x)$ и способа выбора α_k. С точки зрения погрешностей вычисления выгодно брать α_k большим. Так как чем меньше α_k, тем больше влияние ошибок округления при

вычислении разностных отношений (в (1) приходится вычислять разность двух близких чисел и делить на малое число; это всегда связано с потерей точности). Однако для больших α_k ухудшается точность аппроксимации (лемма 1). Можно показать, что можно обеспечить в описанных выше методах сходимость со скоростью геометрической прогрессии, если

$$\alpha_k \leqslant cq^k,$$

где $q < 1$ — некоторое число.

Вопрос о соотношении скоростей сходимости различных вариантов метода довольно сложен. Рассмотрим важный частный случай, который может служить моделью более реалистических ситуаций. Пусть *f(x)* квадратична:

$$f(x) = (Ax,\ x)/2 - (b,\ x), \qquad A > 0, \tag{13}$$

а γ_k выбирается из условия скорейшего спуска:

$$x^{k+1} = x^k - \gamma_k s^k, \quad \gamma_k = \arg\min_{\gamma \geqslant 0} f(x^k - \gamma s^k). \tag{14}$$

Сравним три способа выбора s^k:

- *симметричная разностная аппроксимация градиента*

$$s^k = \sum_{i=1}^{n} (2\alpha)^{-1} [f(x^k + \alpha e_i) - f(x^k - \alpha e_i)]\, e_i = \nabla f(x^k) \tag{15}$$

(последнее равенство в силу (5));

- *покоординатный спуск*

$$s^k = (2\alpha)^{-1} [f(x^k + \alpha e_i) - f(x^k - \alpha e_i)]\, e_i = \nabla f(x^k)_i\, e_i,\ i = k \pmod n \tag{16}$$

- *случайный поиск*

$$s^k = (2\alpha)^{-1} [f(x^k + \alpha h^k) - f(x^k - \alpha h^k)]\, h^k = (\nabla f(x^k),\ h^k)\, h^k, \tag{17}$$

где h^k — равномерно распределенный на единичной сфере вектор.

Таким образом, (14), (15) совпадает с методом наискорейшего спуска, а (14), (16) хорошо известен в линейной алгебре как *метод Гаусса — Зейделя.*

Соотношение скоростей сходимости методов зависит от различных причин; приведем несколько крайних случаев. Если $A = I$, то (14), (15) и (14), (16) приводят к решению за 1 шаг, тогда как метод случайного поиска сходится в среднеквадратичном не быстрее некоторой геометрической прогрессии. Если $(Ax,\ x) = \sum_{i=1}^{n} \lambda_i x_i^2,\ \lambda_i > 0$, то метод (14), (16) конечен, тогда как (14), (15) — нет. Наконец, если задача плохо обусловлена ($\mu \gg 1$), то можно показать, что метод случайного поиска сходится быстрее градиентного (с учетом разницы в числе вычислений *f(x)* на одной итерации методов). Грубо говоря, для таких задач случайное направление в среднем лучше указывает на

решение, чем антиградиент. Метод Гаусса — Зейделя имеет еще один резерв ускорения сходимости — если заменить в нем γ_k на $\alpha\gamma_k$, $1<\alpha<2$ (так называемая *сверхрелаксация),* то оказывается, что в ряде случаев сходимость резко улучшается.

В целом можно рекомендовать в классе поисковых методов описанного типа метод покоординатного спуска как по его простоте, так и по скорости сходимости.

5.3. Интерполяция кубическими сплайнами

Постановка математической задачи

Одной из основных задач оптимизации является задача об интерполяции функций. Пусть на отрезке $a \le \xi \le$ задана сетка

$$\omega = \left\{x_i : x_0 = a < x_1 < \cdots < x_i < \cdots < x_n = b\right\}$$

и в её узлах заданы значения функции *y(x)*, равные

$$y(x_0) = y_0, \cdots, y(x_i) = y_i, \cdots, y(x_n) = y_n.$$

Требуется построить *интерполянту* — функцию *f(x)*, совпадающую с функцией *y(x)* в узлах сетки:

$$f(x_i) = y_i, i = 0,1,\cdots,n. \qquad (1)$$

Основная цель интерполяции — получить быстрый (экономичный) алгоритм вычисления значений *f(x)* для значений *x*, не содержащихся в таблице данных.

Интерполируюшие функции *f(x)*, как правило строятся в виде линейных комбинаций некоторых элементарных функций:

$$f(x) = \sum_{k=0}^{N} c_k \Phi_k(x),$$

где $\{\Phi_k(x)\}$ — фиксированный линейно независимые функции, $c_0, c_1, \cdots, c_n$ — не определенные пока коэффициенты.

Из условия (1) получаем систему из n+1 уравнений относительно коэффициентов $\{c_k\}$:

$$\sum_{k=0}^{N} c_k \Phi_k(x_i) = y_i, i = 0,1,\cdots,n.$$

Предположим, что система функций $\Phi_k(x)$ такова, что при любом выборе узлов

$$a < x_1 < \cdots < x_i < \cdots < x_n = b$$

отличен от нуля определитель системы:

$$\Delta(\Phi) = \begin{vmatrix} \Phi_0(x_0) & \Phi_1(x_0) & \cdots & \Phi_n(x_0) \\ \Phi_0(x_1) & \Phi_1(x_1) & \cdots & \Phi_n(x_1) \\ \cdots & \cdots & \cdots & \cdots \\ \Phi_0(x_n) & \Phi_1(x_n) & \cdots & \Phi_n(x_n) \end{vmatrix}.$$

Тогда по заданным $y_i (i = 1,\cdots,n)$ однозначно определяются коэффициенты $c_k (k = 1,\cdots,n)$.

Изложение метода

Интерполяция кубическими сплайнами является частным случаем кусочно-полиномиальной интерполцией. В этом специальном случае между любыми двумя соседними узлами функция интерполируется

кубическим полиномом. Его коэффициенты на каждом интервале определяются из условий сопряжения в узлах:

$$f_i = y_i, f'(x_i - 0) = f'(x_i + 0), f''(x_i - 0) = f''(x_i + 0), i = 1,2,\cdots,n-1.$$

Кроме того, на границе при $x=x_0$ и $x=x_n$ ставятся условия

$$f''(x_0) = 0, f''(x_n) = 0. \qquad (2)$$

Будем искать кубический полином в виде

$$f(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, x_{i-1} \le \xi \le \xi_i. \qquad (3)$$

Из условия $f_i=y_i$ имеем

$$f(x_{i-1}) = a_i = y_{i-1}, f(x_i) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, h_i = x_i - x_{i-1}, i = 1,2,\cdots,n-1. \qquad (4)$$

Вычислим производные:

$$f'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2, f''(x) = 2c_i + 6d_i(x - x_{i-1}), x_{i-1} \le \xi \le \xi_i,$$

и потребуем их непрерывности при $x=x_i$:

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2, c_{i+1} = c_i + 3d_i h_i, i = 1,2,\cdots,n-1. \qquad (5)$$

Общее число неизвестных коэффициентов, очевидно, равно $4n$, число уравнений (4) и (5) равно $4n$-2. Недостающие два уравнения получаем из условия (2) при $x=x_0$ и $x=x_n$:

$$c_1 = 0, c_n + 3d_n h_n = 0.$$

Выражение из (5)

$$d_i = \frac{c_{i+1} - c_i}{3h_i},$$

подставляя это выражение в (4) и исключая $a_i=y_{i-1}$, получим

$$b_i=\left[\frac{y_i-y_{i-1}}{h}{}_i\right]-\frac{1}{3}h_i(c_{i+1}+2c_i),i=1,2,\cdots,n-1,b_n=\left[\frac{y_n-y_{n-1}}{h}{}_n\right]-\frac{2}{3}h_nc_n.$$

Подставив теперь выражения для b_i, b_{i+1} и d_i в первую формулу (5), после несложных преобразований получаем для определения c_i разностное уравнение второго порядка

$$h_ic_i+2(h_i+h_{i+1})c_{i+1}+h_{i+1}c_{i+2}=3\left(\frac{y_{i+1}-y_i}{h_{i+1}}-\frac{y_{i+1}-y_i}{h_{i+1}}\right),i=1,2,\cdots,n-1. \quad (6)$$

С краевыми условиями

$$c_1=0,c_{n+1}=0. \quad (7)$$

Условие $c_{n+1}=0$ эквивалентно условию $c_n+3d_nh_n=0$ и уравнению $c_{i+1}=c_i+d_ih_i$. Разностное уравнение (6) с условиями (7) можно решить методом прогонки, представив в виде системы линейных алгебраических уравнений вида $A*x=F$, где вектор x соответствует вектору $\{ c_i \}$, вектор F поэлементно равен правой части уравнения (6), а матрица A имеет следующий вид:

$$A=\begin{pmatrix} C_1 & B_1 & 0 & 0 & \cdots & 0 & 0 \\ A_2 & C_2 & B_2 & 0 & \cdots & 0 & 0 \\ 0 & A_3 & C_3 & B_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & B_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & A_n & C_n \end{pmatrix},$$

где

$$A_i=h_i,i=2,\cdots,n,B_i=h_{i+1},i=1,\cdots,n-1$$

и

$$C_i=2(h_i+h_{i+1}),i=1,\cdots,n.$$

Метод прогонки

Метод прогонки основан на предположении, что искомые неизвестные связаны рекуррентным соотношением:

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1} \quad i = 1, \cdots, n-1 \qquad (8)$$

Используя это соотношение, выразим x_{i-1} и x_i через x_{i+1} и подставим в i-е уравнение:

$$\left(A_i \alpha_i \alpha_{i+1} + C_i \alpha_{i+1} + B_i\right) x_{i+1} + A_i \alpha_i \beta_{i+1} + A_i \beta_i + C_i \beta_{i+1} - F_i = 0,$$

где F_i - правая часть i-го уравнения. Это соотношение будет выполняться независимо от решения, если потребовать

$$A_i \alpha_i \alpha_{i+1} + C_i \alpha_{i+1} + B_i = 0$$

$$A_i \alpha_i \beta_{i+1} + A_i \beta_i + C_i \beta_{i+1} - F_i = 0$$

Отсюда следует:

$$\alpha_{i+1} = \frac{-B_i}{A_i \alpha_i + C_i}$$

$$\beta_{i+1} = \frac{F_i - A_i \beta_i}{A_i \alpha_i + C_i}$$

Из первого уравнения получим:

$$\alpha_2 = \frac{-B_1}{C_1}$$

$$\beta_2 = \frac{F_1}{C_1}$$

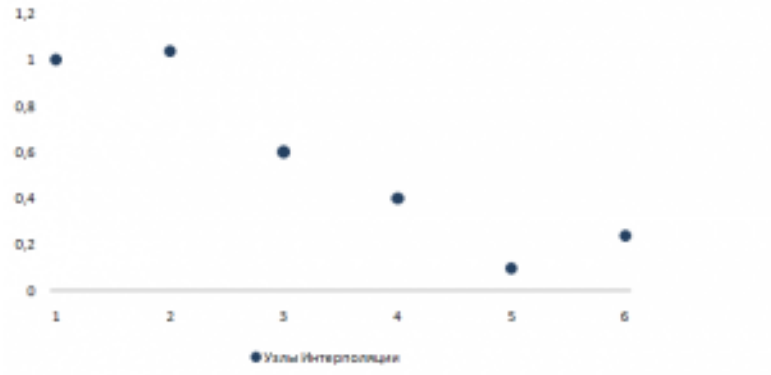
После нахождения прогоночных коэффициентов α и β, используя уравнение (1), получим решение системы. При этом,

$$x_n = \frac{F_n - A_n \beta_n}{C_n + A_n \alpha_n}$$

Пример: интерполирование неизвестной функции

Построим интерполянту для для функции *f*, заданной следующим образом:

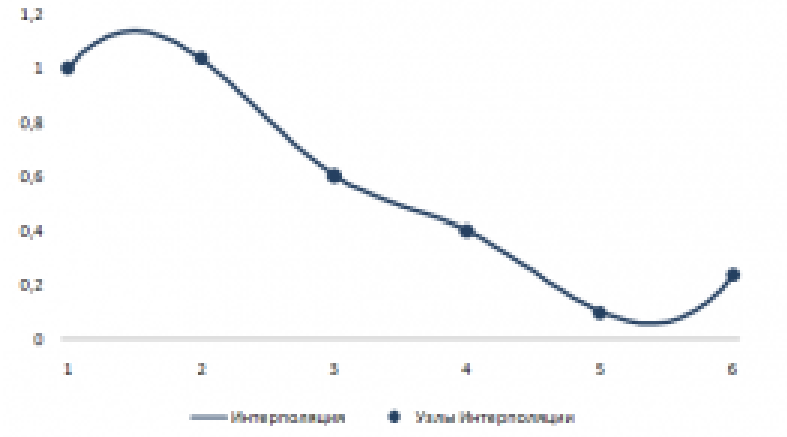
x	$f(x)$
1	1.0002
2	1.0341
3	0.6
4	0.40105
5	0.1
6	0.23975



Вводные значения для задачи интерполяции

В результате интерполяции были рассчитаны следующие коэффициенты интерполянты:

a	b	c	d	**Интервал**
1,0002	-0,140113846	0,440979231	-0,266965385	$1 \le x \le 2$
1,0341	-0,291901538	-0,359916923	0,217718462	$2 \le x \le 3$
0,6	-0,22553	0,293238462	-0,266658462	$3 \le x \le 4$
0,40105	-0,100328462	-0,506736923	0,306015385	$4 \le x \le 5$
0,1	-0,134456154	0,411309231	-0,137103077	$5 \le x \le 6$



Результат интерполяции

Ошибка интерполяции

Нас будет интересовать поведение максимального уклонения сплайна от интерполируемой функции в зависимости от максимального расстояния между соседними узлами интерполирования, т.е. зависимость величины

$$\|s-f\| = \max_{x \in [a,b]} |s(x)-f(x)|$$

от шага *h*, где

$$h = \max_{k=1,2,\cdots,n-1} |x_{k+1}-x_k|.$$

Известно, что если функция [s][*f*](*x*) имеет четыре непрерывные производные, то для ошибки интерполяции определенным выше кубическим сплайном *s*(*x*) верна следующая оценка

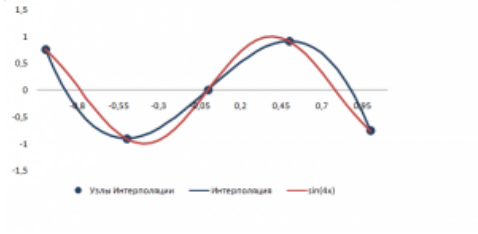
$$\|s-f\| \leq \frac{5}{384} h^4 \left\| \frac{d^4 f}{d f^4} \right\|$$

причем константа $\frac{5}{384}$ в этом неравенстве является наилучшей из возможных

Пример: интерполяция синуса

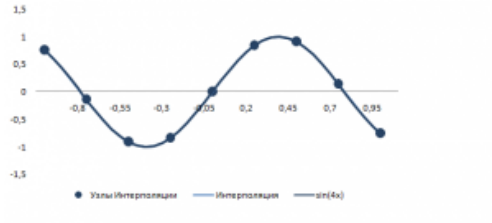
Постром интерполянту функции $f = \sin(4x)$ на отрезке [-1;1], взяв равномерно отстоящие узлы с шагом 0.5 и шагом 0.25, и сравним полученные результаты.

	Ошибка интерполяции	**Оценка ошибки**	**Иллюстрация**
$h = 0.5$	0.429685	3.(3)	



Результат интерполяции sin(4*x*) с шагом 0.5

$h = 0.25$	0.005167	0.208(3)	



Результат интерполяции sin(4x) с шагом 0.25

Как видно из полученных иллюстрации, уже при шаге 0.25 интерполянта визуально ничем не отличается от исходной функции.

5.4. Метод Ньютона

Следующий из рассматриваемых методов однопараметрической оптимизации является градиентным методом второго порядка. В нем при поиске экстремума целевой функции используется ее первые и вторые производные. Этот метод носит название метода Ньютона.

Метод применим для вогнутой (или выпуклой), функции *F(x)*, что соответствует монотонности ее первой производной *f(x)*.

Известно, что если функция *F(x)* имеет локальный минимум (или максимум) в точке $\overline{x}$, то в этой точке градиент функции *F(x)* (вектор ее производных) равен нулю, т.е.

$$F'(x) \equiv f(x) = 0$$

Следовательно, если функция *F(x)* дифференцируема, то для нахождения ее экстремума нужно решить уравнение

$$f(x)=0, \qquad (1)$$

где *f(x)=F'(x)*. $\overline{x}$ - корень уравнения (1), точка, то есть, координата в которой *F'(x)*=0, а функция *F(x)* имеет минимум (или максимум) (рис.1).

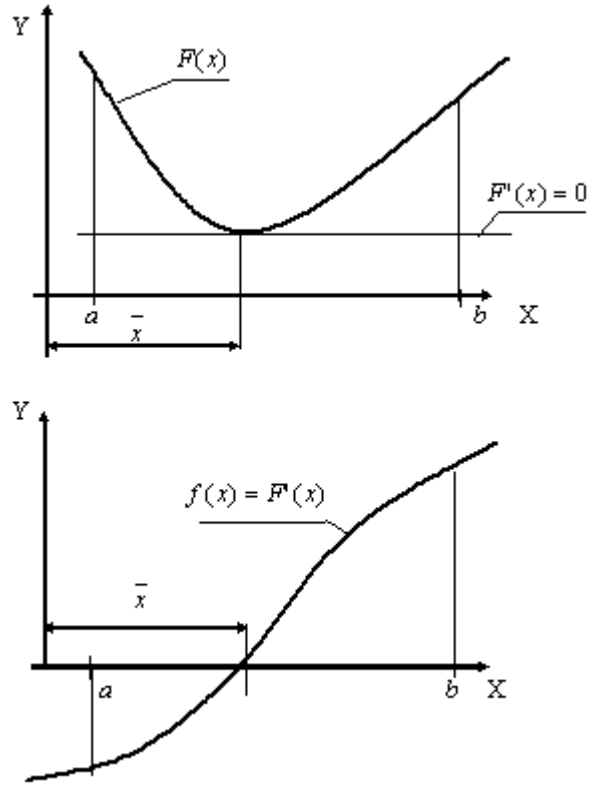


Рис. 1. Вогнутая функция *F(x)* и ее производная *f(x)*.

Алгоритм метода Ньютона сводится к линейному представлению функции *f(x)* и решению уравнения (1).

Разложим функцию *f(x)* в ряд Тейлора:

$$f(x_{i+1}) = f(x_i) + h_i \cdot f'(x_i) + \frac{h_i^2}{2!} \cdot f''(x_i) + \frac{h_i^3}{3!} \cdot f'''(x_i) + \ldots,$$

где $h_i = x_{i+1} - x_i$.

Отбросим члены ряда, содержащие $h_i^2, h_i^3, \ldots$.

В результате имеем:

$$f(x_{i+1}) = f(x_i) + (x_{i+1} - x_i) f'(x_i).$$

Если в точке (x_{i+1}) достигается экстремум функции $F(x)$, то $f(x_{i+1})=0$.

Тогда

$$f(x_i) + (x_{i+1} - x_i) f'(x_i) = 0.$$

Отсюда точка экстремума равна:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{F'(x_i)}{F''(x_i)}. \qquad (2)$$

Для нахождения экстремума функции $F(x)$ необходимо на каждом шаге итерационного процесса поиска определить первую $F1$ и вторую $F2$ производные целевой функции $F(x)$, т.е.

$$F1 = f(x) = F'(x),$$
$$F2 = f'(x) = F''(x).$$

Начальные приближения x рекомендуется выбирать в той точке интервала $[a,b]$, где знаки функции $f(x)$ и ее кривизны $f''(x)$ совпадают, т.е. выполняется условие

$$f(x) \cdot f''(x) > 0, \qquad (3)$$

где

$$f''(x) = F'''(x) = F3$$

Алгоритм метода Ньютона:

1. Выбираем начальную точку x. Если $F'(a) \cdot F'''(a) > 0$, то $x=a$, иначе $x=b$.
2. Находим приближение корня (x_{i+1}) по выражению (2).

3. Итерационный процесс поиска продолжается до тех пор, пока

$$|x_{i+1} - x_i| < \varepsilon. \qquad (4)$$

На основании (2) условие (4) можно записать как

$$\left| x_i = \frac{f(x_i)}{f'(x_i)} - x_i \right| < \varepsilon$$

В результате условие (4) будет иметь вид

$$\left| \frac{f(x_i)}{f'(x_i)} - x_i \right| < \varepsilon$$

В точке экстремума $\overline{x}$ производная $F'(x)$ меняет знак.

Если в точке $\overline{x}$ функция $F(x)$ имеет минимум, то производная $F'(x)$ в окрестности $\overline{x}$ меняет знак с отрицательного на положительный, т.е. $F'(x)$ является возрастающей функцией, значит, $F''(x) > 0$ (рис. 2, a).

Если в точке $\overline{x}$ функция $F(x)$ имеет максимум, то производная $F'(x)$ в окрестности $\overline{x}$ меняет знак с положительного на отрицательный, т.е. $F'(x)$ является убывающей функцией, значит, $F''(x) < 0$ (рис. 2, b).

Следовательно, по знаку $F''(x)$ можно судить: в точке $\overline{x}$ максимум или минимум функции $F(x)$.

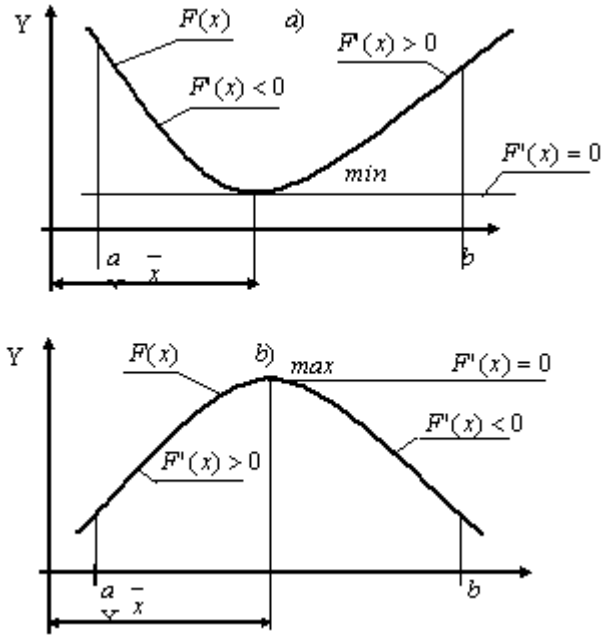


Рис. 2.

Если функция $F(x)$ не дифференцируема или вычисление ее производных очень сложно, то для определения производных функции $F(x)$ можно воспользоваться приблизительными оценками производных с помощью разностных схем:

$$F'(x) = \frac{\Delta F}{h};\ F''(x) = \frac{\Delta F'}{h};\ \text{ и т.д.}$$

Схема алгоритма метода Ньютона представлена на рис. 3.

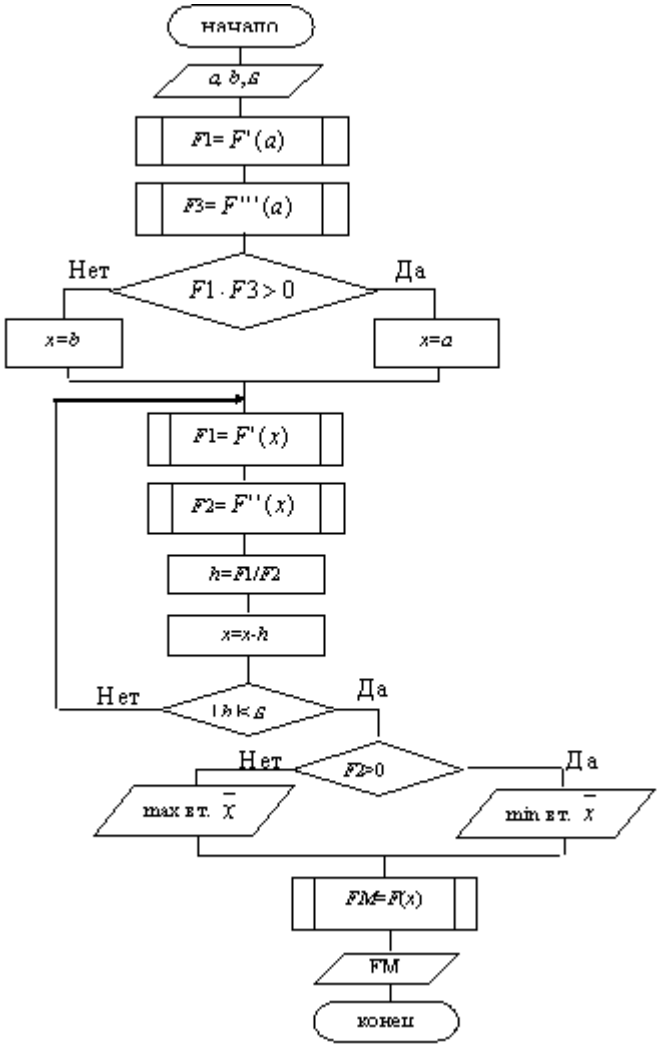


Рис. 3. Схема алгоритма метода Ньютона

На рис.3: $\bar{x}$ - координата точки в которой функция $F(x)$ имеет минимальное (или максимальное) значение, FM - значение, функции $F(x)$ в точке $\bar{x}$.

5.5. Метод касательных (Ньютона)

Метод касательных (Ньютона) — это итерационный численный метод нахождения корня (нуля) заданной функции. Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации. Метод обладает квадратичной сходимостью. Улучшением метода является метод хорд и касательных. Также метод Ньютона может быть использован для решения задач оптимизации, в которых требуется определить нуль первой производной либо градиента в случае многомерного пространства.

Обоснование

Чтобы численно решить уравнение $f(x)=0$ методом простой итерации, его необходимо привести к следующей форме: $x=\varphi(x)$, где φ — сжимающее отображение.

Для наилучшей сходимости метода в точке очередного приближения x^* должно выполняться условие $\varphi'(x^*)=0$. Решение данного уравнения ищут в виде

$$\varphi(x) = x + \alpha(x) f(x),$$

тогда:

$$\varphi'(x^*) = 1 + \alpha'(x^*) f(x^*) + \alpha(x^*) f'(x^*) = 0.$$

В предположении, что точка приближения «достаточно близка» к корню $\tilde{x}$, и что заданная функция непрерывна $(f(x^*) \approx f(\tilde{x}) = 0)$, окончательная формула для $\alpha(x)$такова:

$$\alpha(x) = -\frac{1}{f'(x)}.$$

С учётом этого функция $\varphi(x)$ определяется выражением:

$$\varphi(x) = x - \frac{f(x)}{f'(x)}.$$

Эта функция в окрестности корня осуществляет сжимающее отображение, и алгоритм нахождения численного решения уравнения $f(x)=0$ сводится к итерационной процедуре вычисления:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Приведем доказательство, что функция $\varphi(x)$ в окрестности корня $U_\delta(\bar{x})$ осуществляет сжимающее отображение.

Доказательство: Пусть дана функция вещественного переменного дважды непрерывно дифференцируемая в своей области определения, производная которой нигде не обращается в нуль:

$$f(x)\colon \mathbb{X}\to\mathbb{R},\ f(x)\in C^2(\mathbb{X});\quad \forall x\in\mathbb{X}\ f'(x)\neq 0.$$

И необходимо доказать, что функция $\varphi(x)=x-\frac{f(x)}{f'(x)}$ осуществляет сжимающее отображение вблизи корня уравнения $f(x)=0$. В силу непрерывной дифференцируемости функции $f(x)$ и неравенства нулю её первой производной $\varphi(x)$ непрерывна. Производная $\varphi'(x)$ равна:

$$\varphi'(x)=\frac{f(x)f''(x)}{(f'(x))^2}.$$

В условиях, наложенных на $f(x)$, она также непрерывна. Пусть $\bar{x}$ — искомый корень уравнения: $f(\bar{x})=0$, следовательно в его окрестности $\varphi'(x)\approx 0$:

$$\forall\varepsilon\colon 0<\varepsilon<1,\ \exists\delta>0\ \forall x\in\mathbb{X}\ |x-\bar{x}|<\delta\colon\ |\varphi'(x)-0|<\varepsilon.$$

Тогда согласно теореме Лагранжа:

$$\forall x_1,\ x_2\in U_\delta(\bar{x})\ \exists\xi\in U_\delta(\bar{x})\colon\ |\varphi(x_1)-\varphi(x_2)|=|\varphi'(\xi)||x_1-x_2|<\varepsilon|x_1-x_2|.$$

В силу того, что $\varphi(\bar{x})=\bar{x}$ в этой же дельта окрестности выполняется:

$$\forall x\in U_\delta(\bar{x})\colon\ |\varphi(x)-\bar{x}|<\varepsilon|x-\bar{x}|.$$

Таким образом полученная функция $\varphi(x)$ в окрестности корня $U_\delta(\bar{x})$ осуществляет сжимающее отображение.

По теореме Банаха последовательность приближений стремится к корню уравнения $f(x)=0$.

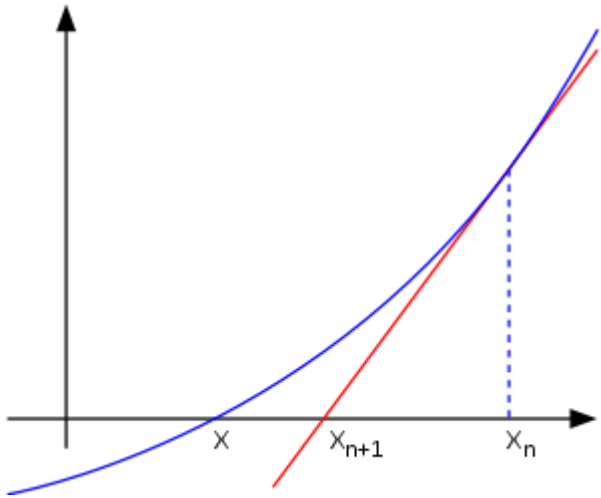


Рис. 1. Иллюстрация метода Ньютона

На рис. 1. представлена иллюстрация метода Ньютона (кривая изображает график функции $f(x)$, нуль которой необходимо найти, прямая — касательную в точке очередного приближения x_n). Здесь мы можем увидеть, что последующее приближение x_{n+1} лучше предыдущего x_n.

Геометрическая интерпретация

Основная идея метода заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка и берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность.

Пусть $f(x): [a,\ b] \to \mathbb{R}$ — определённая на отрезке $[a, b]$ и дифференцируемая на нём вещественнозначная функция. Тогда формула итеративного исчисления приближений может быть выведена следующим образом:

$$f'(x_n) = \operatorname{tg}\alpha = \frac{\Delta y}{\Delta x} = \frac{f(x_n) - 0}{x_n - x_{n+1}} = \frac{0 - f(x_n)}{x_{n+1} - x_n},$$

где α — угол наклона касательной в точке x_n.

Следовательно искомое выражение для x_{n+1} имеет вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Итерационный процесс начинается с некоего начального приближения x_0 (чем ближе к нулю, тем лучше, но если предположения о нахождении решения отсутствуют, методом проб и ошибок можно сузить область возможных значений, применив теорему о промежуточных значениях).

Алгоритм

1. Задается начальное приближение x_0.
2. Пока не выполнено условие остановки, в качестве которого можно взять $|x_{n+1} - x_n| < \varepsilon$ или $|f(x_{n+1})| < \varepsilon$ (то есть погрешность в нужных пределах), вычисляют новое приближение:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Пример

Иллюстрация применения метода Ньютона к функции $f(x) = \cos x - x^3$ с начальным приближением в точке $x_0 = 0{,}5$.

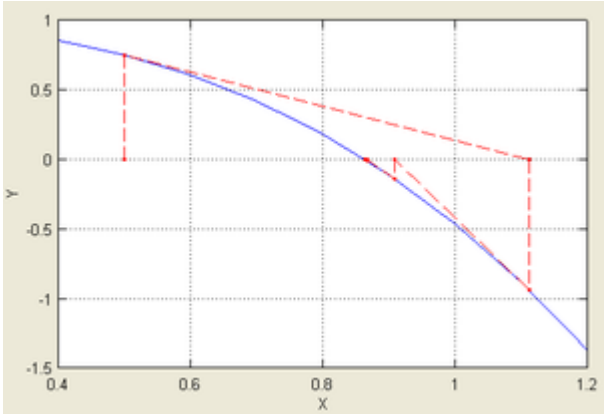


Рис. 2. График последовательных приближений.

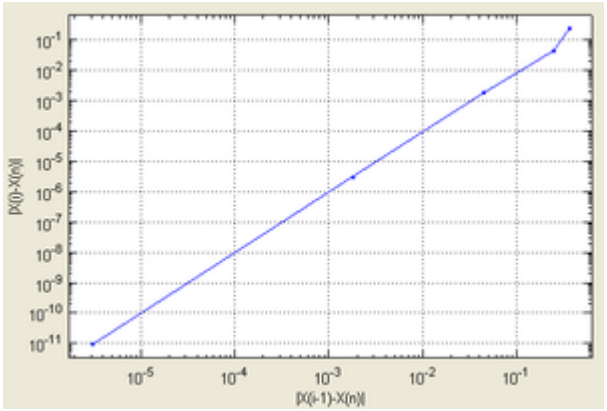


Рис. 3. График сходимости.

Согласно способу практического определения скорость сходимости может быть оценена как тангенс угла наклона графика сходимости, то есть в данном случае равна двум.

Рассмотрим задачу о нахождении положительных x, для которых $\cos x = x^3$. Эта задача может быть представлена как задача нахождения нуля функции $f(x) = \cos x - x^3$. Имеем выражение для производной $f'(x) = -\sin x - 3x^2$. Так как $\cos x \leq 1$ для всех x и $x^3 > 1$ для $x > 1$,

очевидно, что решение лежит между 0 и 1. Возьмём в качестве начального приближения значение $x_0 = 0{,}5$, тогда:

$$\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} &&= 1{,}112\,141\,637\,097,\\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} &&= \underline{0{,}909}\,672\,693\,736,\\
x_3 &= x_2 - \frac{f(x_2)}{f'(x_2)} &&= \underline{0{,}867}\,263\,818\,209,\\
x_4 &= x_3 - \frac{f(x_3)}{f'(x_3)} &&= \underline{0{,}865\,477}\,135\,298,\\
x_5 &= x_4 - \frac{f(x_4)}{f'(x_4)} &&= \underline{0{,}865\,474\,033\,111},\\
x_6 &= x_5 - \frac{f(x_5)}{f'(x_5)} &&= \underline{0{,}865\,474\,033\,102}.
\end{aligned}$$

Подчёркиванием отмечены верные значащие цифры. Видно, что их количество от шага к шагу растёт (приблизительно удваиваясь с каждым шагом): от 1 к 2, от 2 к 5, от 5 к 10, иллюстрируя квадратичную скорость сходимости.

Условия применения

Рассмотрим ряд примеров, указывающих на недостатки метода.

Контрпримеры

- Если начальное приближение недостаточно близко к решению, то метод может не сойтись.

Пусть

$$f(x) = x^3 - 2x + 2.$$

Тогда

$$x_{n+1} = x_n - \frac{x_n^3 - 2x_n + 2}{3x_n^2 - 2}.$$

Возьмём нуль в качестве начального приближения. Первая итерация даст в качестве приближения единицу. В свою очередь, вторая снова даст нуль. Метод зациклится и решение не будет найдено. В общем случае построение последовательности приближений может быть очень запутанным.

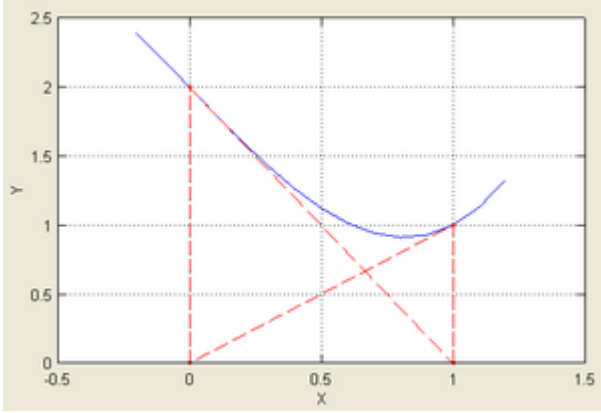


Рис.4. Иллюстрация расхождения метода Ньютона, применённого к функции $f(x)=x^3-2x+2$ с начальным приближением в точке $x_0=0$.

- Если производная не непрерывна в точке корня, то метод может расходиться в любой окрестности корня.

Рассмотрим функцию:

$$f(x) = \begin{cases} x, & x = 0, \\ x + x^2 \sin\left(\frac{2}{x}\right), & x \neq 0. \end{cases}$$

Тогда

$$f'(0) = 1 \text{ и } f'(x) = 1 + 2x\sin(2/x) - 2\cos(2/x)$$

всюду, кроме 0.

В окрестности корня производная меняет знак при приближении *x* к нулю справа или слева. В то время, как $f(x) \geqslant x - x^2 > 0$ для $0<x<1$.

Таким образом *f(x) / f'(x)* не ограничено вблизи корня, и метод будет расходиться, хотя функция всюду дифференцируема, её производная не равна нулю в корне, *f* бесконечно дифференцируема везде, кроме как в корне, а её производная ограничена в окрестности корня.

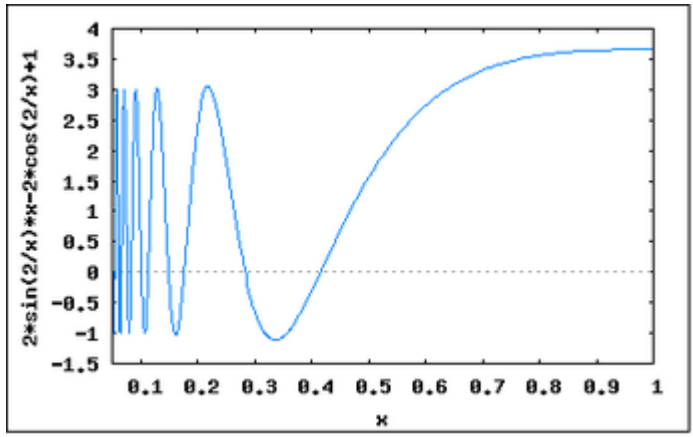


Рис.5. График производной функции $f(x)=x+x^2 \sin(2/x)$ при приближении *x* к нулю справа.

- Если не существует вторая производная в точке корня, то скорость сходимости метода может быть заметно снижена.

Рассмотрим пример:

$$f(x) = x + x^{4/3}.$$

Тогда $f'(x) = 1 + (4/3)x^{1/3}$ и $f''(x) = (4/9)x^{-2/3}$ за исключением x=0, где она не определена.

На очередном шаге имеем x_n:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = \frac{(1/3)x_n^{4/3}}{(1 + (4/3)x_n^{1/3})}.$$

Скорость сходимости полученной последовательности составляет приблизительно 4/3. Это существенно меньше, нежели 2, необходимое для квадратичной сходимости, поэтому в данном случае можно говорить лишь о линейной сходимости, хотя функция всюду непрерывно дифференцируема, производная в корне не равна нулю, и f бесконечно дифференцируема везде, кроме как в корне.

- Если производная в точке корня равна нулю, то скорость сходимости не будет квадратичной, а сам метод может преждевременно прекратить поиск, и дать неверное для заданной точности приближение.

Пусть

$$f(x) = x^2.$$

Тогда $f'(x) = 2x$ и следовательно $x - f(x)/f'(x) = x/2$. Таким образом сходимость метода не квадратичная, а линейная, хотя функция всюду бесконечно дифференцируема.

Ограничения

Пусть задано уравнение $f(x) = 0$, где $f(x)\colon \mathbb{X} \to \mathbb{R}$ и надо найти его решение.

Ниже приведена формулировка основной теоремы, которая позволяет дать чёткие условия применимости. Она носит имя Канторовича.

Теорема Канторовича.

Если существуют такие константы A,B,C, , что:

1. $\frac{1}{|f'(x)|} < A$ *на* [*a, b*]*, то есть f'(x)существует и не равна нулю;*
2. $\left|\frac{f(x)}{f'(x)}\right| < B$ *на* [*a, b*]*, то есть f(x) ограничена;*
3. $\exists f''(x)_{\text{на } [a,\ b],\ \text{и}} |f''(x)| \leqslant C \leqslant \frac{1}{2AB}$*;*

Причём длина рассматриваемого отрезка $|a-b| < \frac{1}{AB}\left(1-\sqrt{1-2ABC}\right)$*. Тогда справедливы следующие утверждения:*

1. *на* [*a, b*] *существует корень x* уравнения* $f(x)=0: \exists x^* \in [a,\ b]: f(x^*)=0$*;*
2. *если* $x_0 = \frac{a+b}{2}$*, то итерационная последовательность сходится к этому корню:* $\left\{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\right\} \to x^*$*;*
3. *погрешность может быть оценена по формуле* $|x^* - x_n| \leqslant \frac{B}{2^{n-1}}(2ABC)^{2^{n-1}}$.

Из последнего из утверждений теоремы в частности следует квадратичная сходимость метода:

$$|x^* - x_n| \leqslant \frac{B}{2^{n-1}}(2ABC)^{2^{n-1}} = \frac{1}{2}\frac{B}{2^{n-2}}\left((2ABC)^{2^{n-2}}\right)^2 = \alpha|x^* - x_{n-1}|^2.$$

Тогда ограничения на исходную функцию *f(x)* будут выглядеть так:

1. функция должна быть ограничена;
2. функция должна быть гладкой, дважды дифференцируемой;
3. её первая производная *f'(x)* равномерно отделена от нуля;
4. её вторая производная *f''(x)* должна быть равномерно ограничена.

Обобщения и модификации

Метод одной касательной

В целях уменьшения числа обращений к значениям производной функции применяют так называемый метод одной касательной.

Формула итераций этого метода имеет вид:

$$x_{n+1} = x_n - \frac{1}{f'(x_0)}f(x_n).$$

Суть метода заключается в том, чтобы вычислять производную лишь один раз, в точке начального приближения x_0, а затем использовать это значение на каждой последующей итерации:

$$\alpha(x) = \alpha_0 = -\frac{1}{f'(x_0)}.$$

При таком выборе α_0 в точке x_0 выполнено равенство:

$$\varphi'(x_0) = 1 + \alpha_0 f'(x_0) = 0,$$

и если отрезок, на котором предполагается наличие корня x^* и выбрано начальное приближение x_0, достаточно мал, а производная $\varphi'(x)$ непрерывна, то значение $\varphi'(x^*)$ будет не сильно отличаться от $\varphi'(x_0) = 0$ и, следовательно, график $y = \varphi(x)$ пройдёт почти горизонтально, пересекая прямую $y=x$, что в свою очередь обеспечит быструю сходимость последовательности точек приближений к корню.

Этот метод можно также рассматривать, как модернизацию метода хорд (секущих), где число γ следует выбрать равным

$$\max_x |\varphi'(x)|.$$

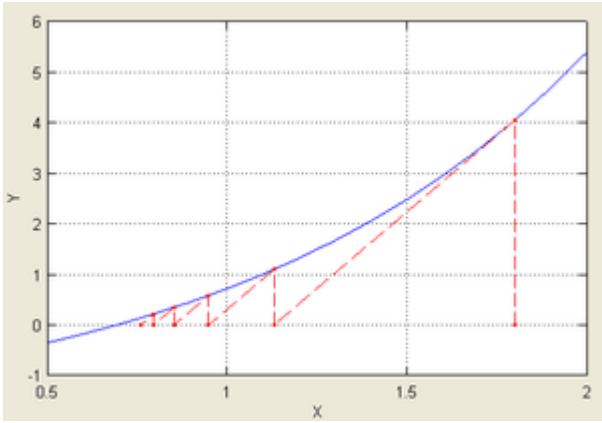


Рис.6. Иллюстрация последовательных приближений метода одной касательной, применённого к функции $f(x)=e^x-2$ с начальным приближением в точке x_0=1,8..

Многомерный случай

Обобщим полученный результат на многомерный случай.

Пускай необходимо найти решение системы:

$$\begin{cases} f_1(x_1,\ x_2,\ \ldots,\ x_n) &= 0, \\ \ldots \\ f_n(x_1,\ x_2,\ \ldots,\ x_n) &= 0. \end{cases}$$

Выбирая некоторое начальное значение $\vec{x}^{[0]}$, последовательные приближения $\vec{x}^{[j+1]}$ находят путём решения систем уравнений:

$$f_i + \sum_{k=1}^{n} \frac{\partial f_i}{\partial x_k}(x_k^{[j+1]} - x_k^{[j]}) = 0, \qquad i = 1,\ 2,\ \ldots,\ n,$$

где

$$\vec{x}^{[j]} = (x_1^{[j]},\ x_2^{[j]},\ \ldots,\ x_n^{[j]}), \quad j = 0,\ 1,\ 2,\ \ldots$$

Применительно к задачам оптимизации

Пусть необходимо найти минимум функции многих переменных $f(\vec{x}) : \mathbb{R}^n \to \mathbb{R}$. Эта задача равносильна задаче нахождения нуля градиента $\nabla f(\vec{x})$. Применим изложенный выше метод Ньютона:

$$\nabla f(\vec{x}^{[j]}) + H(\vec{x}^{[j]})(\vec{x}^{[j+1]} - \vec{x}^{[j]}) = 0, \quad j = 1,\ 2,\ \ldots,\ n,$$

где $H(\vec{x})$ — гессиан функции $f(\vec{x})$.

В более удобном итеративном виде это выражение выглядит так:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - H^{-1}(\vec{x}^{[j]})\nabla f(\vec{x}^{[j]}).$$

Следует отметить, что в случае квадратичной функции метод Ньютона находит экстремум за одну итерацию.

Нахождение матрицы Гессе связано с большими вычислительными затратами, и зачастую не представляется возможным. В таких случаях

альтернативой могут служить квазиньютоновские методы, в которых приближение матрицы Гессе строится в процессе накопления информации о кривизне функции.

Метод Ньютона — Рафсона

Метод Ньютона — Рафсона является улучшением метода Ньютона нахождения экстремума, описанного выше. Основное отличие заключается в том, что на очередной итерации каким-либо из методов одномерной оптимизации выбирается оптимальный шаг:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda_j H^{-1}(\vec{x}^{[j]})\nabla f(\vec{x}^{[j]}),$$

где $$\lambda_j = \arg\min_{\lambda} f(\vec{x}^{[j]} - \lambda H^{-1}(\vec{x}^{[j]})\nabla f(\vec{x}^{[j]})).$$

Для оптимизации вычислений применяют следующее улучшение: вместо того, чтобы на каждой итерации заново вычислять гессиан целевой функции, ограничиваются начальным приближением $H(f(\vec{x}^{[0]}))$ и обновляют его лишь раз в m шагов, либо не обновляют вовсе.

Применительно к задачам о наименьших квадратах

На практике часто встречаются задачи, в которых требуется произвести настройку свободных параметров объекта или подогнать математическую модель под реальные данные. В этих случаях появляются задачи о наименьших квадратах:

$$F(\vec{x}) = \|\vec{f}(\vec{x})\| = \sum_{i=1}^{m} f_i^2(\vec{x}) = \sum_{i=1}^{m} (\varphi_i(\vec{x}) - \mathcal{F}_i)^2 \to \min .$$

Эти задачи отличаются особым видом градиента и матрицы Гессе:

$$\nabla F(\vec{x}) = J^T(\vec{x}) f(\vec{x}),$$

$$H(\vec{x}) = J^T(\vec{x}) J(\vec{x}) + Q(\vec{x}), \qquad Q(\vec{x}) = \sum_{i=1}^{m} f_i(\vec{x}) H_i(\vec{x}),$$

где $J(\vec{x})$ — матрица Якоби вектор-функции $\vec{f}(\vec{x})$, $H_i(\vec{x})$ — матрица Гессе для её компоненты $f_i(\vec{x})$.

Тогда очередное направление $\vec{p}$ определяется из системы:

$$\left[J^T(\vec{x})J(\vec{x}) + \sum_{i=1}^{m} f_i(\vec{x})H_i(\vec{x})\right]\vec{p} = -J^T(\vec{x})f(\vec{x}).$$

Метод Гаусса — Ньютона

Метод Гаусса — Ньютона строится на предположении о том, что слагаемое $J^T(\vec{x})J(\vec{x})$ доминирует над $Q(\vec{x})$. Это требование не соблюдается, если минимальные невязки велики, то есть если норма $\|\vec{f}(\vec{x})\|$ сравнима с максимальным собственным значением матрицы $J^T(\vec{x})J(\vec{x})$. В противном случае можно записать:

$$J^T(\vec{x})J(\vec{x})\vec{p} = -J^T(\vec{x})f(\vec{x}).$$

Таким образом, когда норма $\|Q(\vec{x})\|$ близка к нулю, а матрица $J(\vec{x})$ имеет полный столбцевой ранг, направление $\vec{p}$ мало отличается от ньютоновского (с учётом $Q(\vec{x})$), и метод может достигать квадратичной скорости сходимости, хотя вторые производные и не учитываются. Улучшением метода является алгоритм Левенберга — Марквардта, основанный на эвристических соображениях.

Обобщение на комплексную плоскость

До сих пор в описании метода использовались функции, осуществляющие отображения в пределах множества вещественных

значений. Однако метод может быть применён и для нахождения нуля функции комплексного переменного. При этом процедура остаётся неизменной:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}.$$

Особый интерес представляет выбор начального приближения z_0. Ввиду того, что функция может иметь несколько нулей, в различных случаях метод может сходиться к различным значениям, и вполне естественно возникает желание выяснить, какие области обеспечат сходимость к тому или иному корню. Этот вопрос заинтересовал Артура Кейли ещё в 1879 году, однако разрешить его смогли лишь в 70-х годах двадцатого столетия с появлением вычислительной техники. Оказалось, что на пересечениях этих областей (их принято называть *областями притяжения*) образуются так называемые *фракталы* — бесконечные самоподобные геометрические фигуры. Ввиду того, что Ньютон применял свой метод исключительно к полиномам, фракталы, образованные в результате такого применения, обрели название *фракталов Ньютона* или *бассейнов Ньютона*.

5.6. Метод Коши

Пусть в точке $\overline{x}$ требуется определить направление наискорейшего спуска (то есть направление наибольшего локального уменьшения *f(x)*). Разложим *f(x)* в ряд Тейлора в окрестности точки $\overline{x}$ и отбросим члены второго порядка по Δx и выше.

$$\tilde{f}(x, \overline{x}) = f(\overline{x}) + \nabla f(\overline{x})^T \cdot \Delta x + \dots$$

Локальное уменьшение *f(x)* определяется вторым слагаемым, то есть наибольшее уменьшение *f(x)* будет тогда, когда $\nabla f(\overline{x})^T \cdot \Delta x$ будет иметь наибольшую отрицательную величину. Этого можно добиться

выбором $S^{(k)}$: $S^{(k)} = -\nabla f(\bar{x})$, тогда второе слагаемое примет вид: $-\alpha \cdot \nabla f(\bar{x})^T \cdot \nabla f(\bar{x})$.

Этот случай соответствует наискорейшему локальному спуску

$$x^{(k+1)} = x^{(k)} - \alpha \cdot \nabla f(x^{(k)}).$$

Недостатки:

· остаётся вопрос выбора α;

· вблизи точки минимума медленно сходится, так как $\nabla \to 0$.

α будем находить путём минимизации функции $f(x^{(k+1)})$ в направлении $-\nabla$.

Метод обладает большой надёжностью но медленую сходимость вблизи точки минимума устранить нельзя. Поэтому метод самостоятельно обычно не используется, а используется как предварительная процедура для более сложных методов.

Достоинсиво:

на каждой итерации $f(x^{(k+1)}) \le f(x^{(k)})$ - выполняется свойство убывания функции на каждой итерации.

Алгоритм метода:

1 Задать $x_0, \varepsilon_1, \varepsilon_2, N, M$ - начальное приближение, параметр окончания работы алгоритма Коши, параметр окончания работы одномерного алгоритма, количество переменных и максимальное количество итераций соответственно.

2 Вычислить $\nabla f(x^{(k)})$

3 Если $|\nabla f(x^{(k)})| \le \varepsilon_1$, то $x_k=x^*$ иначе, если $K \ge M$, то $x_k=x^*$. Перейти к п. 4.

4 Решить задачу минимизации функции $f(x^{(k+1)})$ и найти $\alpha^{(k)}$ используя ε_2

5 Вычислить следующее приближение по формуле
$$x^{(k+1)} = x^{(k)} - \alpha \cdot \nabla f(x^{(k)})$$

6 Если $|\Delta x| \le \varepsilon_1$, то $x_k=x^*$ иначе $k=k+1$ и перейти к п. 2.

5.7. Метод Марквардта

Это комбинация методов Ньютона и Коши. Вдали от точки минимума направление определяется по методу Коши, а в окрестности точки минимума – по методу Ньютона.

$$S^{(k)} = -\left[H^{(k)} + \lambda^{(k)} \cdot I\right]^{-1} \cdot \nabla f(x^{(k)}),$$

где: $H^{(k)}$ – матрица Гессе (вторых производных;

I – единичная матрица;

$\lambda^{(k)}$ – параметр, определяющий направление поиска и длину шага.

При этом в формуле

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot S^{(k)} \quad \alpha^{(k)} = 1.$$

На начальном этапе $\lambda^{(k)} \approx 10^4$, при этом второй член в $S^{(k)} = -\left[H^{(k)} + \lambda^{(k)} \cdot I\right]^{-1} \cdot \nabla f(x^{(k)})$ много больше первого, поэтому поиск осуществляется по методу Коши. По мере приближения к точке оптимума $\lambda^{(k)}$ уменьшается и стремится к нулю. Таким образом

вблизи точки оптимума первый член много больше второго и поиск осуществляется по методу Ньютона.

Если после первого шага $f(x^{(1)}) < f(x^{(0)})$, то следует выбрать $\lambda^{(1)} < \lambda^{(0)}$ и реализовать следующий шаг, в противном случае $\lambda^{(0)} = \beta \cdot \lambda^{(0)}$, где $\beta > 1$ и повторить предыдущий шаг.

Алгоритм.

1. Задать x_0 – начальное приближение, M – максимальное количество итераций, N – количество переменных и ε - параметр сходимости.

2. При $k=0$ $\lambda^{(k)} = 10^4$

3. Вычислить компоненты вектора $\nabla f(x^{(k)})$.

4. Если $|\nabla f(x^{(k)})| \le \varepsilon$, то $x_k = x^*$ иначе, если $K \ge M$, то $x_k = x^*$. Перейти к п. 5.

5. Вычислить $S^{(k)}$.

6. Вычислить $x^{(k+1)} = x^{(k)} + S^{(k)}$

7. Если $f(x^{(k+1)}) > f(x^{(k)})$, то перейти к п. 9, иначе перейти к п. 8.

8. Положить $\lambda^{(k+1)} = \frac{1}{2} \cdot \lambda^{(k)}$, $k=k+1$, перейти к п. 3.

9. Положить $\lambda^{(k)} = 2 \cdot \lambda^{(k)}$, перейти к п. 5.

Достоинства метода:

- простота;
- убывание целевой функции;

· быстрая сходимость как вдали от точки оптимума, так и вблизи неё:

· отсутствие поиска вдоль прямой.

Недостаток:

· необходимость вычисления матрицы Гессе на каждой итерации.

Вычислительные эксперименты показали, что метод наиболее эффективен для функций вида суммы квадратов: $f(\bar{x}) = f_1^2(\bar{x}) + f_2^2(\bar{x})$

5.8. Связь методов Ньютона и сопряженных градиентов

Цель раздела - знакомство с методами безусловной оптимизации второго порядка и близкого к ним по эффективности метода сопряжённых градиентов, освоение и сравнение эффективности их применения для конкретных целевых функций.

1. Краткие теоретические сведения

1.1 Методы Ньютона

1.1.1. Общая характеристика

Напомним, что методы Ньютона относятся к методам второго порядка, использующим вторые частные производные целевой функции *f(x)*. Все они являются прямым обобщением известного метода Ньютона отыскания корня уравнения:

$$\varphi(x) = 0, \qquad (1)$$

где $\varphi(x)$ – скалярная функция скалярного аргумента *x*.

Метод Ньютона отыскания корня уравнения описывается следующей рекуррентной формулой:

$$x_{k+1} = x_k - \varphi(x_k) / \varphi'(x_k). \qquad (2)$$

Пусть $\varphi(x)$ – *n*-мерная вектор-функция векторного аргумента *x* той же размерности. Тогда для решения системы уравнений $\varphi(x) = 0$ мы можем использовать итерационный процесс, аналогичный (2):

$$x_{k+1} = x_k - \varphi(x_k)^{-1} \cdot \varphi'(x_k), \qquad (3)$$

где $\varphi'(x_k) = \dfrac{\partial \varphi_i}{\partial x^j}$, (x_k) – квадратная матрица $n \times n$.

Рассмотрим теперь случай, когда вектор-функция $\varphi(x)$ является градиентом некоторой скалярной функции $f(x)$, т.е.

$$\varphi(x) = f'(x).$$

Приравнивая её нулю, приходим к системе уравнений, определяющей координаты стационарных точек функции $f(x)$. Формула метода Ньютона для решения этой системы выглядит так:

$$x_{k+1} = x_k - (f''(x))^{-1} \cdot f'(x), \qquad (4)$$

и получается заменой в (3) $\varphi(x_k)$ на $f'(x)$.

Итерационный процесс (4) строит последовательность точек $\{x_k\}$, которая при определённых предположениях сходится к некоторой стационарной точке x_* функции $f(x)$, т.е. к точке, в которой $f'(x_*) = 0$. Если матрица вторых производных $f''(x_*)$ положительно определена, эта точка будет точкой локального минимума функции $f(x)$.

1.1.2 Метод Ньютона

В методе Ньютона последовательность точек спуска определяется формулой (4). Для текущей точки x_k направление и величина спуска определяется вектором $p_k = -(f''(x_k))^{-1} \cdot f'(x_k)$. Хотя в определении вектора p_k фигурирует обратная к $f''(x_k)$ матрица $(f''(x_k))^{-1}$, на практике нет необходимости вычислять последнюю, так как направление спуска p_k можно найти как решение системы линейных уравнений

$$f''(x_k) \cdot p_k = -f'(x_k) \qquad (5)$$

каким-нибудь из известных методов.

Алгоритм

Шаг 1. На первой итерации, при $k = 0$, вводятся начальное приближение x_0 и условие останова ε_3. Вычисляются градиент $f'(x_0)$ и матрица $f''(x_0)$.

Шаг 2. Определяется направление спуска p_k, как решение системы линейных уравнений $f''(x_k) \cdot p_k = -f'(x_k)$ (например, методом исключений Гаусса).

Шаг 3. Определяется следующая точка спуска: $x_{k+1}=x_k+p_k$.

Шаг 4. Вычисляются в этой точке x_{k+1} градиент $f'(x_{k+1})$ и матрица $f''(x_{k+1})$.

Шаг 5. Если $\|f'(x_{k+1})\| \le \varepsilon_3$, то поиск на этом заканчивается и полагается $x = x_{k+1}$ и $y = f(x_{k+1})$. Иначе $k=k+1$ и переход к шагу 2.

Особенностью метода Ньютона является то, что для квадратичной целевой функции он находит минимум за один шаг, независимо от начального приближения x_0 и степени овражности.

В общем случае, когда минимизируемая функция не квадратичная, вектор $p_k = -(f''(x_k))^{-1} \cdot f'(x_k)$ не указывает в точку её минимума, однако имеет большую составляющую вдоль оси оврага и значительно ближе к направлению на минимум, чем антиградиент. Этим и объясняется более высокая сходимость метода Ньютона по сравнению с градиентными методами при минимизации овражных целевых функций.

Недостатками метода Ньютона является то, что он, во-первых, предполагает вычисление вторых производных и, во-вторых, может расходиться, если начальное приближение находится слишком далеко от минимума.

1.1.3. Методы с регулировкой шага

(методы Ньютона – Рафсона)

Удачный выбор начального приближения x_0 гарантирует сходимость метода Ньютона. Однако отыскание подходящего начального приближения – далеко не простая задача. Поэтому необходимо изменить формулу (4) так, чтобы добиться сходимости независимо от начального приближения. Доказано, что в некоторых предположениях для этого достаточно в методе Ньютона кроме направления движения $(f''(x))^{-1} \cdot f'(x)$ выбирать и длину шага вдоль него. Такие алгоритмы называются методами Ньютона с регулировкой шага (методами Ньютона – Рафсона) и выглядят так:

$$x_{k+1} = x_k - \alpha_k (f''(x_k))^{-1} \cdot f'(x_k). \qquad (6)$$

Как и в градиентных методах величина α_k выбирается так, чтобы обеспечить убывание целевой функции на каждой итерации.

Рассмотрим два способа выбора шага α_k.

Первый из них связан с проверкой неравенства

$$f(x_k + \alpha_k p_k) - f(x_k) \le \delta \cdot \alpha_k (f'(x_k), p_k), \qquad (7)$$

где $p_k = -(f''(x_k))^{-1} \cdot f'(x_k)$ – направление спуска, а $0 < \delta < ½$ – некоторое заданное число, общее для всех итераций. Если это неравенство выполнено при $\alpha_k = 1$, то шаг принимается равным единице и осуществляется следующая итерация. Если нет – дробится до тех пор, пока оно не выполнится.

Алгоритм метода Ньютона – Рафсона с регулировкой шага

Шаг 1. На первой итерации, при $k = 0$, вводятся исходные данные x_0, δ, ε_3. Вычисляются значения градиента $f'(x_0)$ и матрица $f''(x_0)$.

Шаг 2. Присваивается $\alpha = 1$. Определяется направление спуска p_k, как решение системы линейных уравнений $f''(x_k) \cdot p_k = -f'(x_k)$.

Шаг 3. Проверяется условие $f(x_k + \alpha_k p_k) - f(x_k) \le \delta \cdot \alpha_k (f'(x_k), p_k)$. Если оно выполняется, то переход к шагу 4. Иначе дробим значение шага α (например, $\alpha = \alpha/2$) и повторяем шаг 3.

Шаг 4. Определяется следующая точка: $x_{k+1} = x_k + \alpha \cdot p_k$.

Шаг 5. Вычисляются значение градиента $f'(x_{k+1})$ в точке x_{k+1}.

Шаг 6. Если $\|f'(x_{k+1})\| \le \varepsilon_3$, то поиск на этом заканчивается и полагается $x = x_{k+1}$ и $y = f(x_{k+1})$.

Иначе $k = k + 1$ и переход к шагу 2.

Второй метод определения шага α_k в схеме (6), как и в методе наискорейшего спуска состоит в минимизации функции

$$f(x_k + \alpha_k p_k) = \min f(x_k + \alpha_k p_k).$$

$\alpha \ge 0$

Алгоритм метода Ньютона – Рафсона с выбором оптимального шага

Шаг 1. При $k = 0$, вводятся x_0, ε_3. Вычисляются $f'(x_0)$ и $f''(x_0)$.

Шаг 2. Определение направления спуска p_k, как решение системы линейных уравнений $f''(x_k) \cdot p_k = -f'(x_k)$.

Шаг 3. Определяется следующая точка спуска:

$$x_{k+1} = x_k + \alpha p_k,$$

где α - решение задачи одномерной оптимизации: $\min f(x_k + \alpha p_k)$.

Шаг 4. Вычисляются в точке x_{k+1}: $f'(x_{k+1})$ и $f''(x_{k+1})$.

Шаг 5. Если $\|f'(x_{k+1})\| \le \varepsilon_3$, то поиск заканчивается и полагается $x = x_{k+1}$ и $y = f(x_{k+1})$. Иначе $k = k + 1$ и переход к шагу 2.

1.1.3 Модификации метода Ньютона

Значительные трудности, возникающие при практической реализации метода Ньютона, связаны с необходимостью вычислить матрицу $f''(x)$. Мы рассмотрим две модификации метода Ньютона, которые используют не точные значения, а некоторые приближённые аналоги матрицы вторых производных. В результате уменьшается трудоёмкость методов, но ухудшается их сходимость.

В качестве первой модификации метода Ньютона рассмотрим следующий алгоритм:

$$x_{k+1} = x_k - \alpha_k (f''(x_k))^{-1} \cdot f'(x_k), \quad \alpha_k \ge 0. \qquad (8)$$

здесь для построения направления спуска используется один раз вычисленная и обращённая матрица вторых производных $f''(x_0)$.

Алгоритм метода 1 Ньютона

Шаг 1. При $k = 0$, вводятся x_0, ε_3. Вычисляются $f'(x_0)$ и $f''(x_0)$.

Шаг 2. Определение обратной матрицы $(f''(x_0))^{-1}$.

Шаг 3. Определение направления спуска p_k: $p_k = -f'(x_k) \cdot (f''(x_0))^{-1}$.

Шаг 4. Определение следующей точки: $x_{k+1} = x_k + \alpha \cdot p_k$, где α – решение задачи одномерной минимизации функции $\varphi(\alpha) = f(x_k + \alpha \cdot p_k)$, при $\alpha \ge 0$.

Шаг 5. Вычисление в точке x_{k+1} градиента $f'(x_{k+1})$.

Шаг 6. Если $\|f'(x_{k+1})\| \le \varepsilon_3$, то поиск заканчивается и полагается $x = x_{k+1}$ и $y = f(x_{k+1})$.

Иначе $k = k + 1$ и переход к шагу 3.

В рассмотренной схеме для выбора шага α_k используется способ аналогичный используемому в методе наискорейшего спуска.

Но можно воспользоваться и способом, который аналогичен используемому в градиентном методе с дроблением шага.

Если матрица $f''(x)$ положительно определена, то итерационный процесс (δ) является одной из модификаций градиентного спуска, независимо от начального приближения x_0.

Другая модификация метода Ньютона связана с обновлением матрицы вторых производных через определённое количество шагов. Формула вычисления очередной точки x_{k+1}, в этом случае, будет выглядеть следующим образом:

$$x_{jm+i+1} = x_{jm+i} - \alpha_{jm+i} \cdot (f''(x_{jm}))^{-1} \cdot f'(x_{jm+i}),$$

$$\alpha_{jm+i} \geq 0,\ k = jm + i,\ j = 0, 1, 2, \ldots,\quad i = 0, 1, \ldots, m-1.$$

Здесь $m > 0$ – целое число, определяющее количество шагов через которое происходит обновление матрицы вторых производных $f''(x)$. Этот метод занимает промежуточное положение между методом Ньютона и его первой модификацией.

Алгоритм метода 2 Ньютона

Шаг 1. Ввод x_0, ε_3, m. Присвоение $j=0$ и $k=0$. Вычисление градиента $f'(x_0)$.

Шаг 2. Вычисление (обновление) матрицы $f''(x_{jm})$ и обратной матрицы $(f''(x_{jm}))^{-1}$.

Шаг 3. Определение направления спуска p_{jm+1}: $p_{jm+1} = -f'(x_{jm+1}) \cdot (f''(x_{jm}))^{-1}$.

Шаг 4. Определение очередной точки x_{jm+i+1}: $x_{jm+i+1} = x_{jm+i} + \alpha \cdot p_{jm+i}$, где α – решение задачи одномерной минимизации функции $\varphi(\alpha) = f(x_{jm+i} + \alpha \cdot p_{jm+i})$ при $\alpha \geq 0$.

Шаг 5. Вычисление в очередной точке x_{jm+i+1} градиента $f'(x_{jm+i+1})$.

Шаг 6. Если $\|f'(x_{jm+i+1})\| \leq \varepsilon_3$, то поиск закончен; полагаем $x = x_{jm+i+1}$ и $y = f(x_{jm+i+1})$.

Иначе $k = k+1$ и переход к шагу 7.

Шаг 7. $i = i+1$. Если $i = m$, то $j = j+1$, $i = 0$; переход к шагу 2 (т.е. обновляем матрицу $f''(x)$). Иначе переход к шагу 3 (т.е. используем матрицу $f''(x)$, вычисленную на одном из предыдущих шагов).

1.2. Метод сопряженных градиентов

1.2.1. Общая характеристика

Метод сопряженных градиентов относится к группе методов сопряженных направлений. Этот метод как и метод градиентного спуска, является методом первого порядка т. е. использует информацию только первой производной минимизируемой функции. Однако метод сопряженных градиентов отличается от градиентных методов более высокой скоростью сходимости, которая при определенных предположениях относительно целевой функции, приближается к скорости сходимости метода Ньютона.

Два вектора x и y называют H-сопряженными (или сопряженными по отношению к матрице H) или H-ортогональными, если

$$(x, H\cdot y)=0. \qquad (9)$$

Сопряженность можно считать обобщением понятия ортогональности. В самом деле, когда $H=E$, то x и y в соответствии с уравнением (9) ортогональны.

Рассмотрим квадратичную функцию n переменных:

$$f(x) = a + (x,b) + \tfrac{1}{2}(x, H\cdot x) \qquad (10)$$

с положительно определенной $n\times n$ матрицей. Оказывается, что квадратичная функция (10) может быть минимизирована методом сопряженных направлений не более чем за n шагов.

Чтобы воспользоваться этим методом минимизации квадратичной функции (10) нужно знать n взаимно сопряженных направлений $S_0, S_1,\ldots,S_{n-1}$. Эффективность таких направлений – самостоятельная проблема. Существует много взаимно сопряженных направлений $S_0, S_1,\ldots,S_{n-1}$ и способов их построения. Ниже излагается метод сопряженных градиентов Флетчера - Ривса, в котором выбор H-сопряженных направлений осуществляется совместно с одномерной минимизацией $f(x)$ по α.

1.2.2 Метод Флетчера – Ривса

Этот метод использует последовательность направлений поиска, каждая из которых является линейной комбинацией антиградиента в текущей точке и предыдущего направления спуска. Метод изменяется к квадратичной целевой функции

$$f(x)=a+(x,b)+½\,(x,\,H\cdot x).$$

При минимизации ее методом Флетчера - Ривса векторы S_k вычисляются по формулам

$$S_0=-f'(x_0),$$

$$S_k=-f'(x_k)+\beta_{k-1}\cdot S_{k-1},\ \text{при}\ k\geq 1.$$

Величины β_{k-1} выбираются так, чтобы направления S_k , S_{k-1} были H-сопряженными.

Точка x_{k-1} определяется в результате минимизации функции $f(x)$ в направлении S_k, исходящем из точки x_k, т.е.

$$x_{k+1}=x_k+\alpha_k\cdot S_k,$$

где α_k доставляет минимум по α_k функции $f(x_k,\alpha\cdot S_k)$.

Итак, предлагаемая процедура минимизации функции $f(x)$ выглядит следующим образом. В заданной точке x_0 вычисляется антиградиент $S_0=-f'(x_0)$. Осуществляется одномерная минимизация в этом направлении и определяется точка x_1. В точке x_1 снова вычисляется антиградиент $-f'(x_1)$. Так как эта точка доставляет минимум функции $f(x)$ вдоль направления $S_0=-f'(x_0)$, вектор $f'(x_1)$ ортогонален $f'(x_0)$. Затем по известному значению $f'(x_1)$ по формуле (11) вычисляется вектор S_1, который за счет выбора β_0 будет H–сопряженным к S_0. Далее отыскивается минимум функции $f(x)$ вдоль направления S_1 и т.д.

Алгоритм метода Флетчера – Ривса

Шаг 1. При k=0 ввод начального приближения x_0. Вычисление антиградиента $S_0=-f'(x_0)$.

Шаг 2. Решение задачи одномерной минимизации по α функции $f(x_k + \alpha \cdot S_k)$, в результате чего определяется величина шага α_k и точка $x_{k+1}=x_k+\alpha_k \cdot S_k$.

Шаг 3. Вычисление величин $f(x_{k+1})$ и $f'(x_{k+1})$.

Шаг 4. Если $f'(x_{k+1})=0$, то x_{k+1} – решение задачи. Иначе определяем новое направление поиска: S_{k+1} из соотношения :

$$\frac{(f'(x_{k+1}), f'(x_{k+1}) - f'(x_k))}{(f'(x_k), f'(x_k))}$$

$$S_{k+1} = -f'(x_{k+1}) + S_k$$

Далее $k=k+1$ и переход к шагу 2.

1.2.3. Минимизация неквадратичной целевой функции

Метод Флетчера-Ривса может применятся для минимизации и неквадратичных функций. Он является методом первого порядка и в тоже время скорость его сходимости квадратична. Разумеется, если целевая функция не квадратична, метод уже не будет конечным. Поэтому после (n+1)-й итерации процедура повторяется с заменой x_0 на x_{n+1}, а счет заканчивается при $\|f'(x_{k+1})\| \le \varepsilon$, где ε – заданное число. При минимизации неквадратичных функций обычно применяется следующая модификация метода Флетчера-Ривса.

Алгоритм метода Флетчера-Ривса для неквадратичных целевых функций

Шаг 1. При $k = 0$ ввод начального приближения x_0 и условия останова ε_3. Вычисление антиградиента $S_0 = -f'(x_0)$.

Шаг 2. Решение задачи одномерной минимизации по α функции $f(x_k + \alpha \cdot S_k)$, в результате чего определяется величина шага α_k и точка $x_{k+1} = x_k + \alpha_k \cdot S_k$.

Шаг 3. Вычисление величин $f(x_{k+1})$ и $f'(x_{k+1})$.

Шаг 4. Если $\|f'(x_{k+1})\| \le \varepsilon_3$, то точка x_{k+1} – решение задачи и на этом поиск заканчивается. Иначе определяется коэффициент β_k по формуле:

$$\beta_k = \begin{cases} \dfrac{(f'(x), f'(x_{k+1}) - f'(x_k))}{(f'(x_k), f'(x_k))}, & при \dots k+1 \notin I \\ 0, & при \dots k+1 \in I \end{cases}$$

Шаг 5. Вычисление S_{k+1} по формуле $S_{k+1} = -f'(x_{k+1}) + \beta_k \cdot S_k$; $k = k + 1$, переход к шагу 2.

Здесь I – множество индексов, $I = \{0, n, 2n, 3n, \dots\}$. Значения k, для которых $\beta_k = 0$, называют *моментами обновления метода*. Таким образом, обновление метода происходит через каждые n шагов.

1.3. Модификации метода Ньютона.

Придать методу Ньютона свойство глобальной сходимости можно различными способами. Один из них связан с регулировкой длины шага:

$$x^{k+1} = x^k - \gamma_k [\nabla^2 f(x^k)]^{-1} \nabla f(x^k). \qquad (11)$$

Его часто называют *демпфированным методом Ньютона.* Параметр γ_k может выбираться по-разному, например

$$\gamma_k = \arg\min_{\gamma \ge 0} f\left(x^k - \gamma [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)\right) \qquad (12)$$

или γ дробится (умножается на $0<\alpha< 1$), начиная с $\gamma=1$, до выполнения условия

$$f(x^{k+1}) \leqslant f(x^k) - \gamma q \left([\nabla^2 f(x^k)]^{-1} \nabla f(x^k),\ \nabla f(x^k)\right), \quad 0 < q < 1, \quad (13)$$

или условия

$$\| \nabla f(x^{k+1}) \|^2 \leqslant (1 - \gamma q) \| \nabla f(x^k) \|^2, \quad 0 < q < 1. \quad (14)$$

Для гладких сильно выпуклых функций демпфированный метод Ньютона глобально сходится. Что касается скорости сходимости, то на начальных итерациях можно утверждать лишь сходимость со скоростью геометрической прогрессии. При попадании же в окрестность x^*, в которой выполняются условия известной теоремы о сходимости метода Ньютона, будет иметь место квадратичная сходимость.

Возможна и другая модификация (называемая *методом Левенберга — Марквардта),* в которой само направление движения отличается от задаваемого методом Ньютона. Поступим так же, как при одном из обоснований градиентного метода — добавим к аппроксимирующей функции квадратичный штраф за отклонение от точки x^k, т. е. будем искать x^{k+1} из условия минимума

$$\begin{aligned} &f_k(x) + (\alpha_k/2) \| x - x^k \|^2, \\ &f_k(x) = f(x^k) + (\nabla f(x^k),\ x - x^k) + (\nabla^2 f(x^k)(x - x^k),\ x - x^k)/2. \end{aligned} \quad (15)$$

Тогда приходим к методу

$$x^{k+1} = x^k - (\nabla^2 f(x^k) + \alpha_k I)^{-1} \nabla f(x^k). \quad (16)$$

При $\alpha_k=0$ метод переходит в метод Ньютона, при $\alpha_k \to \infty$ направление движения стремится к антиградиенту. Таким образом, (16) представляет собой компромисс между этими двумя методами. За счет выбора α_k можно добиться глобальной сходимости метода.

Метод (16) обладает перед (11) тем преимуществом, что он (как и градиентный метод) пригоден не только для выпуклых функций, тогда как в методе (11) требуется положительная определенность матрицы $\nabla^2 f(x)$.

Есть специальные модификации метода Ньютона, в которых матрица $\nabla^2 f(x^k)$ заменяется на некоторую положительно определенную, если сама $\nabla^2 f(x^k)$ таковой не является.

Однако во всех описанных модификациях метода Ньютона каждая итерация (как и в основном методе Ньютона) требует очень большой вычислительной работы (вычисление $\nabla^2 f(x)$, решение систем линейных уравнений), а скорость сходимости вдали от минимума, вообще говоря, не высока.

Таким образом, попытки «слегка подправить» градиентный метод и метод Ньютона хотя и позволяют устранить некоторые их недостатки, но не меняют положение с наиболее серьезными их дефектами — медленной сходимостью градиентного метода и трудоемкостью метода Ньютона.

5.9. Сравнение методов одномерного поиска

Наилучшими критериями сравнения методов поиска, которые были описаны выше, есть их эффективность и универсальность. Под эффективностью алгоритма понимают число вычислений функции, необходимое для достижения необходимого сужения интервала неопределенности. Из табл. 1 видно, что наилучшим в этом отношении есть метод Фибоначчи, а наиболее плохим - метод общего поиска.

Таблица 1. Сравнение методов одномерного поиска по значением коэффициента дробления интервала неопределенности f

Количество вычислений целевой функции N	Общий поиск	Деление отрезка пополам	Метод дихотомии	Метод золотого сечения	Метод Фибоначчи
1	1,0	1,0	1,0	1,0	1,0
2	0,667	-	0,500	0,618	0,500
3	0,500	0,500	-	0,382	0,333
4	0,400	-	0,250	0,236	0,200
5	0,333	0,250	-	0,146	0,125
6	0,286	-	0,125	0,090	0,077
7	0,250	0,125	-	0,056	0,048
8	0,222	-	0,0625	0,345	0,0294
9	0,200	0,0625	-	0,0213	0,0182

10	0,182	-	0,0312	0,0132	0,0112
11	0,167	0,0312	-	0,00813	0,00694
12	0,154	-	0,0156	0,00502	0,00429
13	0,143	0,0156	-	0,00311	0,00265
14	0,133	-	0,00781	0,00192	0,00164
15	0,125	0,00781	-	0,00119	0,00101
16	0,118	-	0,00391	0,000733	0,000626
17	0,111	0,00391	-	0,000453	0,000387
18	0,105	-	0,00195	0,000280	0,000239
19	0,100	0,00195	-	0,000173	0,000148
20	0,095	-	0,000976	0,000107	0,0000913

Конструктор не с большим удовлетворением использует метод Фибоначчи, так как при его применении необходимо заранее задавать число вычислений значений функции. Однако он может воспользоваться методом золотого сечения. Как правило, методы Фибоначчи и золотого сечения, обладают высокой эффективностью, наиболее подходят для решения одномерных унимодальных задач оптимизации.

Универсальность алгоритма означает, что его можно легко применить для решения самых разнообразных задач. В этом отношении метод Фибоначчи, уступает другим, так как нуждается в отдельном вычислении положения точек, в которых будет определяться значение целевой функции на каждом новом шаге. Этим приходится расплачиваться за повышение эффективности метода. С точки зрения универсальности малоэффективный метод общего поиска имеет по крайней мере одно преимущество - его можно с успехом применять и для неунимодальных функций, если они достаточно

тучные. Нередко заранее не известно, есть ли рассмотренная целевая функция унимодальной. В таких случаях нужно использовать несколько разных алгоритмов и посмотреть, дают ли они все один и тот самый оптимум. Отсюда следует важный вывод, который нужно иметь в виду, решая задачи оптимизации: не существует универсального алгоритма, который позволял бы решать любые задачи. Решая сложные задачи оптимизации, нужно пользоваться разными методами, так как это позволяет увеличить судьбу удобных решений.

Пример 1. *Одномерная минимизация в среде Mathcad*

Для нахождения минимума одномерной функции в среде Mathcad используется функция ***root(f(var1, var2, ...),var1, [a, b]) –*** *возвращает переменную* ***var1,*** *которая лежит между* ***a*** *и* ***b,*** *в которой решаемая функция равна нулю.*

Параметры:

f – *уравнение, которое следует решить;*

var1 *– корень уравнения;*

[a, b] – *отрезок, на котором ищется решение уравнения.*

Например, необходимо найти минимум гладкой унимодальной функции $y=x^2+e^x$*, используя необходимое условие минимума.*

Определим целевую функцию

$$f(x) := x^2 + \exp(x)$$

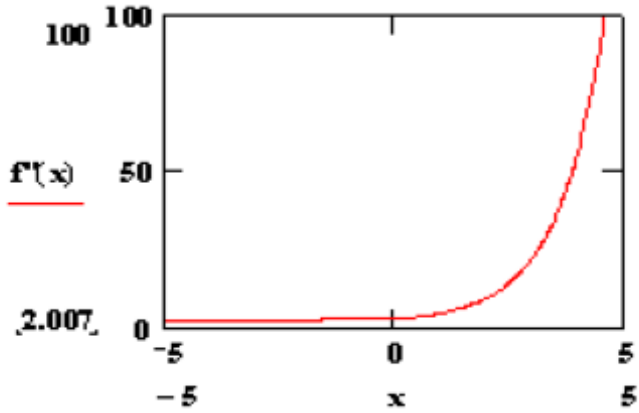
Определим первую производную целевой функции

$$f'(x) := \frac{d}{dx} f(x) \qquad f'(x) \rightarrow 2 \cdot x + \exp(x)$$

Определим вторую производную целевой функции

$$f''(x) := \frac{d^2}{dx^2} f(x) \qquad f''(x) \to 2 + \exp(x)$$

Достаточное условие унимодальности ***(f''(x)>0)*** *выполненная*



Минимум гдадкой функции достигается в стационарной точке ***(f'(x)=0)****.*

Решим уравнение ***f'(x)=0,*** *используя функцию* ***root****; начальное приближение корня пусть равно нулю* ***(x=0)***

$$x := 0$$

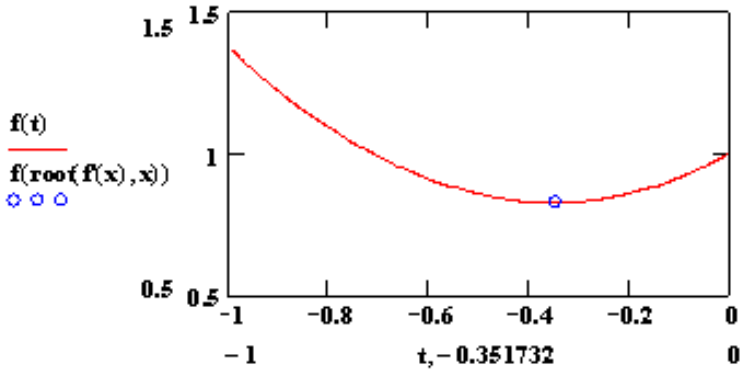
$$root(f'(x), x) = -0.351732$$

Точка минимума ***x=-0.351732***

Значение функции в точке минимума

$$f(root(f'(x), x)) = 0.827184$$

Подтвердим результаты вычислений графически



5.10. Многошаговые методы

В градиентном методе на каждом шаге никак не используется информация, полученная на предыдущих итерациях. Естественно попытаться учесть «предысторию» процесса для ускорения сходимости. Такого рода методы, в которых новое приближение зависит от s предыдущих:

$$x^{k+1} = \varphi_k(x^{k+1},..., x^{k-s+1}), \qquad (1)$$

называются *s-шаговыми.* Градиентный метод и метод Ньютона были одношаговыми, теперь рассмотрим *многошаговые* ($s > 1$) методы.

1. Метод тяжелого шарика. Одним из простейших многошаговых методов является двухшаговый *метод тяжелого шарика*

$$x^{k+1} = x^k - \alpha \nabla f(x^k) + \beta (x^k - x^{k-1}), \qquad (2)$$

где $\alpha > 0$, $\beta \geq 0$ — некоторые параметры. Ясно, что при $\beta = 0$ метод (2) переходит в градиентный. Свое название метод получил из-за следующей физической аналогии. Движение тела («тяжелого шарика») в потенциальном поле при наличии силы трения (или вязкости) описывается дифференциальным уравнением второго порядка

$$\mu \frac{d^2x(t)}{dt^2} = -\nabla f(x(t)) - p\frac{dx(t)}{dt}. \qquad (3)$$

Ясно, что из-за потери энергии на трение тело в конце концов окажется в точке минимума потенциала *f(x)*. Таким образом, тяжелый шарик «решает» соответствующую задачу минимизации. Если рассмотреть разностный аналог уравнения (3), то придем к итерационному методу (2).

Введение инерции движения (член $\beta(x^k - x^{k-1})$) в итерационный процесс может привести к ускорению сходимости. Это видно, например, из рис. 1 — вместо зигзагообразного движения в градиентном методе в данном случае получается более плавная траектория по «дну оврага».

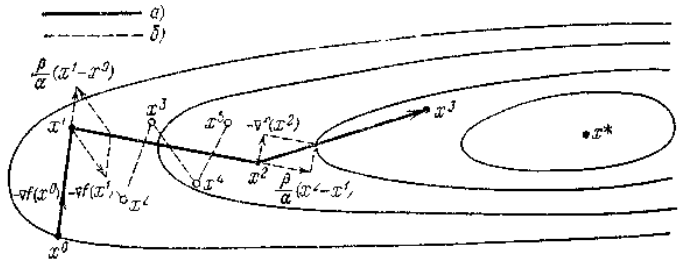


Рис. 1. Метод тяжелого шарика *(а)* и градиентный метод (б).

Эти эвристические соображения подкрепляются следующей теоремой

Теорема 1. *Пусть* x^**— невырожденная точка минимума* $f(x)$, $x \in \mathbb{R}^n$. *Тогда при*

$$0 \leqslant \beta < 1, \quad 0 < \alpha < 2(1+\beta)/L, \quad lI \leqslant \nabla^2 f(x^*) \leqslant LI \qquad (4)$$

найдется $\varepsilon > 0$ *такое, что при любых* x^0, x^1, $\|x^0 - x^*\| \leqslant \varepsilon$, $\|x^1 - x^*\| \leqslant \varepsilon$ *метод* (2) *сходится к* x^* *со скоростью геометрической прогрессии.*

$$\|x^k - x^*\| \leqslant c(\delta)(q+\delta)^k, \quad 0 \leqslant q < 1, \quad 0 < \delta < 1 - q. \qquad (5)$$

Величина q минимальна и равна

$$q^* = \frac{\sqrt{L} - \sqrt{l}}{\sqrt{L} + \sqrt{l}} \quad при \quad \alpha^* = \frac{4}{(\sqrt{L} + \sqrt{l})^2}, \quad \beta^* = \left(\frac{\sqrt{L} - \sqrt{l}}{\sqrt{L} + \sqrt{l}}\right)^2. \qquad (6)$$

Схема доказательства. В данном случае непосредственно применить приемы исследования сходимости, описанные ранее, нельзя, так как все они рассчитаны на одношаговые процессы. Можно, однако, использовать способ увеличения размерности пространства, позволяющий свести многошаговый процесс к одношаговому. Введем $2n$-мерный вектор $z^k = \{x^k - x^*, x^{k-1} - x^*\}$. Тогда итерационный процесс (2) может быть записан в форме

$$z^{k+1} = Az^k + o(z^k), \qquad (7)$$

где квадратная матрица A размерности $2n \times 2n$ и имеет вид

$$A = \begin{pmatrix} (1+\beta)I - \alpha B & -\beta I \\ I & 0 \end{pmatrix}, \quad B = \nabla^2 f(x^*). \tag{8}$$

Пусть $l = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n = L$ — собственные значения матрицы B. Тогда собственные значения ρ_j, $j = 1, \ldots, 2n$, матрицы A совпадают с собственными значениями матриц 2×2 вида

$$\begin{pmatrix} 1+\beta-\alpha\lambda_i & -\beta \\ 1 & 0 \end{pmatrix}.$$

Следовательно, они являются корнями уравнений

$$\rho^2 - \rho(1+\beta-\alpha\lambda_i) + \beta = 0, \qquad i = 1, \ldots, n. \tag{9}$$

Можно показать, что если

$$0 < l \leqslant \lambda_i \leqslant L, \ 0 \leqslant \beta < 1, \ 0 < \alpha < 2(1+\beta)/L, \text{ то } |\rho| < 1,$$

где ρ — любой корень уравнения (9).

Теперь мы можем воспользоваться теоремой о локальной сходимости итерационных процессов вида (7), что дает возможность получить оценку (5). Вычисляя

$$\min_{\alpha, \beta} \max_{1 \leqslant j \leqslant 2n} |\rho_j|,$$

находим приведенные в теореме оптимальные значенияа α^*, β^* и соответствующее им q^*.

Сравним скорость сходимости, даваемую одношаговым и двухшаговым методами при оптимальном выборе параметров. И в том, и в другом случаях имеем сходимость со скоростью геометрической прогрессии, но знаменатель прогрессии для одношагового метода равен

$$q_1 = (L-l)/(L+l), \tag{10}$$

а для двухшагового

$$q_2 = (\sqrt{L} - \sqrt{l})/(\sqrt{L} + \sqrt{l}). \tag{11}$$

Для больших значений числа обусловленности $\mu = L/l$

$$q_1 \approx 1 - 2/\mu, \quad q_2 \approx 1 - 2/\sqrt{\mu}. \tag{12}$$

Поэтому, чтобы приблизиться к решению в e=2,7 ... раз, в одношаговом методе требуется порядка $\mu/2$ итераций, в двух-шаговом — порядка $\sqrt{\mu}/2$. Иными словами, для плохо обусловленных задач метод тяжелого шарика дает выигрыш в $\sim \sqrt{\mu}$ раз по сравнению с

градиентным. Для больших μ эта разница весьма значительна. С вычислительной же точки зрения метод (2) немногим сложнее одношагового.

Правда, подбор оптимальных значений α и β в (2) не прост — формулами (6) непосредственно воспользоваться не удается, так как границы спектра $\nabla^2 f(x^*)$ (числа *l* и *L)* обычно неизвестны.

2. Метод сопряженных градиентов. Рассмотрим другой вариант двухшагового метода — *метод сопряженных градиентов,* в котором параметры находятся из решения двумерной задачи оптимизации:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) + \beta_k (x^k - x^{k-1}), \tag{13}$$

$$\{\alpha_k, \beta_k\} = \operatorname*{argmin}_{\{\alpha, \beta\}} f(x^k - \alpha \nabla f(x^k) + \beta (x^k - x^{k-1})). \tag{14}$$

Для случая квадратичной функции

$$f(x) = (Ax, x)/2 - (b, x), \quad A > 0, \tag{15}$$

эта задача может быть решена явно:

$$\alpha_k = \frac{\|r^k\|^2 (Ap^k, p^k) - (r^k, p^k)(Ar^k, p^k)}{(Ar^k, r^k)(Ap^k, p^k) - (Ar^k, p^k)^2}, \quad r^k = \nabla f(x^k) = Ax^k - b,$$
$$\beta_k = \frac{\|r^k\|^2 (Ar^k, p^k) - (r^k, p^k)(Ar^k, r^k)}{(Ar^k, r^k)(Ap^k, p^k) - (Ar^k, p^k)^2}, \quad p^k = x^k - x^{k-1}. \tag{16}$$

Могло бы показаться, что соотношение методов (13), (14) и (2) такое же, как рассмотренных ранее методов, — если метод скорейшего спуска не дает, как мы видели, выигрыша в скорости сходимости по сравнению с градиентным методом с постоянным оптимальным γ, то и от двухшагового варианта скорейшего спуска (13), (14) трудно ждать существенного ускорения по сравнению с методом тяжелого шарика (2). Оказывается, ситуация здесь иная: так, в квадратичном случае метод (13), (14) (при специальном выборе p^1) является конечным, т. е. дает точный минимум функции (15) за конечное число итераций.

Пусть начальное приближение x^0 произвольно, а x^1 получено из него методом скорейшего спуска:

$$x^1 = x^0 - \frac{\|r^0\|^2}{(Ar^0, r^0)} r^0, \quad r^0 = \nabla f(x^0) = Ax^0 - b. \tag{17}$$

Лемма 1. *Градиенты r^0, r^1, ... в методе* (13), (16), (17) *попарно ортогональны:*

$$(r^i, r^k) = 0, \quad i<k. \tag{18}$$

Доказательство. Воспользуемся индукцией по *k*. Пусть $(r^i, r^k) = 0$ при $0 \leqslant i < k$, $k \geqslant 2$, и $r^i \neq 0$, $i = 0, \ldots, k$. Ортогональность r^0, r^1, r^2 следует непосредственно из определения метода. Тогда, умножая (13) слева на *A,* получаем

$$r^{k+1} = r^k - \alpha_k A r^k + \beta_k (r^k - r^{k-1}).$$

Из $r^i \neq 0$ для $i \leq k$ следует, что $\alpha_k \neq 0$. Поэтому Ar^k есть линейная комбинация r^{k+1}, r^k и r^{k-1}, аналогично Ar^i, $i < k$, есть линейная комбинация r^{i+1}, r^i, r^{i-1} и в силу предположения индукции $(Ar^i, r^j) = 0$, $|i - j| > 1$, $i < k$, $j \leqslant k$. Следовательно,

$$(r^{k+1}, r^i) = (r^k - \alpha_k A r^k + \beta_k (r^k - r^{k-1}), r^i) = 0$$

при $i = 0, ..., k - 2$.

Далее, непосредственно из формул (13), (16) следует, что

$$(r^{k+1}, r^k) = 0, \quad (r^{k+1}, p^k) = 0.$$

Наконец, из (13), заменяя k на k—1, имеем $p^k = -\alpha_{k-1} r^{k-1} + \beta_{k-1} p^{k-1}$. Применяя это соотношение последовательно, получаем, что p^k есть линейная комбинация r^0, r^1, ..., r^{k-1}, причем r^{k-1} входит с коэффициентом $-\alpha_{k-1} \neq 0$. Поэтому из $(r^{k+1}, p^k) = 0$, $(r^{k+1}, r^i) = 0$, $i \leqslant k - 2$, следует, что $(r^{k+1}, r^{k-1}) = 0$.

Итак, для всех $i \leq k$ будет $(r^{k-1}, r^i) = 0$.

Если r^k обращается в 0, то x^k —точка минимума $f(x)$. Но в $\mathbf{R}^n$ не может существовать более n ортогональных ненулевых векторов, поэтому для некоторого $k \leq n$ будет $r^k = 0$. Итак, мы доказали следующий результат.

Теорема 2. *Метод* (13), (16), (17) *дает точку минимума квадратичной функции* $f(x)$ *вида* (15) *за число итераций, не превосходящее* n.

Мы установим в дальнейшем, что если L — некоторое подпространство в $\mathbf{R}^n$, $f(x)$ - выпуклая дифференцируемая функция, то условие

$$(\nabla f(x^*), a) = 0 \quad \text{для всех} \quad a \in L$$

необходимо и достаточно для того, чтобы x^* было минимумом $f(x)$ на L. Отсюда и из леммы 1 следует, что x^k — точка минимума квадратичной функции $f(x)$ вида (5) на подпространстве, проходящем через x^0 и порожденном $r^0, ..., r^{k-1}$. Этот несколько неожиданный факт (мы ищем минимум k раз последовательно на 2-мерных подпространствах, а он оказывается минимумом на всем n-мерном подпространстве) является важнейшей особенностью метода сопряженных градиентов и объясняет его конечность.

Последовательные направления движения p^k в методе сопряженных градиентов удовлетворяют соотношению

$$(Ap^i, p^j) = 0, \quad i \neq j. \tag{19}$$

Действительно, $p^i = x^i - x^{i-1}$, поэтому $Ap^i = Ax^i - Ax^{i-1} =$

$= r^i - r^{i-1}$. С другой стороны, мы уже отмечали, что p^k есть линейная комбинация $r^0, \ldots, r^{k-1}$, $p^k = \sum_{j=0}^{k-1} \mu_j r^j$. Поэтому для $i > k$ имеем $(Ap^i, p^k) = \left(r^i - r^{i-1}, \sum_{j=0}^{k-1} \mu_j r^j\right) = 0$ в силу леммы 1.

Векторы p^i, связанные соотношением (19), называются *сопряженными* или *A-ортогональными* (они ортогональны в метрике, задаваемой матрицей *A)*. Это объясняет название метода — в нем строятся линейные комбинации последовательных градиентов, являющиеся сопряженными.

Отметим, что знание произвольных сопряженных направлений s^i, $i = 1, \ldots, n$, $(As^i, s^j) = 0$, $i \neq j$, позволяет без труда решить систему

$$Ax = b, \quad A > 0. \tag{20}$$

Действительно, будем искать решение в виде

$$x = \sum_{i=1}^{n} \alpha_i s^i.$$

Тогда, подставляя это в (20), умножая скалярно на s^i и используя *A*-ортогональность, имеем

$$\alpha_i = (b, s^i)/(As^i, s^i). \tag{21}$$

Этому решению можно придать рекуррентную форму: зададимся произвольным x^0 и построим $x^{k+1} = x^k + \alpha_k s^k$, где α_k задаются (21). Тогда $x^n = x^*$ — решение (20). Поскольку α_k в (21) можно определить иначе:

$$\alpha_k = \arg\min_{\alpha} f(x^k + \alpha s^k),$$

то мы получаем, что знание системы сопряженных направлений позволяет найти минимум квадратичной функции с помощью n одномерных минимизаций. Этот важный факт неоднократно можно использовать при построении других методов минимизации. В методе сопряженных градиентов сопряженные направления не выбираются заранее, а строятся по рекуррентным формулам.

Если применять метод (13), (14) для неквадратичных функций, то, сопоставляя его с методом скорейшего спуска, нетрудно доказать его глобальную сходимость, а сопоставляя с методом тяжелою шарика, — оценить скорость сходимости.

Методу сопряженных градиентов можно придать и иную форму. Рассмотрим итерационный процесс

$$x^{k+1} = x^k + \alpha_k p^k, \quad \alpha_k = \arg\min_{\alpha} f(x^k + \alpha p^k),$$
$$p^k = -r^k + \beta_k p^{k-1}, \quad \beta_k = \|r^k\|^2 / \|r^{k-1}\|^2, \qquad (22)$$
$$r^k = \nabla f(x^k), \ \beta_0 = 0.$$

Лемма 2. *Для случая квадратичной функции* (15) *методы* (13), (16), (17) *и* (22) *при одинаковом* x^0 *определяют одну и ту же последовательность точек* x^k.

Поскольку p^k в (22) и (16) отличаются лишь скалярными (ненулевыми) множителями, а r^k в (22) и (16) совпадают, то процесс (22) обладает теми же свойствами, что и (13), (16) векторы p^i являются сопряженными, а градиенты r^i — взаимно ортогональны. Из леммы 2 и теоремы 1 следует, что метод (22) дает точку минимума квадратичной функции (15) в $\mathbf{R}^n$ за число итераций, не превосходящее n. Для неквадратичных задач метод (22) проще, чем (13), (14), так как требует решения лишь одномерной (а не двумерной) вспомогательной задачи минимизации. Разумеется, в неквадратичном случае теряется свойство конечности метода и (22) превращается в, вообще говоря, бесконечный итерационный двухшаговый метод.

Обычно для неквадратичных задач метод сопряженных градиентов применяется в несколько иной форме В него вводится процедура *обновления* — время от времени шаг делается не по формуле (22), а как в начальной точке, т. е. по градиенту. Наиболее естественно производить обновление через число итераций, равное размерности пространства:

$$x^{k+1} = x^k + \alpha_k s^k, \quad \alpha_k = \arg\min_{\alpha \geqslant 0} f(x^k + \alpha s^k),$$
$$s^k = -r^k + \beta_k s^{k-1}, \quad r^k = \nabla f(x^k), \qquad (23)$$
$$\beta_k = \begin{cases} 0, & k = 0,\ n,\ 2n,\ \ldots \\ \|r^k\|^2 / \|r^{k-1}\|^2, & k \neq 0,\ n,\ 2n,\ \ldots \end{cases}$$

Нетрудно доказать, что метод сопряженных градиентов с обновлением обладает свойством глобальной сходимости. Оказывается, что в то же время в окрестности минимума он сходится с квадратичной скоростью.

Теорема 3. *Пусть* x^* *— невырожденная точка минимума, и в ее окрестности* $\nabla^2 f(x)$ *удовлетворяет условию Липшица. Тогда для метода* (23) *в окрестности* x^* *справедлива оценка*

$$\|x^{(m+1)n} - x^*\| \leqslant c \|x^{mn} - x^*\|^2.$$

Иначе говоря, по скорости сходимости n шагов метода сопряженных градиентов эквивалентны одному шагу метода Ньютона. Мы не приводим доказательства теоремы, так как оно довольно громоздко. В его основе лежит идея квадратичной аппроксимации $f(x)$ и факт конечности метода для квадратичных функций (см. теорему 2).

Возможны иные вычислительные схемы метода сопряженных градиентов для неквадратичных функций. С одной из них, требующей решения двумерной задачи минимизации на каждом шаге, мы начали анализ этого метода — см. (13), (14). Другие, подобно (22), обычно включают лишь одномерные вспомогательные задачи, но отличаются от (22) правилом выбора β_k. Примером может служить схема

$$x^{k+1} = x^k + \alpha_k s^k, \qquad \alpha_k = \operatorname*{argmin}_{\alpha \geqslant 0} f(x^k - \alpha s^k),$$

$$s^k = -r^k + \beta_k s^{k-1}, \quad \beta_k = \frac{(r^k,\ r^k - r^{k-1})}{\|r^{k-1}\|^2}, \tag{24}$$

$$r^k = \nabla f(x^k), \quad \beta_0 = 0.$$

Как и для (22), здесь возможны варианты либо с обновлением, либо без него. Для квадратичной функции последовательности x^k, порождаемые методами (22) и (24), совпадают.

Как показывает опыт вычислений, для неквадратичного случая несколько более быструю сходимость обычно дает схема (24).

Представляет интерес поведение метода для задач большой размерности (когда число итераций меньше размерности). Оказывается, здесь можно гарантировать лишь сходимость со скоростью геометрической прогрессии даже для квадратичного случая. Пусть A — матрица $n \times n$,

$$lI \leqslant A \leqslant LI, \qquad l > 0, \tag{25}$$

и $f(x)$ — соответствующая ей квадратичная функция на $\mathbf{R}^n$:

$$f(x) = (Ax,\ x)/2 - (b,\ x), \qquad b \in \mathbf{R}^n. \tag{26}$$

Точка x^k может быть представлена в виде

$$x^k - x^* = P_k(A)(x^0 - x^*), \tag{27}$$

где $P_k(A)$ — матричный полином k-й степени вида

$$P_k(A) = I + a_{1k}A + \ldots + a_{kk}A^k. \tag{28}$$

поэтому

$$\|x^k - x^*\|^2 \leqslant 2(f(x^k) - f^*)/l = (AP_k^2(A)(x^0 - x^*),\ x^0 - x^*)/l \leqslant$$
$$\leqslant (L/l)\|x^0 - x^*\|^2 \max_{l \leqslant \lambda \leqslant L} P_k^2(\lambda),$$

где $P_k(\lambda) = 1 + a_{1k}\lambda + \ldots + a_{kk}\lambda^k$ — обычный полином. В силу свойств метода оценка для $f(x^k)=f^*$ справедлива для всех $P_k(\lambda)$, $P_k(0)= 1$, в частности, для

$$P_k(\lambda) = T_k\left(\frac{L+l-2\lambda}{L-l}\right)/T_k\left(\frac{L+l}{L-l}\right).$$

где

$$T_k(\lambda) = [(\lambda+\sqrt{\lambda^2-1})^k + (\lambda-\sqrt{\lambda^2-1})^k]/2, \quad |\lambda|>1;$$
$$T_k(\lambda) = \cos(k \arccos\lambda), \quad |\lambda|\leqslant 1. \qquad (29)$$

Поэтому

$$\|x^k - x^*\| \leqslant 2\left(\frac{L}{l}\right)^{\frac{1}{2}}\left[\left(\frac{\sqrt{L}+\sqrt{l}}{\sqrt{L}-\sqrt{l}}\right)^k + \left(\frac{\sqrt{L}-\sqrt{l}}{\sqrt{L}+\sqrt{l}}\right)^k\right]^{-1}\|x^0 - x^*\| \leqslant$$

$$\leqslant 2\left(\frac{L}{l}\right)^{\frac{1}{2}} q^k \|x^0 - x^*\|, \quad q = (\sqrt{L}-\sqrt{l})/(\sqrt{L}+\sqrt{l}).$$

Можно показать на примерах, что оценка (30) неулучшаема.

Итак, при $k < n$ для метода сопряженных градиентов, примененного для минимизации квадратичной функции, можно гарантировать сходимость со скоростью геометрической прогрессии со знаменателем $q = (\sqrt{L}-\sqrt{l})/(\sqrt{L}+\sqrt{l}) \sim 1 - 2/\sqrt{\mu}$, $\mu = L/l$, т. е. такую же, как для метода тяжелого шарика при оптимальном выборе его параметров. По сравнению с последним в методе сопряженных градиентов нет проблемы выбора параметров — они определяются автоматически, хотя это и требует дополнительных вычислений для решения одномерной задачи минимизации.

Мы видим, что в методе сопряженных градиентов x^k является точкой минимума квадратичной функции $f(x)$ на подпространстве, порожденном первыми k градиентами. Отсюда следует, что никакой метод, использующий только градиенты функции (точнее, в котором шаг делается по линейной комбинации предыдущих градиентов), не может сходиться быстрее. Иными словами, метод сопряженных градиентов является оптимальным по скорости сходимости в классе методов первого порядка. Из полученного выше результата вытекает, что для задач большой размерности с квадратичными функциями $f(x)$, удовлетворяющими условию (25), для всех методов первого порядка нельзя ждать сходимости более высокой, чем скорость геометрической прогрессии со знаменателем $q = (\sqrt{L}-\sqrt{l})/(\sqrt{L}+\sqrt{l})$. Естественно, большая скорость сходимости не может достигаться и в более широком классе сильно выпуклых с константой l функций, градиент которых удовлетворяет

условию Липшица с константой *L*. Факт квадратичной сходимости (теорема 3) имеет место только при числе итераций, существенно большем размерности пространства.

5.11. Краткий анализ методов одномерной минимизации

Методы точечного оценивания

Эти методы учитывают информацию об относительном изменении значений функции в пробных точках. Методы накладывают дополнительные ограничения на функцию: функция должна быть непрерывной и достаточно гладкой.

Основная идея метода: возможность аппроксимации гладкой функции полиномом достаточно высокого порядка и использование этого полинома для оценивания точки оптимума.

Качество этой оценки может быть повышено двумя способами:

1. Увеличением степени полинома;

2. Уменьшением интервала аппроксимации.

Второй способ предпочтительнее, так как построение полинома порядка более 3 – достаточно сложная задача, а сужение интервала для унимодальной функции – достаточно простая.

Использование квадратичной аппроксимации для нахождения оптимума.

Чтобы функция имела минимум внутри отрезка она должна быть по крайней мере квадратичной.

Заданы x_1, x_2, x_3 и соответствующие им y_1, y_2, y_3. Можно задать аппроксимацию полинома вида:

$$q(x) = a_0 + a_1 \cdot (x - x_1) + a_2 \cdot (x - x_2) \cdot (x - x_1)$$

и выбрать a_0, a_1, a_2 так, чтобы

$q(x_1) = y_1$, $q(x_2) = y_2$, $q(x_3) = y_3$.

$$y_1 = q(x_1) = a_0$$

$$y_2 = q(x_2) = a_0 + a_1 \cdot (x_2 - x_1) \Rightarrow a_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

$$y_3 = q(x_3) = a_0 + a_1 \cdot (x_3 - x_1) + a_1 \cdot (x_3 - x_1) \cdot (x_3 - x_2) \Rightarrow a_2 = \frac{1}{x_3 - x_2} \cdot \left(\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1} \right)$$

Найдём стационарную точку $\overline{x}$ полинома *q(x)*

$$\frac{dq}{dx} = a_1 + a_2 \cdot (x - x_2) + a_2 \cdot (x - x_1) = 0$$

$$\overline{x} = \frac{x_2 - x_1}{2} - \frac{a_1}{2 \cdot a_2}$$

Так как функция *y=f(x)* унимодальна на рассматриваемом интервале и полином *q(x)* тоже унимодальная функция, то $\overline{x}$ является приемлемой оценкой истинного оптимума *x**. На этом основан метод Пауэлла.

Метод Пауэлла

Метод основан на последовательном применении процедуры оценивания с использованием квадратичной аппроксимации.

Алгоритм.

1. Задать x_1 и шаг Δx

2. Найти $x_2 = x_1 + \Delta x$. Вычислить $f(x_1)$ и $f(x_2)$.

3. Если $f(x_1) > f(x_2)$, то $x_3 = x_1 + 2 \cdot \Delta x$ иначе $x_3 = x_1 - \Delta x$.

4. Вычислить $f(x_3)$; $F_{\min} = \min\{ f(x_1), f(x_2), f(x_3)\}, X_{\min}$

5. Найти $\overline{x}$.

6. Проверка на окончание поиска. Если условия выполняются, то поиск окончен, иначе перейти к п. 7.

7. Принять за x_1 наилучшую из точек $\overline{x}$ и $X_{\min}$. Перейти к п. 2.

Метод Ньютона-Рафсона

Повышение эффективности метода за счёт использования информации о производной накладывает дополнительные ограничения на функцию. Кроме унимодальности функция должна быть непрерывной и дважды дифференцируемой.

Пусть $f(x)$ - непрерывная и дважды дифференцируемая функция.

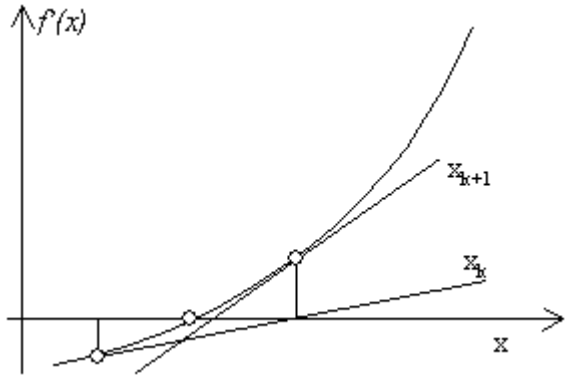
Требуется найти корень уравнения $f'(x)=0$.

Зададим x_1 – начальную точку поиска. Построим линейную аппроксимацию функции $f'(x)$ в точке x_1. Для этого разложим $f'(x)$ в ряд Тейлора в точке x_1 и отбросим все члены второго порядка и выше.

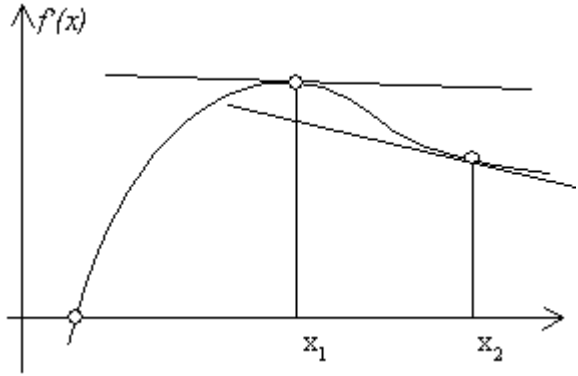
$$\tilde{f}'(x; x_1) = f'(x_1) + f''(x_1) \cdot (x - x_1) + o(x - x_1^2)$$

$$x = x_{k+1}$$

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$



Сходимость метода зависит от выбора начальной точки и вида функции.



Условие выхода $|f'(x)| \le \varepsilon$ не сходится

Метод средней точки

Определяются две точки *L, R* в которых производные имеют разные знаки *f'(L)* <0, *f'(R)* >0. Искомый оптимум находится между ними. Делим интервал пополам:

$$Z = \frac{L+R}{2}.$$

Если *f'(Z)* >0 то исключаем (*Z, R*). Если *f'(Z)* <0 то исключаем (*L, Z*).

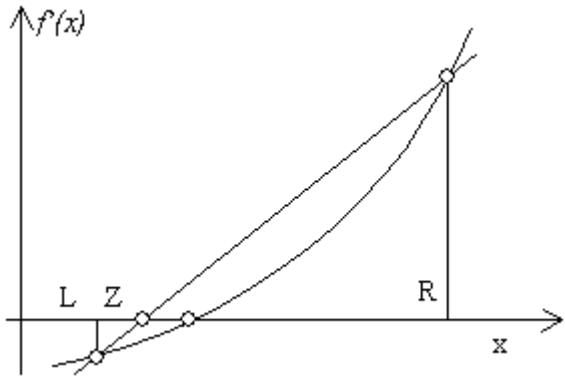
Алгоритм поиска минимума на (*a, b*).

1. $L = a; R = b; f'(a) < 0; f'(b) > 0$
2. Вычислить *Z; f'(Z)*
3. Если $|f'(Z)| \le \varepsilon$, то закончить поиск.
4. Исключить соответствующий интервал. Перейти к п. 2.

Метод секущих

Метод ориентирован на нахождение решения уравнения *f'(x)* =0 на заданном интервале (*a, b*). Метод похож на метод Ньютона, но строится не касательная, а секущая.

$$Z = R - \frac{f'(R)}{\left[f'(R) - f'(L)\right]/(R-L)}$$



В отличие от метода средней точки метод секущих использует информацию не только о знаке производной, но и о значениях в пробных точках.

Метод с использованием кубической аппроксимации

Функция $f(x)$ апроксимируется полиномом третьего порядка. Находится стационарная точка $\bar{x}$ этого полинома. Эта точка заключается в интервал (x_1, x_2) такой, что производные в x_1, x_2 имеют разные знаки.

Построим полином

$$q(x) = a_0 + a_1 \cdot (x - x_1) + a_2 \cdot (x - x_1) \cdot (x - x_2) + a_3 \cdot (x - x_1)^2 \cdot (x - x_2)$$

a_0, a_1, a_2, a_3 находятся так, чтобы значения функции и значения производной были: $q(x)$ и $q'(x)$, и совпадали бы с $f(x)$ и $f'(x)$ соответственно в точках x_1 и x_2.

$$f_1 = f(x_1) = q(x_1) = a_0 \Rightarrow a_0 = f_1$$

$$f_2 = f(x_2) = q(x_2) = a_0 + a_1 \cdot (x_2 - x_1) \Rightarrow a_1 = \frac{f_1 - f_2}{x_2 - x_1}$$

$$f_1' = q'(x_1) = a_1 + a_2 \cdot (x_1 - x_2) \Rightarrow a_2 = \left(f_1' - \frac{f_2 - f_1}{x_2 - x_1} \right) \cdot \frac{1}{x_1 - x_2} = \frac{f_1' - a_1}{x_1 - x_2}$$

$$f_2' = a_1 + a_2 \cdot (x_2 - x_1) + a_3 \cdot (x_2 - x_1)^2 \Rightarrow a_3 = \frac{f_2' - a_1 - a_2 \cdot (x_2 - x_1)}{(x_2 - x_1)^2}$$

$$\frac{dq}{dx} = 0$$

$$\bar{x} = \begin{cases} x_2, M < 0 \\ x_2 - M \cdot (x_2 - x_1), 0 \le M \le 1 \\ x_1, M > 1 \end{cases}$$

$$M = \frac{f_2' + \omega - z}{f_2' - f_1' + 2 \cdot \omega}; z = \frac{3 \cdot (f_1 - f_2)}{x_2 - x_1} + f_1' + f_2'; \omega = \begin{cases} (z^2 - f_1' \cdot f_2')^{1/2}, x_1 < x_2 \\ -(z^2 - f_1' \cdot f_2')^{1/2}, x_1 > x_2 \end{cases}$$

Формула для ω обеспечивает надлежащий выбор одного из двух корней квадратного уравнения.

Для значений *M*, заключённых в интервале от 0 до 1 формула для $\bar{x}$ гарантирует, что $\bar{x}$ всегда будет между x_1 и x_2.

Метод с использованием кубической аппроксимации

Алгоритм

1. Задать x_0 – начальное приближение, Δ -шаг поиска и ε_1, ε_2 погрешности по функции и аргументу.

2. Вычислить f' в x_0. Если $f'(x_0)$<0, то Δ>0 и $x_{k+1} = x_k + 2^k \cdot \Delta$, иначе Δ<0 и какая-нибудь своя формула для вычисления x_{k+1}. $k = 1,2,3...$

3. Вычислять $f'(x_{k+1})$ до тех пор, пока не получим x_m в которой $f'(x_{m-1}) \cdot f'(x_m) \le 0$.

$x_1 = x_{m-1}; x_2 = x_m$ Вычислить f_1, f_2, f_1', f_2'.

4. Вычислить $\bar{x}$ (см. выше).

5. Если $f(\bar{x}) < f(x_1)$, то перейти к п. 6 иначе $\bar{x} = \bar{x} - \frac{1}{2} \cdot (\bar{x} - x_1)$ и так вычислять, пока не выполнится условие $f(\bar{x}) < f(x_1)$.

6. Проверка на окончание $|f'(\bar{x})| \le \varepsilon_1$ и $\frac{|\bar{x} - x_1|}{|x_2 - x_1|} \le \varepsilon_1$. Если выполняется, то конец вычислений, иначе если $f'(\bar{x}) \cdot f'(x_1) < 0$,то $x_2 = \bar{x}$ или, если $f'(\bar{x}) \cdot f'(x_2) < 0$,то $x_1 = \bar{x}$ и перейти к п. 4.

Сравнение методов

Для быстрого получения предварительных результатов (начальной точки для применения других методов), а также, если требуется надёжная работа алгоритма при неизвестной заранее целевой функции, лучше использовать методы исключения интервалов.

Если требуется точное решение, необходимо воспользоваться градиентными методами (особенно кубической аппроксимацией).

С другой стороны, если требуются высокая точность, но функция не задана аналитически, лучше пользоваться методами точечного оценивания, так как при использовании градиентных методов накапливается погрешность при конечно-разностной аппроксимации производных.

Если сравнить методы с точки зрения поставленной задачи и вида функции, то при минимуме информации о функции следует использовать метод исключения интервалов.

Если функция квадратичная или близка к таковой, то следует использовать метод Пауэлла

Если функция дважды дифференцируемая, непрерывная и задана аналитически, то следует использовать градиентные методы.

Методы точечного оценивания при прочих равных условиях (интервалы, гладкая функция) быстрее методов исключения интервалов.

6. Методы многомерной безусловной оптимизации

6.1. Введение в методы многомерной оптимизации

6.1.1. Основные понятия и определения

В этой части рассматриваются фундаментальные понятия и конкретные методы, которые используют при поиске безусловных минимумов функций нескольких сменных. Изложенное основывается на материале разделов одномерных методов, поскольку одномерные методы играют очень важную роль при исследовании функции нескольких переменных.

На первый взгляд может показаться, что отличие между методами многомерного и одномерного поиска заключается лишь в том, что первые требуют большего объема вычислений, и, что в принципе методы, которые пригодны для функций одной переменной, можно применять и для функций многих переменных. Однако это не так, поскольку многомерное пространство качественно отличается от одномерного. Прежде всего с увеличением числа измерений уменьшается вероятность унимодальности целевой функции. Кроме

того, множество элементов, которые образовывают многомерное пространство, значительно мощнее множества элементов одномерного пространства. Объем вычислений, которые необходимы для сужения интервала неопределенности в многомерном пространстве, является степеневой функцией, показатель которой равен размерности пространства. Так, если в случае одномерного пространства для достижения f=0,1 требуется вычислить 19 значений целевой функции, то в случае двухмерного пространства это число составляет 361, трехмерного – 6859, четырехмерного – 130321, а пятимерного – 2476099! Поскольку при выборе оптимальной конструкции нередко нужно иметь дело с пятью и больше переменными, серьезность трудностей, обусловленных многомерностью, становится очевидной. Вначале рассмотрим *вопрос анализа (в статике)* с использованием положения линейной алгебры и дифференционного вычисления, а также условий, которые (в достаточно общих возможных случаях) позволяют идентифицировать точки оптимума. Такие условия используют для проверки избранных точек и дают возможность выяснить, есть ли эти точки точками минимума или седловыми точками. При этом задача выбора указанных точек остается за пределами этого анализа; основное внимание отдается решению вопроса о том, отвечают ли исследуемые точки решению многомерной задачи безусловной оптимизации, в которой необходимо минимизировать $f(x)$, $x \in R^N$, (1) когда ограничения отсутствуют на x , где x - вектор *управляемых переменных* размерности N, f — скалярная *целевая функция*. Обычно допускается, что x_i (для всех значений i= 1,2,3,...,N) могут принимать любые значения, хотя иногда в практических целях область значений x выбирается в виде дискретного множества. Кроме того, часто удобно допускать, что функция f и ее производные существуют и непрерывны везде, хотя мы знаем, что оптимумы могут достигаться в точках разрыва f или ее *градиента.*

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \ldots, \frac{\partial f}{\partial x_N} \right]^T . \qquad (1)$$

Нужно иметь в виду, что функция f может принимать минимальные значения в точке $\bar{x}$, в которой f или ∇f разрываются. Кроме того, этой точки ∇f может не существовать. Для того, чтобы построить систему конструктивных критериев оптимальности, необходимо (как можно меньше на первой стадии исследования) исключить из

рассмотрения подобные ситуации, которые очень усложняют анализ. В ряде случаев приходится ограничиваться лишь идентификацией *локальных* оптимумов, поскольку нелинейная целевая функция f не всегда имеет выпуклый характер, может быть мультимодальной. На рис. 1 изображены линии уравнения функции Химмельблау.

Функция Химмельблау:

$$f(x) = [x12 + x2 - 11]2 + [x1 + x22 - 7]2 \qquad (2)$$

Нетрудно видеть, что эта функция имеет четыре разных минимума.

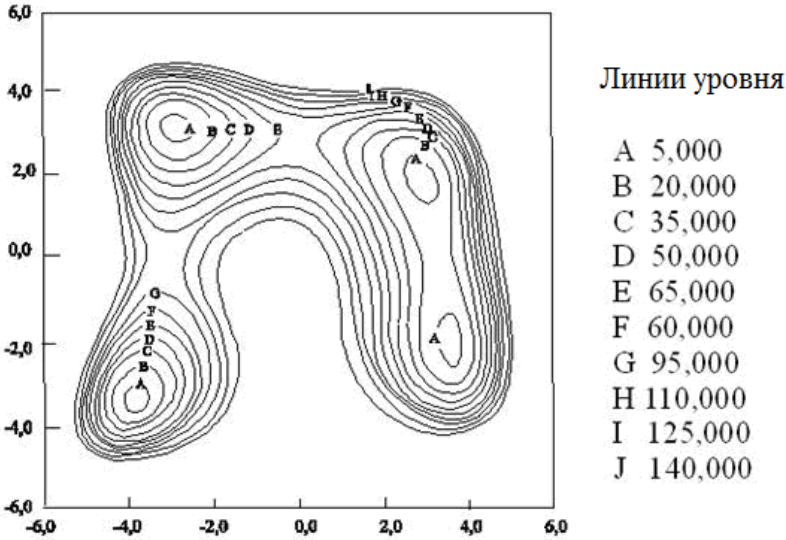


Рис. 1. Линии уровня мультимодальной функции

Дальше подойдем к *вопросу анализа* (в динамике), которое формулируется таким образом: если точка $x^{(0)}$ не удовлетворяет требованиям, которые определяются критериями оптимальности, то как получить (хорошее) новое приближение $x^{(1)}$ к решению x? Попытка дать ответ на этот вопрос приводит к необходимости рассмотрения ряда методов. Методы, которые рассматриваются, классифицируются, как мы знаем, в соответствии с тем, используется ли информация о производных исследуемой функции.

Задача безусловной оптимизации состоит в нахождении минимума или максимума функции в отсутствие каких-либо ограничений. Несмотря на то что большинство практических задач оптимизации содержит ограничения, изучение методов безусловной оптимизации важно с нескольких точек зрения. Многие алгоритмы решения задачи с ограничениями предполагают сведение ее к последовательности задач безусловной оптимизации. Другой класс методов основан на поиске подходящего направления и последующей минимизации вдоль этого направления. Обоснование методов безусловной оптимизации может быть естественным образом распространено на обоснование процедур решения задач с ограничениями.

Задача многомерной безусловной оптимизации формулируется в виде:

$$\min f(x), \qquad x \in X$$

где $x=\{x^{(1)}, x^{(2)},..., x^{(n)}\}$ – точка в n-мерном пространстве $X=R^n$, то есть целевая функция $f(x)=f(x^{(1)},...,f(x^{(n)})$ – функция n аргументов.

Численные методы отыскания минимума, как правило, заключаются в построении последовательности точек $\{x_k\}$, удовлетворяющих условию $f(x_0)>f(x_1)>...>f(x_n)>...$.

Методы построения таких последовательностей называются *методами спуска*. В этих методах точки последовательности $\{x_k\}$ вычисляются по формуле:

$$x_{k+1} = x_k + \alpha_k p_k,\ k=0,1,2,\dots ,$$

где p_k – направление спуска, α_k – длина шага в этом направлении.

Различные методы спуска отличаются друг от друга способами выбора направления спуска p_k и длины шага α_k вдоль этого направления. Алгоритмы безусловной минимизации, как мы уже говорили, принято делить на классы в зависимости от максимального порядка производных минимизируемой функции, вычисление которых предполагается. Так, методы, использующие только значения самой целевой функции, как мы знаем, относят к *методам нулевого порядка* (иногда их называют также *методами прямого поиска*); если требуется

вычисление первых производных минимизируемой функции, то мы имеем дело с *методами первого порядка*; если же дополнительно используются вторые производные, то это *методы второго порядка* и т. д.

6.2. Постановка задачи многомерной оптимизации.

Пусть скалярная функция *f(x)* определена на множестве $x \in X$, где множество X принадлежит некоторому метрическому пространству. Говорят, что на элементе (точке) $\overline{x} \in X$ функция $f(x)$ имеет *локальный минимум,* если существует такая конечная ε-окрестность точки $\overline{x}$, что для всех $x \in X$, удовлетворяющих $||x - \overline{x}|| < \varepsilon$, выполняется неравенство

$$f(\overline{x}) \leq f(x). \qquad (1)$$

Такая точка $\overline{x}$ называется точкой *локального минимума* Если указанное неравенство выполняется как строгое при $\overline{x} \neq x$, то говорят, что $\overline{x}$ — точка *строгого* локального минимума. Подобных локальных минимумов у функции $f(x)$ может быть много. Если выполняется

$$f(\overline{x}) = \inf_x f(x), \qquad (2)$$

то говорят, что $f(\overline{x})$ является *глобальным* (абсолютным) минимумом *f(x)* на заданном множестве *X,* т.е. $f(x) > /(\overline{x})$ для всех $x \in X$. Всякая точка глобального минимума является и точкой локального минимума, но не наоборот

Поиск хотя бы одной точки минимума $\overline{x}$ и минимума $f(\overline{x})$ называется *минимизацией* функции *f(x).* Нахождение точки максимума сводится к задаче минимизации при помощи замены *f(x)* на *—f(x)*

В дальнейшем будем предполагать, что множество X компактно (т. е. из каждого бесконечного и ограниченного его подмножества можно выделить сходящуюся последовательность) и замкнуто (т. е. предел любой сходящейся последовательности его элементов принадлежит этому множеству). В частности, если множество X само является пространством, то это пространство должно быть банаховым. Будем также предполагать, что функция *f(x)* непрерывна или, по крайней мере, кусочно-непрерывна

Если перечисленные требования не выполняются то поиск минимума затруднителен. Например, если *f(x)* не является кусочно-непрерывной

функцией, то единственный способ состоит в переборе всех ючек *x,* на которых определена *f(x).*

Заметим, что чем более жестким требованиям удовлетворяет *f(x)* (например, требованию существования непрерывных производных различного порядка), тем легче строить численные алгоритмы.

Если множество *X* является числовой осью, то задача минимизации состоит в поиске минимума функции одного вещественного переменного *(одномерная минимизация)*

Если же *X* есть *n*-мерное векторное пространство, то говорят о поиске минимума функции *n* переменных *(многомерная минимизация).*

В случае когда *X* — пространство функции *x*(*t*), то задачу (1) называют задачей на *минимум функционала*. Для решения этих задач используются методы вариационного исчисления.

Глобальный минимум может быть определен только тогда, когда вычислены все локальные минимумы: наименьший из них и есть глобальный. Поэтому в основном рассматривают задачу поиска локальных минимумов.

Из курса математического анализа известно, что в точке минимума удовлетворяется уравнение

$$\frac{\partial f}{\partial x}=0. \qquad (3)$$

Для задачи одномерной минимизации $\frac{\partial f}{\partial x}$ является обычной производной $\frac{df}{dx}$. Тогда уравнение (3) становится одним нелинейным (в общем случае) уравнением с одним неизвестным, которое может быть решено каким-либо из численных методов вычисления нулей нелинейных уравнений.

В случае многомерной минимизации уравнение (3) представляет собой систему нелинейных уравнении

$$\frac{\partial f}{\partial x_i} = 0, \quad 1 \le i \le n,$$

которая решается специальными методами. (Заметим, что при минимизации функционалов уравнение (3) оказывается дифференциальным или интегро-дифференциальным)

Однако на практике указанные уравнения являются сложными и для них известные итерационные методы решения нелинейных уравнений сходятся медленно или вообще не сходятся. Поэтому разработаны методы решения задачи (1) без приведения ее к виду (3).

Если множество *X* является пространством, то говорят о *безусловной минимизации* функции *f(x).*

Если же множество *X* принадлежит какому-либо пространству то задачу (1) называют задачей на *минимум в ограниченной обгасти.* Когда множество *X* выделяется из пространства системой ограничений типа равенств и/или неравенств то говорят об *условной минимизации* и задачу (1) называют задачей на *условный экстремум* (или задачей *математического программирования)*

Задачи математического программирования по виду функции *f(x)* разбиваются на следующие классы

— функция *f(x)* линейная и ограничения линейные: задача *линейного программирования;*

— функция *f(x)* не линейная и/или ограничения нелиненые (или ограничения нелинейные, а *f(x)* —линейная функция) задача *нелинейного программирования*

В свою очередь если ограничения линейны, то задача нелинейного программирования может быть разбита на следующие подклассы:

— *f(x)* дробно-рациональная функция: задача *дробно-рационального программирования,*

— *f(x)* выпуклая квадратичная функция: задача *квадратичного программирования.*

Все перечисченные задачи называют еще *задачами оптимизации.*

Отдельный класс оптимизационных задач представляют задачи *оптимального цправления.* Если в задачах оптимального управления процесс оптимизации можно представить в виде ряда последовательных этапов (шагов), то такие задачи называют *многошаговыми* задачами оптимизации (управления). Для их решения используются методы *динамического программирования,* которые применимы к непрерывной модели многошаговою процесса оптимизации, когда управления и векторы состояния могут непрерывно изменяться. Однако для многих экономических и производственных задач характерной является дискретная модель, когда величины, описывающие процесс, могут принимать только дискретный ряд значений. В таких задачах применяются дискретные методы динамического программирования

Оптимизационная задача называется *детерминированной* в том случае, если погрешностями вычисления или экспериментального определения значении функций *f(x)* можно пренебречь В противном случае оптимизационная задача называется *стохастической.* Для этого класса задач разработаны специальные методы

Необходимые и достаточные условия экстремума

1) Если целевая функция $f(x) \equiv f(x_1,..., x_n)$, $x \in R^n$, то минимизация $f(x)$ приводит к задаче:

$$x^* \in loc\min_{R^n} f(x); \qquad f(x^*) = \min_{h_\epsilon(x^*)} f(x). \tag{4}$$

Введем в рассмотрение градиент и гессиан функции f:

$$\vec{g}(x) = grad f = \vec{\nabla} f = \left\{ \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right\},$$

$$G(x) = \operatorname{hess} f(x) = \left\| \frac{\partial^2 f}{\partial x_i \partial x_j} \right\| \quad — \quad \text{симметричная матрица вторых производных } f(x_1, .., x_n).$$

Тогда разложение функции $f(\vec{x})$ в ряд Тейлора в окрестности точки при $\Delta \vec{x} = h\vec{p}$; $||\vec{p}|| = 1$; $h = \Delta x$ имеет вид:

$$f(\vec{x} + h\vec{p}) = f(\vec{x}) + df(\vec{x}) + \frac{1}{2} d^2 f(\vec{x}) + o\left(\|\delta x\|^2\right) =$$

$$= f(\vec{x}) + (grad f \cdot \vec{p})\, h + \frac{h^2}{2} \underbrace{(\operatorname{hess} f(\vec{x}) \cdot \vec{p}, \vec{p})}_{(G\vec{p}, \vec{p})} + o\left(h^2\right).$$

Величина $(\vec{g}(\vec{x}), \vec{p}) \equiv \dfrac{\partial f}{\partial p}$ — производная f в точке $\vec{x}$ по направлению $\vec{p}$; $(G\vec{p}, \vec{p}) = (\operatorname{hess} f\, \vec{p}, \vec{p}) \equiv K(\vec{p})$ — "кривизна" поверхности $u = f(\vec{x})$ в точке $\vec{x}$ по направлению $\vec{p}$.

2) Необходимые и достаточные условия минимума для дважды дифференцируемой функции $f(x_1,..., x_n)$. Напомним

необходимое условие экстремума	достаточное условие экстремума
$\|\vec{g}(x^*)\| = 0$	$\|\vec{g}(x^*)\| = 0$
$\operatorname{hess} f(x^*) \geq 0$	$\operatorname{hess} f(x^*) > 0$

(матрица $A > 0$, если $\forall x \neq 0 (Ax,x) > 0$ — положительно определенная квадратичная форма.

Заметим, что по существу многокритериальная задача отличается от обычной задачи оптимизации только наличием нескольких целевых функций вместо одной.

6.3. Критерий оптимальности для функции многих переменных

Разложим $f(\bar{x})$ в ряд Тейлора в окрестности некоторой точки $\vec{x}$ (здесь и далее через x обозначается вектор $\vec{x}$).

$$f(x) = f(\bar{x}) + \nabla f(x)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(\bar{x}) \Delta x + o_3(\Delta x),$$

где $\bar{x}$ - точка разложения в пространстве R^n;

$\Delta x = x - \bar{x}$ - величина изменения x;

$\nabla f(x)^T$ - n-мерный вектор столбец первых частных производных $f(x)$, вычисляемый в $\bar{x}$;

$\nabla^2 f(\bar{x}) = H(\bar{x})$ - матрица Гессе – симметричная матрица $n \times n$ вторых частных производных $f(x)$. Элемент матрицы Гессе, расположенный на пересечении i-ой строки и j-го столбца равен $\frac{\partial^2 f}{\partial x_i \partial x_j}$

$o_3(\Delta x)$ - члены порядка выше второго по Δx. Ими можно пренебречь.

Запишем изменение функции Δf в соответствии с изменением аргументом Δx:

$$\Delta f = f(x) - f(\bar{x}) = \nabla f(\bar{x})^T \Delta x + \frac{1}{2} \Delta x^T H(\bar{x}) \Delta x$$

Для всех точек в окрестности минимума $\Delta f \geq 0$.

Определение 1.

Точка $\bar{x}$ является **точкой глобального минимума**, если $\Delta f \geq 0$ выполняется для $\forall x \in R^n$. Обозначим её как x^{**}.

Определение 2.

Точка $\bar{x}$ является **точкой локального минимума**, если $\Delta f \geq 0$ выполняется в некоторой δ-окрестности точки $\bar{x}$. Обозначим её как x^*.

Если Δf больше 0, меньше 0 или равно нулю в зависимости от выбора δ -окрестности, то $\bar{x}$ - седловая точка.

Для того, чтобы знак Δf не менялся при произвольном варьировании Δx нужно, чтобы Δf ($\bar{x}$)=0, то есть $\bar{x}$ была бы стационарная точка функции Δf. Если это выполняется, то

$$\Delta f = \frac{1}{2} \cdot \Delta x^T \cdot H(\bar{x}) \cdot \Delta x$$

Теперь знак Δf определяется квадратичной формой:

$$f(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij} \cdot x_i \cdot y_j = x^T Q(x)$$

такая $f(x)$, что ;

$$Q(x) = \Delta x^T \cdot \nabla^2 f(\bar{x}) \cdot \Delta x$$

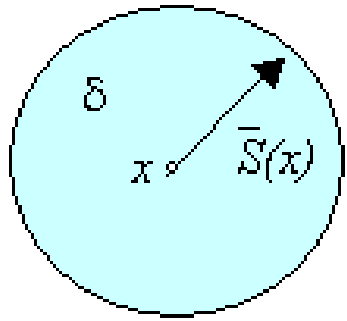
Стационарная точка $\bar{x}$ является точкой локального минимума, если матрица Гессе $H(\bar{x})$ положительно полуопределена, то есть $Q(x) \geq 0$ для всех *x*.

Стационарная точка $\bar{x}$ есть точка локального максимума, если $H(\bar{x})$ отрицательно полуопределена, то есть $Q(x) \leq 0$ для всех *x*.

Стационарная точка $\bar{x}$ есть седловая точка, если $H(\bar{x})$ не определена (главная диагональ и главные определители не равны нулю).

Анализ $\bar{x}$ можно провести в другом аспекте.

Рассмотрим стационарную точку $\bar{x}$ с δ-окрестностью и векторами, исходящими из $\bar{x}$.



При этом любую точку $\tilde{x}$ из этой δ-окрестности можно получить как

$$\tilde{x} = \bar{x} + \alpha \cdot \vec{S}(\bar{x}),$$

где α - коэффициент.

Путём соответствующего подбора α и $\vec{S}$ можно получить все точки δ -окрестности. Подставим это значение в Δf (вместо Δx подставим $\tilde{x} - \bar{x}$)

Получим

$$\Delta f = \frac{\alpha^2}{2} \cdot S^T \cdot \nabla^2 f(x) \cdot S$$

Теперь мы можем определить направление *S,* определить как направление спуска, подъёма или общего вида.

Спуск – если $\Delta^2 f(x)$ положительно полуопределена.

Подъём – если $\Delta^2 f(x)$ отрицательно полуопределена.

Общего вида – если $\Delta^2 f(x)$ не определена.

Необходимые и достаточные условия оптимальности.

Необходимые.

Для наличия в точке $\overline{x}$ локального минимума необходимо, чтобы

$$\Delta f(\overline{x}) = 0 \text{ и } \Delta^2 f(\overline{x}) \geq 0.$$

Достаточные.

Если выполняются необходимые условия оптимальности, то этого достаточно чтобы $\overline{x} = x^*$ (была локальным минимумом).

Если *f(x)* выпуклая, то x^* является и x^{**}.

Пример. Критерии оптимальности

Рассмотрим функцию

$$f(x) = 1x_1^2 + 4x_1x_2^3 - 10x_1x_2 + x_2^2,$$

линии уровня которой изображены на рис. 1.

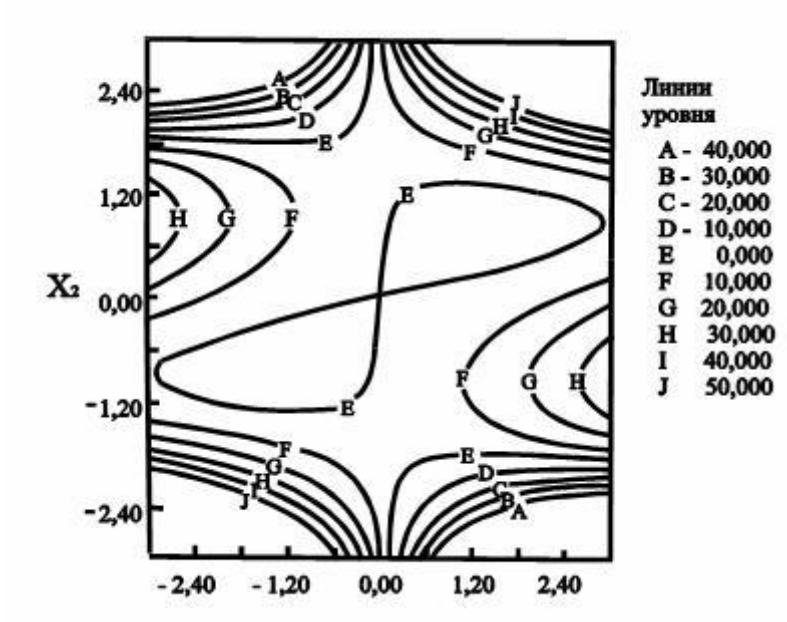


Рис. 1. Линии уровня нелинейной функции двух переменных

необходимо классифицировать точку $\overline{x}=[0,0]^{\mathrm{T}}$.

$$f(x) = 2(x_1)^2 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$$

Решение.

$$\frac{\partial f}{\partial x_1} = 4x_1 + 4x_2^3 - 10x_2,$$

$$\frac{\partial f}{\partial x_2} = 12x_1x_2^2 - 10x_1 + 2x_2$$

,

$$\nabla f(\bar{x}) = [0,0]^T.$$

Отсюда, точка $\bar{x}$ — стационарная.

$$\frac{\partial^2 f}{\partial x_1^2} = 4$$

,

$$\frac{\partial^2 f}{\partial x_2^2} = 24x_1x_2 + 2 = +2,$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = 12x_2^2 - 10 = -10.$$

Отсюда,

$$\nabla^2 f(\bar{x}) = H_f(\bar{x}) = \begin{bmatrix} +4 & -10 \\ -10 & +2 \end{bmatrix}.$$

Матрица $\nabla^2 f(\bar{x})$ является неопределенной, так как квадратичная форма $z^T H_f z$ принимает положительное значение при z=(0,1) и отрицательное значение при z=(1,1). Поэтому $\bar{x}$ представляет собой *седловую точку*, которая и изображена на рис. 1.

6.4 Квадратичная функция аргумента $\vec{x}$

Опираясь на тейлоровское разложение естественно в качестве удобной апроксимации гладкой функции *f(x)* в окрестности некоторой точки (в том числе и точки возможного экстремума) использовать квадратичную функцию $\Psi(\vec{x})$:

$$\Psi(\vec{x}) = \frac{1}{2}(A\vec{x},\vec{x}) + \left(\vec{b},\vec{x}\right) + c,$$

где A - симметричная, невырожденная матрица $A = A^T$, $detA \neq 0$. Установим вид градиента $\vec{\nabla}\ \Psi$ и гессиана G = hess Ψ функции $\Psi(\vec{x})$:

$$\Psi(\vec{x}+h\vec{p}) = \frac{1}{2}(A(\vec{x}+h\vec{p}),\vec{x}+h\vec{p}) + \left(\vec{b},\vec{x}+h\vec{p}\right) + c =$$

$$= \left\{\frac{1}{2}(Ax,x) + (b,x) + c\right\} + h\left\{\frac{1}{2}(A\vec{x},\vec{p}) + \frac{1}{2}(A\vec{p},\vec{x}) + \left(\vec{b},\vec{p}\right)\right\} + \frac{h^2}{2}(A\vec{p},\vec{p}) =$$

$$= \Psi(\vec{x}) + h\left(A\vec{x}+\vec{b},\vec{p}\right) + \frac{h^2}{2}(A\vec{p},\vec{p}) \quad \text{, т.е.}$$

$$grad\Psi = A\vec{x} + \vec{b}; \quad \text{hess}\,\Psi(\vec{x}) = A - const. \tag{1}$$

Стационарная дочка для $\Psi(\vec{x})$ удовлетворяет условию:

$$grad\Psi(x^*) = 0 \quad \Leftrightarrow \quad Ax^* + b = 0 \quad \Leftrightarrow \quad Ax^* = -b \text{ -- СЛАУ} \tag{2}$$

Решение системы (2) зависит от ранта матрицы A. В случае совместной системы решение может быть и не единственным.

В окрестности стационарной точки $\vec{x}^*$:

$$\Psi(\vec{x}) = \Psi(\vec{x}^* + h\vec{p}) = \Psi(\vec{x}^*) + \frac{h^2}{2}(A\vec{p},\vec{p}).$$

И поведение квадратичной функции определяется только свойствами матрицы A. Если A — симметричная невырожденная матрица, то сущеествует ортонормированный базис (ОНБ) из собственных векторов матрицы A. Пусть $\{\lambda_i,\ \vec{x}_i\}$ собственные значения и собственные векторы матрицы A. $\{\vec{x}_i\}$ - ОНБ. Разложим направление $\vec{p}$ по базису $\{\vec{x}_i\}$ — $\vec{p} = \sum_{i=1}^{n} \alpha_i \vec{x}_i$, тогда

$$\Psi(\vec{x}^* + h\vec{p}) = \Psi(\vec{x}^*) + \frac{h^2}{2}\left(\sum_i \alpha_i\lambda_i\vec{x}_i, \sum_j \alpha_j\vec{x}_j\right) = \Psi(\vec{x}^*) + \frac{h^2}{2}\sum_i \lambda_i\alpha_i^2.$$

Характер изменения $\Psi(\vec{x})$ при движении вдоль $\vec{x}_k$ полностью определяется шагом λ_k. Если $A>0$, то все $\lambda_i > 0$ и x^* точка минимума.

6.5 Рельеф поверхности целевой функции *f(x)*. Поверхности уровня

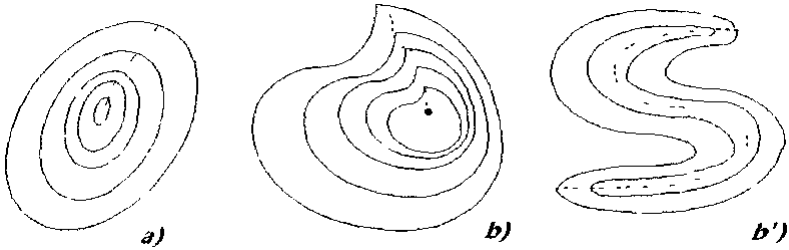
Трудности и проблемы задачи минимизации, характерные для общего случая, столь же ясно проявляются и при рассмотрении минимизации функции двух переменных *f(x, y)*. Геометрию поверхности $z = f(x, y)$ представляют с помощью "плоских" линий уровня

$$L_0 = \{(x, y) : \quad f(x, y) = f(x_0, y_0) = f_0 = const\},$$

являющихся проекциями на плоскость *OXY* сечения поверхности $z = f(x, y)$ плоскостью $z_0 = f_0$.

Выделяют три основных типа рельефа поверхности.

а) *котловинный* - линии уровня похожи на концентричекские эллипсы с главными осями параллельными собственным векторам ness *f(x, y)*. В малой окрестности невырожденного минимума *(x*, y*)* hess $f(x, y) > 0$ и рельеф поверхности именно *котловиниый*.



б) *овражный* - если линия уровня кусочно-гладкая, то геометрическое место точек (ГМТ) излома по всем линиям уровня называют *истинным оврагом* (если угол излома направлен в сторону возрастания функции) или *истинным гребнем* (если угол излома направлен в сторону убывания функции).

Однако чаще приходится иметь дело с *разрешимыми* оврагами и гребнями (ГМТ наибольшей кривизны - рисунок ***b'))***. Например, одна из стандартных тестовых функций многомерной минимизации (функция Розенброка)

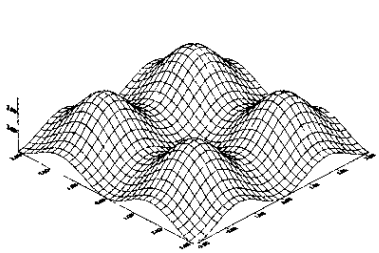
$$f(x,y) = 100\,(y - x^2)^2 + (1 - x)^2$$

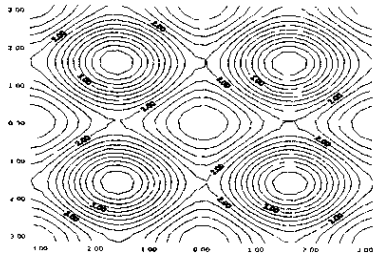
обладает пологим серповидным ("банановидным") ущельем и имеет абсолютный минимум в точке x^* (1.1).

в) *неупорядоченный тип рельефа* — характеризуется наличием многих максимумов, минимумов и седловин. Приведем в качестве примера функцию

$$f(x,y) = (1 + \sin^2 x)\,(1 + \sin^2 y)$$

с достаточно неупорядоченным рельефом:





Если рассматривать дифференцируемую в каждой точке функцию $f(\vec{x})$, то её производная по направлению $\vec{p}$

$$\frac{\partial f}{\partial p} = (\text{grad } f,\ \vec{p}\,) = \vec{g} \cdot \vec{p}$$

обладает характерными свойствами на поверхности уровня

- производная по направлению радиента - максимальна;
- вдоль линии уровня $\dfrac{\partial f}{\partial p}$ - равна нулю и градиент $\vec{g}$ перпендикулярен линии уровня в каждой точке.

6.6. Введение в методы безусловной минимизации функций многих переменных

6.6.1. Вводные понятия

Пусть заданы множество *X,* принадлежащее некоторому метрическому пространству, и скалярная функция *f(x),* определенная на этом множестве *X.* Напомним, что задача на минимум функции *f(x)*записывается в виде

$$f(x) \to \min, \quad x \in X. \tag{1}$$

В этой записи функцию $f(x)$ называют *целевой функцией, X — допустимым множеством,* любой элемент $x \in X$ - *допустимой точкой* задачи (1)

Поиск максимума функции $f(x)$ на X эквивалентен задаче вычисления минимума функции - $f(x)$ и записывается в виде

$$-f(x) \to \min, \quad x \in X. \tag{2}$$

Точки минимума и максимума называют *точками экстремума,* а задачи (1) и (2) называются *экстремальными задачами.* Вопрос о существовании решений этих задач, как мы занем, базируется на теореме Вейерштрасса:

Пусть X — компакт в евклидовом n-мерном пространстве $\mathbf{R}^n$ (т.е. X — замкнутое ограниченное множество), а $f(x)$ — непрерывная функция на X. Тогда существует точка глобального минимума $f(x)$ на X

Теорема Вейерштрасса имеет важное следствие: если функция $f(x)$ непрерывна на $\mathbf{R}^n$ и $\lim\limits_{\|x\|_2 \to \infty} f(x) \to +\infty$, то $f(x)$ достигает своего глобального минимума на любом замкнутом подмножестве в $\mathbf{R}^n$.

Мы будем иметь дело с конечномерными задачами, когда допустимое множество X совпадает с $\mathbf{R}^n$, т.е. когда задача (1) является задачей безусловной минимизации функций многих переменных. Дадим ряд определений

Градиентом функции $f(x)$ называется векгор первых частных производных

$$\operatorname{grad} f = \operatorname{grad} f(\boldsymbol{x}) = \boldsymbol{f}'(\boldsymbol{x}) = \left(\frac{\partial f}{\partial \boldsymbol{x}_1}, \ldots, \frac{\partial f}{\partial x_n} \right).$$

Антиградиентом функции $f(x)$ называется вектор первых частных производных, взятых со знаком минус, т.е — grad f.

Матрицей Гессе функции $f(x)$называется матрица вторых частных производных

$$f''(\boldsymbol{x}) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{i,j=1,\ \ n}.$$

Ниже будем предполагать, что смешанные производные функции $f(x)$ второго порядка непрерывны; следовательно, имеем

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i} \left(\frac{\partial f}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left(\frac{\partial f}{\partial x_i} \right) = \frac{\partial^2 f}{\partial x_j \partial x_i},$$

а это означает, что матрица Гессе является симметричной.

Функция *f(x)* называется *дифференцируемой* в точке *x**, если она имеет в этой точке полный дифференциал, т.е. для полного приращения *f(x)* в точке *x** имеет место равенство

$$\Delta f = f(\boldsymbol{x}^* + \Delta\boldsymbol{x}) - f(\boldsymbol{x}^*) = (f'(\boldsymbol{x}).\Delta\boldsymbol{x}) + o(\|\Delta\boldsymbol{x}\|_2).$$

Здесь и далее под (•,•) подразумевается скалярное произведение векторов. Заметим, что если все частные производные непрерывны, то функция дифференцируема. Разложение в ряд Тейлора функции *f(x)* в точке *x** имеет вид

$$f(\boldsymbol{x}^* + \Delta\boldsymbol{x}) = f(\boldsymbol{x}^*) + (f'(\boldsymbol{x}), \Delta\boldsymbol{x}) + \frac{1}{2}(f''(\boldsymbol{x})\Delta\boldsymbol{x}\ \Delta\boldsymbol{x}) + o\left(\|\Delta\boldsymbol{x}\|_2^2\right)$$

В приведенной записи удержаны три члена разложения. Полезны следующие частные случаи этого разложения.

а) Формула Лагранжа:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + (f'(\boldsymbol{x}^* + \alpha\boldsymbol{h}).\boldsymbol{h}), \quad \boldsymbol{h} = \boldsymbol{x} - \boldsymbol{x}^*, \quad 0 < \alpha < 1.$$

б) Формула Ньютона-Лейбница:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + \int_0^1 (f'(\boldsymbol{x}^* + \alpha\boldsymbol{h}), \boldsymbol{h})\, d\alpha, \quad \boldsymbol{h} = \boldsymbol{x} - \boldsymbol{x}^*, \quad 0 < \alpha < 1.$$

в) Формула Тейлора с остаточным членом в форме Лагранжа:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + (f'(\boldsymbol{x}^*), \boldsymbol{h}) + \frac{1}{2}(f''(\boldsymbol{x}^* + \alpha\boldsymbol{h})\boldsymbol{h}, \boldsymbol{h}),$$

$$\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{x}^*, \quad 0 < \alpha < 1.$$

Частную производную функции *f(x)* по x_i в точке *x** можно представить в виде

$$\frac{\partial f}{\partial x_i}(\boldsymbol{x}^*) = \lim_{\alpha \to 0} \frac{f(\boldsymbol{x}^* + \alpha\, \boldsymbol{e}_i) - f(\boldsymbol{x}^*)}{\alpha}$$

где e_i — вектор-столбец, у которого *i*-я координата равна единице, а остальные равны нулю.

Функция *f(x)* называется *дифференцируемой* в точке *x*,* если градиент *f'(x*)* существует и при всех достаточно малых $\boldsymbol{h} \in \mathbf{R}^n$ справедливо представление

$$f(\boldsymbol{x}^* + \boldsymbol{h}) = f(\boldsymbol{x}^*) + (f'(\boldsymbol{x}^*), \boldsymbol{h}) + o(\|\boldsymbol{h}\|_2)$$

Функция *f(x)* называется *дважды дифференцируемой* в точке *x**, если матрица Гессе *f''(x*)* существует и симметрична и при всех достаточно малых $\boldsymbol{h} \in \mathbf{R}^n$ справедливо представление

$$f(\boldsymbol{x}^* + \boldsymbol{h}) = f(\boldsymbol{x}^*) + (f'(\boldsymbol{x}^*), \boldsymbol{h}) + \frac{1}{2}(f''(\boldsymbol{x}^*)\boldsymbol{h}, \boldsymbol{h}) + o\left(\|\boldsymbol{h}\|_2^2\right)$$

Величина

$$f'(\boldsymbol{x}^*, \boldsymbol{h}) = \lim_{\alpha \to 0+} \frac{f(\boldsymbol{x}^* + \alpha \boldsymbol{h}) - f(\boldsymbol{x}^*)}{\alpha}, \quad \|\boldsymbol{h}\|_2 = 1$$

называется производной функции *f(x)* в точке *x** по *направлению* вектора ***h.*** Функция *f(x)* называется дифференцируемой в точке *x** по направлению вектора ***h***, если величина *f'(x*,**h**)* существует и конечна. Если функция *f(x)* дифференцируема в точке *x**, то она дифференцируема в точке *x** но направлению любого вектора ***h***, причем выполняется равенство

$$f'(x^*, \boldsymbol{h}) = (f'(x^*), \boldsymbol{h}).$$

Условие, которому *необтодимо* должна удовлетворять точка локального минимума (необходимое условие локальной оптимальности), дается следующей теоремой

Теорема 1. *Пусть функция f(x) дифференцируема в точке* $\bar{x} \in \mathbf{R}^n$. *Если x — точка локального минимума, то*

$$\text{grad } f(\bar{x}) = f'(\bar{x}) = 0.$$

Доказательство. Если $\bar{x}$ точка локального минимума, то по определению существует такая ε-окрестность этой точки (ε -шар), что

$$f(\bar{x}) \le f(\bar{x} + \alpha \boldsymbol{h}),$$

где ***h*** — любой вектор из $\mathbf{R}^n$ и $\| (\bar{x} + \alpha \boldsymbol{h}) - \bar{x} \|_2 \le \varepsilon$, т.е. выполняется неравенство $\| \alpha \boldsymbol{h} \|_2 \le \varepsilon$. Поскольку *f(x)* дифференцируема, то

$0 \le f(\bar{x} + \alpha \boldsymbol{h}) - f(\bar{x}) = (f'(\bar{x}), \alpha \boldsymbol{h}) + o(\| \alpha \boldsymbol{h} \|_2)$.

Разделим обе части неравенства на α*:*

$$(f'(\bar{x}), \boldsymbol{h}) + \frac{o(\|\alpha \boldsymbol{h}\|_2)}{\alpha} \ge 0$$

и перейдем к пределу при $\alpha \to 0$:

$$(f'(\bar{x}), \boldsymbol{h}) \ge 0.$$

Это неравенство верно при любых ***h***, в том числе и для вектора $\boldsymbol{h} = -f'(\bar{x})$, для которого имеем

$$-f'(\bar{x}), f'(\bar{x})) = -\|f'(\bar{x})\|_2^2 \ge 0.$$

Следовательно, $\|f'(\bar{x})\|_2 = 0$ т. е. $f'(\bar{x}) = 0$. Теорема доказана

Определение Точка $\bar{x}$, для которой $f'(\bar{x}) = 0$, называется *стационарной* точкой функции *f(x).*

Стационарная точка не обязательно является точкой минимума, поскольку $f'(\bar{x})=0$ — только необходимое, но не достаточное условие оптимальности. Приведем пример, когда стационарная точка не является точкой минимума. Рассмотрим функцию

$$f(\boldsymbol{x}) = x_1^3 + x_2^3 - 3x_1x_2, \quad \boldsymbol{x} \in \boldsymbol{R}^n$$

Градиент этой функции имеет вид

$$f'(\boldsymbol{x}) = \left(3x_1^2 - 3x_2, 3x_2^2 - 3x_1\right).$$

Выпишем решения уравнения $f'(x) = 0$:

$$\bar{\boldsymbol{x}}^{(1)} = \left(\bar{x}_1^{(1)}, \bar{x}_2^{(1)}\right) = (0.0), \quad \bar{\boldsymbol{x}}^{(2)} = \left(\bar{x}_1^{(2)}, \bar{x}_2^{(2)}\right) = (1,1).$$

Точка $\bar{x}^{(1)}$ является стационарной, но не является точкой минимума, т.е. нет такого ε-шара с центром $\bar{x}^{(1)}$, для которого при всех x:$\|x - \bar{x}^{(1)}\| < \varepsilon$ выполнено неравенство $f(x) \geq f(\bar{x}^{(1)})$. Действительно, для любой точки

$$\boldsymbol{x} = \boldsymbol{x}^{(1)} + \boldsymbol{\varepsilon} \text{ (где } \boldsymbol{\varepsilon} = (\varepsilon, \varepsilon) \text{ и } 0 < \varepsilon < 3/2)$$

имеем

$$\begin{aligned} f(\boldsymbol{x}) &= \left(\bar{x}_1^{(1)} + \varepsilon\right)^3 + \left(x_2^{(1)} + \varepsilon\right)^3 - 3\left(\bar{x}_1^{(1)} + \varepsilon\right)\left(\bar{x}_2^{(1)} + \varepsilon\right) = \\ &= \varepsilon^3 + \varepsilon^3 - 3\varepsilon\varepsilon = 2\varepsilon^3 - 3\varepsilon^2 = \varepsilon^2(2\varepsilon - 3) < 0 = f(\bar{\boldsymbol{x}}^{(1)}). \end{aligned}$$

Отметим, что каждая точка минимума является стационарной.

Теорему 1 называют необходимым условием оптимальности *первого порядка.* Для выявления посторонних стационарных точек может использоваться необходимое условие оптимальности *второго порядка:*

Теорема 2. *Пусть функция f(x) дважды дифференцируема в точке* $\bar{x} \in \mathbf{R}^n$. *Если* $\bar{x}$ *— точка локального минимума, то матрица Гессе* $f''(\bar{x})$ *неотрицательно определена, т.е.*

$$(f''(\bar{x})\boldsymbol{h},\boldsymbol{h}) \geq 0$$

при всех $\boldsymbol{h} \in \mathbf{R}^n$.

Доказательство. Поскольку $\bar{x}$ — точка локального минимума, то

$$f(\bar{x}) \leq f(\bar{x} + \alpha\boldsymbol{h})$$

для достаточно малых α. По определению дважды дифференцируемой функции имеем

$$\begin{aligned} 0 \leq f(\bar{\boldsymbol{x}} + \alpha\boldsymbol{h}) - f(\bar{\boldsymbol{x}}) &= (f'(\bar{\boldsymbol{x}}), \alpha\boldsymbol{h}) + \frac{1}{2}(f''(\bar{\boldsymbol{x}})\alpha\boldsymbol{h}, \alpha\boldsymbol{h}) + \\ &+ o\left(\|\alpha\boldsymbol{h}\|_2^2\right). \end{aligned}$$

Поскольку $f'(\bar{x}) = 0$, то

$$0 \le f(\boldsymbol{x} + \alpha \boldsymbol{h}) - f(\boldsymbol{x}) = \frac{1}{2}\alpha^2 (f''(\bar{\boldsymbol{x}})\boldsymbol{h}, \boldsymbol{h}) + o(\alpha^2)$$

при всех достаточно малых α. Поделим обе части последнего неравенства на α^2 и перейдем к пределу при $\alpha \to 0$:

$$\frac{1}{2}(f''(\bar{\boldsymbol{x}})\boldsymbol{h}, \boldsymbol{h}) \ge 0.$$

Следовательно, приходим к заключению: если $\bar{x}$ — точка локального минимума, то матрица $f''(\bar{x})$ неотрицательно определена. Теорема доказана.

Теперь сформулируем *достаточное* условие локальной оптимальности.

Теорема 3. *Пусть функция f(x) дважды дифференцируема в точке* $\bar{x} \in \mathbf{R}^n$ *и пусть* $f'(\bar{x}) = 0$, *а матраца* $f''(\bar{x})$ *положительно определена, т.е.*

$$(f''(\bar{x})\boldsymbol{h},\boldsymbol{h}) > 0$$

при всех $\boldsymbol{h} \in \mathbf{R}^n$, $\boldsymbol{h} \neq 0$. *Тогда* $\bar{x}$ *— точка строгого локального минимума.*

Доказательство. Наши рассуждения будем проводить от противного. Пусть в $\mathbf{R}^n$ существует такая последовательность $\{x^k\}$, что

$$x^k \neq \bar{x}, \quad x^k \to \bar{x}, \quad f(x^k) \le f(\bar{x}).$$

Представим x^k в виде

$$\boldsymbol{x}^k = \bar{\boldsymbol{x}} + \alpha_k \boldsymbol{h}^k, \quad \alpha_k = \|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2, \quad \boldsymbol{h}^k = \frac{\boldsymbol{x}^k - \bar{\boldsymbol{x}}}{\alpha_k}.$$

Поскольку $\|\boldsymbol{h}^k\|^2 = 1$ (т. е. множество векторов $\boldsymbol{h}^k$ ограничено), то из последовательности $\boldsymbol{h}^k$ можно выделить сходящуюся подпоследовательность. Для определенности будем считать, что это сама последовательность $\boldsymbol{h}^k$, т.е. $\boldsymbol{h}^k \to \boldsymbol{h} \neq 0$. Из определения дважды дифференцируемой функции имеем

$$\begin{aligned} 0 \;&\ge\; f(\boldsymbol{x}^k) - f(\boldsymbol{x}) = f(\boldsymbol{x} + \alpha_k \boldsymbol{h}^k) - f(\bar{\boldsymbol{x}}) = \\ &= \; (f'(\bar{\boldsymbol{x}}), \alpha_k \boldsymbol{h}^k) + \frac{1}{2}(f''(\boldsymbol{x})\alpha_k \boldsymbol{h}^k, \alpha_k \boldsymbol{h}^k) + o\left(\left\|\alpha_k \boldsymbol{h}^k\right\|_2^2\right) = \\ &= \; \frac{1}{2}\alpha_k^2 (f''(\boldsymbol{x})\boldsymbol{h}^k, \boldsymbol{h}^k) + o(\alpha_k^2). \end{aligned}$$

Поделим обе части этого неравенства на α^2_k и перейдем к пределу при $\alpha \to 0$:

$$0 \ge (f''(\bar{x})\boldsymbol{h}^k, \boldsymbol{h}^k).$$

Полученное неравенство противоречит условию теоремы. Следовательно, $\overline{x}$ — точка строгого локального минимума. Теорема доказана.

Вернемся к анализу стационарных точек рассмотренного выше примера. Предварительно напомним формулировку *критерия Сильвестра:* симметричная матрица положительно (неотрицательно) определена тогда и только тогда, когда все ее ведущие миноры положительны (неотрицательны).

Матрица Гессе для функции $f(x) = x^3_1 + x^3_2 — 3x_1 x_2$ имеет вид

$$f''(\boldsymbol{x}) = \begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}.$$

Для ранее найденных стационарных точек $\overline{x}^{(1)} = (0,0)$ и $\overline{x}^{(2)} = (1, 1)$ имеем

$$f''(\bar{\boldsymbol{x}}^{(1)}) = \begin{pmatrix} 0 & -3 \\ -3 & 0 \end{pmatrix}. \quad f''(\boldsymbol{x}^{(2)}) = \begin{pmatrix} 6 & -3 \\ -3 & 6 \end{pmatrix}.$$

По критерию Сильвестра матрица $f''(\overline{x}^{(1)})$ не является неотрицательно определенной, т.е. необходимое условие оптимальности второго порядка не выполняется. Таким образом, еще раз показано, что точка $\overline{x}^{(1)}$ не является точкой минимума.

Что же касается матрицы $f''(\overline{x}^{(2)})$, то по критерию Сильвестра эта матрица положительно определена. Это означает, что $\overline{x}^{(2)}$ — точка минимума по достаточному условию оптимальности.

6.6.2. Общие сведения о численных методах безусловной минимизации

Напомним, что методы безусловной минимизации, использующие информацию только о значениях минимизируемой функции, называются методами *нулевого порядка.* Если при этом используются значения первых и вторых производных минимизируемой функции, то такие методы называют методами *первого* и *второго* порядков соответственно.

Алгоритм минимизации называют *последовательным,* если каждое следующее приближение к точке минимума строится через предыдущие приближения.

Для записи методов минимизации используются соотношения вида

$$x^{k+1} = x^k + \alpha_k \boldsymbol{h}^k, \quad \alpha_k \in \mathbf{R}, \quad k=0,1,2.....$$

Каждый конкретный алгоритм минимизации определяется заданием *начальной* точки x^0 (начального приближения к точке минимума),

правилами выбора векторов $\boldsymbol{h}^k$ и чисел α_k, а также *критериями окончания счета.* Вектор $\boldsymbol{h}^k$ задает *направление* (k+1)-го шага алгоритма, а коэффициент α_k — *длину* этого шага.

Название метода минимизации определяется способом выбора векторов $\boldsymbol{h}^k$, в то время как модификации метота связаны с различными способами выбора α_k. Термины *шаг метода* и *итерация метода* эквивалентны

Если метод гарантирует получение точки минимума за *конечное* число шагов, то его называют *конечношаговым.* Такие методы удается построить для специальных типов задач (например, для задач линейного и квадратичного программирования). Если же достижение решения гарантируется лишь в пределе, то соответствующий метод называется *беконечношаговым.*

Говорят, что метод

$$x^{k+1} = x^k + \alpha_k \boldsymbol{h}^k$$

сходится если $x^k \to \overline{x}$ при $k \to \infty$, где $\overline{x}$ - точка минимума функции $f(x)$. Если $f(x^k) \to f(\overline{x})$, то говорят что метод сходится *по функции,* а последовательность $\{ x^k \}$ называют *минимизирующей.* Отметим, что минимизирующая последовательность может не сходиться к точке минимума.

Говорят, что вектор $\boldsymbol{h}$ задает *направление убывания* функции $f(x)$ в точке x, если $f(x + \alpha \boldsymbol{h}) < f(x)$при всех достаточно малых $\alpha > 0$. Сам вектор $\boldsymbol{h}$ называют *направлением убывания.* Если при всех достаточно малых $\alpha > 0$ выполняется $f(x + \alpha \boldsymbol{h}) > f(x)$, то вектор $\boldsymbol{h}$ называют *направлением возрастания*

Сформулируем достаточный и необходимый признак направления убывания.

Теорема 4. *Пусть функция $f(x)$ дифференцируема в точке $x \in \mathbf{R}^n$. Если вектор $\boldsymbol{h}$ удовлетворяет условию*

$$(f'(x), \boldsymbol{h}) < 0,$$

то $\boldsymbol{h}$ — направление убывания функции $f(x)$ в точке x. Если $\boldsymbol{h}$ - направление убывания функции $f(x)$ в точке x, то выпопняется неравенство

$$(f'(x), \boldsymbol{h}) \le 0.$$

Доказательство. Пусть $(f'(x), \boldsymbol{h}) < 0$. По определению дифференцируемой функции можно записать, что

$$\begin{aligned} f(\boldsymbol{x} + \alpha \boldsymbol{h}) - f(\boldsymbol{x}) &= (f'(\boldsymbol{x}), \alpha \boldsymbol{h}) + o(\|\alpha \boldsymbol{h}\|_2) = \\ &= \alpha \left((f'(\boldsymbol{x}), \boldsymbol{h}) + \frac{o(\alpha)}{\alpha} \right) \end{aligned} \qquad (3)$$

Поскольку $(f'(x),\boldsymbol{h}) < 0$ по предположению теоремы, то начиная с некоторого достаточно малого значения α имеем неравенство $(f'(x),\boldsymbol{h}) + o(\alpha)/\alpha < 0$, т.е. $f(x + \alpha\boldsymbol{h}) - f(x) < 0$. Следовательно, $\boldsymbol{h}$ — направление убывания.

Вторую часть утверждения теоремы докажем от противного. Пусть $\boldsymbol{h}$ задает направление убывания в точке x. однако $(f'(x),\boldsymbol{h}) > 0$. Тогда из (3) следует, что в действительности $\boldsymbol{h}$ является направлением возрастания. Полученное противоречие показывает, что должно быть выполнено неравенство $(f'(x), \boldsymbol{h}) \leq 0$, если $\boldsymbol{h}$ — направление убывания. Теорема доказана

Метод $x^{k+1} = x^k + \alpha_k\boldsymbol{h}^k$ называют *методой спуска,* если вектор $\boldsymbol{h}$ задает направление убывания функции $f(x)$ в точке x^k, а число α_k положительно и таково, что $f(x^{k+1}) < f(x^k)$.

Простейшим примером метода спуска является *градиентный метод,* в котором $\boldsymbol{h}^k = -f'(x^k)$. Действительно, предположим, что $f'(x^k) \neq 0$. Тогда вектор $-f'(x^k)$ есть направление убывания в силу достаточного признака поскольку

$$(f'(x^k), -f'(x^k)) = -\|f'(x^k)\|_2^2 < 0.$$

Напомним что вектор $\boldsymbol{h}^k = -f'(x^k)$ называют *антиградиентом.*

Теперь рассмотрим два подхода к выбору шага α_k по направлению убывания минимизируемой функции

Первый из них называют *дроблением шага.* Пусть $\boldsymbol{h}^k$ - направление убывания. Выберем некоторые постоянные $\beta>0$ и $0<\lambda<1$. Полагаем вначале $\alpha=\beta$ и проверим условие

$$f(x^k + \alpha\boldsymbol{h}^k) < f(x^k). \qquad (4)$$

Если это условие не выполняется, то осуществляем дробление шага $\alpha = \lambda\beta$ и вновь проверяем условие (4). Процесс дробление шага продолжаем до тех пор, пока условие (4) не окажется выполненным Первое α, при котором это условие выполнено, принимается за α_k. Описанный процесс не может быть бесконечным, поскольку $\boldsymbol{h}^k$ — направление убывания.

Если при $\alpha = \beta$ условие (4) выполнено, то полезно увеличить шаг: $\alpha = \mu\beta$, $\mu > 1$. Если будет выполнено

$$f(x^k + \alpha\boldsymbol{h}^k) < f(x^k + \beta\boldsymbol{h}^k),$$

то текущее значение α опять умножается на μ и так до тех пор, пока значение функции не перестанет уменьшаться. Последнее α, при котором произошло уменьшение, берется в качестве α_k.

На практике часто выбирают $\lambda = 1/2$ и $\mu = 2$. Величину β относят к *параметрам управления* процессом минимизации и подбирают в зависимости от характера поведения минимизируемой функции вблизи x^k. Полезно также ограничить сверху увеличение шага.

Согласно второму подходу выбор длины шага по направлению убывания осуществляется из условия минимизации функции вдоль этого направления.

$$f(\boldsymbol{x}^k + \alpha_k \boldsymbol{h}^k) = \min_{\alpha} f(\boldsymbol{x}^k + \alpha \boldsymbol{h}^k) = \min_{\alpha} f(\alpha).$$

Для методов спуска минимум берется по $\alpha > 0$. Такой способ выбора α_k является наилучшим, поскольку при нем не только выполняется условие (4), но и обеспечивается достижение наименьшего значения *f(x)* вдоль заданного направления убывания. Недостаток данного подхода состоит в том, что на каждом шаге требуется решение одномерной задачи минимизации, что приводит к дополнительному увеличению объема вычислений.

6.6.3. Скорость сходимости. Критерии окончания счета

Эффективность применяемого метода минимизации характеризуют при помощи понятия *скорости сходимости.*

Говорят, что метод сходится к точке минимума $\bar{x}$ *линейно* (с линейной скоростью, или со скоростью геометрической прогрессии), если существуют такие постоянные $q \in (0, 1)$ и k_0, что

$$\left\| \boldsymbol{x}^{k+1} - \bar{\boldsymbol{x}} \right\| \le q \left\| \boldsymbol{x}^k - \boldsymbol{x} \right\| \quad \text{при} \quad k \ge k_0$$

Скорость сходимости становится *сверхлинейной,* если

$$\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x} \right\| \le q_{k+1} \left\| \boldsymbol{x}^k - \boldsymbol{x} \right\|, \quad q_k \to 0+ \quad \text{при} \quad k \to \infty.$$

Говорят, что имеет место *квадратичная* скорость сходимости, если существуют такие постоянные $c \ge 0$ и k_0, что

$$\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x} \right\| \le c \left\| \boldsymbol{x}^k - \bar{\boldsymbol{x}} \right\|^2 \quad \text{при} \quad k \ge k_0.$$

Иногда указанные неравенства заменяют на неравенства

$$\left\| \boldsymbol{x}^{k+1} - \bar{\boldsymbol{x}} \right\| \le c_1 q^{k+1}, \quad q \in (0, 1), \quad k \ge k_0,$$

$$\left\| \boldsymbol{x}^{k+1} - \bar{\boldsymbol{x}} \right\| \le c_2 q_{k+1} q_k \dots q_1, \quad q_k \to 0+,$$

$$\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x} \right\| \le c_3 q^{2^{k+1}}, \quad q \in (0, 1), \quad k \ge k_0.$$

Большинство теорем о сходимости методов минимизации доказываются в предположении *выпуклости* целевой функции, а скорость сходимости устанавливается в предположении ее *сильной* выпуклости. Для невыпуклых задач методы обычно позволяют отыскивать только локальные решения (точнее говоря, стационарные точки). Требования, которые накладываются в теоремах сходимости на минимизируемую функцию, называют *областью применимости*

метода. Часть из них формулируют требования к начальному приближению.

На практике часто используют следующие критерии окончания счета:

$$\left\| x^{k+1} - x^k \right\| \le \varepsilon,$$
$$\left\| f(x^{k+1}) - f(x^k) \right\| \le \varepsilon,$$
$$\left\| f'(x^{k+1}) \right\| \le \varepsilon,$$

где ε — заданная абсолютная точность, с которой ищется точка минимума, а в качестве нормы может быть выбрана любая векторная норма. Как правило, требуют одновременного выполнения указанных критериев.

В тех случаях, когда желательно достижение относительной точности δ, используются такие критерии:

$$\left\| x^{k+1} - x^k \right\| \le \delta \left(1 + \left\| x^{k+1} \right\|\right),$$
$$\left\| f(x^{k+1}) - f(x^k) \right\| \le \delta \left(1 + \left\| f(x^{k+1}) \right\|\right),$$
$$\left\| f'(x^{k+1}) \right\| \le \delta \left(1 + \left\| f'(x^{k+1}) \right\|\right).$$

Иногда применяют *комбинированные* критерии, объединяющие контроль по абсолютной и относительной погрешностям. В пользу такого подхода можно высказать следующие соображения. Рассмотрим неравенство

$$\left\| x^{k+1} - x^k \right\| \le \varepsilon + \delta \left\| x^k \right\|, \tag{5}$$

где x^{k+1} и x^k два последовательных приближения к точке минимума.

Если задана только допустимая абсолютная погрешность ε (т.е. δ = 0), то тем самым фиксируется разряд приближенных значений координат точки минимума, соответствующий требуемым самым младшим верным цифрам этих значений. Однако если задать абсолютную погрешность без учета величины порядка искомого минимума и длины разрядной сетки используемой вычислительной машины, то контроль точности вычислений по абсолютной погрешности может оказаться невозможным. Например, если вычисления проводятся с семью десятичными разрядами и искомый минимум (для одномерного случая) равен 55555.55, то задание абсолютной погрешности, равной 10^{-4} окажется бессмысленным и приведет к зацикливанию итерационного процесса. Поэтому если мы хотим, чтобы четвертый разряд приближенного значения минимума соответствовал самой младшей верной цифре, то в данном примере мы должны положить абсолютную погрешность равной 10. Такое задание абсолютной погрешности в отрыве от величины порядка искомого

минимума и количества разрядов, с которыми проводятся вычисления, может показаться нелепым, поскольку обычно абсолютная погрешность используется для задания количества верных цифр после точки, отделяющей целую часть от дробной.

Таким образом, чтобы разумно задать абсолютную погрешность вычислений, нужно предварительно знать величину порядка нормы решения и учитывать величину нормы начального приближения.

Если задана только допустимая относительная погрешность δ (т.е. ε = 0), то тем самым фиксируется общее требуемое количество верных цифр в приближенных значениях координат точки минимума. Однако если искомый минимум мал и значение $\| x^k \|$ становится слишком близким к нулю, то даже при разумном задании δ неравенство (5) может никогда не достигаться или же при вычислении $\delta\| x^k \|$ может произойти образование машинного нуля (потеря значимости).

Поясним на примере одномерной минимизации, почему это неравенство может никогда не достигаться, даже если в машинном представлении произведение $\delta| x^k|$ не равно нулю и итерационный процесс гарантированно сходится. Для этого напомним фундаментальное свойство систем представления чисел с плавающей точкой: расстояние между числом x и соседним по отношению к нему числом не меньше masheps • $|x|/\beta$ и не больше masheps • $|x|$, если только само число x или соседнее число не равны нулю. Здесь β — основание системы счисления машины, а машинно-зависимый параметр masheps (называемый машинным эпсилон) характеризует относительную точность машинной арифметики.

Таким образом, если $\delta| x^k|$ окажется меньше masheps • $|x|/\beta$, то неравенство (5) при $\varepsilon = 0$ никогда не будет достигаться, а основанный на нем итерационный процесс никогда не завершится. Если мы хотим, чтобы $| x^{k+1}|$ и $| x^k|$ стали максимально близкими друг к другу, т.е. стали соседними числами, то критерий точности должен быть таким:

$$\left|x^{k+1} - x^k\right| \leq \text{masheps} \cdot \max\left(\left|x^{k+1}\right|, \left|x^k\right|\right).$$

Конечно, данный критерий неприменим для небольшой окрестности нуля, в которой происходит образование машинного нуля при вычислении правой части этого неравенства. Отметим, что расстояние от нуля до правого (левого) соседнего числа не связано с машинным эпсилон и представляет собой самостоятельный машинно-зависимый параметр.

Часто применяют следующие две модификации рассмотренного комбинированного критерия:

$$\left\|x^{k+1} - x^k\right\| \leq \max\left\{\varepsilon, \delta \left\|x^k\right\|\right\}$$

или

$$\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^{k}\right\| \leq \begin{cases} \varepsilon, & \text{если } \left\|\boldsymbol{x}^{k}\right\| \leq 1; \\ \delta\left\|\boldsymbol{x}^{k}\right\|, & \text{если } \left\|\boldsymbol{x}^{k}\right\| > 1. \end{cases}$$

Таким образом, применение критерия типа (5) или его модификаций позволяет избегать тех тупиковых ситуаций, которые могут возникнуть, если задавать только абсолютную или только относительную погрешности, и дает возможность задавать требуемое количество верных знаков в приближенном решении, не заботясь о величине его порядка.

6.6.4. Выпуклые множества и выпуклые функции

Пусть $\mathbf{R}^n$ — n-мерное евклидово пространство вещественных векторов $\boldsymbol{x} = (x_1, x_2,..., x_n)^{\mathrm{T}}$. Множество $X \in \mathbf{R}^n$ называется *выпуклым,* если вместе с любыми двумя точками $\boldsymbol{x}^{(1)}$ и $\boldsymbol{x}^{(2)}$ оно содержит и отрезок, соединяющий эти точки; это означает, что

$$\lambda\boldsymbol{x}^{(1)} + (1-\lambda)\,\boldsymbol{x}^{(2)} \in X, \quad \lambda \in [0.1].$$

На числовой прямой $\mathbf{R}^1$ выпуклыми множествами являются всевозможные промежутки (сама прямая, отрезки, интервалы, полупрямые).

Функция $f(\boldsymbol{x})$, определенная на некотором выпуклом множестве $X \in \mathbf{R}^n$, называется выпуклой на X, если выполнено неравенство

$$f(\lambda\boldsymbol{x}^{(1)} + (1-\lambda)\,\boldsymbol{x}^{(2)}) \leq \lambda f(\boldsymbol{x}^{(1)}) + (1-\lambda)\,f(\boldsymbol{x}^{(2)})$$

при всех $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \in X$, $X \in [0,1]$. Если это неравенство строгое, то $f(\boldsymbol{x})$ называют строго выпуклой функцией на X. Функция $f(\boldsymbol{x})$ называется *вогнутой,* если — $f(\boldsymbol{x})$ выпукла. Геометрически выпуклость означает, что любая хорда графика $f(\boldsymbol{x})$ располагается выше кривой $f(\boldsymbol{x})$.

Задача минимизации (оптимизации) называется выпуклой, если X — выпуклое множество, а $f(\boldsymbol{x})$ — выпуклая на X функция.

Теорема 5. *Если задача минимизации выпукла, то любое ее локальное решение является также глобальным*

Доказательство. Пусть $\overline{x}$ — точка локального минимума, т.е. при некотором $\varepsilon > 0$ имеем $f(\overline{x}) \leq f(\boldsymbol{x})$ для всех

$$\boldsymbol{x} \in X \bigcap U_\varepsilon(\bar{\boldsymbol{x}}), \text{ где } U_\varepsilon(\bar{\boldsymbol{x}}) = \{\boldsymbol{x} \in \boldsymbol{R}^n \mid \|\boldsymbol{x} - \bar{\boldsymbol{x}}\| \leq \varepsilon\}$$

— шар радиуса ε с центром в точке $\boldsymbol{x}$.

Для любого $\boldsymbol{x} \in X$, $\boldsymbol{x} \neq \overline{x}$, положим

$$\lambda = \min\{\varepsilon/(\|\boldsymbol{x} - \boldsymbol{x}\|), 1\}.$$

Тогда

$$\lambda \boldsymbol{x} + (1 - \lambda)\,\bar{\boldsymbol{x}} \in X \bigcap U_\varepsilon(\boldsymbol{x}).$$

Действительно, имеет место неравенство

$$\|\lambda \boldsymbol{x} + (1 - \lambda)\,\boldsymbol{x} - \bar{\boldsymbol{x}}\| = \lambda\,\|\boldsymbol{x} - \bar{\boldsymbol{x}}\| \le \varepsilon.$$

Следовательно, в силу выпуклости $f(\boldsymbol{x})$ имеем

$$f(\bar{\boldsymbol{x}}) \le f(\lambda \boldsymbol{x} + (1 - \lambda)\,\bar{\boldsymbol{x}}) \le \lambda f(\boldsymbol{x}) + (1 - \lambda)\,f(\boldsymbol{x}).$$

Отсюда заключаем, что $f(\overline{x}) \le f(\boldsymbol{x})$. Теорема доказана.

Для выпуклых задач необходимые условия оптимальности являются также и достаточными.

Теорема 6. *Пусть функция $f(\boldsymbol{x})$— выпукла на X и дифференцируема в точке $\overline{x} \in X$. Если $f'(\overline{x}) = 0$, то $\overline{x}$ — точка минимума $f(\boldsymbol{x})$ на X.*

Доказательство. В силу выпуклости $f(\boldsymbol{x})$ имеем

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\,\bar{\boldsymbol{x}}) \le \lambda f(\boldsymbol{x}) + (1 - \lambda)\,f(\boldsymbol{x}), \quad \lambda \in [0, 1].$$

Отсюда

$$f(\boldsymbol{x}) - f(\boldsymbol{x}) \ge \frac{f(\bar{\boldsymbol{x}} + \lambda(\boldsymbol{x} - \bar{\boldsymbol{x}})) - f(\boldsymbol{x})}{\lambda}.$$

Разложим $f(\overline{x} + \lambda(\boldsymbol{x} - \overline{x}))$ в ряд Тейлора:

$$f(\boldsymbol{x}) - f(\boldsymbol{x}) \ge \frac{(f'(\boldsymbol{x}), \lambda(\boldsymbol{x} - \bar{\boldsymbol{x}})) + o(\lambda\|\boldsymbol{x} - \boldsymbol{x}\|_2)}{\lambda} = \frac{o(\lambda)}{\lambda}$$

После предельного перехода при $\lambda \to 0$ получим $f(\boldsymbol{x}) - f(\overline{x}) \ge 0$. Отсюда $f(\boldsymbol{x}) \le f(\overline{x})$. Теорема доказана

Из этой теоремы следует, что для выпуклых задач оптимизации отыскание стационарной точки означает отыскание точки побального минимума

Для выявления выпуклости функции можно воспользоваться следующим критерием, если функция $f(\boldsymbol{x})$ дважды дифференцируема на выпуклом множестве $X \subset \mathbf{R}^n$ и матрица ее вторых производных $f''(\boldsymbol{x})$ положительно определена при всех $\boldsymbol{x} \in X$, то $f(\boldsymbol{x})$ является выпуклой функцией на множестве X.

Если к матрице $f''(\boldsymbol{x})$ применить критерии Сильвестра, то критерий выпуклости формулируется так - если все ведущие миноры матрицы $f''(\boldsymbol{x})$ положительны при всех $\boldsymbol{x} \in X$, то функция $f(\boldsymbol{x})$ выпукла на множестве X.

Укажем еще одно полезное свойство выпуклых задач.

Теорема 7. *Пусть рассматривается выпуклая задача оптимизации. Тогда множество ее решений X* = { $\bar{x}$ } выпукло. Если при этом функция f(x) строго выпукла на X, то решение задачи единственно, т. е. множество X* состоит из одной точки.*

Доказательство. Пусть $\boldsymbol{x}_1$ и $\boldsymbol{x}_2$ принадлежат X^* и $\lambda \in [0, 1]$. Тогда $f(\boldsymbol{x}_1) = f(\boldsymbol{x}_2) = f(\bar{x})$. В силу выпуклости функции $f(\boldsymbol{x})$ имеем:

$$f(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) \le \lambda f(\boldsymbol{x}_1) + (1-\lambda) f(\boldsymbol{x}_2) = f(\bar{x})$$

Поскольку $f(\bar{x})$ - минимальное значение $f(\boldsymbol{x})$ на X, то это неравенство может выполниться только как равенство. Следовательно, $\lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2$ — точка минимума. Значит, по определению, множество X^* выпукло.

Пусть теперь функция $f(\boldsymbol{x})$ строго выпукла. Если предположить, что в X^* существуют две различные точки $\boldsymbol{x}_1$ и $\boldsymbol{x}_2$, то при $\lambda \in [0, 1]$ приведенное выше неравенство должно быть строгим, что невозможно, поскольку $f(\bar{x})$ — минимальное значение $f(\boldsymbol{x})$ на X. Теорема доказана

6.6.5. Квадратичные функция

Во многих задачах оптимизации рассматриваются *квадратичные* функции, т.е. функции вида

$$f(\boldsymbol{x}) = \sum_{i,j=1}^{n} c_{ij} x_i x_j + \sum_{j=1}^{n} b_j x_j .$$

Положим $a_{ij} = c_{ij} + c_{ji}$. Тогда матрица $A = (a_{ij})$ будет симметричной. С ее помощью квадратичную функцию можно представить в виде

$$f(\boldsymbol{x}) = \frac{1}{2}(A\boldsymbol{x}, \boldsymbol{x}) + (\boldsymbol{b}, \boldsymbol{x}),$$

где $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ и $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)^T$.

Градиент и матрица Гессе квадратичной функции представляются следующим образом:

$$\text{grad } f(\boldsymbol{x}) = f'(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}, \quad f''(\boldsymbol{x}) = A.$$

Чтобы квадратичная функция была выпуклой на R^n, достаточно, чтобы матрица A была положительно определена

В случае минимизации выпуклой квадратичной функции выбор шага α_k, на $(k + 1)$-й итерации по направлению убывания может быть осуществлен из следующих соображений. Запишем

$$
\begin{aligned}
p(\alpha) &= f(\boldsymbol{x}^k + \alpha \boldsymbol{h}^k) = \\
&= \frac{1}{2}\left(A(\boldsymbol{x}^k + \alpha \boldsymbol{h}^k), \boldsymbol{x}^k + \alpha \boldsymbol{h}^k\right) + \left(\boldsymbol{b}, \boldsymbol{x}^k + \alpha \boldsymbol{h}^k\right) = \\
&= \frac{1}{2}\left((A\boldsymbol{x}^k, \boldsymbol{x}^k) + \alpha\,(A\boldsymbol{x}^k, \boldsymbol{h}^k) + \alpha\,(A\boldsymbol{h}^k, \boldsymbol{x}^k) + \right. \\
&\quad \left. + \alpha^2 (A\boldsymbol{h}^k, \boldsymbol{h}^k)\right) + (\boldsymbol{b}, \boldsymbol{x}^k) + \alpha\,(\boldsymbol{b}, \boldsymbol{h}^k) = \\
&= \frac{1}{2}\,(A\boldsymbol{h}^k, \boldsymbol{h}^k)\,\alpha^2 + (A\boldsymbol{x}^k + \boldsymbol{b}, \boldsymbol{h}^k)\,\alpha + \left(\frac{1}{2} A\boldsymbol{x}^k + \boldsymbol{b}, \boldsymbol{x}^k\right).
\end{aligned}
$$

Здесь мы воспользовались равенством $(A\boldsymbol{h}^k, \boldsymbol{x}^k) = (A\boldsymbol{x}^k, \boldsymbol{h})$, поскольку A — симметричная матрица.

Итак, мы выписали квадратный трехчлен $p(\alpha)$. Его минимум достигается при том значении α, которое может быть получено из уравнения $p'(\alpha) = 0$:

$$(A\boldsymbol{h}^k, \boldsymbol{h}^k)\,\alpha + (A\boldsymbol{x}^k + \boldsymbol{b}, \boldsymbol{h}^k) = 0.$$

Отсюда получаем, что

$$\alpha_k = -\frac{(A\boldsymbol{x}^k + \boldsymbol{b}, \boldsymbol{h}^k)}{(A\boldsymbol{h}^k, \boldsymbol{h}^k)}.$$

Полученное значение α_k неотрицательно, поскольку числитель не положителен по признаку убывания, а знаменатель строго больше нуля в силу положительной определенности матрицы A.

Если квадратичная функция выпукла, то точку минимума можно также найти из уравнения $f'(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b} = 0$, т.е. решая систему линейных алгебраических уравнений с симметричной положительно определенной матрицей.

6.6.6. Градиентные методы

Рассмотрим методы безусловной минимизации, основанные на идее замены минимизируемой функции $f(\boldsymbol{x})$ в окрестности очередного приближения $\boldsymbol{x}^k$ первым членом (линейной частью) ее разложения в ряд Тейлора. Такие методы называют *градиентными,* поскольку при вычислении $\boldsymbol{x}^{k+1}$ используются производные функции $f(\boldsymbol{x})$ первого порядка.

Градиентные методы относятся к классу методов спуска, в которых два последовательных приближения к точке минимума связаны соотношением

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{h}^k,$$

где $\boldsymbol{h}^k$ — направление убывания функции $f(\boldsymbol{x})$ в точке и α_k — длина шага по направлению убывания $\boldsymbol{h}^k$. Вектор $\boldsymbol{h}^k$ берется равным антиградиенту функции $f(x)$ в точке $\boldsymbol{x}^k$, т.е. $\boldsymbol{h}^k = -f'(\boldsymbol{x}^k)$:

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k - \alpha_k f'(\boldsymbol{x}^k), \quad \alpha_k >0, \quad k = 0,1,2,.... \tag{6}$$

В пользу такого выбора направления убывания могут быть высказаны следующие соображения. В предположении, что функция $f(\boldsymbol{x})$ дифференцируема на $\boldsymbol{R}^n$, рассмотрим линейную часть приращения $f(\boldsymbol{x})-f(\boldsymbol{x}^k)$:

$$\begin{aligned} f(\boldsymbol{x}) &= f(\boldsymbol{x}^k + (\boldsymbol{x} - \boldsymbol{x}^k)) = \\ &= f(\boldsymbol{x}^k) + (f'(\boldsymbol{x}^k).\, \boldsymbol{x} - \boldsymbol{x}^k) + o\left(\left\|\boldsymbol{x} - \boldsymbol{x}^k\right\|_2\right). \end{aligned} \tag{7}$$

Все возможные направления перемещении от точки $\boldsymbol{x}^k$ с конечным шагом α образуют шар X радиуса α с центром в точке $\boldsymbol{x}^k$: $X = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^k\|_2 \le \alpha\}$. Наша цель найти такое направление убывания, при котором на границе этого шара выполнялись условия, чтобы $f(\boldsymbol{x}) < f(\boldsymbol{x}^k)$ и чтобы разность $f(\boldsymbol{x}^k) - f(\boldsymbol{x})$ при этом была наибольшей (т.е. чтобы при фиксированной длине шага по искомому направлению достигалось наименьшее значение $f(\boldsymbol{x})$).

Из (7) можно заключить, что эта разность будет наибольшей, если мы минимизируем по x на шаре X линейную часть приращения $f(\boldsymbol{x}) - f(\boldsymbol{x}^k)$, равную $(f'(\boldsymbol{x}^k), \boldsymbol{x} - \boldsymbol{x}^k)$. Воспользовавшись неравенством Коши-Буняковекого, запишем

$$(f'(\boldsymbol{x}^k), \boldsymbol{x} - \boldsymbol{x}^k) \ge -\left\|f'(\boldsymbol{x}^k)\right\|_2 \left\|\boldsymbol{x} - \boldsymbol{x}^k\right\|_2 \ge -\alpha \left\|f'(\boldsymbol{x}^k)\right\|_2.$$

Легко видеть, что нижняя грань последнего неравенства достигается при

$$\boldsymbol{x} = \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \frac{\alpha\, f'(\boldsymbol{x}^k)}{\left\|f'(\boldsymbol{x}^k)\right\|_2} \in X.$$

Таким образом, приходим к выводу, что при фиксированной длине шага α минимум линейной части разложения функции $f(\boldsymbol{x})$ в ряд Тейлора в окрестности точки $\boldsymbol{x}^k$ достигается, если направление вектора $\boldsymbol{h} = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$ совпадает с направлением антиградиента $-f'(\boldsymbol{x}^k)$. Это означает, что направление ангиградиента является самым выгодным из всех направлений убывания.

Для квадратичной функции градиентный метод (6) принимает вид $x^{k+1}=x^k - \alpha_k(A\boldsymbol{x}^k+\boldsymbol{b})$.

В численных расчетах шаг α_k по направлению убывания может быть получен методом дробления шага, рассмотренном в п. 6.6.2. Если же α_k выбирается при помощи одномерной минимизации функции $f(\boldsymbol{x}^k + \alpha \boldsymbol{h})$ вдоль антиградиента, то такая модификация градиентного метода называется методом *наискорейшего спуска,* при котором достигается

максимальное уменьшение функции $f(\boldsymbol{x})$ вдоль направления ее антиградиента. Для квадратичных функций соответствующее значение α_k, приведено в п. 6.6. 5.

Градиентный метод сходится к точке минимума линейно, т.е. со скоростью геометрической прогрессии. Если на текущем шаге итерации наименьшее и наибольшее собственные значения матрицы Гессе мало отличаются друг от друга, то знаменатель прогрессии уменьшаемся, а скорость сходимости увеличивается. Если же эти собственные значения значительно отличаются, то направление антиградиента может сильно отклоняться от направления в точку минимума; из-за этого движение к минимуму приобретает зигзагообразный характер и сходимость замедляется.

Чувствительность градиентного метода минимизации к погрешностям вычислений повышается в окрестности точки минимума, когда норма градиента мала. Поэтому градиентный метод и его модификации лучше использовать в начальной стадии поиска минимума, чем на его заключительном этапе.

6.6.7. Метод Ньютона многомерной минимизации

Если в окрестности очередного приближения $\boldsymbol{x}^k$ мы разложим минимизируемую функцию $f(\boldsymbol{x})$ в ряд Тейлора и возьмем квадратичную часть этого разложения, то получим метод второго порядка (метод Ньютона), который использует информацию о вторых производных функции $f(\boldsymbol{x})$. Эмот метод применяется для безусловной минимизации выпуклых дважды дифференцируемых функций и при определенных условиях обеспечивает более быструю, нежели градиентный метод и его модификации, скорость сходимости.

Пусть функция $f(\boldsymbol{x})$ выпукла и дважды дифференцируема на $\boldsymbol{R}^n$, причем матрица $f''(\boldsymbol{x})$ не вырождена на $\boldsymbol{R}^n$. Исходя из определения дважды дифференцируемой функции, можно выписать следующее разложение для $f(\boldsymbol{x})$ в окрестности точки $\boldsymbol{x}^k$:

$$\begin{aligned} f(\boldsymbol{x}) - f(\boldsymbol{x}^k) \;=\; & (f'(\boldsymbol{x}^k), \boldsymbol{x} - \boldsymbol{x}^k) + \\ & + \frac{1}{2}\,(f''(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k), \boldsymbol{x} - \boldsymbol{x}^k) + \\ & + o\left(\|\boldsymbol{x} - \boldsymbol{x}^k\|_2^2\right). \end{aligned}$$

Обозначим квадратичную часть приращения $f(\boldsymbol{x}) - f(\boldsymbol{x}^k)$ через

$$f_k(\boldsymbol{x}) = (f'(\boldsymbol{x}^k), \boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2}\,(f''(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k), \boldsymbol{x} - \boldsymbol{x}^k).$$

Найдем точку $\boldsymbol{x}^{k+1}$, в которой достигается минимум функции $f_k(\boldsymbol{x})$. По предположению функция $f(\boldsymbol{x})$ выпукла; значит, матрица $f''(\boldsymbol{x})$ положительно определена. Поскольку $f''_k(\boldsymbol{x}) = f''(\boldsymbol{x}^k)$, то $f''_k(\boldsymbol{x})$ — также положительно определенная матрица. Следовательно, функция $f_k(\boldsymbol{x})$ выпукла в силу необходимого и достаточного условия выпуклости. Отсюда заключаем, что по теоремам 5 и 6 необходимое и достаточное условие ее минимума имеет вид

$$f'_k(\boldsymbol{x}) = f'(\boldsymbol{x}^k) + f''(\boldsymbol{x}^k)(\boldsymbol{x}-\boldsymbol{x}^k) = 0.$$

Теперь решим полученную систему линейных уравнений, получим точку минимума функции $f'(\boldsymbol{x})$ и возьмем ее в качестве очередного приближения $\boldsymbol{x}^{k+1}$ к точке минимума исходной функции $f(\boldsymbol{x})$:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - [f''(\boldsymbol{x}^k)]^{-1} f'(\boldsymbol{x}^k). \tag{8}$$

Здесь $[f''(\boldsymbol{x}^k)]^{-1}$ — матрица, обратная к матрице вторых производных $f''(\boldsymbol{x}^k)$. Выписанное соотношение называют методом Ньютона.

При достаточно хорошем приближении метод (8) имеет квадратичную скорость сходимости. Поэтому его удобно применять на завершающем этапе минимизации при уточнении приближения к точке минимума, найденного каким-либо другим, менее трудоемким способом. Если начальное приближение выбрано неудачно, то сходимость отсутствует. Указанный недостаток устраняется, если применить следующую модификацию метода Ньютона, называемую *модифицированным* методом Ньютона (или методом Ньютона с регулировкой шага):

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha_k \left[f''(\boldsymbol{x}^k)\right]^{-1} f'(\boldsymbol{x}^k), \quad \alpha_k > 0. \tag{9}$$

При $\alpha_k = 1$ итерационный метод (9) совпадает с классическим методом (8). Легко видеть, что эти методы относятся к классу методов спуска $x^{k+1} = x^k + \alpha_k \boldsymbol{h}^k$, где вектор направления убывания $\boldsymbol{h}^k$ находится из решения линейной системы $f''(\boldsymbol{x}^k)\,\boldsymbol{h}^k = -f'(x^k)$. Отсюда следует, что в практических расчетах на каждой итерации нет необходимости обращать матрицу $f''(\boldsymbol{x}^k)$: достаточно решить указанную линейную систему. Выбор шага α_k по направлению убывания можно осуществлять либо методом дробления шага, рассмотренном в п. 6.6.2, либо при помощи одномерной минимизации функции $f(\boldsymbol{x}^k + \alpha \boldsymbol{h}^k)$ вдоль направления убывания.

Может быть показано, что модифицированный метод Ньютона (9) сходится при любом начальном приближении $x^0 \in \boldsymbol{R}^n$, причем скорость сходимости будет сверхлинейной или квадратичной в зависимости от свойств функцни $f(\boldsymbol{x})$. Таким образом, с помощью регулировки шага по направлению убывания преодолевается недостаток метода (8), связанный с необходимостью выбора хорошего начального приближения.

Если по каким-либо причинам сложно вычислять матрицу $f''(x^k)$, то можно строить ее аппроксимации при помощи формул численного дифференцирования. Построенные при таком подходе методы называют *квазиньютоновскими.* Остановимся на этом вопросе подробнее.

Поскольку матрица $f''(x^k)$ содержит частные производные второго порядка, то достаточно рассмотреть случай функции двух переменных $f(x,y)$. Для аппроксимации производных

$$\frac{\partial^2 f}{\partial x^2} \text{ и } \frac{\partial^2 f}{\partial y^2}$$

воспользуемся известными соотношениями

$$\frac{\partial^2 f}{\partial x^2} = \frac{f(x-h,y) - 2f(x,y) + f(x+h,y)}{h^2} + O\left(h^2\right),$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{f(x,y-h) - 2f(x,y) + f(x,y+h)}{h^2} + O\left(h^2\right).$$

Здесь h - малый параметр, определяющий погрешность выписанных формул численного дифференцирования.

Теперь выведем разностное соотношение для аппроксимации смешанной производной

$$\frac{\partial^2 f}{\partial x \partial y}.$$

Для произвольной достаточно гладкой функции $g(x, y)$ введем в рассмотрение разностные операторы

$$g_x = \frac{g(x+h,y) - g(x,y)}{h}, \quad g_{\bar{x}} = \frac{g(x,y) - g(x-h,y)}{h},$$

$$g_y = \frac{g(x,y+h) - g(x,y)}{h}, \quad g_{\bar{y}} = \frac{g(x,y) - g(x,y-h)}{h}.$$

Имеем

$$g_x = \frac{\partial g}{\partial x} + \frac{h}{2}\frac{\partial^2 g}{\partial x^2} + O\left(h^2\right),$$

$$f_y = \frac{\partial f}{\partial y} - \frac{h}{2}\frac{\partial^2 f}{\partial y^2} + O\left(h^2\right).$$

Используя эти разложения для $g = f_{\bar{y}}$, получим

$$(f_y)_x = \frac{\partial^2 f}{\partial x \partial y} + \frac{h}{2}\frac{\partial^3 f}{\partial^2 x \partial y} - \frac{h}{2}\frac{\partial^3 f}{\partial x \partial^2 y} + O\left(h^2\right)$$

Аналогично можно получить, что

$$(f_y)_x = \frac{\partial^2 f}{\partial x \partial y} - \frac{h}{2}\frac{\partial^3 f}{\partial^2 x \partial y} + \frac{h}{2}\frac{\partial^3 f}{\partial x \partial^2 y} + O(h^2).$$

Сложив два последних соотношения, получаем, что

$$\frac{1}{2}\left((f_{\bar{y}})_x + (f_y)_{\bar{x}}\right) = \frac{\partial^2 f}{\partial x \partial y} + O(h^2),$$

где

$$(f_{\bar{y}})_x = \frac{f(x+h,y) - f(x,y) - f(x+h,y-h) + f(x,y-h)}{h^2},$$

$$(f_y)_{\bar{x}} = \frac{f(x,y+h) - f(x,y) - f(x-h,y+h) + f(x-h,y)}{h^2}.$$

Эти разностные соотношения получаются последовательным применением приведенных выше разностных операторов.
Для квадратичной функции

$$f(\boldsymbol{x}) = \frac{1}{2}(A\boldsymbol{x},\boldsymbol{x}) + (\boldsymbol{b},\boldsymbol{x}), \quad f'(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}, \quad f''(\boldsymbol{x}) = A$$

метод (8) примет вид

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - A^{-1}(A\boldsymbol{x}^k + \boldsymbol{b}),$$

т.е. при любом начальном приближении точное решение достигается за одну итерацию. Если применяется метод (9) с регулировкой шага при помощи одномерной минимизации вдоль направления убывания, то α_k для квадратичной функции выражается явно (см. п. 6.6.5).

6.7. Многомерный поиск без использования производных (прямые методы минимизации).

Рассмотрим методы решения минимизации функции нескольких переменных *f*, которые опираются только на вычисление значений функции *f(x)*, не используют вычисление производных, т.е. прямые методы минимизации. Важно отметить, что для применения этих методов не требуется не только дифференцируемости целевой функции, но даже аналитического задания. Нужно лишь иметь возможность вычислять или измерять значения *f* в произвольных точках. Такие ситуации часто встречаются в практически важных задачах оптимизации. В основном все описанные методы заключаются в следующем. При заданном векторе *x* определяется допустимое направление *d*. Затем, отправляясь из точки *x*, функция *f* минимизируется вдоль направления *d* одним из методов одномерной

минимизации. Задача линейного поиска заключается в минимизации *f(x+lym*d)* при условии, что *lym* принадлежит *L*, где *L* обычно задается в форме *L=El, L={lym: lym* ≥ 0*}* или *L={l: a≤lym≤b}*. Будем предполагать, что точка минимума *lym** существует. Однако в реальных задачах это предположение может не выполняться. Оптимальное значение целевой функции в задаче линейного поиска может быть не ограниченным или оптимальное значение функции конечно, но не достигается ни при каком *lym*.

Подавляющее число реальных задач оптимизации, представляющих практический интерес, являются многомерными: в них целевая функция зависит от нескольких аргументов, причем иногда их число может быть весьма большим. Математическая постановка таких задач аналогична их постановке в одномерном случае: ищется наименьшее (наибольшее) значение целевой функции, заданной на некотором множестве *G* возможных значений ее аргументов. Как и в одномерном случае, характер задачи и соответственно возможные методы решения существенно зависят от той информации о целевой функции, которая нам доступна в процессе ее исследования. В одних случаях целевая функция задается аналитической формулой, являясь при этом дифференцируемой функцией. Тогда можно вычислить ее частные производные, получить явное выражение для градиента, определяющего в каждой точке направления возрастания и убывания функции, и использовать эту информацию для решения задачи. В других случаях никакой формулы для целевой функции нет, а имеется лишь возможность определить ее значение в любой точке рассматриваемой области (с помощью расчетов, в результате эксперимента и т.д.). В таких задачах в процессе решения мы фактически можем найти значения целевой функции лишь в конечном числе точек, и по этой информации требуется приближенно установить ее наименьшее значение для всей области.

6.7.1. Метод Хука – Дживса

Этот метод был разработан в 1961 году, но до сих пор является весьма эффективным и оригинальным. Поиск состоит из последовательности шагов исследующего поиска вокруг базисной точки, за которой в случае успеха следует поиск по образцу.

Описание этой процедуры представлено ниже:

А. Выбрать начальную базисную точку b_1 и шаг длиной h_j для каждой переменной x_j, $j = 1, 2, ..., n$. В приведенном ниже алгоритме для каждой переменной используется шаг h, однако указанная выше модификация тоже может оказаться полезной.

Б. Вычислить $f(x)$ в базисной точке b_1 с целью получения сведений о локальном поведении функции $f(x)$. Эти сведения будут использоваться для нахождения подходящего направления поиска по образцу, с помощью которого можно надеяться достичь большего убывания значения функции. Функция $f(x)$ в базисной точке b_1 находится следующим образом:

1. Вычисляется значение функции $f(b_1)$ в базисной точке b_1.

2. Каждая переменная по очереди изменяется прибавлением длины шага.

Таким образом, мы вычисляем значение функции $f(b_1 + h_1e_1)$, где e_1 - единичный вектор в направлении оси x_1. Если это приводит к уменьшению значения функции, то b_1 заменяется на $b_1 + h_1e_1$. В противном случае вычисляется значение функции $f(b_1 – h_1e_1)$, и если ее значение уменьшилось, то b_1 заменяем на b_1-h_1e_1. Если ни один из проделанных шагов не приводит к уменьшению значения функции, то точка b_1 остается неизменной и рассматриваются изменения в направлении оси x_2, т.е. находится значение функции $f(b_1 + h_2e_2)$ и т.д. Когда будут рассмотрены все n переменные, мы будем иметь новую базисную точку b_2.

3. Если $b_2 = b_1$, т.е. уменьшение функции не было достигнуто, то исследование повторяется вокруг той же базисной точки b_1, но с уменьшенной длиной шага. На практике удовлетворительным является уменьшение шага (шагов) в десять раз от начальной длины.

4. Если $b_2 \neq b_1$, то производится поиск по образцу.

В. При поиске по образцу используется информация, полученная в процессе исследования, и минимизация функции завершается поиском в направлении, заданном образцом. Эта процедура производится следующим образом:

1. Разумно двигаться из базисной точки b_2 в направлении b_2 - b_1, поскольку поиск в этом направлении уже привел к уменьшению значения функции. Поэтому вычислим функцию в точке образца

$$P_1 = b_1 + 2(b_2 - b_1) \quad (1)$$

В общем случае

$$P_j = b_j + 2(b_{j+1} - b_i) \quad (2)$$

2. Затем исследование следует продолжать вокруг точки P_1 (P_j).

3. Если наименьшее значение на шаге В,2 меньше значения в базисной точке b_2 (в общем случае b_{j+1}), то получают новую базисную точку b_3 (b_{j+2}), после чего следует повторить шаг В,1. В противном случае не производить поиск по образцу из точки b_2 (b_{j+1}) а продолжить исследования в точке b_2 (b_{j+1}).

Г. Завершить этот процесс, когда длина шага (длины шагов) будет уменьшена до заданного малого значения.

Ниже приведена блок-схема данного метода.

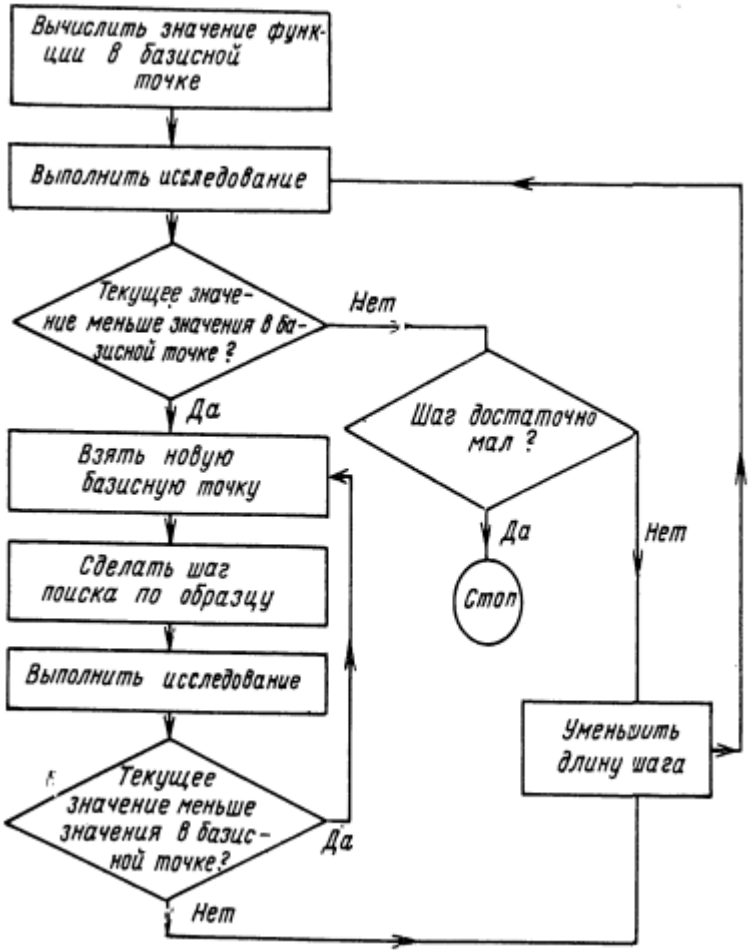

Вычислить значение функции в базисной точке
Выполнить исследование
Текущее значение меньше значения в базисной точке ?
Нет
Да
Шаг достаточно мал ?
Да
Нет
Стоп
Взять новую базисную точку
Сделать шаг поиска по образцу
Выполнить исследование
Уменьшить длину шага
Текущее значение меньше значения в базисной точке?
Да
Нет

Рис. 1.

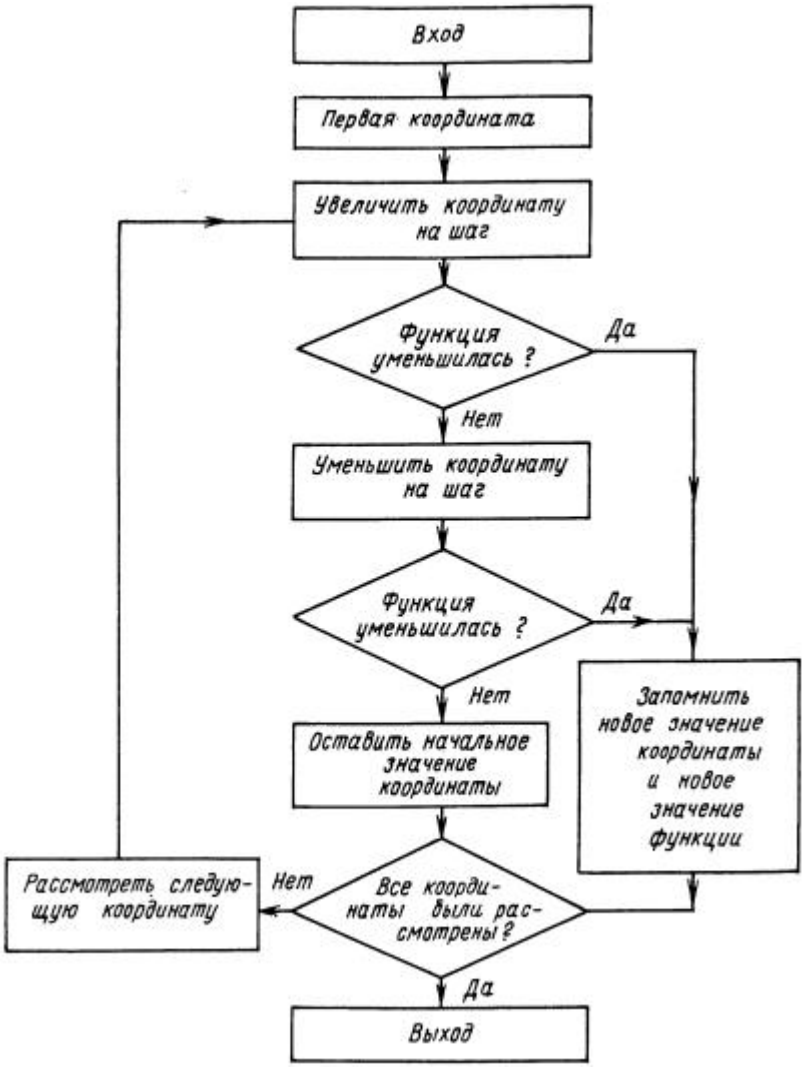

Вход
Первая координата
Увеличить координату на шаг
Функция уменьшилась ?
Да
Нет
Уменьшить координату на шаг
Функция уменьшилась ?
Да
Нет
Запомнить новое значение координаты и новое значение функции
Оставить начальное значение координаты
Рассмотреть следующую координату
Нет
Все координаты были рассмотрены?
Да
Выход

Рис. 2.

6.7.2. Метод Нелдера – Мида.

Метод Нелдера — Мида (называется также поиском по деформируемому многограннику) является развитием симплексного метода Спендли, Хекста и Химсворта. Множество (n+1)-й равноудаленной точки в n-мерном пространстве называется регулярным симплексом. Эта конфигурация рассматривается в методе Спендли, Хекста и Химсворта. Следовательно, в двумерном пространстве симплексом является равносторонний треугольник, а в трехмерном пространстве — правильный тетраэдр. Идея метода состоит в сравнении значений функции в (n + 1) вершинах симплекса и перемещении симплекса в направлении оптимальной точки с помощью итерационной процедуры. В симплексном методе, предложенном первоначально, регулярный симплекс использовался на каждом этапе. Нелдер и Мид предложили несколько модификаций этого метода, допускающих, чтобы симплексы были неправильными. В результате получился очень надежный метод прямого поиска, являющийся одним из самых эффективных, если $n \leq 6$.

В методе Спендли, Хекста и Химсворта симплекс перемещается с помощью трех основных операций: отражения, растяжения и сжатия. Смысл этих операций станет понятным при рассмотрении шагов процедуры.

А. Найдем значения функции $f_1=f(x_1), f_2=f(x_2)$ ... $f_{n+1}=f(x_{n+1})$ в вершинах симплекса.

Б. Найдем наибольшее значение функции f_h, следующее за наибольшим значением функции f_g наименьшее значение функции f_l и соответствующие им точки x_h, x_g, x_l.

В. Найдем центр тяжести всех точек, за исключением точки x_h. Пусть центром тяжести будет

$$x_0 = \frac{1}{n} \sum_{i \neq h} x_i \qquad (3)$$

и вычислим $f(x_0)=f_0$.

Г. Удобнее всего начать перемещение от точки x_h. Отразив точку x_h относительно точки x_0, получим точку x_r и найдем $f(x_r) = f_r$. Операция отражения иллюстрируется рис. 3. Если $a>0$ - коэффициент отражения, то положение точки x_r определяется следующим образом:

$$x_r - x_0 = a(x_0 - x_h)$$

т.е.

$$x_r = (1+a)x_0 - ax_h \qquad (4)$$

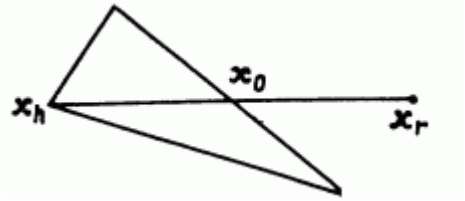


Рис. 3.

Замечание:

$$a = |x_r - x_0| / |x_0 - x_h|$$

Д. Сравним значения функций f_r и f_l.

1. Если $f_r < f_l$, то мы получили наименьшее значение функции. Направление из точки x_0 в точку x_r наиболее удобно для перемещения. Таким образом, мы производим растяжение в этом направлении и находим точку x_e и значение функции $f_e = f(x_e)$. Рисунок 4 иллюстрирует операцию растяжения симплекса.

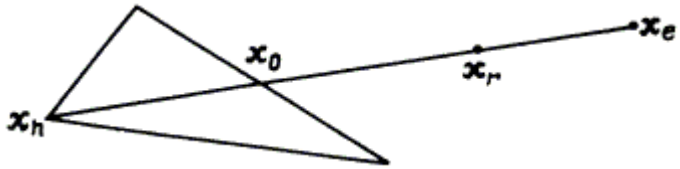


Рис. 4.

Коэффициент растяжения γ>1 можно найти из следующих соотношений:

$$x_e - x_0 = \gamma(x_r - x_0)$$

т.е.

$$x_e = \gamma x_r + (1-\gamma)x_0 \qquad (5)$$

Замечание:

$$\gamma = |x_e - x_0| / |x_r - x_0|$$

а) Если $f_e < f_l$, то заменяем точку x_h на точку x_e и проверяем (n + 1)-ую точку симплекса на сходимость к минимуму (см. шаг Б). Если сходимость достигнута, то процесс останавливается; в противном случае возвращаемся на шаг Б.

б) Если $f_e > f_l$, то отбрасываем точку x_e. Очевидно, мы переместились слишком далеко от точки x_0 к точке x_r. Поэтому следует заменить точку x_h на точку x_r, в которой было получено улучшение (шаг Д, 1), проверить сходимость и, если она не достигнута, вернуться на шаг Б.

2. Если $f_r > f_l$, но $f_r < f_g$, то x_r является лучшей точкой по сравнению с другими двумя точками симплекса и мы заменяем точку x_h на точку x_r и, если сходимость не достигнута, возвращаемся на шаг Б, т.е. выполняем пункт 1,6, описанный выше.

3. Если $f_r > f_e$ *и* $f_r > f_g$, перейдем на шаг Е.

Е. Сравним значения функций f_r и f_h.

1. Если $f_r > f_h$, то переходим непосредственно к шагу сжатия Е,2.

Если $f_r < f_h$, то заменяем точку x_h на точку x_r и значение функции f_h на значение функции f_r. Запоминаем значение $f_r > f_g$ из шага Д,2, приведенного выше. Затем переходим на шаг Е,2.

В этом случае $f_r > f_h$, поэтому ясно, что мы переместились слишком далеко от точки x_h к точке x_0. Попытаемся исправить это, найдя точку x_c (а затем f_c) с помощью шага сжатия, показанного на рис. 5.

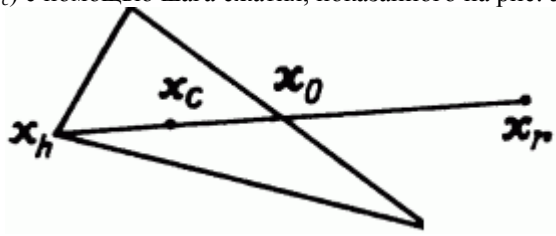


Рис. 5.

3. Если $f_r > f_h$, то сразу переходим к шагу сжатия и находим точку x_c из соотношения

$$x_c - x_0 = \beta(x_h - x_0)$$

где $\beta(0 < \beta < 1)$ - коэффициент сжатия. Тогда

$$x_c = \beta x_h + (1-\beta)x_0 \qquad (6)$$

Если $f_r < f_h$, то сначала заменим точку x_h на точку x_r, а затем произведем сжатие. Тогда точку x_c найдем из соотношения

$$x_c - x_0 = \beta(x_r - x_0)$$

т.е.

$$x_c = \beta x_r + (1-\beta)x_0 \qquad (7)$$

(рис. 6).

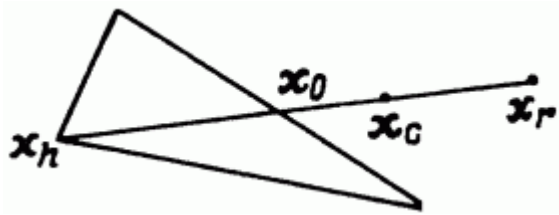


Рис. 6.

Ж. Сравним значения функций f_c и f_h.

1. Если $f_c < f_h$, то заменяем точку x_h на точку x_c и если сходимость не достигнута, то возвращаемся на шаг Б.

2. Если $f_c > f_h$, то очевидно, что все наши попытки найти значение меньшее f_h закончились неудачей, поэтому мы переходим на шаг 3.

3. На этом шаге мы уменьшаем размерность симплекса делением пополам расстояния от каждой точки симплекса до x_1 - точки, определяющей наименьшее значение функции.

Таким образом, точка x_j заменяется на точку

$$x_i = {}^1/_2(x_i - x_l),$$

т.е. заменяем точку x_i точкой

$${}^1/_2(x_i - x_l) \qquad (8)$$

Затем вычисляем f_i для i = 1, 2, ...,(n+1), проверяем сходимость и, если она не достигнута, возвращаемся на шаг В.

И. Проверка сходимости основана на том, чтобы стандартное отклонение (n + 1)-го значения функции было меньше некоторого заданного малого значения e. В этом случае вычисляется

$$\sigma^2 = \sum_{i=I}^{n+1}(f_i - \overline{f})^2/(n+1) \qquad (9)$$

где

$$\overline{f} = \sum f_i/n+1.$$

Если $\sigma < e$, то все значения функции очень близки друг к другу, и поэтому они, возможно, лежат вблизи точки минимума функции x_1. Исходя из этого, такой критерий сходимости является разумным, хотя Бокс, Дэвис и Свенн предлагают то, что они считают более "безопасной" проверкой.

Шаги этой процедуры представлены в виде блок-схемы на рис. 7.

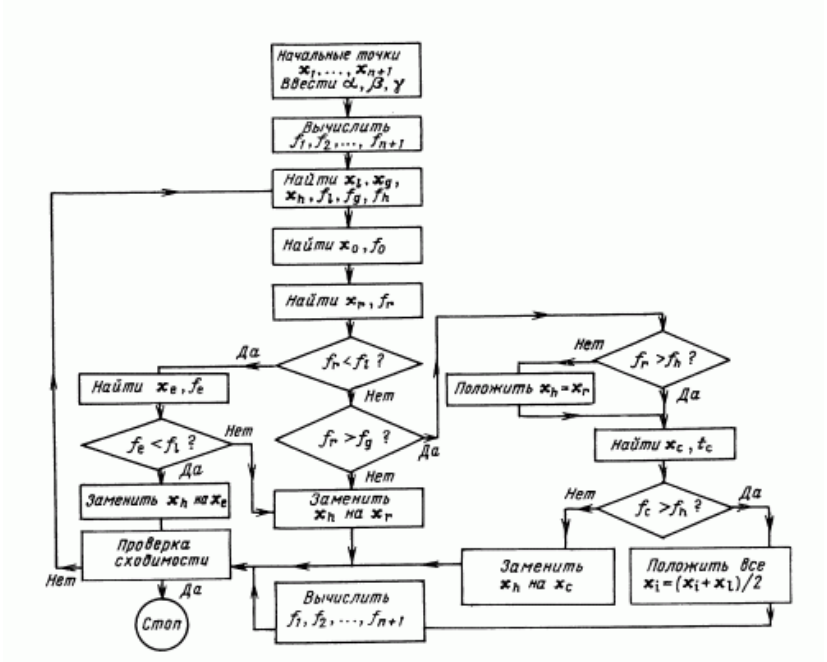


Рис. 7.

Коэффициенты α, β, γ в вышеприведенной процедуре являются соответственно коэффициентами отражения, сжатия и растяжения. Нелдер и Мид рекомендуют брать α=1, β=0,5, γ=2. Рекомендация основана на результатах экспериментов с различными комбинациями значений. Эти значения параметров позволяют методу быть эффективным, но работать в различных сложных ситуациях.

Начальный симплекс выбирается на наше усмотрение. В данном случае точка x_1 является начальной точкой, затем формируются точки

$$x_2 = x_1 + ke_1$$
$$x_3 = x_1 + ke_2 \qquad (10)$$
$$x_{n+1} = x_1 + ke_n$$

где k - произвольная длина шага, а e_j - единичный вектор.

6.7.3. Метод полного перебора (метод сеток)

Многомерные задачи, естественно, являются более сложными и трудоемкими, чем одномерные, причем обычно трудности при их решении возрастают при увеличении размерности. Для того чтобы лучше почувствовать это, возьмем самый простой по своей идее приближенный метод поиска наименьшего значения функции. Покроем рассматриваемую область сеткой G с шагом h (рис. 8) и определим значения функции в ее узлах. Сравнивая полученные числа между собой, найдем среди них наименьшее и примем его приближенно за наименьшее значение функции для всей области.

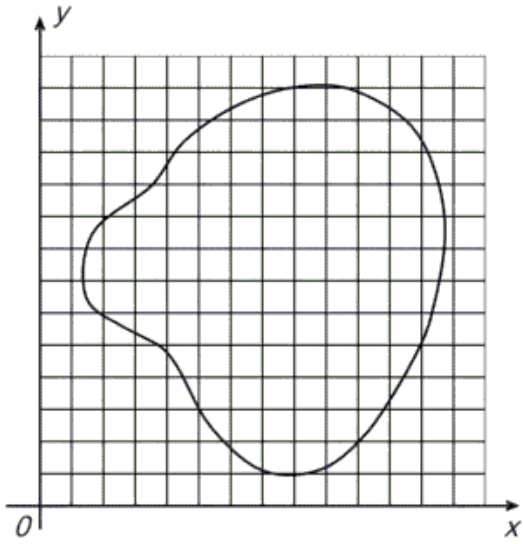


Рис. 8.

Как мы уже говорили выше, данный метод используется для решения одномерных задач. Иногда он применяется также для решения двумерных, реже трехмерных задач. Однако для задач большей размерности он практически непригоден из-за слишком большого времени, необходимого для проведения расчетов. Действительно, предположим, что целевая функция зависит от пяти переменных, а область определения G является пятимерным кубом, каждую сторону которого при построении сетки мы делим на 40 частей. Тогда общее число узлов сетки будет равно $41^5 \approx 10^8$. Пусть вычисление значения функции в одной точке требует 1000 арифметических операций (это немного для функции пяти переменных). В таком случае общее число операций составит 10^{11}. Если в нашем распоряжении имеется ЭВМ с быстродействием 1 млн. операций в секунду, то для решения задачи с помощью данного метода потребуется 10^5 секунд, что превышает сутки непрерывной работы. Добавление еще одной независимой переменной увеличит это время в 40 раз. Проведенная оценка показывает, что для больших задач оптимизации метод сплошного перебора непригоден. Иногда сплошной перебор заменяют случайным поиском. В этом случае точки сетки просматриваются не подряд, а в случайном порядке. В результате поиск наименьшего значения целевой функции существенно ускоряется, но теряет свою надежность.

6.7.4. Метод покоординатного спуска

Рассмотрим функцию двух переменных. Ее линии постоянного уровня представлены на рис. 9, а минимум лежит в точке (x_1^*, x_2^*). (Напомним, что линией постоянного уровня называется кривая в двумерном сечении пространства параметров (в данном случае в плоскости (x_1, x_2), значение функции на которой - константа). Простейшим методом поиска является метод покоординатного спуска. Из точки A мы производим поиск минимума вдоль направления оси x_1 и, таким образом, находим точку B, в которой касательная к линии постоянного уровня параллельна оси x_1. Затем, производя поиск из точки B в направлении оси x_2, получаем точку C, производя поиск параллельно оси x_1, получаем точку D, и т.д. Таким образом, мы приходим к оптимальной точке. Любой из одномерных методов, описанных ранее, может быть использован здесь для поиска вдоль оси. Очевидным образом эту идею можно применить для функций n переменных.

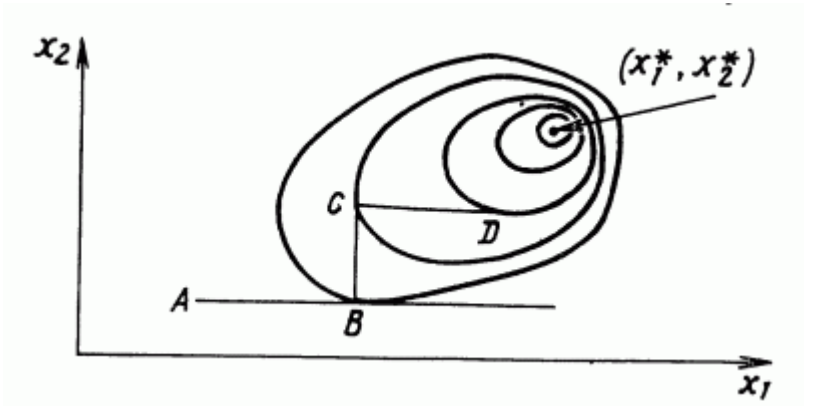


Рис. 9.

Рассмотрим данный метод более детально на примере некоторой целевой функции.

Пусть нужно найти наименьшее значение целевой функции $u=f(M)=f(x_1,x_2,...,x_n)$. Здесь через M обозначена точка n-мерного пространства с координатами $x_1,x_2,...,x_n$: $M=(x_1,x_2,...,x_n)$. Выберем какую-нибудь начальную точку $M_0 = (x_1^0, x_2^0, \ldots, x_n^0)$ и рассмотрим функцию f при фиксированных значениях всех переменных, кроме первой: $f(x_1^0, x_2^0, \ldots, x_n^0)$. Тогда она превратится в функцию одной переменной x_1. Изменяя эту переменную, будем двигаться от начальной точки $x_1 = x_1^0$ в сторону убывания функции, пока не дойдем до ее минимума при $x_1 = x_1^1$, после которого она начинает возрастать. Точку с координатами $f(x_1^1, x_2^0, x_3^0, \ldots, x_n^0)$ обозначим через M^1, при этом $f(M^0) \geq f(M^1)$.

Фиксируем теперь переменные:

$$x_1 = x_1^1, x_3 = x_3^0, ..., x_n = x_n^0$$

и рассмотрим функцию f как функцию одной переменной $x_2 : f(x_1^1, x_2, x_3^0, \ldots, x_n^0)$. Изменяя x_2, будем опять двигаться от начального значения $x_2 = x_2^0$ в сторону убывания функции, пока не дойдем до минимума при $x_2 = x_2^1$. Точку с координатами $(x_1^1, x_2, x_3^0, \ldots, x_n^0)$ обозначим через M^2, при этом $f(M^1) \geq f(M^2)$. Проведем такую же минимизацию целевой функции по переменным $x_3,x_4,\ldots,x_n$. Дойдя до переменной x_n, снова вернемся к x_1 и продолжим процесс.

Эта процедура вполне оправдывает название метода. С ее помощью мы построим последовательность точек $M^0,M^1,M^2,\ldots$ которой соответствует монотонная последовательность значений функции $f(M^0) \geq f(M^1) \geq f(M^2) \cdots$ Обрывая ее на некотором шаге k, можно приближенно принять значение функции $f(M^k)$ за ее наименьшее значение в рассматриваемой области (рис. 10).

Отметим, что данный метод сводит задачу поиска наименьшего значения функции нескольких переменных к многократному решению одномерных задач оптимизации. Если целевая функция $f(x_1,x_2,\ldots,x_n$ задана явной формулой и является дифференцируемой, то мы можем вычислить ее частные производные и использовать их для определения направления убывания функции по каждой переменной и поиска соответствующих одномерных минимумов.

На рис. 10 изображены линии уровня некоторой функции двух переменных $u=f(x,y)$. Вдоль этих линий функция сохраняет постоянные значения, равные 1, 3, 5, 7, 9. Показана траектория поиска ее наименьшего значения, которое достигается в точке O, с помощью метода покоординатного спуска.

При этом нужно ясно понимать, что рисунок служит только для иллюстрации метода. Когда мы приступаем к решению реальной задачи оптимизации, такого рисунка, содержащего в себе готовый ответ, у нас, конечно, нет.

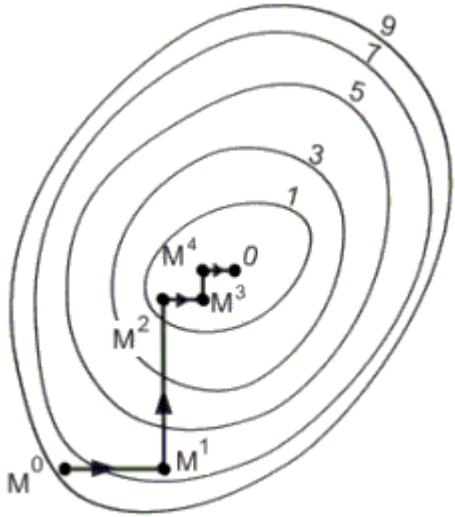


Рис. 10.

Теоретически данный метод эффективен в случае единственного минимума функции. Но на практике он оказывается слишком медленным. Поэтому были разработаны более сложные методы, использующие больше информации на основании уже полученных значений функции. Было предложено несколько функций, которые из-за своих свойств являются тестовыми для таких методов. Ниже приведено несколько примеров таких функций.

Функция Розенброка:

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2; \quad x^* = (1;1). \quad (11)$$

Функция Пауэлла:

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2X_3)^4 + 10(x_1 - x_4)^4; \quad x^* = (0;0;0;0). \quad (12)$$

Двумерная экспоненциальная функция:

$$f(x_1, x_2) = \sum_a [(e^{-ax_1} - e^{-ax_2}) - (e^{-a} - e^{-10a})]^2, \quad (13)$$

где $a = 0,1(0,1)1^*;\ x^* = (1;10)$.

6.8. Методы оптимизации первого порядка

6.8.1. Спуск по координатам

Все методы минимизации сводятся к построению траектории спуска $\{ M_k \}$ вдоль которой целевая флнкция убывает:

$$f(M_{k+1}) < f(M_k)$$

(или возрастает).

Опишем *координатный спуск.* Выберем нулевое приближение $M_0(\overset{(0)}{x_1}, \ldots. \overset{(0)}{x_n})$ и зафиксируем все значения координат, кроме первой, тогда $f(\vec{x})$

$$f(x_1, \overset{(0)}{x_2}, \ldots., \overset{(0)}{x_n}) \equiv \varphi_1(x_1)$$

становится функцией одного переменного.

Используя методы минимизации функции одного переменного, найдем точку ее минимума $\overset{(0)}{x_1}$ и совершим шаг из M_0 в $M_0^{(1)}(\overset{(1)}{x_1}, \overset{(0)}{x_2}, \ldots., \overset{(0)}{x_n})$.

На k-м шаге спуска: Из точки $M_0^{(k-1)}(\overset{(1)}{x_1}, \ldots, \overset{(1)}{x_{k-1}}, \overset{(0)}{x_k}, \ldots., \overset{(0)}{x_n})$.

спускаемся по x_k минимизируя

$$\varphi_k(x_k) \equiv f(\overset{(1)}{x_1}, \ldots, \overset{(1)}{x_{k-1}}, x_k, \overset{(0)}{x_{k+1}}, \ldots., \overset{(0)}{x_n}),$$

$$\overset{(1)}{x_k} : \varphi(\overset{(1)}{x_k}) = \min_{x_k} \varphi_k(x_k) \quad (1)$$

в точку

$$M_0^{(k)}(\overset{(1)}{x_1},\ldots,\overset{(1)}{x_k},\overset{(0)}{x_{k+1}},\ldots,\overset{(0)}{x_n}).$$

И так до тех пор, пока не выполним один цикл спуска по координатам Последнюю точку спуска назовем

$$M_1 \equiv M_0^{(n)}(\overset{(1)}{x_1},\ldots,\overset{(1)}{x_n}) \equiv M_1(\overset{(0)}{x_1},\ldots,\overset{(0)}{x_n}).$$

Траектория $\{M_k\}$ - траектория спуска, поскольку

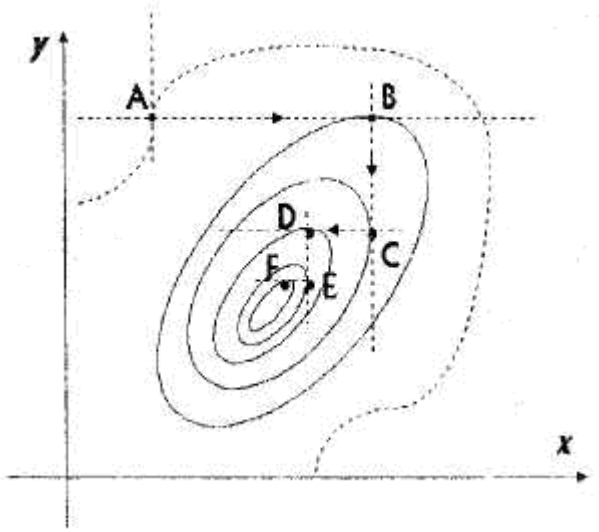
$$f(M_k) \le f(M_{k-1}).$$

В силу ограниченности снизу значений $f(x)$ значением $f(x^*) \equiv f^*$ (мы предполагаем, что экстремум существует), то

$$f_k \ge f^* \Rightarrow \lim_{k\to\infty} f_k = \tilde{f} \ge f^*$$

Будет ли здесь равенство, т. е. сходится ли спуск по координатам к минимуму и как быстро, зависит от функции $f(\vec{x})$ и выбранного начального приближения $\vec{x}_0$ (оно должно попасть в область влияния локального экстремума)

Рассмотрим трактовку координатного спуска на примере функции двух переменных:

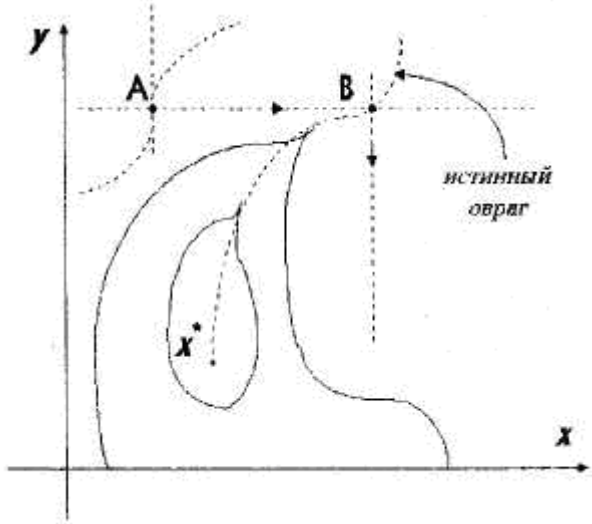


Двигаясь по прямой AB мы пересекаем линии уровня (x,y) = cons, при этом $f(x, y)$ либо возрастает, либо убывает в зависимости от направления движения. Только в одной точке B, где данная прямая

касается линии уровня, функция *f(x,y)* имеет минимальное значение в данном направлении (экстремум по *x* или по *y)*. Найдя такую точку, завершаем спуск по данному направлению.

Заметим, что в координатном спукске соответствующие направления взаимно ортогональны.

Если в рельефе наличествует «истинный», то спуск (в данном случае первый же спуск в точку *B)* приводит к попаданию на "дно" оврага. А поскольку он ориентирован достаточно произвольно, то дальнейший спуск может оказаться невозможным. Хотя минимум еще и не достигнут.



Если же $f(\vec{x})$ достаточно гладкая функция и минимум невырожден, hess f $(\vec{x}^*) > 0$, то в окрестности $\vec{x}^*$ рельеф котловинный и координатный спуск ведет нас к локальному минимуму при произвольном начальном приближении $\vec{x}_0$ в этой окрестности.

Рассмотрим достаточные условия сходимости координатного спуска на примере функции ДВУХ переменных:

Теорема 1. *Пусть D – множество уровня, ограниченное линией уровня f(x,y) = f_0, т.е.*

$$D = \{(x,y): \quad f(x,y) \le f(x_0,y_0)\},$$

замкнутая ограниченная область и в D функция f(x,y) дважды дифференцируема, причем

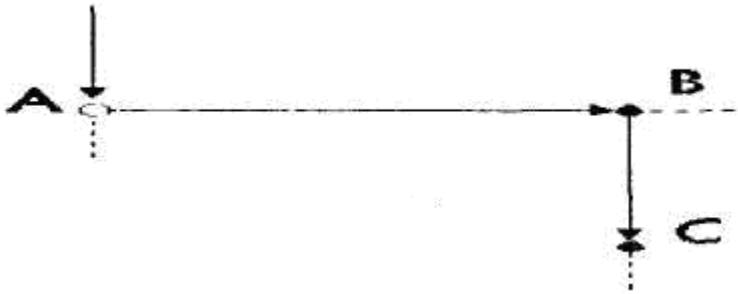
$$f_{xx} \ge a > 0;\ f_{yy} \ge b > 0;\ |f_{xy}| \le c \ и \ ab > c^2. \quad (2)$$

($G(x,y)\geq d>0$ в D. Используя критерий Сильвестра можно сформулировать многомерный аналог этого условия.)

Тогда траектория координатного спуска $\{M_k\}$ (1) *из произвольной точки* $M_0 \in D$ *сходится к локальному минимуму* x^* *в области* D.

Доказательство. Докажем сходимость grad $f(M_k)$ на траектории спуска $\{ M_k \}$. Проследим за изменением $| f_x |$ и $| f_y |$ на траектории спуска $\{M_k\}$. Поскольку $f(x,y)$ вдоль траектории спуска не возрастает, то все точки $M_k \in D_0$. Пусть предыдущий цикл спусков закончился в точке A, тогда

$$f_y(A) = 0. \quad |f_x(A)| = U \neq 0.$$



Попав в точку экстремума B на прямой AB получим следующие компоненты градиета

$$|f_y(B)| = V \neq 0 \quad f_x(A) = 0.$$

Теперь нетрудно получить, что

$$\begin{cases} U = |f_x(B) - f_x(A)| = |f_{xx}(\xi)| \cdot |x_B - x_A| \geq a \cdot \rho(A,B) \\ V = |f_y(B) - f_y(A)| = |f_{xy}(\eta)| \cdot |x_B - x_A| \leq c \cdot \rho(A,B) \end{cases} \Rightarrow \quad Uc \geq Va.$$

Спустившись далее по направлению BC в точку экстремума C, найдём

$$\begin{cases} V = |f_y(C) - f_y(B)| = |f_{yy}(\xi)| \cdot |y_C - y_B| \geq b \cdot \rho(B,C) \\ W = |f_x(C) - f_x(B)| = |f_{xy}(\eta)| \cdot |y_C - y_B| \leq c \cdot \rho(B,C) \end{cases} \Rightarrow \quad Vc \geq Wb.$$

Окончательно, за один цикл спуска, получаем

$$W \leq \frac{c}{b} V \leq \frac{c^2}{ab} U = q \cdot U,$$

причём, в силу условий теоремы 1, $q < 1$.

Итак, за один цикл спуск $| f_x |$ уменьшился в q раз. Аналогично, со сдвигом на 1/2 цикла, $| f_y |$ уменьшится в q раз. Выполнив n циклов координатного спуска получим, что

$$|f_x|_{(n)} \le q^n |f_x|_{(0)} \quad \Longrightarrow \quad |f_x| \underset{n\to\infty}{\longrightarrow} 0 \text{ и } |f_y| \underset{n\to\infty}{\longrightarrow} 0.$$

Далее, в окрестности точки экстремума x^* компоненты градиента можно разложить по формуле Тейлора

$$\begin{aligned} f_x(M) &= \underbrace{f_x(M^*)}_{\equiv 0} + \frac{\partial f_x}{\partial x}(M^*)\cdot\Delta x + \frac{\partial f_x}{\partial y}(M^*)\cdot\Delta y + \cdots \\ f_y(M) &= \underbrace{f_y(M^*)}_{\equiv 0} + \frac{\partial f_y}{\partial x}(M^*)\cdot\Delta x + \frac{\partial f_y}{\partial y}(M^*)\cdot\Delta y + \cdots \end{aligned} \quad \left| \begin{array}{l} \Delta x = x - x^* \\ \Delta y = y - y^*. \end{array} \right.$$

Пренебрегая в разложении слагаемыми высших порядков, получаем линейную систему относительно приращений координат Δx и Δy. По условию теоремы 1 гессиан $G(M^*) > 0$, тем самым полученная система совместна и можно выразить Δx и Δy через линейную комбинацию компонент градиента в точке $M = M_{(n)}$. При этом Δx, $\Delta y \to 0$ на траектории $\{M_k\}$, $M_k \to M^*$.

Итак:

- Вблизи точки экстремума M^* сходимость координатою спуска и по координатам, и по градиенту *линейная* (достаточно медленная, что с практической точки зрения плохо).
- По "циклам"' спусков можно делать ускорения по методу Эйткена;
- При попадании траектории спуска в разрешимый овраг расчет практически невозможен (слишком медленная сходимость при произвольной ориентации оврага относительно координатных осей) Поэтому выгоднее использовать методы, обладающие повышенным порядком точности

6.8.2. Градиентные методы

Во многих алгоритмах многомерной оптимизации так или иначе используется информация о градиентах. Проиллюстрируем это положение следующим простым примером. Представим себе, что альпинисту завязали глаза и сказали, что он должен добраться к вершине “унимодальной” горы. Даже если он не будет ничего видеть, он может это сделать, если все время будет двигаться вверх. Хотя любая тропа, которая ведет вверх, в конце-концов приведет его к вершине, кратчайшей из них будет самая крутая, если, правда, альпинист не натолкнется на вертикальный обрыв, который необходимо будет обойти. (Математическим эквивалентом обрыва на поверхности, которую создает целевая функция, являются те ее места, где поставленны условные ограничения). Вообразим, что задача оптимизации не содержит ограничений. Позднее мы включим их в схему поиска. Метод оптимизации, в основе которого лежит идея

движения по самой крутой тропе, называется *методом наискорейшего подъема или наибыстрейшего спуска. Вектор градиента перпендикулярный линии уровня и указывает направление к новой точке в пространстве проектирования.* Отметим, что градиентный метод в отличие от метода касательной к линии уровня можно использовать к любой унимодальной функции, а не только тех, в которых это свойство явным образом выражено.

В общем случае для траектории спуска $\{M_k\}$: $f_{k+1}<f_k$ при минимизации достаточно гладких функций можно сформулировать *достаточные* условия сходимости соответствующего метода спуска, характеризующие изменение функции f и ее градиента $\vec{g}$ =grad f на траектории $\{ M_k \}$

Пусть очередной шаг совершается вдоль направления $\vec{p}_k$ и приводит нас в точку M_{k+1}:

$$\vec{x}_{k+1} = \vec{x}_k + \vec{p}_k h_k.$$

Шаг h_k выбирается из условия минимальности $f(M)$ вдоль $\vec{p}_k$

$$h_k = \varphi(h_k) = \min_h \varphi(h) = \min_h f(\vec{x_k} + h\vec{p_k})$$

Сформулируем достаточные условия сходимости метода спуска **Теорема 2**. *Пусть*

1) $f(\vec{x})$ *дважды дифференцируемая функция;*

2) *множество уровня*

$$D(f(\vec{x}_0)) = \{\vec{x} : \ f(\vec{x}) \le f(\vec{x}_0)\}$$

ограничено и замкнуто;

3) *на каждой итерации*

а) направление $\vec{p}_k$ - *«существенное направление спуска»*

$$\exists \beta < 0, \quad \vec{p}_k \ \vec{g}_k \le \beta < 0$$

б) $f(x)$ *«существенно убывает», (т.е. выбрано соответствующее ограничение на шаг)*

$$\exists \mu_1, \quad \mu_2 \quad 0 < \mu_1 \le \mu_2 \le 1$$

$$-\mu_1 h_k \underbrace{\vec{g_k} \ \vec{p_k}}_{\text{отриц}} \le \Phi_k - \Phi_{k+1} \le -\mu_2 h_k \underbrace{\vec{g_k} \ \vec{p_k}}_{\text{число}}$$

Тогда

$$\lim_{k\to\infty} \| \vec{g}_k \| = 0; \ (M_k \to M^*)$$

т.е. метод спуска обладает сходимостью (как правило — линейной).

В основном соответствующие методы спуска отличаются выбором очередного направления $\vec{p}_k$ и шага h_k.

Метод "наискорейшего" спуска. Рассмотрим линейную аппроксимацию целевой функции) $f(\vec{x})$ в окрестности точки $\vec{x}_k$. Опираясь на формулу Тейлора:

$$\Phi(\overrightarrow{x_k} + \overrightarrow{p}) = \Phi(\overrightarrow{x_k}) + (grad\Phi(\overrightarrow{x_k}), \overrightarrow{p}) + o(\|\overrightarrow{p}\|),$$

с определенной точки зрения (локально!) естественно искать направление, по которому $\frac{\partial f}{\partial p} = \vec{g}_k \cdot \vec{p}$ наибольшее по модулю отрицательное число. Это направление в первом порядке по $\|\vec{p}\|$ обеспечивает наибольшее убывание функции *f*.

Итак, необходимо найти направление $\vec{p}$

$$\begin{cases} \min(\overrightarrow{g_k} \cdot \overrightarrow{p}) \\ \|\overrightarrow{p}\| = 1 \end{cases}$$

задача на условный экстремум для $\vec{p}$

Решение полученной задачи зависит от вида рассматриваемой нормы. Если выбрать *C*-энергетичеекую норму $\|\vec{p}\|^{2} = (C\vec{p}, \vec{p})$, где $C > 0$ и симметрична, тогда направление $\vec{p}$ (с точностью до нормировочной *Const)*

$$\vec{p} = -C^{-1} \cdot \vec{g}_k .$$

Для евклидовой нормы - $C = E$ и $p = -\vec{g}_k$, что приводит нас к *методу наискорейшего спуска.*

$$\begin{cases} \overrightarrow{x}_{k+1} = \overrightarrow{x_k} - h_k \overrightarrow{g_k} \\ h_k : \varphi(h_k) = \min\limits_{h} f(\overrightarrow{x_k} - h\overrightarrow{g_k}) \end{cases} \qquad (1)$$

Замечания:

1) При таком выборе $\vec{p}_k$ и h_k (1) траектория спуска перпендикулярна линии уровня $f(x_k)$ в точке x_k.

2) Но сходимости *наискорейший спускав* лучше, чем координатный спуск, т.е. он обладает лишь линейной сходимостью.

3) Анализ сходимости наискорейшего спуска на квдратичной функции с симметричной и положительно определенной матрицей (что характерно для гессиана в окрестности невырожденного минимума)

$$\Psi(x) = \frac{1}{2}(Ax, x) + (\vec{b}, x) + C: \quad A > 0, \quad A^T = A$$

дает лишь линейную сходимость. Поскольку $A>0$, $A^T = A$ следовательно все собственные значения матрицы A положительны $\forall_i \ \lambda_i(A)>0$. Сходимосгь метода наискорейшего спуска характеризуют величиной

$$\varkappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} = \|A\| \cdot \|A^{-1}\| = CondA$$

$$\Psi(x_{k+1}) - \Psi(x^*) \simeq \left(\frac{\varkappa - 1}{\varkappa + 1}\right)^2 (\Psi(\vec{x_k}) - \Psi(x^*)). \qquad (2)$$

Полученная оценка скорости сходимости, например для $\varkappa$ = 100 (хорошая обусловленность матрицы *A)* даёт $q \sim 0,96$(!) и нужны сотни итераций для уменьшения погрешности на порядок.

Расчетные формулы наискорейшего спуска (1) в этом случае принимают вид:

$$\vec{g} = A\vec{x} + \vec{b}: \ hess\Psi = A \quad \Rightarrow \quad \vec{p_k} = -\vec{g_k},$$

$$\psi(h) = \Psi(\vec{x} + h\vec{p}_k) = \Psi(\vec{x}) + h(Ax + b, \vec{p}_k) + \frac{h^2}{2}(A\vec{p}_k, \vec{p}_k),$$

$$\frac{\partial \psi}{\partial h} = 0 \Leftrightarrow h_k = \left\{ \begin{array}{c} \text{получить} \\ \text{самостоятельно} \\ \text{расчетные} \\ \text{формулы} \end{array} \right\}. \qquad (3)$$

Тем не менее:

1) Необходимо бесконечное число итераций для нахождения экстремума даже в случае квадратичной функции.

2) Метод наискорейшего спуска не рекомендуется как серьезная минимизационная процедура. Дело в том, что свойство наискорейшего спуска является лишь *локальным* свойством, поэтому необходима частая смена направлений спуска и относительно малый шаг движения по каждому направлению, что и приводит в итоге к неэффективной вычислительной процедуре (например в случае разрешимого оврага).

3) Метод наискорейшею спуска невозможно адаптировать для использования информации о вторых производных $f(\vec{x})$.

Чтобы лучше понять идею градиентных методов, более конкретно остановимся на свойствах градиентов. Рассмотрим систему независимых единичных векторов e_1, e_2, e_3, ..., e_N, которые направленны вдоль осей координат $x_1, x_2, x_3, \ldots, x_N$, которые есть в то же время проектными параметрами. Вектор градиента произвольной целевой функции $F = (x_1, x_2, x_3, \ldots, x_N)$ имеет вид

$$\frac{\partial F}{\partial x_1}e_1 + \frac{\partial F}{\partial x_2}e_2 + \ldots + \frac{\partial F}{\partial x_N}e_N,$$

где частные производные вычисляются в рассматриваемой точке. Этот вектор направлен вверх, в направлении подъема; обратный ему вектор указывает направление спуска. Единичный вектор градиента часто представляют в виде

$$s_1e_1 + s_2e_2 + s_3e_3 + \ldots + s_Ne_N,$$

где

$$s_j = \frac{\dfrac{\partial F}{\partial x_j}}{\left[\sum_{j=1}^{N}\left(\dfrac{\partial F}{\partial x_j}\right)^2\right]^{1/2}}. \qquad (4)$$

Иногда характер целевой функции бывает достаточно хорошо известен, чтобы можно было вычислить компоненты вектора градиента путем непосредственного дифференцирования. Если таким способом частные производные получить невозможно, то можно найти их приближенные значения в непосредственной окрестности рассматриваемой точки:

$$\frac{\partial F}{\partial x_j} = \frac{F(x_1, x_2, \ldots, x_j + \Delta, \ldots, x_N) - F(x_1, x_2, \ldots, x_j, \ldots, x_N)}{\Delta}.$$

Здесь Δ - небольшое смещение в направлении x_i. Эту формулу часто называют "*приближением секущей*". Полученную информацию о

направлении градиента можно использовать различным образом для построения алгоритма поиска.

Постановка задачи оптимизации градиентными методами: минимизация функции $F(x_1, x_2, x_3, \ldots, x_N)$ с N проектными параметрами с помощью ЭВМ решается итерационными методами. Решение задачи начинается с выбора начальных значений $x_i^{[0]}$ ($i = 1, 2, \ldots, N$), которые как обычно определяются из условий решаемой задачи, и потом строят последовательные приближения, используя итерационную формулу :

$$x_i^{[j+1]} = x_i^{[j]} + \lambda^{[j]} s_i^{[j]}, \quad (i = 1, 2, \ldots, N;\ j = 0, 1, 2, \ldots), \qquad (5)$$

где $\lambda^{[j]}$ - величина шага итерации по каждому из параметров x_i;

$s_i^{[j]}$ - параметр выбора “направления”, который обычно определяется по итерационной формуле.

Данная формула обеспечивает сходимость исследуемой функции к некоторому решению $\overline{x}_k$ при $j \to \infty$. Величина шага $\lambda^{[j]}$ на каждой j-й итерации определяется одним из методов оптимизации однопараметрической оптимизации, например методом деления отрезка пополам или методом “золотого сечения” или Фибоначчи.

Наискорейший подъем с использованием одномерного поиска

В некоторых методах поиска информация о градиенте используется для ведения одномерного поиска в направлении наискорейшего подъема или спуска, причем используется соотношение

$$x_{i,нов} = x_{i,ст} + \lambda s_i, \qquad (6)$$

где λ- величина шага, значение которого определяются в направлении градиента.

Получив одномерный оптимум в направлении данного градиента, находят новый градиент и повторяют процесс до тех пор, пока следующие вычисления позволяют улучшать полученный результат. Главное преимущество этого метода заключается в том, что параметр λ можно использовать в качестве независимой переменной для поиска по методу Фибоначчи, и это обеспечивает высокую эффективность метода. Другое важное преимущество методов, которые рассматриваются, заключается в том, что они позволяют отходить от седловин точек поверхности, которая описывается целевой функцией (рис. 1).

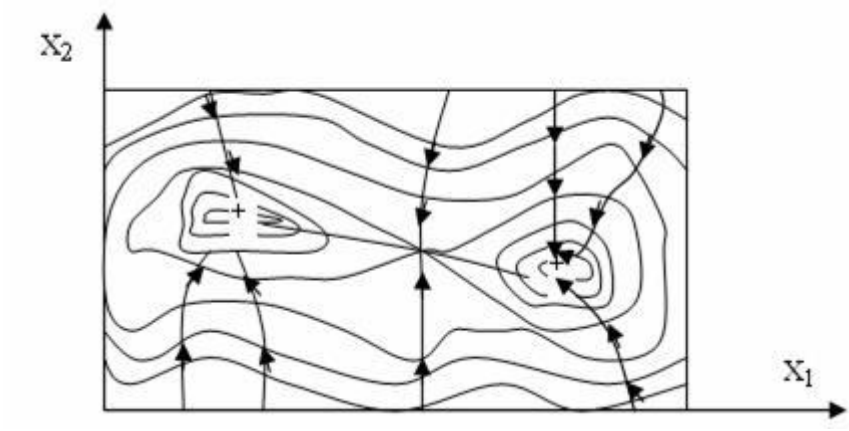


Рис. 1. Бимодальная целевая функция

Отметим, однако, что, как видно из рисунку, для мультимодальных функций градиентные методы позволяют найти лишь локальный оптимум. Поэтому, если характер поверхности недостаточно хорошо известен, то необходимо подвергнуть испытанию несколько начальных точек и убедиться, что во всех случаях получается одно и то же оптимальное решение. Другой причиной, которая снижает эффективность градиентных методов, являются излом линий уровня целевой функции. Так как такие точки соответствуют разрыву в наклоне линии контура, то здесь возможны ошибки в определении направления дальнейшего поиска. Поэтому поиск может замедлиться и идти зигзагами поперек линии излома, а время необходимое для получения решения, будет на столько большим, что счет придется прекратить. В действительности большинство исследуемых поверхностей имеет одну или больше линий излома, которые нередко

проходят через точку оптимума. Поэтому, натолкнувшись на линию излома, нужно в дальнейшем двигаться вдоль нее.

Алгоритм наискорейшего спуска

Данный алгоритм основан на использовании итерационной формулы

$$x_i^{[j+1]} = x_i^{[j]} + \lambda^{[j]} s_i^{[j]},$$

где

$$s_i^{[j]} = -\frac{\partial F}{\partial x_i},$$

причем все производные вычисляются при $\lambda_i = x_i^{[j]}$;

$\lambda^{[j]}$ - величина шага, значение которого изменяется (уменьшается или вычисляется) методом половинного деления.

Алгоритм метода наискорейшего спуска:

1. Выбираем начальные значения координат вектора

$$\overline{x^0} = \left(x_1^{(0)}, x_2^{(0)}, \ldots x_N^{(0)}\right)$$

и начальные значения шага итерационного процесса λ, которые обычно выбираются из условий решаемой конкретной задачи. Хотя общих правил выбора $\bar{x}^0$ нет, однако если есть дополнительная информация об области расположения минимума целевой функции, то $\bar{x}^0$ выбираем в этой области.

2. Задаем номер итерации $k = 1$.

3. Вычисляем значение целевой функции в точке с координатами $\bar{x}^0$.

4. Вычисляем значение градиента s_i.

5. Вычисляем норму вектора градиента *NG*.

6. Если |*NG*| < заданной ε, то итерационный процесс заканчивается и оптимум найден.

7. Если условие |*NG*| < ε не выполняется, то определяются новые координаты вектора $\bar{x}^1$, которые получаются при движении к минимуму целевой функции с шагом λ (рис. 2).

8. Сравниваем два значения целевой функции в двух точках с координатами векторов $\bar{x}^0$ и $\bar{x}^1$ по формуле

$$f(\bar{x}^1) < f(\bar{x}^0), \qquad (7)$$

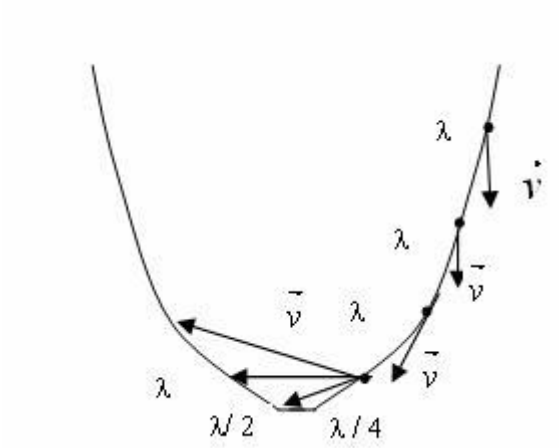


Рис. 2. Последовательность движения к минимуму с заданным шагом λ.

9. Если условие не выполняется, то шаг был выбран неверно, т.е. с этим шагом перескочили через оптимум и шаг нужно уменьшить, например, в два раза λ=1/2 и переходим к пункту 7 (рис. 2).

10. Если условие (7) выполняется, то запоминаем координаты вектора $\bar{x}^1$ и переходим к пункту 4.

Схема алгоритма описанного метода представлена на рис. 3.

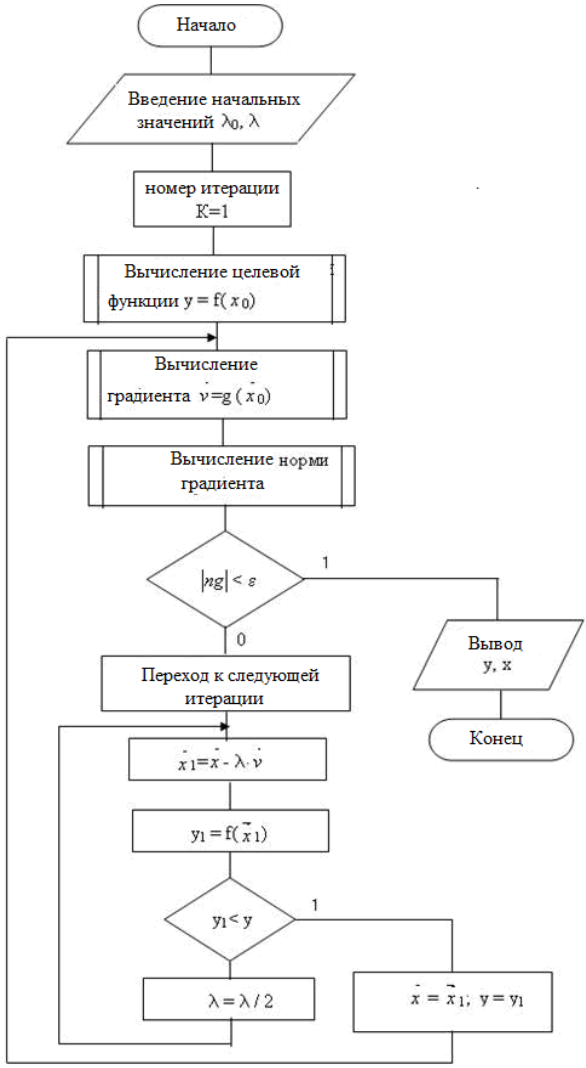


Рис. 3. Схема алгоритма метода наискорейшего спуска

6.8.3. Метод Флетчера – Ривса

Этот метод позволяет найти минимум нелинейной целевой функции многих переменных вида

$$M = F(x_1, x_2, \ldots, x_N)$$

при отсутствии ограничений. Метод основан на применении частных производных целевой функции по независимым переменным и переопределен для исследования унимодальних функций. С его помощью можно исследовать и мультимодальные функции, однако в этом случае следует брать несколько входных точек и проверять, одинаково или во всех случаях решение. Схема алгоритма метода Флетчера - Ривса представленная на рис.4.

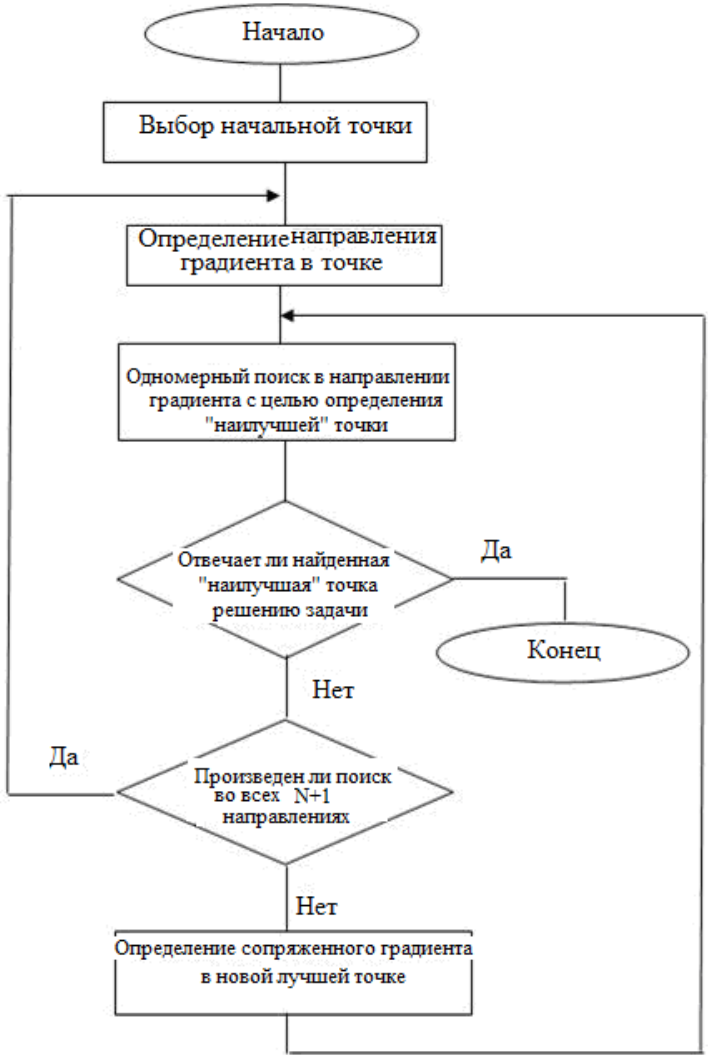


Рис. 4. Схема алгоритма метода Флетчера - Ривса

Выполняется он следующим образом. Вначале выбирается подходящая начальная точка пространства проектирования и путем вычисления компонент вектора градиента определяется направление наискорейшего спуска. Индекс k=1 соответствует входной точке.

После этого в направлении наискорейшего спуска ведется одномерный поиск по формуле

$$s_i^{(k)} = \frac{-\left(\frac{\partial F}{\partial x_i}\right)^{(k)}}{\left[\sum_{j=1}^{N}\left(\frac{\partial F}{\partial x_j}\right)^2\right]^{\frac{1}{2}}}, \; i = 1, 2, ..., N,$$

$$x_{i,\,new} = x_{i,\,cm} + \lambda s_i, \; i = 1, 2, ..., N,$$

где λ – смещение в направлении вектора градиента. Найдя минимум в этом направлении, определяют направления новых единичных векторов, которые несколько отличаются от направления нового вектора градиента и представляют собой линейные комбинации вектора градиента на данном шаге и вектора градиента, полученного на предыдущем шаге. Новые компоненты единичных векторов записываются в виде

$$x_i^{(k+1)} = \frac{-\frac{\partial F^{(k+1)}}{\partial x_i} + \beta^{(k)} s_i^{(k)}}{\left\{\sum_{j=1}^{N}\left[-\left(\frac{\partial F}{\partial x_j}\right)^{(k+1)} + \beta^{(k)} s_j^{(k)}\right]^2\right\}^{\frac{1}{2}}}, \; i = 1, 2, ..., N, \qquad (8)$$

где

$$\beta^{(k)} = \frac{\sum_{i=1}^{N}\left[\left(\frac{\partial F}{\partial x_i}\right)^{(k+1)}\right]^2}{\sum_{i=1}^{N}\left[\left(\frac{\partial F}{\partial x_i}\right)^{(k)}\right]^2}. \qquad (9)$$

Индекс k указывает на последовательность вычислений в процессе итераций. Новые направления называются «сопряженными» и соответствуют текущей локальной квадратичной аппроксимации функции, а фактически представляют собой движение по дну оврага (рис. 5).

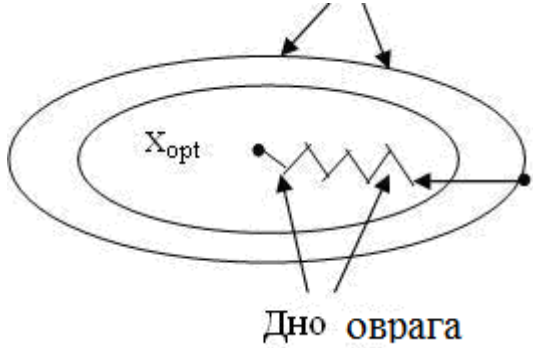


Рис. 5. Изменение направлений движения $\overline{S}_i$ по дну оврага

После этого по новому направлению (другому склону оврага) проводят одномерный поиск и, найдя минимум, проверяют, достигнута ли необходимая степень сходимости. Если проверка показывает, что это так, то счет прекращается. В противном случае определяют новые сопряженные направления, k увеличивают на единицу и продолжают процесс до тех пор, пока не будет обеспечена сходимость или пока поиск не будет проведен по всему N +1 направлениям. Закончив цикл поиска по N +1 направлениям, начинают новый цикл, в котором опять используется направление наискорейшего спуска. Особенность этого алгоритма заключается в том, что он позволяет использовать преимущества градиентных методов, которые проявляются при исследовании целевой функции с прерывистыми производными. Так как N+1 направлений поиска второй совокупности отличаются от направлений единичных векторов градиента, то поиск не «зависает на перегибе», а идет вдоль линии, которая соединяет точки перегибов линии уровня, которая, как правило, проходит через точку оптимума. Вообще можно утверждать, что методы, основанные на определении новых направлений поиска на основе накопленных данных о локальном поведении функции, по самой своей природе более эффективны, чем методы, в которых направление поиска задается

заранее. Именно поэтому метод Флетчера - Ривса обладает большими преимуществами по сравнению с методами наискорейшего спуска или подъема. Его недостаток заключается в том, что он является более сложным чем указанные методы, и требует разработки более сложных программ.

6.8.4. Метод Девидона – Флетчера – Пауэлла

Метод Девидона - Флетчера - Пауэлла представляет собой алгоритм оптимизации, приспособленный для отыскания безусловного минимума целевой функции, которая зависит от нескольких переменных и имеет вид

$$M = F(x_1, x_2, \ldots, x_N) \tag{10}$$

Необходимые частные производные целевой функции по независимым переменным. Поскольку в основе метода лежит допущение об унимодальности целевой функции, в тех случаях, когда есть основания допускать, что она не является таковой, необходимо брать несколько входных точек. На рис. 6 представлена схема алгоритма метода Девидона - Флетчера - Пауэлла.

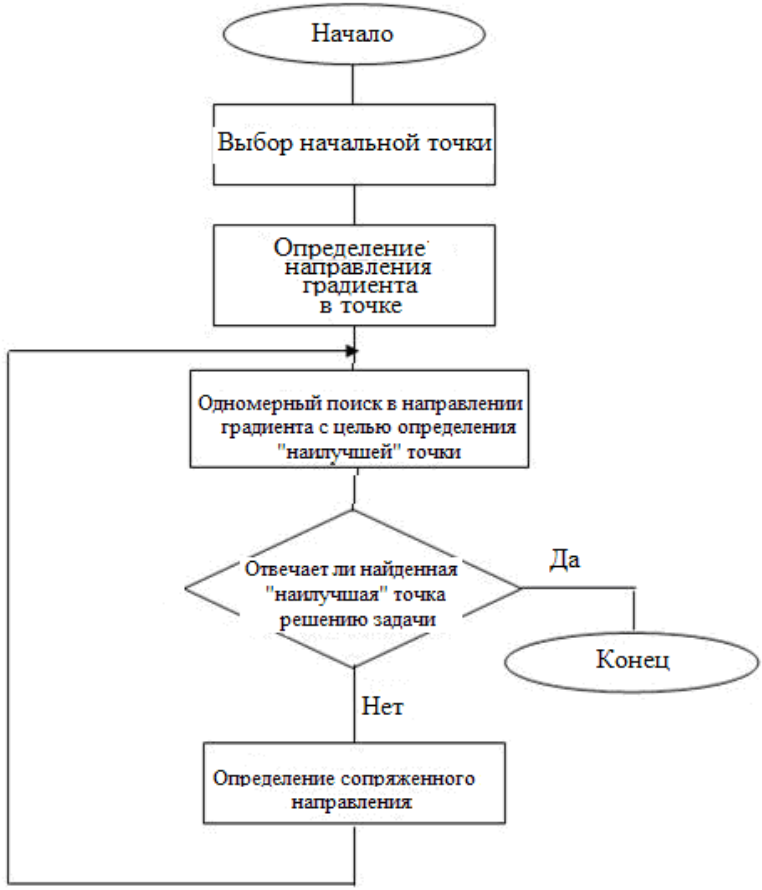


Рис. 7. Схема алгоритма метода Девидона – Флетчера – Пауэлла

Вначале в пространстве проектирования выбирают пригодную начальную точку. После этого, вычисляя состав вектора градиента определяют направление поиска.

$$s_i^{(k)} = \frac{\sum_{j=1}^{N} \underline{H}_{i,j} \left(\frac{\partial F}{\partial x_j} \right)^{(k)}}{\left\{ \sum_{i=1}^{N} \left[\sum_{j=1}^{N} \underline{H}_{i,j} \frac{\partial F}{\partial x_j} \right]^2 \right\}^{\frac{1}{2}}}, \; i = 1, 2, \ldots, N,$$

Здесь k – номер итерации, а $\underline{H}_{i,j}$ – элементы симметричной положительно определенной матрицы размерности $N \times N$. В процессе итераций эта матрица обращается в матрицу, обратную матрицы Гессе, элементами которой являются вторые частичные производные целевой функции. Поскольку обычно матрица заранее неизвестна, то в качестве начальной можно воспользоваться любой симметричной положительно определенной матрицей. Как правило, берут простейшую из них - единичную матрицу. В этом случае поиск начинается вдоль линии наискорейшего спуска. Одномерный поиск ведется вдоль входного направления в соответствии с соотношением

$$x_{i,new} = x_{i,cm} + \lambda S_i, i = 1,2,\ldots,N, \qquad (11)$$

где λ – величина шага в направлении поиска. Найдя одномерный оптимум, проверяют результат на сходимость и, если она достигнута, поиск прекращают. В противном случае для дальнейшего поиска выбирают новое направление, причем используют бывшее соотношение и новую матрицу $\underline{H}$, которая определяется формулой

$$\underline{H}^{(k+1)} = \underline{H}^{(k)} + \underline{A}^{(k)} - \underline{B}^{(k)}. \qquad (12)$$

Элементы матриц $\underline{A}^{(k)}$ и $\underline{B}^{(k)}$, которые имеют размерность $N \times N$ и вычисляются по формулам

$$\underline{A}^{(k)} = \frac{\Delta x^{(k)} \left(\Delta x^{(k)} \right)}{\left(\Delta x^{(k)} \right) \Delta G^{(k)}}, \qquad (13)$$

$$\underline{B}^{(k)} = \frac{\underline{H}^{(k)}\underline{\Delta G}^{(k)}\left(\underline{\Delta G}^{(k)}\right)^{t}\underline{H}^{(k)}}{\left(\underline{\Delta G}^{(k)}\right)^{t}\underline{H}^{(k)}\underline{\Delta G}^{(k)}} \tag{14}$$

где верхним индексом t обозначенны транспонированные матрицы, а $\underline{\Delta x}^{(k)}$ и $\underline{\Delta G}^{(k)}$ – векторы-столбцы разностей значений x_i и градиентов в двух точках. Векторы-столбцы определяются выражениями

$$\underline{\Delta x}^{(k)} = \underline{x}^{(k+1)} - \underline{x}^{(k)},$$
$$\underline{\Delta G}^{(k)} = \frac{\partial F^{(k+1)}}{\partial \underline{x}} - \frac{\partial F^{(k)}}{\partial \underline{x}}.$$

В соответствии с правилами матричного вычисления числительные выражений для $\underline{A}^{(k)}$ и $\underline{B}^{(k)}$ представляют собой матрицы размерности N×N, а знаменатели являются скалярами. Определив новое направление поиска, проводят одномерный поиск и продолжают итерационный процесс. При выполнении алгоритма, который описывается, поиск после первой попытки ведется в тех направлениях, в которых целевая функция в ближайшей окрестности имеет значения, которые приближаются к оптимальному. Лишь в редких случаях эти направления совпадают с направлением градиента. Поэтому данный алгоритм часто называют методом «отклоненного» градиента. Указанное свойство метода Девидона – Флетчера – Пауэлла позволяет обходить трудности, которые связаны с разрывами производных в пространстве проектирования. Считается, что этот метод является наиболее эффективным из всех градиентных методов. В отличие от метода Флетчера – Ривса он дает полную информацию о кривизне поверхности целевой функции в точке минимума, однако при этом требуется больший объем памяти и большее время счета для обработки матрицы $\underline{H}$.

6.8.5. Метод конфигураций Хука – Дживса

Этот метод облегчает поиск и не требует вычисления производных. Поиск ведется вдоль линий разрыва производных в предположении, что смещения в пространстве проектирования, которые оказались удачными на ранней стадии поиска, могут привести к успеху и на его

более поздних стадиях. Метод Хука - Дживса переопределен для поиска минимума унимодальной функции многих переменных

$$M = F(x_1, x_2, \ldots, x_N) \tag{15}$$

при отсутствии ограничений. На рис. 8 представленная схема алгоритма этого метода.

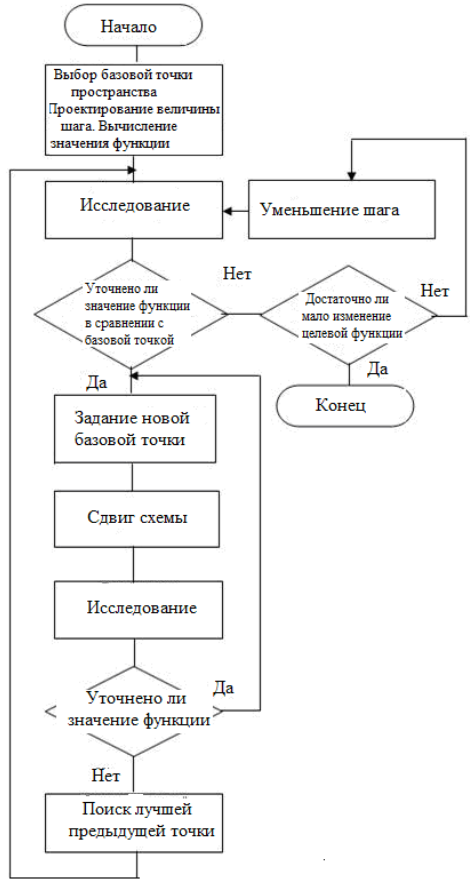


Рис. 8. Схема алгоритма метода конфигураций Хука - Дживса

Выполняется он следующим образом. Вначале выбирается входная базовая точка пространства проектирования и величины шагов, которые будут использованы при исследовании функции. После этого в соответствии со схемой рис. 9 проводится исследование с заданным приростом в направлениях, соответствующих всем независимым переменным. Там, где получено уточненное значение функции, размещают новую временную базовую точку. Закончив этап исследования, выбирают новую базовую точку и выполняют «сдвиги схемы». Эта операция заключается в экстраполяции вдоль линии, которая соединяет новую и бывшую базовые точки. Расстояние сдвига новой базовой точки несколько превышает расстояние между двумя бывшими базовыми точками. Математически экстраполяция определяется формулой (16)

$$x_{i,0}^{(k+1)} = x_i^{(k+1)} + \alpha\left(x_i^{(k+1)} - x_i^{(k)}\right) \qquad (16)$$

где $x_{i,0}^{(k+1)}$ – новая временная базовая точка, или «точка роста», I – переменный индекс, k – порядковый номер стадии поиска, а α – коэффициент усиления, значение которого больше или равно единице. После этого исследуют окрестность новой временной базовой точки, чтобы выяснить, не содержит ли она точку, приняв которую за следующую базовую можно приблизиться к оптимальному решению. Этот поиск также ведется по схеме, которая показана на рис. 9.

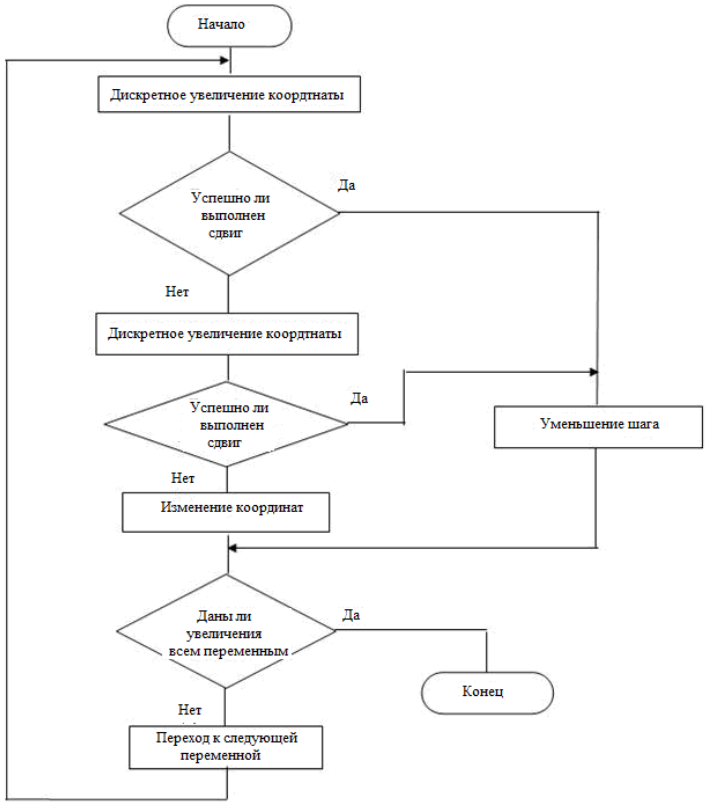


Рис. 9. Алгоритм исследования целевой функции на основе метода Хука-Дживса

Если найденная временная точка роста или одна из соседних с ней точек имеет преимущество перед другими, то вся процедура повторяется с использованием ее в качестве базовой. Благодаря введению коэффициента усиления, каждое следующее исследование окрестности точки осуществляется на все большем и большем отдалении от входной точки до тех пор, пока в процессе поиска не окажется пройденным пик или линия разрыва производной. В этом случае возвращаются к предыдущей «лучшей базовой точке», суживают область исследования и повторяют весь процесс снова. Если шаг, который уменьшается, последовательно оказывается меньшим за

некоторую заранее заданную величину и при этом отсутствует заметное изменение значения целевой функции, поиск прекращается. После нескольких изменений направления поиска метод Хука - Дживса обеспечивает совпадение распределения расчетных точек с линией разрыва производных. Обычно после завершения выбора схемы поиска сдвига на каждом следующем шаге увеличивается, пока не превысит величину входного шага в 10 или даже в 100 раз. Поэтому в случае, когда сдвиг оказывается неудачным, единственное средство продолжить поиск - возвратиться к наиболее удачной из базовых точек и начать все сначала. Тот факт, что данный алгоритм обладает свойством «ускоряться», оказывает содействие повышению его общей эффективности. Второе преимущество метода Хука - Дживса - возможность получения с его помощью приближенного решения, качество которого непрерывно повышается на всех стадиях численного решения. Особенно явным образом преимущества подобных средств оказываются при отыскании екстремумов на гиперповерхностях, которые содержат глубокие узкие впадины, т.е. тогда, когда градиентние методы неэффективны.

6.8.6. Метод конфигураций Розенброка

Метод конфигураций Розенброка основан на поиске минимума вдоль линий разрыва производных и часто оказывается эффективным, когда другие методы не позволяют получить решения. Его нередко называют «методом вращения осей координат», поскольку исследование в окрестности выбранной точки ведется именно таким методом. В отличие от предыдущих методов, в которых входным переменным предоставляют независимые приросты, в методе Розенброка система координат поворачивается так, чтобы одна из осей была направлена вдоль линии разрыва производных, положение которой определяется в результате предыдущего исследования. Остальные оси образуют с ней ортогональную систему координат. Метод Розенброка основан на предположении об унимодальности целевой функции и переопределен для отыскания минимума функции многих переменных вида

$$M = F(x_1, x_2, \ldots, x_N) \qquad (17)$$

при отсутствии ограничений. На рис. 10 показанная схема алгоритма, который используется в этом методе.

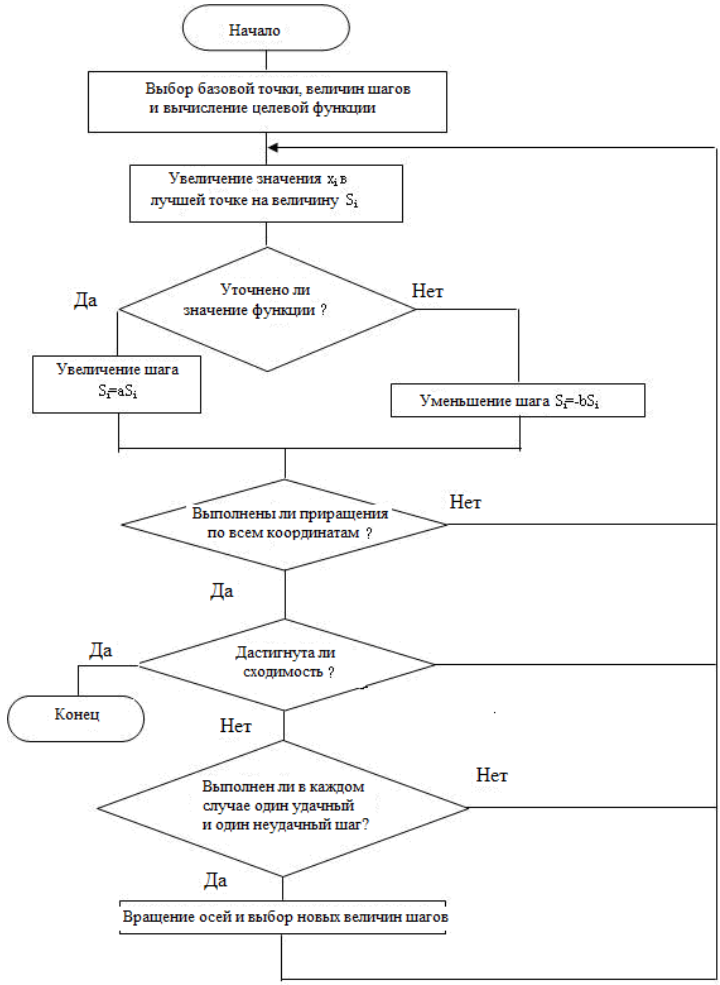


Рис. 10. Блок-схема алгоритма метода конфигураций Розенброка

Выполняется он следующим образом. Вначале выбирают начальную точку, задают начальные величины шагов $S_i (i = 1,2,..,N)$ и вычисляют целевую функцию. После этого каждой переменной x_i дают прирост S_i в направлении, параллельном к соответствующей оси координат в пространстве проектирования, и снова вычисляют

целевую функцию F . Если ее новое значение оказывается меньшим за предыдущее, то сдвиг считается удачным и следующий шаг увеличивается в соответствии с формулой

$$S_j = \alpha S_j, \tag{18}$$

где α>1. Если же новое значение F оказывается большим за предыдущее, то сдвиг считается неудачным и следующий шаг определяется по формуле

$$S_j = -\beta S_j, \tag{19}$$

где β<1 .

Осуществив сдвиг по всем переменным, проверяют сходимость и, если она достигнута, поиск прекращают. В противном случае вводят дополнительную проверку, чтобы выяснить, были ли сделанные хотя бы один успешный и единственных чисел безуспешный сдвиг в каждом направлении. Если сходимость не достигнута, то вся процедура повторяется, начиная с первой переменной. При этом оси вращают так, чтобы входное направление поиска совпало с наиболее перспективным из прежде рассмотренных направлений. После этого выбирают новые значения шагов и продолжают поиск по всем переменным, пользуясь новой системой координат.

В отличие от других, данный алгоритм нацелен на поиск оптимальной точки в каждом направлении, а не просто на фиксированном сдвиге по всему направлению. Величина шага в процессе поиска непрерывно изменяется в зависимости от конфигурации рельефа поверхности.

6.9. Методы второго порядка

Ньютоновские методы. Эта группа методов основана на более точной аппроксимации целевой функции в окрестности точки $\vec{x}_k$

$$f(\vec{x}_k + \vec{p}\ ) = \underbrace{f(x_k) + g_k \cdot \vec{p} + \frac{1}{2}(G_k \vec{p} \cdot \vec{p})}_{\Psi(\vec{p})} + o(\|\vec{p}\ \|^2).$$

Минимизируемая функция $\Psi(\vec{p}\ )$. Соответствующее направление и шаг берут из условия минимума $\Psi(\vec{p}\ )$:

$$\left.\begin{array}{l} grad\Psi = 0 \quad \Leftrightarrow \quad G_k \vec{p} + \vec{g_k} = 0; \quad \Leftrightarrow \quad \underbrace{\vec{p_k} = -G_k^{-1} \cdot \vec{g_k}}_{\text{Ньютоновское направление}} \\ \vec{x}_{k+1} = \vec{x_k} + \vec{p_k} = x_k - G_k^{-1} \cdot \vec{g_k} \end{array}\right\} \quad (1)$$

• Для квадратичной целевой функции $\Psi(\vec{p}\ )$ метод (1) решает задачу минимизации за одну (!) итерацию.

• В окрестности невырожденного экстремума имеет *квадратичную* сходимость (гессиан $G_k > 0$ и симметричен).

• Ньютоновское направление - это направление *наискорейшего* спуска в G-энергетической метрике

$$\|\vec{p}\| = \sqrt{(G\vec{p}, \vec{p})}.$$

• Существенным является то, что на каждом шаге необходимо решать систему линейных уравнений (1) для определения *нътоновского направления* очередной итерации.

• При модификации метода Ньютона, когда гессиан фиксируется на определенное число итераций G_{k_0} - в методе Ньютона-Рафсона — существенен алгоритмический выигрыш, но при этом обеспечена лишь линейная сходимостъ метода.

Метод сопряженных градиентов. Методы *координатного спуска* или *наискорейшего спуска* требовали даже для минимизации квадратичной функции бесконечного чиста итераций.

Опираясь на тейлоровское разложение в окрестности невырожденного экстремума x^* выгодно строить методы спуска, которые, но крайней мере, эффективны для квадратичных функций.

Такими методами, не требующими решения СЛАУ (1) на каждом итерационном шаге для определения направления спуска, являются методы *сопряженных направлений.* Для квадратичной функции $\Psi(\vec{x}\ )$:

$$\Psi(x) = \frac{1}{2}(Ax, x) + (b, x) + c, \quad A > 0, \quad A^T = A$$

они позволяют не более чем за n шагов спуска получить её минимум. Напомним:

Симметричная положительноопределенная матрица $A>0$, $A^T=A$ позволяет ввести "A-энергетическую" норму вектора

$$\|x\|_A = \sqrt{(Ax, x)}$$

и соответствующеее скалярное произведение

$$(x, y)_A = (Ax, y) = (x. Ay).$$

Определение. *Векторы, ортогональные в А-энергетическом смысле, называются сопряженнымы относительно матрицы А.*

$$x \underset{A}{\perp} y \quad \Leftrightarrow \quad (x. y)_A = (Ax. y) = (x, Ay) = 0.$$

Сопряженные векторы обладают рядом "хороших" свойств:

1) *Если* $\{x_i\}_k$ — *система сопряженных векторов и* $k \le n$, *то эта сисстема векторов — линейно независима.*

Действительно, пусть

$$\overrightarrow{x_1} = \sum_{i=2}^{k} \alpha_i \overrightarrow{x_i}$$

- ненулевая комбинация остальных векторов. Тогда

$$(x_1, Ax_1) = (x_1, A\sum_{i=2}^{k} \alpha_i \overrightarrow{x_i}) = \sum_{i=2}^{k} \alpha_i (x_1. Ax_i) \equiv 0$$

но $A > 0$ и следовательно $\vec{x}_1$ нулевой вектор, что невозможно.

2) Если число векторов в рассматриваемой системе $k=n$, то $\{x_i\}_k$ — сопряженный базис. Можно считать его сопряженным ОНБ. т.е. $(x_i, x_j)_A = \delta_{ij}$. Разложим направление $\vec{p}$ по ОНБ $\{x_i\}_k$ и рассмотрим квадратичную функцию на этом направлении

$$\Psi(\overrightarrow{x} + \overrightarrow{p}) = \Psi(\overrightarrow{x}) + (Ax + b, \overrightarrow{p}) + \frac{1}{2}(A\overrightarrow{p}, \overrightarrow{p}) = \left| p = \sum \alpha_i \overrightarrow{x_i} \right| =$$

$$= \Psi(\overrightarrow{x}) + (Ax + b, \sum_i \alpha_i \overrightarrow{x_i}) + \frac{1}{2}\left(A\sum_i \alpha_i \overrightarrow{x_i}, \sum_k \alpha_k \overrightarrow{x_k}\right) =$$

$$= \underbrace{\sum_i \left\{\frac{1}{2}\alpha_i^2 + \alpha_i(Ax + b, x_i)\right\}}_{n \text{ независимых слагаемых}} + \Psi(\overrightarrow{x}); \qquad (2)$$

Движение по каждому из сопряженных направлений x_i изменяет только одно слагаемое в сумме (2) и, тем самым, за не более, чем n шагов приводит к минимуму функции Ψ.

Существуют различные способы построения сопряженных относительно A направлений, в частности - метод *сопряженных градиентов* (метод Флетчера-Рнвса)- приводит к одной из наиболее эффективных процедур многомерной численной минимизации.

Рассмотрим снова квадратичную аппроксимацию $\Psi(x)$ целевой фрикции *f(x)* в окрестности точки $\vec{x}_k$ *:*

$$\Phi(\vec{x}_k + \vec{p}) = \underbrace{\Phi(x_k) + (grad\Phi(x_k), \vec{p}) + \frac{1}{2}(hess\Phi(x_k)\vec{p}, \vec{p})}_{\Psi_k(\vec{p})} + o(\|\vec{p}\|^2)$$

На каждом *цикле* итерационных шагов для построения *сопряженного базиса* будем использовать одну и ту же матрицу $G_k \equiv$ hess $f(x_k)$. При этом мы будем считать, что находимся в достаточно малой окрестности точки минимума x^*, где $G(x_k) > 0$.

В методе *сопряженных градиентов* совокупность сопряженных относительно $G \equiv G(x_k)$ направлений строится следующим образом. Опишем процедуру построения одного цикла минимизации, содержащего n шагов и точно минимизирующего $\Psi_k(\vec{p})$.

Цикл движения: $M_k \equiv \overset{(1)}{M_k} \xrightarrow[\vec{p_1}]{} \overset{(2)}{M_k} \xrightarrow[\vec{p_2}]{} \cdots \xrightarrow[\vec{p}_{n-1}]{} \overset{(n)}{M_k} \xrightarrow[\vec{p_n}]{} M_{k+1}$

1-ый шаг: $\vec{p_1} = -\vec{g_1}$; $\overset{(2)}{x_k} = \overset{(1)}{x_k} + h_1\vec{p_1}$; $h_1 : \psi(h_1) = \min\limits_h \Psi_k\left(\overset{(1)}{x_k} + h\vec{p_1}\right)$;

2-ой шаг: $\vec{p_2} = -\vec{g_2} + \alpha_1\vec{p_1}$; $\alpha_1 = \frac{(g_2, g_2)}{(g_1, g_1)}$; $\vec{p_2} \perp \vec{p_1}$ отн-но G_k (3)

Пусть $\vec{p}_1, \ldots, \vec{p}_s$ G_k - сопряженная система векторов

$S+1$-ый шаг:

$$\left.\begin{aligned} &\vec{p}_{S+1} = -\vec{g}_{S+1} + \alpha_S\vec{p_S} \text{ (все остальные} \alpha_i = 0) \\ &\alpha_S = \frac{g^2_{S+1}}{g^2_S}; \text{ (из сообр. } (\vec{p}_{S+1}, \vec{p_S})_{G_k} = 0) \\ &\vec{x}_{S+2} = \vec{x}_{S+1} + h_{S+1} \cdot \vec{p}_{S+1} \\ &h_{S+1} : \psi(h_{S+1}) = \min_h \Psi_k\left(\vec{x}_{S+1} + h\vec{p}_{S+1}\right) \end{aligned}\right\}$$

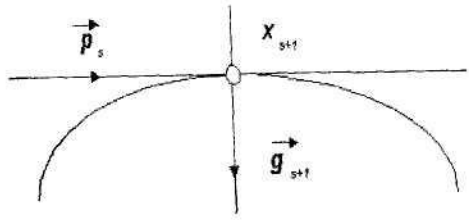
Покажем, что (3) определяет систему сопряженных относительно G_k векторов движения $\{ \vec{p}_s \}n$.

а) Проверить самостоятельно 2 й шаг;

б) 1: $\vec{g}_{s+1}$ ортогонально всем предыдущим $\vec{p}_j$ при $j \le s$, ибо спускаясь на предыдущем, S-ом шаге, мы пришли в точку

$$\vec{x}_{S+1} = \vec{x}_S + h_S \vec{p}_S$$

вдоль направления $\vec{p}_s$.



Но эта точка — $\vec{x}_{s+1}$ — точка "минимума", т.е.

$$\vec{g}_{S+1} \perp \vec{p}_S, \quad (\vec{g}_{S+1}, \vec{p}_S) = 0.$$

ЕСЛИ проследить "вглубь" траектории, то

$$\vec{x}_{S+1} = \vec{x}_S + h_S \vec{p}_S = \vec{x}_{S-1} + h_{S-1}\vec{p}_{S-1} + h_S \vec{p}_S = \cdots = \vec{x}_{j+1} + \sum_{j+1}^{S} h_i \vec{p}_i, \quad 1 \le j \le S-1.$$

Тогда

$$G x_{S+1} = G x_{j+1} + \sum_{j+1}^{S} h_i G \vec{p}_i.$$

Добавим слева и справа по $\vec{b} \equiv \vec{g}\ (M_k)$, и учтем, что $G_k \equiv G$; $Gx + b \equiv \vec{g}\ (x)$. Таким образом

$$\vec{g}_{S+1} = \vec{g}_{j+1} + \sum_{j+1}^{S} h_i G p_i.$$

Тогда

$$\overrightarrow{g}_{S+1}\cdot\overrightarrow{p_j}=\underbrace{\overrightarrow{g}_{j+1}\cdot\overrightarrow{p_j}}_{=0\text{ для этого шага}}+\sum_{j+1}^{S}h_i\cdot\underbrace{(Gp_i,p_j)}_{=0\text{ в силу индукции}}\quad\Rightarrow$$

$$\Rightarrow\quad(\vec{g}_{S+1},\vec{p}_j)=0,\ 1\le j\le S-1;\ S.$$

2: Покажем, что вектор $\vec{g}_{s+1}$ ортогонален всем градиентам $\vec{g}_j$, $j=\overline{1,S}$. Имеем

$$\overrightarrow{p_j}=-g_j+\alpha_{j-1}\overrightarrow{p}_{j-1}\quad\Leftrightarrow\quad\underbrace{\overrightarrow{g}_{S+1}\cdot\overrightarrow{p_j}}_{=0}=-(g_{S+1},g_j)+\underbrace{\alpha_{j-1}\cdot(\overrightarrow{g}_{S+1},\overrightarrow{p}_{j-1})}_{=0}$$

т.о.

$$(\overrightarrow{g}_{S+1},\overrightarrow{g_j})=0,\ j=\overline{1,S}.$$

3: Рассмотрим очередное направление:

$$\overrightarrow{p}_{S+1}=-\overrightarrow{g}_{S+1}+\alpha_S\overrightarrow{pS}:\quad\alpha_S=\frac{g_{S+1}^2}{g_S^2}$$

и покажем, что $\vec{p}_{s+1}$ сопряжено всем $\vec{p}_j$, $j=\overline{1,S}$. Оно сопряжено, по крайней мере, со всеми $\vec{p}$ до предыдущего, т.е.

$$(\overrightarrow{p}_{S+1},\overrightarrow{p_j})_G=0,\ j=\overline{1,S-1}.$$

Действительно, поскольку $j\le S$, то

$$(\overrightarrow{p}_{S+1},G\overrightarrow{p_j})=\left(-\overrightarrow{g}_{S+1}+\alpha_S\overbrace{\overrightarrow{pS},G\overrightarrow{p_j}}^{\text{сопряжены}}\right)=-\left(g_{S+1},G\frac{x_{j+1}-x_j}{h_j}\right)=$$

$$=-\left(g_{S+1},\frac{(Gx_{j+1}+b_k)-(Gx_j+b_k)}{h_j}\right)=-\left(g_{S+1},\frac{g_{j+1}-g_j}{h_j}\right)=0.$$

Предыдущее направление:

$$(\overrightarrow{p}_{S+1},G\overrightarrow{pS})=-(g_{S+1},Gp_S)+\alpha(p_S,Gp_S)=$$

$$=-\left(g_{S+1},G\frac{x_{S+1}-x_S}{h_S}\right)+\alpha_S\left(-g_S+\alpha_{S-1}\overrightarrow{p}_{S-1},\frac{g_{S+1}-g_S}{h_S}\right)=$$

$$=-\frac{g_{S+1}^2}{h_S}+\frac{g_{S+1}^2}{g_S^2}\frac{g_S^2}{h_S}=0$$

Метод Флетчера Ривса обладает квадратичной сходимостью, в достаточно малой окрестности точки $\vec{x}^*$. Рестарт в точке M_k осуществляется по антиградиенту ($-\vec{g}_k$).

Это один из наиболее эффективных методов численной минимизации функций многих пепременных.

7. Методы анализа многомерной безусловной оптимизации

7.1. Анализ методов прямого поиска

Методы разделяются на методы прямого поиска и градиентные. ***Методы прямого поиска*** используют только значение функции, разделяются на ***эвристические и теоретические***. Теоретические методы инвариантны.

Среди эвристических методов: поиск по симплексу и его модификация – метод Нелдера-Мида, а также метод Хука-Дживса.

Среди теоретических: метод сопряжённых направлений Пауэлла (основан на фундаментальном свойстве параллельного подпространства).

Градиентные методы:

- с использованием первой и второй производной;
- сопряжённых градиентов;
- квазиньютоновские методы.

7.1. Анализ методов прямого поиска

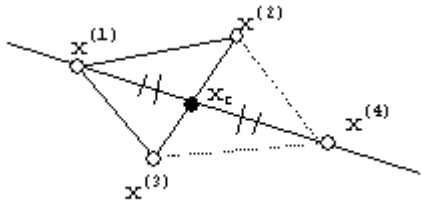
Метод поиска по симплексу

Метод основан на том, что экспериментальным образцом, содержащим наименьшее количество точек, является симплекс.

Регулярный симплекс в N-мерном пространстве – это многогранник, образованный N+1 равноотстоящими точками – вершинами симплекса.

Важное свойство симплекса - это то, что новый симплекс можно построить на любой грани исходного путём переноса выбранной вершины на некоторое расстояние вдоль прямой, соединяющей эту вершину с центром тяжести остальных вершин симплекса.

Пример для двухмерного случая.



в точке $x^{(1)}$ наихудшее значение функции; в точке x_c центр тяжести.

Работа алгоритма начинается с построения регулярного симплекса в пространстве независимых переменных задачи и оценивания значения целевой функции в его вершинах. Затем точка с наибольшим значением функции отражается через центр тяжести остальных точек. Новая точка используется как вершина нового симплекса. Итерации продолжаются до тех пор, пока либо не будет накрыта точка минимума, либо не начнётся циклическое движение по двум или более симплексам. При этом следует пользоваться тремя правилами:

· Если точка с наибольшим значением функции получена на предыдущей итерации, то вместо неё берётся точка со следующим по величине значением функции.

· Если некоторая вершина симплекса не исключается более, чем на N итерациях, то уменьшить размеры симплекса с помощью некоторого коэффициента и построить новый симплекс, используя в качестве

базовой точку с наименьшим значением функции. Количество итераций не исключения вершины: $M=1{,}65ЧN+0{,}05ЧN^2$.

· Поиск заканчивается, когда размеры симплекса и разности значений функции в вершинах станут достаточно малы.

Реализация алгоритма использует две основные процедуры: построение регулярного симплекса при заданной базовой точке и масштабном множителе (шаге) симплекса; расчёт отражённой точки.

Пусть $x^{(j)}$ - точка для отражения.

$$x_c = \frac{1}{N}\sum_{\substack{i=0 \\ i\neq j}}^{N} x^{(i)}$$

центр масс.

Все точки прямой, проходящей через $x^{(j)}$ и x_c определяются формулой $x = x^{(j)} + \lambda \cdot (x_c - x^{(j)})$ при λ=0 $x= x^{(j)}$, при λ=1 $x=x_c$.

Для получения нового регулярного симплекса λ=2, тогда $x = 2 \cdot x_c - x^{(j)}$.

Достоинства метода:

· простота;

· малое количество заранее установленных параметров;

· алгоритм эффективен и тогда, когда ошибки в определении значения целевой функции достаточно велики, так как в нём используется наибольшее значение целевой функции, а не наименьшее.

Недостатки метода:

· возникают трудности связанные с масштабированием задачи (в реальных задачах разные переменные часто не сопоставимы между собой по значениям);

· алгоритм работает медленно (не используется информация предыдущих итераций);

· не существует простого способа изменения размеров симплекса без пересчёта всех значений целевой функции.

Метод Нелдера-Мида

Это модифицированный метод поиска по симплексу (или метод деформируемого многоугольника). Он частично устраняет недостатки предыдущего.

Регулярность симплекса удобна при построении исходного симплекса, но нет оснований сохранения регулярности в процессе поиска. Было предложено деформировать симплекс, используя информацию с предыдущих итераций.

Деформирование осуществляется с помощью трёх операций:

операция	коэффициент для процедур в методе
отражение	1
сжатие	0,5
растяжение	2

Используемые процедуры.

Регуляризация симплекса.

$x^{(0)}$- начальная точка, h – шаг.

$$X = \begin{Bmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_N^{(0)} \\ x_1^{(0)} + h & x_2^{(0)} & \cdots & x_N^{(0)} \\ x_1^{(0)} & x_2^{(0)} + h & \cdots & x_N^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(0)} & x_2^{(0)} & \cdots & x_N^{(0)} - h \end{Bmatrix} = \begin{Bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N} \\ x_{3,1} & x_{3,2} & \cdots & x_{3,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N+1,1} & x_{N+1,2} & \cdots & x_{N+1,N} \end{Bmatrix}$$

Расчёт значений функции в вершинах симплекса.

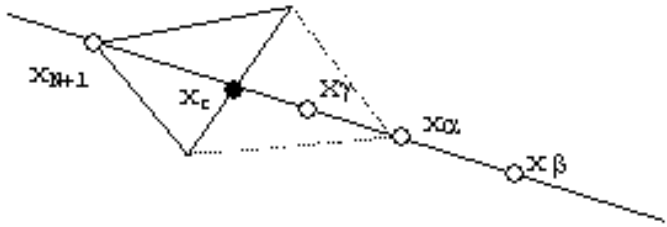
$$f_1 = f(x_{1,1}, x_{1,2}, \ldots, x_{1,N})$$
$$f_1 = f(x_{2,1}, x_{2,2}, \ldots, x_{2,N})$$
$$\ldots$$
$$f_1 = f(x_{N+1,1}, x_{N+1,2}, \ldots, x_{N+1,N})$$

Сортировка симплекса.

Точки симплекса нумеруются в порядке возрастания значений функции. Лучшая точка имеет номер 1, а худшая – номер N+1.

Нахождение пробной точки (на прямой, соединяющей худшую точку и центр масс).

Возможно получение трёх различных точек:



x_α - получается в результате симметричного отражения худшей точки симплекса относительно центра масс остальных точек.

x_β - результат растяжения симплекса, лежит на расстоянии в два раза большем, чем x_α от центра масс.

x_γ - результат сжатия симплекса, лежит в два раза ближе к центру масс, чем точка x_α.

Редукция симплекса.

Все точки симплекса сближаются к лучшей точке на половину расстояния.

На каждой итерации действия алгоритма описываются набором следующих правил:

Рассчитывается x_α.

Если $f(x_\alpha) < f(x^{(1)})$, то выполняется растяжение симплекса и находится точка x_β. Лучшая из точек x_α, x_β записывается на место $x^{(N+1)}$ и производится сортировка симплекса.

Если $f(x_\alpha) > f(x^{(1)})$ и $f(x_\alpha) < f(x^{(N+1)})$, то точка x_α записывается на место $x^{(N+1)}$ и производится сортировка симплекса.

Если $f(x_\alpha) > f(x^{(1)})$, то производится сжатие симплекса и находится точка x_γ. Если $f(x_\gamma) < f(x^{(N+1)})$, то x_γ записывается на место $x^{(N+1)}$ в противном случае производится редукция симплекса.

Недостаток: метод работает работает эффективно при $N \leq 6$.

Метод поиска Хука-Дживса

Стратегию поиска по симплексу можно усовершенствовать путём введения множества векторов, задающих направления поиска. Эти вектора должны быть линейно-независимы и образовывать базис в пространстве независимых переменных. Этому удовлетворяет система координатных направлений.

Метод Хука-Дживса - это комбинация исследующего поиска по направлениям и поиска по образцу.

Исследующий поиск: задаётся величина шага, которая может быть разной для разных координатных направлений и изменяться в процессе

поиска. Если значение целевой функции в пробной точке не превышает значение в исходной, то шаг поиска рассматривается как успешный. В противном случае, необходимо вернуться в предыдущую точку и сделать шаг в противоположном направлении. После перебора всех N координат исследующий поиск заканчивается. Полученная точка называется базовой.

Поиск по образцу: заключается в реализации единственного шага из полученной базовой точки вдоль прямой, соединяющей её с предыдущей базовой точкой.

Новая точка строится по формуле:

$$x_p^{(k+1)} = x^{(k)} + (x^{(k)} - x^{(k-1)}),$$

где:

$x^{(k)}$ - текущая базовая точка;

$x^{(k-1)}$ - предыдущая базовая точка;

$x_p^{(k+1)}$ - точка, построенная при движении по образцу;

$x^{(k+1)}$ - новая базовая точка.

Если движение по образцу не приводит к уменьшению целевой функции, то точка $x_p^{(k+1)}$ фиксируется в качестве временной базовой точки и вновь проводится исследующий поиск из этой точки. Если в результате получается точка со значением функции меньшим, чем в $x^{(k)}$, то она рассматривается как новая базовая точка $x^{(k+1)}$.

Если исследующий поиск неудачен, то нужно вернуться в $x^{(k)}$ и провести поиск в противоположном направлении. Если он также не приводит к успеху, то нужно уменьшить величину шага и возобновить исследующий поиск. Поиск завершается, когда величина шага становится достаточно малой.

Алгоритм метода.

1. Определить начальную точку $x^{(0)}$, приращения по координатным направлениям D_i, i=1,...,N, коэффициент уменьшения шага $\alpha>1$ и параметр окончания поиска ε.

2. Провести исследующий поиск.

3. Проверка успешности исследующего поиска. Если успешно, перейти к шагу 5, если нет, продолжать поиск.

4. Проверка на окончание поиска:

$$|\Delta x| \le \varepsilon$$

Если условие выполняется, поиск прекратить, если не выполняется, уменьшить шаг D и перейти к шагу 2.

$$\Delta_i = \Delta_i / \alpha$$

5. Провести поиск по образцу, то есть найти точку

$$x_p^{(k+1)} = x^{(k)} + (x^{(k)} - x^{(k-1)})$$

6. Провести исследующий поиск из точки $x_p^{(k+1)}$ и получить точку $x^{(k+1)}$.

7. Если $f(x^{(k+1)}) < f(x^{(k)})$,

то: $x^{(k-1)} = x^{(k)}$;

$x^{(k)} = x^{(k+1)}$;

goto 5.

иначе, goto 4.

Пример.

$f(x) = 8x_1^2 + 4x_1x_2 + 5x_2^2$

1) $x^{(0)} = [-4;-4]$; $f(x^{(0)}) = 272$;

$D = [1,1]$; $\alpha = 2$; $\varepsilon = 10^{-4}$.

2) (исследующий поиск) $x_2 = -4$, дадим приращение x_1.

(-3;-4) $f(-3;-4) = 200$ (удачно)

фиксируем $x_1 = -3$, дадим приращение x_2.

(-3;-3) $f(-3;-3) = 153$ (удачно)

базовая точка $x^{(1)} = [-3;-3]$; $f(x^{(1)}) = 153$.

3) (поиск по образцу)

$x_p^{(2)} = x^{(1)} + (x^{(1)} - x^{(0)}) = [-2;-2]$

$f(x_p^{(2)}) = 68$

4) (исследующий поиск)

$x^{(2)} = [-1;-1]$

$f(x^{(2)}) = 17 < f(x^{(1)})$ (удачно)

$x^{(2)}$ - базовая точка для проведения посика по образцу.

5) $x_p^{(3)} = x^{(2)} + (x^{(2)} - x^{(1)}) = [0;0]$ (минимум)

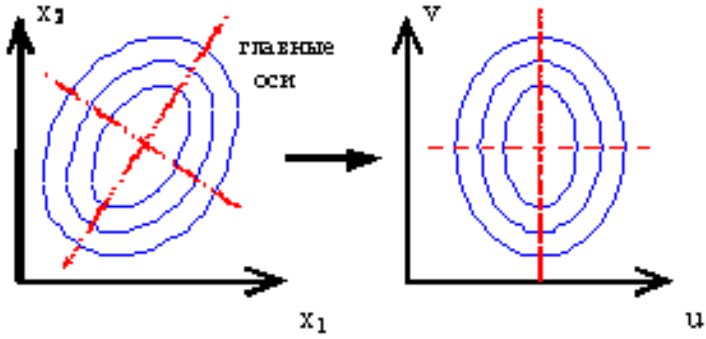
Достоинства метода: простая стратегия поиска, вычисление только значений функции, небольшой объём требуемой памяти.

Недостатки: алгоритм основан на циклическом движении по координатам. Это может привести к вырождению алгоритма в бесконечную последовательность исследующих поисков без поиска по образцу.

Метод сопряжённых направлений Пауэлла

Ориентирован на исследование квадратичных функций. Подходит также и для других функций после разложения в ряд Тейлора в окрестности точки оптимума.

Основная идея: если квадратичную функцию *n* переменных привести к виду суммы полных квадратов, то её оптимум может быть найден в результате *n* одномерных поисков по преобразованным координатным направлениям.



Процедура преобразования квадратичной функции

$$q(x) = a + b^T x + \frac{1}{2} x^T C x$$

К виду суммы полных квадратов эквивалентна нахождению такой матрицы преобразования T, которая приводит матрицу квадратичной формы $x^T Cx$ к диагональному виду. Квадратичная форма $Q(x) = x^T Cx$ путём преобразования $x=Tz$ приводится к виду:

$$Q(x) = z^T Dz,$$

где D - диагональная матрица.

$$x = Tz = t_1 z_1 + t_2 z_2 + \ldots + t_N z_N,$$

то есть вместо координат вектора x в стандартной координатной системе используются его координаты в новой системе, задаваемой векторами t_j. Поскольку t совпадают с главными осями квадратичной формы, то матрица D диагональна.

Итак, с помощью преобразования переменных квадратичной функции строится новая система координат, совпадающая с главными осями квадратичной функции, следовательно одномерный поиск точки оптимума в преобразованных координатах z эквивалентен поиску вдоль каждой из осей квадратичной функции. Таким образом, для нахождения оптимума достаточно провести n одномерных поисков вдоль векторов t_j.

Метод сопряжённых направлений Пауэлла

Пример.

$f(x) = 4x_1^2+3x_2^2-4x_1x_2+x_1$

$$x_1 = z_1 + \frac{1}{2}z_2, \quad x_2 = z_2.$$

(к сумме полных квадратов)

$$x = \begin{vmatrix} 1 & 1/2 \\ 0 & 1 \end{vmatrix} z$$

$f(z) = 4z_1^2+2z_2^2+z_1+1/2(z_2)$

$x^{(0)} = [0;0]$ $t_1 = [1;0]$ $t_2 = [1/2;1]$

(столбцы преобразований)

Точку оптимума найдём двумя одномерными поисками из начальной точки в этих направлениях.

$f(x^{(1)} = x^{(0)}+l^{(1)}t_1) \rightarrow \min$

$x^{(1)} = x^{(0)}+l^{(1)}t_1 = [-1/8;0]$

Из $x^{(1)}$ проводим поиск в направлении t_2.

$f(x^{(1)}+l^{(2)}t_2) \rightarrow \min$

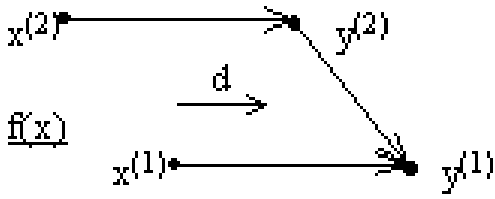
$x^{(2)} = x^{(1)}+l^{(2)}t_2 = [-3/16;-1/8]$

Таким образом, остаётся открытым вопрос о построении системы векторов t_j, вдоль которой осуществляется поиск. Она называется системой сопряжённых направлений.

Пусть C - симметричная матрица $n \times n$. Направления $S^{(1)}$, $S^{(2)}$, ... , $S^{(r)}$, где $r \in n$ называются C-сопряжёнными, если они линейно-независимы и выполняются равенства:

$$S^{(i)T}CS^{(j)} = 0 \qquad i \geq j$$

Для построения системы сопряжённых направлений будем использовать свойства параллельного подпространства.



Пусть задана квадратичная функция $f(x)$, две произвольные, несовпадающие точки $x^{(1)}$ и $x^{(2)}$ и направление d. Если точка $y^{(1)}$ минимизирует функцию $f(x^{(1)}+l_1 d)$, а точка $y^{(2)}$ минимизирует функцию $f(x^{(2)}+l_2 d)$, то направление $y^{(2)}-y^{(1)}$ является сопряжённым с d.

Для построения системы сопряжённых направлений лучше использовать [0;0] и систему координатных векторов. Рассмотрим $x^{(0)}$, $e^{(1)} = [1,0]$, $e^{(2)} = [0,1]$. Найдём значение $l^{(0)}$, которому соответствует минимум

$$f(x^{(0)}+l^{(0)}e^{(1)})$$

$$x^{(1)} = x^{(0)}+l^{(0)}e^{(1)}$$

Найдём $l^{(1)}$, которому соответствует минимум

$$f(x^{(1)}+l^{(1)}e^{(2)})$$

и точку

$$x^{(2)} = x^{(1)}+l^{(1)}e^{(2)}$$

Найдём $l^{(2)}$, которому соответствует минимум

$$f(x^{(2)}+l^{(2)}e^{(1)})$$

и точку

$$x^{(3)} = x^{(2)}+l^{(2)}e^{(1)}$$

$x^{(3)}$-$x^{(1)}$ сопряжено с $e^{(1)}$. По этим двум направлениям и производим поиск.

Метод сопряжённых направлений Пауэлла

Алгоритм метода.

1. Задать $x^{(0)}$, $e^{(1)}$, $e^{(2)}$, ... , $e^{(n)}$

2. Минимизировать $f(x)$ при последовательном движении по n+1 направлению. При этом полученная ранее точка минимума берётся в качестве исходной, а направление $S^{(n)}$ используется как при первом, так и при последнем поиске.

3. Определить новое сопряжённое направление по свойствам параллельного подпространства.

4. Заменить $S^{(1)}$ на $S^{(2)}$ и т.д., $S^{(n)}$ заменить новым сопряжённым направлением и goto 2. Провести всё это n^2 раз.

Алгоритм работает, если функция квадратична.

Пример.

$f(x) = 2x_1^3+4x_1x_2^3-10x_1x_2+x_2^2$

1) $x^{(0)} = [5;2]$; $f(x^{(0)}) = 314$;

$S^{(1)} = [1;0]$; $S^{(2)} = [0;1]$

2) Найдём l, при котором

$f(x^{(0)}+lS^{(2)}) \rightarrow \min$

$l = -0.81$

$x^{(1)} = x^{(0)}+lS^{(2)} = [5;2]-0.81[0;1] = [5;1.19]$

$f(x^{(1)}) = 250$

В направлении $S^{(1)}$:

l: $f(x^{(1)}+lS^{(1)}) \rightarrow \min$

$l = -3.26$

$x^{(2)} = x^{(1)}+lS^{(1)} = [1.74;1.19]$

$f(x^{(2)}) = 1.1$

l: $f(x^{(2)}+lS^{(2)}) \rightarrow \min$

$l = -0.098$

$x^{(3)} = [1.74;1.092]$

$f(x^{(3)}) = 0.72$

3) $x^{(3)}-x^{(1)} = S^{(3)}$ (сопряжённое с $x^{(2)}$)

$S^{(3)} = [-3.26;-0.098]$

$$S^{(3)} = \frac{S^{(3)}}{\left|S^{(3)}\right|} = \left[-0.999;-0.03\right]$$

(пронормировали)

4) $S^{(1)}$ исключаем, $S^{(1)} = S^{(2)}$, $S^{(2)} = S^{(3)}$.

Теперь найдём

l: $f(x^{(3)}+lS^{(2)}) \rightarrow \min$

$l = 0.734$

$x^{(4)} = x^{(3)}+lS^{(2)} = [1.006;1.07]$

$f(x^{(4)}) = -2.86$

Если бы данная функция была квадратичной, то поиск был бы завершён, а в данном случае необходим искусственный выход из цикла.

7.2. Анализ методов первого и второго порядков

Во всех этих методах предполагается $f(\vec{x})$, $\nabla f(\vec{x})$ и $\nabla^2 f(\vec{x})$ существуют и непрерывны. Все эти методы основаны на итерационной процедуре, определяемой формулой:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot S^{(k)}(x),$$

где $x^{(k)}$ - текущее приближение к решению;

$S^{(k)}(x)$ или $S^{(k)}$ - направление поиска;

$\alpha^{(k)}$ - параметр, характеризующий длину шага в направлении $S^{(k)}$.

Градиентные методы различаются только способом определения $\alpha^{(k)}$ и $S^{(k)}$. $\alpha^{(k)}$ обычно определяется путём решения задачи оптимизации *f(x)* в направлении $S^{(k)}$. Направление $S^{(k)}$ зависит от того, как аппроксимируется функция *f(x).*

Метод Коши

Пусть в точке $\bar{x}$ требуется определить направление наискорейшего спуска (то есть направление наибольшего локального уменьшения *f(x)*). Разложим *f(x)* в ряд Тейлора в окрестности точки $\bar{x}$ и отбросим члены второго порядка по Δx и выше.

$$\tilde{f}(x,\bar{x}) = f(\bar{x}) + \nabla f(\bar{x})^T \cdot \Delta x + \dots$$

Локальное уменьшение *f(x)* определяется вторым слагаемым, то есть наибольшее уменьшение *f(x)* будет тогда, когда $\nabla f(\bar{x})^T \cdot \Delta x$ будет иметь наибольшую отрицательную величину. Этого можно добиться выбором $S^{(k)}$: $S^{(k)} = -\nabla f(\bar{x})$, тогда второе слагаемое примет вид: $-\alpha \cdot \nabla f(\bar{x})^T \cdot \nabla f(\bar{x})$.

Этот случай соответствует наискорейшему локальному спуску $x^{(k+1)} = x^{(k)} - \alpha \cdot \nabla f(x^{(k)})$.

Недостатки:

- остаётся вопрос выбора α;
- вблизи точки минимума медленно сходится, так как $\nabla \to 0$.

α будем находить путём минимизации функции $f(x^{(k+1)})$ в направлении $-\nabla$.

Метод обладает большой надёжностью, но медленую сходимость вблизи точки минимума устранить нельзя. Поэтому метод самостоятельно обычно не используется, а используется как предварительная процедура для более сложных методов.

Достоинсиво:

на каждой итерации $f(x^{(k+1)}) \le f(x^{(k)})$ - выполняется свойство убывания функции на каждой итерации.

Алгоритм метода:

1 Задать $x_0, \varepsilon_1, \varepsilon_2, N, M$ - начальное приближение, параметр окончания работы алгоритма Коши, параметр окончания работы одномерного алгоритма, количество переменных и максимальное количество итераций соответственно.

2 Вычислить $\nabla f(x^{(k)})$

3 Если $|\nabla f(x^{(k)})| \le \varepsilon_1$, то $x_k=x^*$ иначе, если $K \ge M$, то $x_k=x^*$. Перейти к п. 4.

4 Решить задачу минимизации функции $f(x^{(k+1)})$ и найти $\alpha^{(k)}$ используя ε_2.

5 Вычислить следующее приближение по формуле $x^{(k+1)} = x^{(k)} - \alpha \cdot \nabla f(x^{(k)})$

6 Если $|\Delta x| \le \varepsilon_1$, то $x_k=x^*$ иначе $k=k+1$ и перейти к п. 2.

Метод Ньютона

Используется квадратичная аппроксимация *f(x)*. Разложим функцию в ряд Тейлора и оставим члены второго порядка:

$$\tilde{f}(x, x^{(k)}) = f(x^{(k)}) + \nabla f(x^{(k)})^T \cdot \Delta x + \frac{1}{2} \cdot \Delta x^T \cdot \nabla^2 f(x^{(k)}) \cdot \Delta x + \dots$$

Нужно, чтобы в каждой вновь получаемой точке $x^{(k+1)}$ градиент аппроксимирующего полинома был равен нулю:

$$\nabla f(x^{(k)}) \cdot \Delta x + \nabla^2 f(x^{(k)})^T \cdot \Delta x = 0; \quad x^{(k+1)} = x - \frac{\nabla f(x^{(k)})}{\nabla^2 f(x^{(k)})}.$$

Метод Ньютона обладает медленной сходимостью вдали от точки минимума, но хорошо сходится вблизи неё.

Модифицированный метод Ньютона

Исследования показывают, что, если целевая функция не квадратичная, то метод Ньютона ненадёжен, то есть если x_0 находится на значительном расстоянии от точки оптимума, то шаг может быть таким большим, что приведёт к несходимости.

Введём параметр длинны шага $\alpha^{(k)}$, который определяется из задачи минимизации функции $f(x^{(k+1)})$, теперь

$$x^{(k+1)} = x - \alpha^{(k)} \frac{\nabla f(x^{(k)})}{\nabla^2 f(x^{(k)})}.$$

Такая формула обеспечивает убывание функции от итерации к итерациии.

Метод Марквардта

Это комбинация методов Ньютона и Коши. Вдали от точки минимума направление определяется по методу Коши, а в окрестности точки минимума – по методу Ньютона.

$$S^{(k)} = -\left[H^{(k)} + \lambda^{(k)} \cdot I\right]^{-1} \cdot \nabla f(x^{(k)}),$$

где: $H^{(k)}$ – матрица Гессе (вторых производных);

I – единичная матрица;

$\lambda^{(k)}$ – параметр, определяющий направление поиска и длину шага.

При этом в формуле $x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot S^{(k)}$ $\alpha^{(k)} = 1$.

На начальном этапе $\lambda^{(k)} \approx 10^4$, при этом второй член в $S^{(k)} = -\left[H^{(k)} + \lambda^{(k)} \cdot I\right]^{-1} \cdot \nabla f(x^{(k)})$ много больше первого, поэтому поиск осуществляется по методу Коши. По мере приближения к точке оптимума $\lambda^{(k)}$ уменьшается и стремится к нулю. Таким образом вблизи точки оптимума первый член много больше второго и поиск осуществляется по методу Ньютона.

Если после первого шага $f(x^{(1)}) < f(x^{(0)})$, то следует выбрать $\lambda^{(1)} < \lambda^{(0)}$ и реализовать следующий шаг, в противном случае $\lambda^{(0)} = \beta \cdot \lambda^{(0)}$, где $\beta > 1$ и повторить предыдущий шаг.

Алгоритм.

1. Задать x_0 – начальное приближение, M – максимальное количество итераций, N – количество переменных и ε - параметр сходимости.

2. При $k=0$ $\lambda^{(k)} = 10^4$

3. Вычислить компоненты вектора $\nabla f(x^{(k)})$.

4. Если $|\nabla f(x^{(k)})| \le \varepsilon$, то $x_k = x^*$ иначе, если $K \ge M$, то $x_k = x^*$. Перейти к п. 5.

5. Вычислить $S^{(k)}$.

6. Вычислить $x^{(k+1)} = x^{(k)} + S^{(k)}$

7. Если $f(x^{(k+1)}) > f(x^{(k)})$, то перейти к п. 9, иначе перейти к п. 8.

8. Положить $\lambda^{(k+1)} = \frac{1}{2} \cdot \lambda^{(k)}$, $k=k+1$, перейти к п. 3.

9. Положить $\lambda^{(k)} = 2 \cdot \lambda^{(k)}$, перейти к п. 5.

Достоинства метода:

· простота;

· убывание целевой функции;

· быстрая сходимость как вдали от точки оптимума, так и вблизи неё:

· отсутствие поиска вдоль прямой.

Недостаток:

· необходимость вычисления матрицы Гессе на каждой итерации.

Вычислительные эксперименты показали, что метод наиболее эффективен для функций вида суммы квадратов: $f(\bar{x}) = f_1^2(\bar{x}) + f_2^2(\bar{x})$.

Численная аппроксимация градиентов

Способ: конечная разность вперёд.

$$\left.\frac{\partial f(x)}{\partial x_i}\right|_{x=\bar{x}} = \frac{f(\bar{x} + \varepsilon \cdot e^{(i)}) - f(\bar{x})}{\varepsilon}$$

$e^{(i)}$ – единичный орт того направления, по которому берём производную.

Эта формула основана на определении частной производной и при малых значениях ε даёт достаточно точное значение. Выбор ε осуществляется в зависимости от вида функции *f(x)*. Величина ε должна быть одновременно достаточно большой, чтобы не получить ноль в числителе, и достаточно малой для получения необходимой точности.

Способ: центральная конечная разность.

$$\left.\frac{\partial f(x)}{\partial x_i}\right|_{x=\bar{x}} = \frac{f(\bar{x}+\varepsilon\cdot e^{(i)}) - f(\bar{x}-\varepsilon\cdot e^{(i)})}{2\cdot\varepsilon}$$

Эта формула более точна, чем предыдущая при одних и тех же *f(x)* и ε, но требует дополнительного вычисления значения функции.

Способ: разность вперёд.

$$\left.\frac{\partial f(x)}{\partial x_i}\right|_{x=\bar{x}} = \frac{f(\bar{x}) - f(\bar{x}-\varepsilon\cdot e^{(i)})}{\varepsilon}$$

Формула аналогична разности назад.

Методы сопряжённых градиентов

В методе сопряжённых направлений Пауэлла для построения системы сопряжённых направлений использовались только значения целевой функции. В рассматриваемых методах для получения этой системы используется квадратичная аппроксимация целевой функции и значения компонент градиентов. Эти методы обеспечивают убывание целевой функции от итерации к итерации. Методы ориентированы на исследование квадратичных функций.

Свойство квадратичной функции, на котором основаны методы.

Пусть *q(x)* – квадратичная функция и есть две произвольные несовпадающие точки $x^{(0)}$ и $x^{(1)}$, тогда: $\nabla q(x) = c\cdot x + b = g(x)$.

$g(x^{(0)})=c\Delta x^{(0)}+b$, $g(x^{(1)})=c\Delta x^{(1)}+b$.

Найдём изменение градиента при переходе из $x^{(0)}$ в $x^{(1)}$:

$$\Delta g(x) = g(x^{(1)}) - g(x^{(0)}) = c \cdot (x^{(1)} - x^{(0)})$$

$$\Delta g(x) = c \cdot \Delta x$$

Метод Флетчера-Ривса

Пусть дана целевая квадратичная функция

$$q(x) = a + b^T \cdot x + \frac{1}{2} \cdot x^T \cdot c \cdot x$$

и итерации производятся по формуле

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot S^{(k)}(x).$$

В данном методе $S^{(k)}$ ищется по формуле:

$$S^{(k)} = -g^{(k)} + \sum_{i=0}^{k-1} \gamma^i \cdot S^i,$$

где

$$g^{(k)} = \nabla f(x^{(k)}).$$

Величины γ^i выбираются так, чтобы новое направление $S^{(k)}$ было сопряжено со всеми предыдущими направлениями. При этом критерием окончания поиска является выполнение условия: $\nabla q(x^{(k+1)})^T \cdot S^{(k)} = 0$.

Определим γ^i.

Рассмотрим первое направление. k=1.

$$S^{(1)} = -g^{(1)} + \gamma^0 \cdot S^0 = -g^{(1)} - \gamma^0 g^{(0)}$$

Наложим условия C-сопряжённости направлений $S^{(1)}$ и $S^{(0)}$:

$$S^{(1)^T} \cdot C \cdot S^{(0)} = 0$$

$$\left[-g^1 - \gamma^{(0)} \cdot g^{(0)}\right]^T \cdot C \cdot S^{(0)} = 0$$

На первой итерации $$S^{(0)} = \frac{\Delta x}{a^{(0)}}$$

$$\left[-g^1 - \gamma^{(0)} \cdot g^{(0)}\right]^T \cdot C \cdot \frac{\Delta x}{a^{(0)}} S^{(0)} = 0 \quad ; \quad C \cdot \frac{\Delta x}{a^{(0)}} = \Delta g \quad ; \quad \frac{C}{a^{(0)}} -$$

константа.

$$\left[-g^1 - \gamma^{(0)} \cdot g^{(0)}\right]^T \cdot \Delta g = 0$$

Отсюда можем найти $\gamma^{(0)}$

$$\gamma^{(0)} = -\frac{\Delta g^T \cdot g^{(1)}}{\Delta g^T \cdot g^{(0)}}; \Delta g = g^{(1)} - g^{(0)}$$

$$g^{(1)^T} \cdot g^{(1)} + \gamma^{(0)} \cdot g^{(0)^T} \cdot g^{(1)} - g^{(1)^T} \cdot g^{(0)} - \gamma^{(0)} \cdot g^{(0)^T} \cdot g^{(0)} = 0;$$

При соответствующем выборе $\alpha^{(0)}$ и использовании условия $\nabla q(x^{(k+1)})^T \cdot S^{(k)} = 0$ имеем $g^{(1)^T} \cdot g^{(0)} = 0$.

$$\gamma^{(0)} = \frac{\left\|g^{(1)}\right\|^2}{\left\|g^{(0)}\right\|^2}$$, где ||…|| - норма вектора.

Определим следующее направление

$$S^{(2)} = -g^{(2)} - \gamma^0 \cdot g^{(0)} + \gamma^1 \cdot g^{(1)}$$

Выберем γ^0 и γ^1 так, чтобы вектора $S^{(0)}$, $S^{(1)}$ и $S^{(2)}$ были C-сопряжены.

$$S^{(2)T} \cdot C \cdot S^{(0)} = 0$$

$$S^{(2)T} \cdot C \cdot S^{(1)} = 0$$

Самостоятельно доказать, что все $\gamma^i=0$ для $i=0...k-2$.

$$S^{(k)} = -g^{(k)} + \frac{\left\|g^{(k)}\right\|^2}{\left\|g^{(k-1)}\right\|^2} \cdot S^{(k-1)}$$

Если функция квадратична, то для нахождения минимума нужно найти N-1 направлений и провести N одномерных поисков вдоль прямой.

Метод Поллака-Ребьера

В предыдущем методе:

- функция квадратична;
- нет погрешностей при поиске по прямой.

Метод основан на точной процедуре поиска вдоль прямой (точно находим $\alpha^{(k)}$), но целевая функция может быть общего вида.

$$\gamma^{(k)} = \frac{\Delta g(x^{(k)})^T \cdot g(x^{(k)})}{\left\| g(x^{(k-1)}) \right\|^2},$$

где

$$\Delta g(x^{(k)}) = g(x^{(k)}) - g(x^{(k-1)})$$

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot S^{(k)}(x)$$

$$S^{(k)} = \frac{-\nabla f(x^{(k)})}{g(x^{(k)})} + \gamma^{(k)} \cdot S^{(k-1)}$$

Квазиньютоновские методы

В этих методах обратная матрица Гессе аппроксимируется другой матрицей – метрикой. Метрика изменяется на каждой итерации и поэтому методы так же называются методами с переменной метрикой.

$$S^{(k)} = -A^{(k)} \cdot \nabla f(x^{(k)})$$

$A^{(k)}$ – матрица $n \times n$ - метрика.

$A^{(k+1)} = A^{(k)} + A^{c(k)}$, где $A^{c(k)}$ корректирующая матрица.

Нужно построить последовательность $A^{(0)}$, $A^{(1)}$, $A^{(2)}$… и так далее, которая давала бы приближение к обратной матрице Гессе.

Метод Дэвидона- Флетчера- Пауэлла

$$A^{(k)} = A^{(k-1)} + \frac{\Delta x^{(k-1)} \cdot \Delta x^{(k-1)^T}}{\Delta x^{(k-1)^T} \cdot \Delta g^{(k-1)}} - \frac{A^{(k-1)} \cdot \Delta g^{(k-1)} \cdot \Delta g^{(k-1)^T} \cdot A^{(k-1)}}{\Delta g^{(k-1)^T} \cdot A^{(k-1)} \cdot \Delta g^{(k-1)}};$$

$A^{(0)} = I$

Обеспечивает убывание целевой функции от итерации к итерации.

Самостоятельно показать, что $\Delta f(x) \le 0$.

Метод Бройдена-Флетчера-Шенно

$$A^{(k+1)} = \left[I - \frac{\Delta x^{(k)} \cdot \Delta g^{(k)T}}{\Delta x^{(k)T} \cdot \Delta g^{(k)}} \right] \cdot A^{(k)} \cdot \left[I - \frac{\Delta x^{(k)} \cdot \Delta g^{(k)T}}{\Delta x^{(k)T} \cdot \Delta g^{(k)}} \right] + \frac{\Delta x^{(k)} \cdot \Delta x^{(k)T}}{\Delta x^{(k)T} \cdot \Delta g^{(k)}}$$

Метод обладает слабой по сравнению с ДФП чувствительностью к погрешности одномерного поиска.

7.3. Обобщённый алгоритм

Схожесть градиентных методов позволяет построить обобщённый алгоритм.

1. Задать n – число переменных, M – максимальное число итераций; x(0) – начальное приближение; ε_1 - параметр окончания работы градиентного алгоритма; ε_2 - параметр окончания одномерного поиска. k=0.

2. Вычислить $\nabla f(x^{(k)})$.

3. Если $|\nabla f(x^{(k)})| \le \varepsilon_1$, то $x_k=x^*$ иначе, если $K \ge M$, то $x_k=x^*$. Перейти к п. 4.

4. Вычислить $S(x^{(k)})$, используя различные способы вычисления.

5. Если $\nabla f(x^{(k)})^T \cdot S(x^{(k)}) < 0$, то прейти к п. 6, иначе $S^{(k)} = -\nabla f(x^{(k)})$

6. Решить задачу одномерного поиска и найти $\alpha^{(k)}$ используя ε_2

7. Найти $x^{(k+1)} = x^{(k)} + a^{(k)} \cdot S^{(k)}(x)$

8. Если $f(x^{(k+1)}) > f(x^{(k)})$, то $x_k = x^*$, иначе перейти к п. 9.

9. Если $|\Delta x| \le \varepsilon_1$, то $x_k = x^*$, иначе $k = k+1$, перейти к п. 2.

Свойства сходимости методов

Определение. Метод называется сходящийся, если неравенство

$$\left|\frac{\varepsilon^{(k+1)}}{\varepsilon^{(k)}}\right| \le 1$$

выполняется на каждой итерации, где $\varepsilon^{(k)} = x^{(k)} - x^*$.

Определение. Алгоритм обладает сходимостью порядка r, если отношение

$$\lim_{k\to\infty} \frac{\left|\varepsilon^{(k+1)}\right|}{\left|\varepsilon^{(k)}\right|^r} = C$$

выполняется (конечно). Если r=1, то алгоритм обладает линейной скоростью сходимости. Если ещё при этом C=0, то алгоритм обладает суперлинейной скоростью сходимости. Если r=2, то скорость квадратичная.

8. Методы оптимизации овражных функций

Методы оптимизации овражных функций - численные методы отыскания минимумов функций многих переменных. Пусть задана ограниченная снизу дважды непрерывно дифференцируемая по своим аргументам функция

$$J(x) = J(x_1, \ldots, x_m),$$

для которой известно, что при некотором векторе $x^* = (x_1^*, \ldots, x_m^*)^T$ (T - знак транспонирования) она принимает наименьшее значение. Требуется построить последовательность векторов

$$\{x_n\}, \; x_n = (x_{1n}, \ldots, x_{mn})^T,$$

такую, что

$$\lim_{n \to \infty} J(x_n) = J(x^*).$$

Существует много методов, позволяющих получить указанную последовательность векторов. Однако общим недостатком большинства алгоритмов является резкое ухудшение их свойств в случаях, когда поверхности уровня минимизируемой функции $J(x) = \text{const}$ имеют структуру, сильно отличающуюся от сферической. В этом случае некоторую область Q, в которой норма вектора-градиента

$$J'(x) = \left(\frac{\partial J}{\partial x_1}, \ldots, \frac{\partial J}{\partial x_m} \right)^T$$

существенно меньше, чем в остальной части пространства, называют дном оврага, а саму функцию - овражной функцией. Если размерность пространства аргументов минимизируемой функции больше двух, то структура поверхностей уровня овражных функций может оказаться весьма сложной. Появляются (m-k)-мерные овраги, где число k изменяется от 1 до m-1. В трехмерном пространстве, например, возможны одномерные и двумерные овраги.

Функции овражного типа локально характеризуются плохой обусловленностью матриц двух производных (матриц Гессе)

$$J''(x) = \left\| \frac{\partial^2 J(x)}{\partial x_i \partial x_j} \right\|, \quad i, \; j = 1, \ldots, m,$$

что приводит к сильному изменению функции *J(x)* вдоль направлений, совпадающих с собственными векторами матрицы Гессе для больших собственных чисел, и к слабому изменению вдоль других направлений, отвечающих малым собственным значениям матрицы Гессе. Большинство известных методов оптимизации позволяет достаточно быстро попадать на дно оврага, приводя иногда к существенному уменьшению значения функции *J(x)* по сравнению с его значением в начальной точке (спуск на дно оврага). Однако далее процесс резко замедляется и практически останавливается в некоторой точке из *Q,* которая может быть расположена очень далеко от истинной точки минимума.

Дважды непрерывно дифференцируемая по своим аргументам функция *J(x)* называется овражной функцией, если существует некоторая область $G \subset \mathbb{R}^m$, где собственные значения матрицы Гессе *J''(x)*, упорядоченные в любой точке $x \in G$ по убыванию модулей, удовлетворяют неравенствам

$$0 < \left| \min_i \lambda_i(x) \right| \ll \lambda_1(x). \qquad (1)$$

Степень овражности характеризуется числом

$$S = \lambda_1 / \left| \min_{\lambda_i \neq 0} \lambda_i \right|. \qquad (2)$$

Если собственные значения *J''(x)* в области *G* удовлетворяют неравенствам

$$|\lambda_m(x)| \leqslant \ldots \leqslant |\lambda_{m-r+1}(x)| \ll \lambda_{m-r}(x) \leqslant \ldots \leqslant \lambda_1(x),$$

то число *r* называется размерностью оврага функции *J(x)* при $x \in G$.

Системы дифференциальных уравнений, описывающие траекторию спуска овражной функции *J(x)*,

$$\frac{dx}{dt} = -J'(x), \quad x(0) = x_0, \qquad (3)$$

являются *жесткими дифференциальными системами.* В частности, когда функция *J(x)*сильно выпуклая и матрица Гессе положительно определена (все ее собственные значения строго больше нуля), неравенства (1) совпадают с известным требованием плохой обусловленности матрицы Гессе

$$k(J''(x)) = \max_i \lambda_i(x) / \min_i \lambda_i(x) \gg 1.$$

В этом случае спектральное число обусловленности совпадает со степенью овражности.

8.1. Метод покоординатного спуска

$$J(x_{1,k+1}, \ldots, x_{i-1,k+1}, x_{i,k+1}, x_{i+1,k}, \ldots, x_{m,k}) = \min_y J(x_{1,k+1}, \ldots, x_{i-1,k+1}, y, x_{i+1,k}, \ldots, x_{m,k}),$$
$$k = 0, 1, 2, \ldots, \qquad (4)$$

несмотря на простоту и универсальность, в овражной ситуации эффективен лишь в редких случаях ориентации оврагов вдоль координатных осей.

Существующая модернизация метода (4), состоящая в использовании процедуры вращения осей координат так, чтобы одна из осей была направлена вдоль $x_k - x_{k-1}$, после чего начинается поиск на (k+1)-м шаге. Такой подход приводит к тому, что одна из осей имеет тенденцию выстраиваться вдоль образующей дна оврага, позволяя в ряде случаев весьма успешно проводить минимизацию функций с одномерными оврагами. В случае многомерных оврагов метод непригоден.

Схема метода наискорейшего спуска задается разностным уравнением

$$x_{k+1} = x_k - h_k J'_k, \quad J'_k = J'(x_k), \tag{5}$$

где h_k выбирается из условия

$$J(x_{k+1}) = \min_{h>0} J(x_k - hJ'_k).$$

Для сильно выпуклой овражной функции, в частности квадратичной

$$J(x) = \frac{1}{2} x^T D x - b^T x, \tag{6}$$

последовательность $\{x_k\}$ построенная алгоритмом (5), сходится к точке минимума функции x^* по закону геометрической прогрессии

$$\| x_k - x^* \| \leqslant C q^k,$$

где C=const,

$$q = [k(J''(x^*)) - 1]/[k(J''(x^*)) + 1].$$

Так как для овражной функции $k(J''(x^*)) \gg 1$, то $q \simeq 1$ и сходимость практически отсутствует.

Аналогичная картина наблюдается и для простой градиентной схемы

$$x_{k+1} = x_k - hJ''_k, \quad J_{k+1} = J(x_{k+1}), \quad h = \text{const}. \tag{7}$$

Ускорение ее сходимости основано на использовании результатов предыдущих итераций для уточнения дна оврага. Может быть использован градиентный метод (7) с вычислением на каждой итерации отношения $q = \|J'_k\| / \|J'_{k-1}\|$. Когда оно

устанавливается около некоторого постоянного значения q=1, делается большой ускоряющий шаг согласно выражению

$$x_{k+1} = x_k - \frac{h}{1-q} J'_k.$$

Далее из точки x_{k+1} продолжается спуск градиентным методом до следующего ускоряющего шага.

Различные версии метода параллельных касательных основаны на выполнении ускоряющего шага вдоль направления $x_{k+2} - x_k$, задаваемого точками x_k, x_{k+2} в градиентном методе. В методе "тяжелого шарика" очередное приближение имеет вид

$$x_{k+1} = x_k - \alpha J'_k + \beta (x_k - x_{k-1}).$$

В методе оврагов предлагается провести локальные спуски градиентным методом (7) из двух случайно выбранных исходных точек, а затем выполнить ускоряющий шаг по направлению, задаваемому двумя полученными на дне оврага точками.

Все эти методы немногим сложнее градиентного метода (7) и построены на его основе. Ускорение сходимости получается для одномерных оврагов. В более общих случаях многомерных оврагов, где сходимость этих схем резко замедляется, приходится обращаться к более мощным методам квадратичной аппроксимации, в основе которых лежит метод Ньютона

$$x_{k+1} = x_k - (J''_k)^{-1} J'_k, \; J''_k = J''(x_k). \qquad (8)$$

Точка минимума функции (6) удовлетворяет системе линейных уравнений

$$D\boldsymbol{x} = \boldsymbol{b}, \qquad (9)$$

и при условии абсолютной точности всех вычислений для квадратичной функции метод Ньютона независимо от степени овражности (2) и размерности оврагов приводит к минимуму за один шаг. На самом деле, при больших числах обусловленности *k(D*) при ограниченной разрядности вычислений задача получения решения (9) может быть некорректной, и небольшие деформации элементов матрицы *D* и вектора ***b*** могут приводить к большим вариациям *x**.

При умеренных степенях овражности в выпуклой ситуации метод Ньютона часто оказывается более предпочтительным по скорости сходимости, чем другие, например, градиентные, методы.

Большой класс квадратичных (квазиньютоновских) методов основан на использовании сопряженных направлений. Эти алгоритмы для случая минимизации выпуклой функции оказываются весьма эффективными, ибо, имея квадратичное окончание, они не требуют вычисления матрицы двух производных.

Иногда итерации строятся по схеме

$$x_{k+1} = x_k - (\beta_k E + J''_k)^{-1} J'_k, \qquad (10)$$

где *E*- единичная матрица. Скаляр β_k подбирается так, чтобы матрица $J''_k + \beta_k E$ была положительно определенной и чтобы

$$\| x_{k+1} - x_k \| \leqslant \varepsilon_k.$$

Существует ряд аналогичных подходов, основанных на получении строго положительно определенных аппроксимаций матрицы Гессе. При минимизации овражных функций такие алгоритмы оказываются малоэффективными из-за трудностей в подборе параметров β_k, ε_k и т. д. Выбор этих параметров основан на информации о величине наименьших по модулю собственных значений матрицы Гессе, а при реальных вычислениях и большой степени овражности эта информация сильно искажена.

Более целесообразно обобщение метода Ньютона на случай минимизации овражных функций проводится на базе непрерывного

принципа оптимизации. Функции $J(x)$ ставится в соответствие дифференциальная система (3), интегрируемая системным методом (см. *Жесткая дифференциальная система*). Алгоритм минимизации принимает вид

$$\left.\begin{gathered} x_{k+1}=x_k-\Phi\left(2^N h_k^0\right) J_k', \\ \Phi\left(2^N h_k^0\right)=\int_0^{2^N h_k^0} \exp\left(-J_k''\tau\right) d\tau, \\ J_{k+1}=\min_N J\left(x_k-\Phi\left(2^N h_k^0\right) J_k'\right), \\ h_k^0 \leqslant 1/\|J_k''\|, \\ \Phi\left(h_k^0\right)=h_k^0\left[E-\frac{h_k^0}{2} J_k''+\frac{\left(h_k^0\right)^2}{6}\left(J_k''\right)^2-\ldots\right], \\ \Phi\left(2^{s+1} h_k^0\right)=\Phi\left(2^s h_k^0\right)\left[2E-J_k''\Phi\left(2^s h_k^0\right)\right], \\ s=0,\ 1,\ \ldots,\ N-1. \end{gathered}\right\} \quad (11)$$

Предложен алгоритм минимизации овражной функции, основанный на использовании свойств жестких систем. Пусть функция $J(x)$ в окрестности x_0 аппроксимируется квадратичной функцией (6). Матрица D и вектор $\boldsymbol{b}$ вычисляются, например, с помощью конечно-разностной аппроксимации. Из представления элементов матрицы

$$d_{ij}=\sum_{s=1}^{m} u_{is}u_{js}\lambda_s,$$

где

$$u_s=(u_{1s},\ \ldots,\ u_{ms})^T,\quad s=1,\ \ldots,\ m,$$

ортонормированный базис собственных векторов D, следует, что неточное измерение этих элементов искажает информацию о малых собственных значениях плохо обусловленной матрицы, а следовательно, приводит к некорректности задачи минимизации функции (6). Вместе с тем система дифференциальных уравнений спуска для овражной функции (6)

$$\frac{dx}{dt} = -Dx + b, \quad x(0) = x_0$$

имеет решение, в котором в силу условия (1) слагаемые с сомножителями $\exp(-\lambda_1 t)$ оказывают влияние лишь на малом начальном отрезке длиной $\tau_{\text{пс}} = O(\lambda_1^{-1})$. Другими словами, компоненты вектора $x(t)$ удовлетворяют равенству

$$x^T(t)\,u_1 - \lambda_1^{-1} b^T u_1 = (x_0^T u_1 - \lambda_1^{-1} b^T u_1)\exp(-\lambda_1 t),$$

быстро переходящему в стационарную связь

$$\sum_{i=1}^{m} u_{i1}\bar{x}_i - \lambda_1^{-1}\sum_{i=1}^{m} b_i u_{i1} = 0, \qquad (12)$$

где $\bar{x}_i$ - компоненты вектора, удовлетворяющие равенству (12). Это свойство используется в алгоритме. Выражая j-ю компоненту вектора $\bar{x}$, которой соответствует максимальная компонента вектора u_1, через остальные компоненты, вместо функции $J(x)$, получают новую функцию с аргументом размерности (m-1):

$$J\Big(\bar{x}_1, \ldots, \bar{x}_{j-1}, -u_{j1}^{-1}\sum_{i=1,\, i\neq j}^{m}(u_{i1}\bar{x}_i - \lambda_1^{-1} b_i u_{i1}), x_{j+1}, \ldots, x_m\Big) = \bar{J}(\bar{x}_1, \ldots, \bar{x}_{j-1}, \bar{x}_{j+1}, \ldots, \bar{x}_m). \qquad (13)$$

По функции (13) с помощью конечноразностной аппроксимации находится новая матрица $\bar{D}$ порядка (m-1) и вектор $\bar{b}$. Здесь важно не только и не столько понижение размерности пространства поиска, сколько уменьшение степени овражности, т. к. при минимизации новой функции в подпространстве, ортогональном вектору u_1, большое собственное значение уже не оказывает влияния на вычислительный процесс. Самым существенным моментом здесь является требование

получения $\overline{D}$ и $\overline{b}$ по функции (13), а не по матрице D и вектору $\boldsymbol{b}$. Коэффициенты связи (12) находят степенным методом, как коэффициенты любого уравнения системы

$$D^k x = D^{k-1} b.$$

Если степень овражности не понижается или понижается незначительно, то процесс исключения координат вектора x продолжается рекурсивно до необходимого ее уменьшения.

Сравнение методов многомерной безусловной оптимизации

Существуют два пути сравнения: теоретическое исследование сходимости и численные эксперименты.

Метод Пауэлла - суперлинейная скорость сходимости.

Метод Коши - линейная скорость.

Метод Ньютона - квадратичная.

Методы сопряжённых градиентов - линейная скорость сходимости.

Квазиньютоновские - квадратичная скорость.

В результате численных экспериментов Химмельблау («Нелинейное программирование») методы распределяются по количеству вычислений значений функции, устойчивости, машинному времени. Устойчивость характеризует ширину круга задач (успешно решаемых).

Лучшие методы: ДФП, Пауэлла, Бройдена-Флетчера-Шенно.

Другие исследователи сравнивали градиентные методы. При этом учитывалось влияние параметров сходимости методов одномерного поиска, положительная определённость матрицы H для квазиньютоновских методов и точность определения компонент градиента.

Выводы:

1) превосходство квазиньютоновских методов при решении задач с функциями общего вида;

2) на эти методы точность вычислений на ЭВМ оказывает большее влияние, чем на методы сопряжённых градиентов.

Функция Розенброка:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

(общепринятая тестовая) $f(1,1) = 0$ (минимум)

Комбинация метода Коши и метода деления пополам обеспечивает наибольшую точность при больших затратах машинного времени. Самая эффективная с точки зрения вычисления значений функции - комбинация Бройдена-Флетчера-Шенно и кубической аппроксимации.

9. Влияние помех на поведение методов безусловной минимизации

Цель настоящего раздела — выяснить поведение методов безусловной минимизации дифференцируемых функций при наличии помех. Оказывается, что чувствительность методов к помехам различна. Грубо говоря, чем эффективнее метод в идеальном случае (без помех), тем более чувствителен он к разного рода ошибкам. Можно модифицировать методы, сделав их работоспособными в условиях помех. При этом априорная информация о помехах (их уровень, закон распределения и т. д.) может быть эффективно использована.

9.1. Источники и типы помех

1. Источники помех. В реальных задачах применить методы минимизации «в чистом виде» нельзя — ситуация неизбежно осложняется наличием разного рода ошибок и погрешностей. Перечислим некоторые из причин их возникновения.

В простейшем случае, когда минимизируемая функция и ее градиент заданы формулами, ошибки возникают вследствие *погрешностей вычисления,* связанных с округлением при выполнении арифметических действий на ЭВМ. В результате $f(x^k)$, $\nabla f(x^k)$ и т. д. вычисляются с некоторой ошибкой, т. е. вместо вектора $\nabla f(x^k)$ мы получаем вектор $s^k = \nabla f(x^k)+r^k$. Здесь помеха r^k является детерминированной (ошибки округления в ЭВМ не носят случайного характера) и можно оценить ее уровень $\|r^k\|\leq\varepsilon$, так как законы образования погрешностей округления хорошо изучены. Величину ε обычно можно считать постоянной (не зависящей от x^k) и, как правило, не слишком большой. В случае необходимости ε можно уменьшить, производя вычисления с двойной точностью.

В ряде задач значения $f(x^k)$ и $\nabla f(x^k)$ получаются не с помощью вычислений, а в *результате измерений.* Такова ситуация при оптимизации на реальном объекте (экстремальное регулирование, планирование эксперимента). Тогда помехи носят случайный характер, свойственный погрешностям измерений. При этом обычно бывает доступна информация об уровне и статистической природе помехи.

Нередко (особенно в задачах адаптации, обучения, распознавания и т. д.) проблема оптимизации ставится следующим образом. Нужно минимизировать детерминированную функцию $f(x)$
типа *среднего риска:*

$$f(x) = \mathbf{M}Q(x, \omega) = \int Q(x, \omega)\, d\mathbf{P}(\omega), \qquad (1)$$

где функция $Q(x, \omega)$ известна, однако распределение $\mathbf{P}(\omega)$ не задано. Дана лишь выборка $\omega_1,..., \omega_k$ из этого распределения. Тогда точное вычисление $f(x)$ и $\nabla f(x)$ в принципе невозможно. В качестве приближенного значения этих величин можно взять

$$\frac{1}{k}\sum_{i=1}^{k} Q(x, \omega_i) \quad \text{и} \quad \frac{1}{k}\sum_{i=1}^{k} \nabla_x Q(x, \omega_i), \qquad (2)$$

или более просто

$$Q(x, \omega_k) \quad \text{и} \quad \nabla_x Q(x, \omega_k). \qquad (3)$$

В этом случае значения функции и градиента содержат случайную помеху. Если брать в качестве приближений для $f(x^k)$ и $\nabla f(x^k)$ величины $Q(x^k, \omega_k)$ и $\nabla_x Q(x^k, \omega_k)$, то помехи будут независимы в различных точках.

Аналогичная ситуация возникает в *методе Монте-Карло,* когда задача заключается в минимизации $f(x)$ вида (1) и распределение $\mathbf{P}(\omega)$ известно, однако вычисление интеграла (1) слишком трудоемко. Тогда

можно точные значения $f(x)$ и $\nabla f(x)$ заменить выборочными значениями, как и выше.

В ряде задач ошибки возникают из-за того, что значения функции и градиента вычисляются по упрощенным или приближенным формулам. Нередко точное вычисление требует громоздкого расчета функций влияния, решения сложных вспомогательных задач, учета взаимодействия всех параметров и т. д. Все эти вычисления нецелесообразно (а иногда и невозможно) проводить полностью. Их упрощение и огрубление приводят к погрешностям в определении функции и градиента. Это так называемые *неустранимые погрешности.*

Наконец, во многих методах ошибки возникают не из-за приближенного вычисления функции или градиента, а из-за необходимости решения вспомогательных задач, которое не может быть осуществлено точно. Например, в методе Ньютона на каждом шаге нужно решать систему линейных уравнений, что неизбежно сопряжено с ошибками; в методе сопряженных градиентов требуется проводить одномерную минимизацию, что также может быть сделано лишь приближенно и т. д. В таком случае говорят о *погрешностях метода.*

2. Типы помех. Как мы видели выше, ошибки при вычислении функции и градиента могут иметь различное происхождение и различную природу. Несколько упрощая реальную ситуацию, можно выделить следующие основные типы помех. Всюду ниже речь идет о вычислении градиента, когда вместо точного значения $\nabla f(x^k)$ нам доступен вектор

$$s^k = \nabla f(x^k) + r^k \tag{4}$$

где r^k — помехи. Случай приближенного вычисления $f(x)$ исследуется аналогично.

а) *Абсолютные детерминированные помехи* удовлетворяют условию

$$||r^k|| \le \varepsilon, \tag{5}$$

т. е. градиент вычисляется с заданной абсолютной ошибкой. Предполагается, что про помехи не известно ничего, кроме этого условия. В частности, вектор r^k может не являться случайным, либо он может быть коррелирован с предыдущими помехами и т. д. Такая ситуация характерна для погрешностей вычислений и систематических ошибок измерений.

б) *Относительные детерминированные помехи* удовлетворяют условию

$$\| r^k \| \leqslant \varepsilon \| \nabla f(x^k) \|. \tag{6}$$

Иначе говоря, градиент вычисляется с относительной ошибкой. В остальном, как и выше, о природе r^k ничего не известно. Такие помехи возникают, например, при использовании приближенных формул, дающих фиксированную относительную ошибку.

в) *Абсолютные случайные помехи.* Предположим, что помехи r^k случайны, независимы при различных x, центрированы и имеют ограниченную дисперсию:

$$\mathbf{M}r^k = 0, \quad \mathbf{M}\| r^k \|^2 \leqslant \sigma^2. \tag{7}$$

Помехи такого типа характерны для задач, в которых градиент отыскивается в результате измерений на реальном объекте (экстремальное регулирование, планирование эксперимента), а также для задач с функцией типа среднего риска (1).

г) *Относительные случайные помехи* обладают теми же свойствами, что и в п в), однако их дисперсия убывает по мере приближения к точке минимума:

$$\mathbf{M}r^k = 0, \quad \mathbf{M}\| r^k \|^2 \leqslant \tau \| \nabla f(x^k) \|^2. \tag{8}$$

На практике часто встречаются и другие типы помех например случайные помехи с *систематической ошибкой* ($||\mathbf{M}^k||\leq\varepsilon$) или случайные ограниченные помехи ($\mathbf{M}r^k = 0$, $||r^k||\leq\varepsilon$). Однако их можно рассматривать как комбинацию основных типов, описанных выше. Поэтому мы ограничимся этими наиболее важными классами помех. Иногда (особенно в теоретических работах) предполагают, что уровень помех ε_k зависит от номера итерации и $\varepsilon_k \to 0$ при $k \to \infty$, Такое предположение представляется не очень реалистическим. В некоторых случаях можно добиться его выполнения путем повышения точности вычислений и уменьшения погрешности метода.

9. 2. Градиентный метод при наличии помех

1. Постановка задачи. Рассмотрим градиентный метод минимизации дифференцируемой функции $f(x)$ на $\mathbf{R}^n$ в ситуации, когда градиент вычисляется с ошибкой:

$$x^{k+1} = x^k - \gamma_k s^k, \quad s^k = \nabla f(x^k) + r^k. \tag{1}$$

Относительно помех r^k будут делаться предположения об их принадлежности одному из классов, описанных выше. Функция $f(x)$ будет предполагаться сильно выпуклой (с константой L) и с градиентом, удовлетворяющим условию Липшица (с константой L) — этот класс функций наиболее важен. Нас будет интересовать поведение обычного градиентного метода $\gamma_k \equiv \gamma$ при наличии помех, а также вопрос

о целесообразном выборе длины шага в условиях помех. Обоснование методов будет вестись с помощью общих теорем.

2. Абсолютные детерминированные помехи.

Теорема 1. *Пусть* $\|r^k\| \leqslant \varepsilon$, $\gamma_k \equiv \gamma$. *Тогда найдется* $\overline{\gamma} > 0$ *такое, что при* $0 < \gamma < \overline{\gamma}$ *в методе* (1) *будет*

$$\|x^k - x^*\| \leqslant \rho + q^k \|x^0 - x^*\|, \tag{2}$$

где $0 \leqslant q < 1$, $\rho = \rho(\varepsilon) \to 0$ *при* $\varepsilon \to 0$, x^* *— точка минимума* $f(x)$.

Доказательство. Введем функцию Ляпунова

$$V(x) = \frac{1}{2}\left(\|x - x^*\| - \frac{1}{l}\varepsilon\right)_+^2. \tag{3}$$

Учитывя, что (3) дифференцируема и имеет результат

$$\nabla V(x) = (\|x - x^*\| - \varepsilon/l)_+ \|x - x^*\|^{-1}(x - x^*), \quad \nabla V(x),$$

который удовлетворяет условию Липшица с константой 1, получаем

$$(\nabla V(x^k), s^k) = \left(\|x^k - x^*\| - \frac{1}{l}\varepsilon\right)_+ \frac{(\nabla f(x^k) + r^k, x^k - x^*)}{\|x^k - x^*\|} \geqslant$$

$$\geqslant \left(\|x^k - x^*\| - \frac{1}{l}\varepsilon\right)_+ (l\|x^k - x^*\| - \varepsilon) = 2lV(x^k),$$

$$\|s^k\|^2 = \|\nabla f(x^k) + r^k\|^2 \leqslant (L\|x^k - x^*\| + \varepsilon)^2 \leqslant$$

$$\leqslant a + bV(x^k) \leqslant a + (b/(2l))(\nabla V(x^k), s^k),$$

где a, b — некоторые константы, причем $a \to 0$ при $\varepsilon \to 0$.

Как нетрудно проверить на примерах, оценка (2) не является завышенной. Таким образом, наличие аддитивных помех приводит к тому, что градиентный метод с постоянным γ перестает сходиться к точке минимума. Он дает лишь возможность попасть в некоторую окрестность минимума, размеры которой тем меньше, чем меньше уровень помех. Сходимость к этой окрестности происходит со скоростью геометрической прогрессии.

Мы не выписывали выше точных значений констант (величин ρ, γ, q), интересуясь лишь качественной картиной процесса. Рассмотрим эти значения указаны для случая квадратичной функции.

Пусть $f(x) = (Ax, x)/2 - (b, x)$, $lI \leqslant A \leqslant LI$, $l > 0$, $\|r^k\| \leqslant \varepsilon$, $0 < \gamma < 2/L$. Тогда в методе (1) будет $\|x^{k+1} - x^*\| \leqslant q\|x^k - x^*\| + \gamma\varepsilon$, $q = \max\{|1 - \gamma l|, |1 - \gamma L|\}$. Используя известную лемму, получим оценку $\|x^k - x^*\| \leqslant \gamma\varepsilon/(1 - q) + q^k(\|x^0 - x^*\| - \gamma\varepsilon/(1 - q))$. В частности, при $\gamma = 2/(L + l)$ отсюда следует

$$\|x^k - x^*\| \leqslant \frac{\varepsilon}{l} + \left(\|x^0 - x^*\| - \frac{\varepsilon}{l}\right)\left(\frac{L - l}{L + l}\right)^k.$$

3. Относительные детерминированные помехи.

Теорема 2. *Пусть* $\|r^k\| \leqslant \alpha\|\nabla f(x^k)\|$, $\alpha < 1$, $\gamma_k \equiv \gamma$. *Тогда*

найдется $\bar{\gamma} > 0$ *такое, что при* $0 < \gamma < \bar{\gamma}$ *метод* (1) *сходится к* x^* *со скоростью геометрической прогрессии.*

Доказательство. Возьмем в качестве функции Ляпунова $V(x) = f(x) - f(x^*)$. Тогда

$$(\nabla V(x^k), s^k) = (\nabla f(x^k), \nabla f(x^k) + r^k) \geqslant (1-\alpha)\|\nabla f(x^k)\|^2 \geqslant$$
$$\geqslant (1-\alpha)\, 2lV(x^k),$$
$$\|s^k\|^2 \leqslant \|\nabla f(x^k)\|^2 (1+\alpha)^2 \leqslant 2(1+\alpha)^2 LV(x^k).$$

Таким образом, градиентный метод устойчив к относительным ошибкам, если их уровень менее 100%. Причина этого очевидна— всякое направление, составляющее с антиградиентом острый угол, является направлением убывания $f(x)$ и может быть использовано в качестве направления движения вместо градиента.

4. Абсолютные случайные помехи. Пусть помехи r^k случайны, независимы, $\mathbf{M}r^k = 0$ и $\mathbf{M}\|r^k\|^2 \leqslant \sigma^2$.

Теорема 3. *Найдется* $\bar{\gamma} > 0$ *такое, что при* $\gamma_k \equiv \gamma$, $0 < \gamma < \bar{\gamma}$ *в методе* (1)

$$\mathbf{M}(f(x^k) - f^*) \leqslant \rho(\gamma) + \mathbf{M}(f(x^0) - f^*) q^k, \qquad (4)$$
где $q < 1$, $\rho(\gamma) \to 0$ *при* $\gamma \to 0$.

Если

$$\gamma_k \to 0, \quad \sum_{k=0}^{\infty} \gamma_k = \infty, \qquad (5)$$

то $\mathbf{M}\|x^k - x^*\|^2 \to 0$. *Если же*

$$\sum_{k=0}^{\infty} \gamma_k^2 < \infty, \quad \sum_{k=0}^{\infty} \gamma_k = \infty, \qquad (6)$$

то $x^k \to x^*$ *п. н. Наконец, если* $\gamma_k = \gamma/k$, $\gamma > (2l)^{-1}$, *то*

$$\mathbf{M}(f(x^k) - f^*) \leqslant \frac{L\sigma^2\gamma^2}{2(2l\gamma - 1)k} + o\left(\frac{1}{k}\right). \qquad (7)$$

Доказательство. Возьмем $V(x) = f(x) - f^*$. Тогда

$$(\nabla V(x^k), \mathbf{M}s^k) = (\nabla f(x^k), \nabla f(x^k)) \geqslant 2lV(x^k),$$
$$\mathbf{M}\|s^k\|^2 = \|\nabla f(x^k)\|^2 + \mathbf{M}\|r^k\|^2 \leqslant \sigma^2 + (\nabla V(x^k), \mathbf{M}s^k).$$

Мы увидим далее (теорема 4), что вышеприведенные оценки не завышены, поэтому теорема 3 дает основания для следующих выводов. Во-первых, обычный вариант градиентного метода (с $\gamma_k \equiv \gamma$) при наличии аддитивных случайных помех не сходится к точке минимума,

а приводит лишь в окрестность минимума. Размеры этой области тем меньше, чем меньше γ. Во-вторых, выбирая убывающие γ_k, можно сделать метод сходящимся в том или ином вероятностном смысле (в среднем при $\gamma_k \to 0$ и почти наверное при

$$\sum_{k=0}^{\infty} \gamma_k^2 < \infty).$$

В-третьих, скорость сходимости при этом довольно медленна (порядка $O(1/k)$). Как мы увидим в дальнейшем, более высокой скорости сходимости нельзя добиться ни при каком выборе γ_k.

Уточним теорему 3 для квадратичной функции и помех постоянного уровня. Пусть

$$f(x) = (Ax, x)/2 - (b, x), \quad lI \leqslant A \leqslant LI, \quad l > 0,$$
$$\mathbf{M} r^k = 0, \quad \mathbf{M} r^k (r^k)^T = \sigma^2 I. \tag{8}$$

Будем считать, что начальное приближение x^0 случайно и симметрично распределено вокруг x^*: $\mathbf{M}(x^0 - x^*)(x^0 - x^*)^T = \alpha I$.

Теорема 4. *При любом* $0 < \gamma < 2/L$, $\gamma_k \equiv \gamma$ *в методе* (1) *при условиях* (8) *для величины*

$$U_k = \mathbf{M}(x^k - x^*)(x^k - x^*)^T \tag{9}$$

справедливы соотношения

$$U_k \to U_\infty = \gamma\sigma^2 A^{-1}(2I - \gamma A)^{-1}, \tag{10}$$
$$\| U_k - U_\infty \| \leqslant \| U_0 - U_\infty \| q^k, \quad q = \max\{(1 - \gamma l)^2, (1 - \gamma L)^2\} < 1. \tag{11}$$

Если $\gamma_k = \gamma/k$, $\gamma > (2l)^{-1}$, *то*

$$U_k = \frac{1}{k} B(\gamma) + o\left(\frac{1}{k}\right), \quad B(\gamma) = \gamma\sigma^2 \left(2A - \frac{1}{\gamma} I\right)^{-1}. \tag{12}$$

Величина $\| B(\gamma) \|$ *минимальна при* $\gamma = 1/l$,

$$\| U_k \| = \frac{1}{k} \frac{\sigma^2}{l^2} + o\left(\frac{1}{k}\right). \tag{13}$$

5. Относительные случайные помехи. Пусть помехи r^k такие же, как в предыдущем пункте, но их дисперсия удовлетворяет условию

$$\mathbf{M}\| r^k \| \leq \alpha \| \nabla f(x) \|^2. \tag{14}$$

Теорема 5. *При любом* α *существует* $\overline{\gamma}$ *такое, что при* $\gamma_k \equiv \gamma$, $0 < \gamma < \overline{\gamma}$ *методе* (1) *будет*

$$\mathbf{M}\| x^k - x^* \|^2 \leqslant c q^k, \quad q < 1. \tag{15}$$

Мы видим, что наличие случайных относительных помех любого уровня не приводит к нарушению сходимости.

Итак, в зависимости от типа помех их присутствие может либо сохранять, либо нарушать сходимость градиентного метода. Иногда сходимость можно восстановить за счет регулировки длины шага.

9. 3. Другие методы минимизации при наличии помех

1. Метод Ньютона. Вопрос о поведении метода Ньютона при наличии помех значительно более сложен, чем тот же вопрос для градиентного метода. Дело в том, что в этом методе может быть несколько источников помех (вычисление $\nabla f(x)$, $\nabla^2 f(x)$, обращение $\nabla^2 f(x)$) и их природа может быть различна (например, случайные ошибки в вычислении градиента и систематические в обращении матрицы). Мы не будем стараться рассмотреть все возможные ситуации, а остановимся на нескольких характерных примерах, интересуясь лишь качественным анализом процесса.

Пусть в результате всех вычислений (градиента, гессиана, решения системы линейных уравнений) получается вектор, отличающийся от истинного:

$$s^k = [\nabla^2 f(x^k)]^{-1} \nabla f(x^k) + r^k, \tag{1}$$

где r^k — помеха, и делается шаг

$$x^{k+1} = x^k - s^k. \tag{2}$$

Предположим, что помеха может содержать систематическую ошибку:

$$\| r^k \| \leqslant \varepsilon. \tag{3}$$

Как мы знаем, метод Ньютона сходится локально в некоторой области U. Ясно, что если в больше диаметра U, то сходимости заведомо нет — при любом x^0, сколь угодно близком к x^*, процесс выходит из U. Таким образом, возникает ситуация, которой не было в градиентном методе: при достаточно высоком уровне абсолютных помех метод Ньютона может вести себя бессмысленным образом (например, $||x^k - x^*||$ может возрастать) при любом x^0.

Возникновение систематических ошибок в методе Ньютона неизбежно, даже если $\nabla f(x)$ и $\nabla^2 f(x)$ вычисляются точно. Дело в том, что если число обусловленности μ точки минимума велико (а именно тогда применение метода Ньютона наиболее целесообразно), то матрица $\nabla^2 f(x^k)$ оказывается плохо обусловленной. Поэтому результат решения системы линейных уравнений $\nabla^2 (\{x^k) z = \nabla f(x^k)$ для определения шага метода отличается от точного решения вследствие

ошибок округления в ЭВМ. Это отличие (для плохо обусловленных систем) может быть значительным и приводит к развалу метода Ньютона.

Присутствие случайных или относительных ошибок не столь катастрофично, но может повлечь существенное замедление метода Ньютона. Пусть, например, требуется минимизировать квадратичную функцию

$$f(x) = (Ax,\ x)/2 - (b,\ x), \quad A > 0, \tag{4}$$

причем матрицы A и A^{-1} вычисляются точно, а градиент содержит случайную ошибку:

$$s^k = \nabla f(x^k) + r^k = Ax^k - b + r^k, \quad \mathbf{M}r^k = 0, \quad \mathbf{M}\|r^k\|^2 = \sigma^2. \tag{5}$$

Рассмотрим метод

$$x^{k+1} = x^k - \gamma_k A^{-1} s^k, \tag{6}$$

являющийся обобщением метода Ньютона за счет введения параметра γ_k. Как мы увидим в дальнейшем, этот метод ни при каком способе выбора γ_k не может сходиться быстрее чем $O(1/k)$. Но скорость сходимости такого же порядка может обеспечить гораздо более простой градиентный метод. Таким образом, здесь теряется основное преимущество метода Ньютона — его высокая скорость сходимости. Аналогичная ситуация возникает при наличии относительной ошибки. Если, например, градиент вычисляется с относительной ошибкой, то метод Ньютона может сходиться лишь со скоростью геометрической прогрессии. Лишь при высокой точности вычислений метод Ньютона сохраняет свои преимущества.

2. Многошаговые методы. Ограничимся вновь анализом некоторых характерных частных случаев. Начнем с *метода тяжелого шарика.* Можно показать, что при наличии абсолютных детерминированных помех в определении градиента он сходится в область вокруг минимума. Громоздкая выкладка показывает, что для квадратичной функции размер этой области, вообще говоря, больше, чем для градиентного метода. Приведем аналогичный результат, относящийся к абсолютным случайным помехам. Пусть

$$\begin{aligned} &f(x) = (Ax,\ x)/2 - (b,\ x), \quad lI \leqslant A \leqslant LI, \quad l > 0, \\ &s^k = \nabla f(x^k) + r^k = Ax^k - b + r^k, \quad \mathbf{M}r^k = 0, \quad \mathbf{M}r^k (r^k)^T = \sigma^2 I, \end{aligned} \tag{8}$$

причем помехи r^k взаимно независимы. Как можно показать, метод тяжелого шарика с постоянными коэффицентами

$$x^{k+1} = x^k - \alpha s^k + \beta (x^k - x^{k-1}) \tag{9}$$

в такой ситуации не сходится к $x^* = A^{-1}b$, а приводит лишь в область вокруг x^*. Поэтому рассмотрим метод с переменными коэффициентами, который удобно записать в форме

$$x^{k+1} = x^k - \alpha_k y^k, \quad y^{k+1} = y^k - \beta_k (y^k - s^k). \tag{10}$$

Наряду с ним рассмотрим градиентный метод

$$x^{k+1} = x^k - \gamma_k s^k. \tag{11}$$

Ограничимся коэффициентами вида

$$\alpha_k = \frac{1}{k}\alpha, \quad \beta_k = \frac{1}{k}\beta, \quad \gamma_k = \frac{1}{k}\gamma. \tag{12}$$

Теорема 1. *При любом выборе* α, β *метод* (10), (12) *сходится асимптотически не быстрее (в смысле величины* $\| M (x^k - x^*)(x^k - x^*)^T \|$), *чем метод* (11) *с* $\gamma_k = 1/(kl)$.

Таким образом, метод тяжелого шарика, превосходящий градиентный метод по скорости сходимости для задач без помех, является относительно менее эффективным при наличии помех.

Этот вывод относится только к асимптотическому поведению метода. На начальных итерациях, когда относительная величина помех мала, двухшаговый метод может превосходить одношаговый, как и для задач без помех.

Примерно такова же ситуация с методом сопряженных градиентов. Полный анализ его поведения при наличии помех очень сложен. При этом разные его варианты по-разному реагируют на ошибки. Можно показать, что при абсолютных и относительных помехах метод сопряженных градиентов вблизи минимума теряет преимущества перед градиентным. Лишь если помехи удовлетворяют условию типа (7), то метод сопряженных градиентов сохраняет свои достоинства.

3. Другие методы. Квазиньютоновские методы очень чувствительны к ошибкам вычисления градиента. Действительно, в них восстанавливается матрица $A = \nabla^2 f(x)$ по измерениям градиента:

$$Ap^i \approx y^i, \quad p^i = x^{i+1} - x^i, \; y^i = \nabla f(x^{i+1}) - \nabla f(x^i), \; i = 0, \ldots, k-1. \tag{13}$$

Если шаги малы (x^{i+1} близко к x^i), а измерения $\nabla f(x^i)$ содержат ошибки, то матрица восстанавливается плохо. Для задач со случайными аддитивными помехами с этим эффектом можно бороться путем увеличения числа измерений — нужно восстанавливать не по n значениям $\nabla f(x)$, как в детерминированном случае, а по $N > n$ замерам. При этом можно выписать рекуррентные формулы,. Для неслучайных помех такой прием, вообще говоря, не приводит к повышению точности.

Совершенно аналогичные замечания относятся и к методу секущих — чтобы сделать его работоспособным при наличии случайных помех, нужно брать число базисных точек заметно большее, чем размерность пространства.

Однако нужно помнить, что возможности всех методов, основанных на квадратичной аппроксимации, весьма ограничены в задачах с помехами — даже знание точной матрицы вторых производных не спасает положения.

9.4. Прямые методы

1. **Постановка задачи**. Пусть в произвольной точке x^k измеряется значение $f(x^k)$ с ошибкой η_k. По-прежнему будем говорить об *абсолютной (относительной) детерминированной ошибке,* если $|\eta_k| \leqslant \varepsilon \quad (|\eta_k| \leqslant \alpha(f(x^k) - f(x^*)))$, и об *абсолютной (относительной) случайной ошибке,* если η_k случайны, независимы, $\mathbf{M}\eta_k = 0$ и $\mathbf{M}\eta_k^2 \leqslant \sigma^2 \quad (\mathbf{M}\eta_k^2 \leqslant \tau(f(x^k) - f(x^*)))$. Задача заключается в изучении влияния разного рода ошибок на прямые методы минимизации и в модификации этих методов для преодоления влияния помех.

2. Разностные методы при случайных помехах. Рассмотрим методы некоторые методы в ситуации со случайными помехами. Начнем с наиболее типичного примера — *метода Кифера — Вольфовица* (метода разностной аппроксимации градиента):

$$x^{k+1} = x^k - \gamma_k s^k, \quad s^k = \sum_{i=1}^{n} \frac{1}{2\alpha_k} (\tilde{f}(x^k + \alpha_k e_i) - \tilde{f}(x^k - \alpha_k e_i)) e_i, \quad (1)$$

e_i — координатные орты. Здесь и далее

$$\tilde{f}(x) = f(x) + \eta, \quad (2)$$

причем случайные ошибки η независимы в различных точках и

$$\mathbf{M}\eta = 0, \quad \mathbf{M}\eta^2 \leqslant \sigma^2. \quad (3)$$

Обсудим вопрос о выборе пробных и рабочих шагов α_k, γ_k. Обозначим

$$s^k - \nabla f(x^k) = g^k + \xi^k,$$

где g^k — систематическая, а ξ^k — случайная ошибки. Если $f(x)$ дважды дифференцируема, а $\nabla^2 f(x)$ удовлетворяет условию Липшица, то в соответствии с известной леммой

$$\| g^k \| \leqslant c\alpha_k^2. \quad (4)$$

Для случайной составляющей погрешности оценки градиента имеем

$$\mathbf{M}\xi^k = 0, \quad \mathbf{M}\| \xi^k \|^2 \leqslant \sigma^2/(2\alpha_k^2). \quad (5)$$

Таким образом, при уменьшении α_k убывает систематическая погрешность, но растет случайная. Покажем, прежде всего, что можно так регулировать α_k, γ_k, чтобы обеспечить сходимость.

Теорема 1. *Пусть f(x) сильно выпукла и дважды дифференцируема,* $\nabla^2(x)$ *удовлетворяет условию Липшица, выполнено* (3) *и для* α_k, γ_k *справедливы соотношения*

$$\sum_{k=0}^{\infty} \gamma_k = \infty, \quad \sum_{k=0}^{\infty} \gamma_k \sigma_k^4 < \infty, \quad \sum_{k=0}^{\infty} \gamma_k^2 \alpha_k^2 < \infty, \quad \sum_{k=0}^{\infty} \gamma_k^2 \alpha_k^{-2} < \infty. \quad (6)$$

Тогда в методе (1) $x^k \to x^*$ *п. н. и* $\mathbf{M}\| x^k - x^* \|^2 \to 0$. *Если при этом* $\gamma_k = \gamma/k$, $\alpha_k = \alpha k^{-1/6}$ *и* γ *достаточно велико, то* $\mathbf{M}\| x^k - x^* \|^2 = O\left(k^{-2/3}\right)$.

Можно получить аналогичный результат для несимметричной разностной аппроксимации градиента при менее жестких предположениях о гладкости *f(x)*.

Таким образом, при наличии аддитивных случайных помех в измерении функции для сходимости следует и пробные, и рабочие шаги стремить к 0, причем пробные шаги следует уменьшать медленнее. Асимптотическая скорость сходимости зависит от выбора α_k, γ_k гладкости *f(x)* и вида разностной аппроксимации, однако она не превосходит $O(k^{-s})$, $s < 1$. Эти же выводы справедливы и для более общих алгоритмов.

Приведем более точные оценки скорости сходимости для квадратичной функции при постоянных аддитивных помехах:

$$f(x) = (Ax, x)/2 - (b, x), \quad A \geqslant lI > 0, \quad x \in \mathbf{R}^n,$$
$$\tilde{f}(x) = f(x) + \eta, \quad \mathbf{M}\eta = 0, \quad \mathbf{M}\eta^2 = \sigma^2, \quad (7)$$

где помехи η независимы в различных точках. Сопоставим *метод Кифера — Вольфовица (градиентный)*

$$x^{k+1} = x^k - \gamma_k s^k,$$
$$s^k = \sum_{i=1}^{n} \frac{1}{2\alpha_k} [\tilde{f}(x^k + \alpha_k e_i) - \tilde{f}(x^k - \alpha_k e_i)] e_i \quad (8)$$

и метод случайного поиска

$$x^{k+1} = x^k - \gamma_k s^k,$$
$$s^k = (2\alpha_k)^{-1} [\tilde{f}(x^k + \alpha_k h^k) - \tilde{f}(x^k - \alpha_k h^k)] h^k, \quad (9)$$

где h^k — случайный вектор, равномерно распределенный на единичной сфере (и не зависящий от η). Поскольку для квадратичной функции систематическая ошибка в разностной аппроксимации градиента равна 0 при любом α_k, здесь не нужно стремить α_k к 0. Будем считать, что в

(8) и (9) $\alpha_k \equiv \alpha > 0$. Используя изветную теорему, нетрудно доказать, что в методе (8) при $\gamma_k = \gamma/k$, $\gamma > 1/(2l)$

$$\mathbf{M}(x^k - x^*)(x^k - x^*)^T = \frac{1}{k} \cdot \frac{\gamma\sigma^2}{2\alpha^2}\left(2A - \frac{1}{\gamma} I\right)^{-1} + o\left(\frac{1}{k}\right), \qquad (10)$$

$$\mathbf{M}(x^k - x^*)(x^k - x^*)^T = \frac{1}{k} \cdot \frac{\gamma\sigma^2}{2\alpha^2}\left(2A - \frac{n}{\gamma} I\right)^{-1} + o\left(\frac{1}{k}\right). \qquad (11)$$

Отсюда следует, что если брать γ_k в (8) в n раз большим, чем в (9), то n шагов метода (9) будут асимптотически эквивалентны одному шагу метода (8). Учитывая, что трудоемкость мотода (8) в n раз больше, чем метода (9), получаем, что в данной ситуации методы (8) и (9) эквивалентны по их асимптотической эффективности. Этот вывод не зависит от обусловленности или каких-либо других свойств A.

Отметим в заключение, что к асимптотическим оценкам типа приведенных в теореме 1, следует относиться с большой осторожностью. Например, выбор $\alpha_k = \gamma k^{-1/6}$ означает, что нужно сделать миллион итераций, чтобы уменьшить пробный шаг в 10 раз. Поэтому практически счет будет происходить при постоянном α_k.

3. Другие методы. Для задач с помехами перестают быть работоспособными все методы, построенные на одномерных минимизациях (например, методы сопряженных направлений), поскольку такую минимизацию нельзя осуществить. Более перспективными являются методы, в которых строится нелокальная аппроксимация функции по ее значениям в ряде точек (типа симплексного поиска или метода барицентрических координат). Влияние помех сказывается в том, что эти методы перестают работать в окрестности минимума, где уровень помех сравним с приращениями функции. Если помехи случайны и центрированы, то методы можно модифицировать так, что они останутся работоспособными и в указанной области. Общая идея такой модификации — использовать большее число точек для построения аппроксимации функции, чем в детерминированном случае. Это позволяет усреднять помехи и получать все более точную аппроксимацию. Например, в симплексном методе можно многократно проводить вычисления функции в каждой вершине симплекса, сопоставляя точность оценки значений функции с их разностью в различных вершинах.

Более экономный способ заключается в пересчете аппроксимации после каждого нового измерения. Опишем схему подобных методов на упрощенной модели Пусть можно предполагать, что функция $f(x)$, $x \in \mathbf{R}^n$, аффинна в некоторой области: $f(x) \approx (a, x) + \beta$ и уже вычислены

ее значения с помехой в k $(k \geq n+1)$ точках: $y_i = (a, x^i) + \beta + \eta_i$, $i = 1, \ldots, k$, где η_i — случайные независимые помехи, $\mathbf{M}\eta_i = 0$, $\mathbf{M}\eta_i^2 = \sigma^2$. Рассмотрим $(n+1)$-мерные векторы $z^i = \{x^i, 1\}$, $c^* = \{a, \beta\}$ и запишем измерения в виде $y_i = (c^*, z^i) + \eta_i$. Найдем оценку для c^* методом наименьших квадратов, т. е.

$$c^k = \arg\min_c \sum_{i=1}^{k} (y_i - (c, z^i))^2 = \left(\sum_{i=1}^{k} z^i (z^i)^T\right)^{-1} \left(\sum_{i=1}^{k} z^i y_i\right) =$$

$$= \Gamma_k \sum_{i=1}^{k} z^i y_i, \quad \Gamma_k = \left(\sum_{i=1}^{k} z^i (z^i)^T\right)^{-1}. \qquad (12)$$

Этому методу можно придать рекуррентную форму — новое измерение в точке x^{k+1}: $y_{k+1} = (c^*, z^{k+1}) + \eta_{k+1}$, $z^{k+1} = \{x^{k+1}, 1\}$, может быть учтено с помощью следующей формулы:

$$c^{k+1} = c^k - \Gamma_{k+1} z^{k+1} ((c^k, z^{k+1}) - y_{k+1}),$$

$$\Gamma_{k+1} = \Gamma_k - \frac{\Gamma_k z^{k+1} (\Gamma_k z^{k+1})^T}{1 + (\Gamma_k z^{k+1}, z^{k+1})}, \quad k \geqslant n+1, \qquad (13)$$

$$\Gamma_{n+1} = \left(\sum_{i=1}^{n+1} z^i (z^i)^T\right)^{-1}.$$

Таким образом, на каждом шаге не нужно заново вычислять оценку для аппроксимирующей функции, решая систему линейных уравнений (12), а достаточно использовать простую рекуррентную формулу (13). Оценка c^k может быть использована для реализации шага спуска: $x^{k+1} = x^k - \gamma_k a^k$, $c^k = \{a^k, \beta_k\}$, и проверки согласованности линейной модели функции с измерениями. Разумеется, в реальных задачах линейная модель функции правомерна лишь локально, и метод минимизации должен включать «забывание» информации, полученной на ранних итерациях.

Совершенно аналогичные способы могут быть применены для восстановления квадратичной аппроксимации функции по результатам измерений, содержащих случайную ошибку.

9.5. Оптимальные методы при наличии помех

1. Потенциальные возможности итеративных методов при наличии помех. Для детерминированных «невозмущенных» задач, как мы видели, существует множество методов, каждому из которых присуща своя скорость сходимости. Так, для гладких сильно выпуклых функций метод тяжелого шарика сходится быстрее градиентного, метод сопряженных градиентов — быстрее метода тяжелого шарика,

метод Ньютона — еще более быстро и т. д. Вопрос об оптимальном в смысле скорости сходимости методе здесь весьма сложен. Оказывается, наличие помех в определенном смысле упрощает ситуацию — оно ограничивает возможности любых методов минимизации. В этом случае существует некая предельная скорость сходимости, которая не может быть превзойдена. Тот метод, для которого эта предельная скорость достигается, естественно считать оптимальным.

Начнем с результатов, устанавливающих *потенциальные возможности* по скорости сходимости произвольных итеративных алгоритмов (не обязатпьно связанных с минимизацией) при наличии случайных помех Рассмотрим итерационный процесс в $\mathbf{R}^n$:

$$x^{k+1} = x^k - \gamma_k s^k, \quad s^k = R(x^k) + \xi^k, \tag{1}$$

где $\gamma_k \geq 0$ — детерминированные скалярные множители, $R(x)$ — некоторая функция, а ξ^k — случайные помехи, предполагающиеся независимыми и центрированными ($\mathbf{M}\xi^k = 0$). Начальное приближение x^0 может быть либо детерминированным, либо случайным, в последнем случае предполагается, что $\mathbf{M}\|x^0\|^2 < \infty$ и x^0, ξ^i независимы. Предположим, что существует единственная точка x^* такая, что $R(x^*) = 0$ и $R(x)$ удовлетворяет условию линейного роста:

$$\|R(x)\| \leqslant L\|x - x^*\|. \tag{2}$$

Теорема 1. *Пусть для всех k*

$$\mathbf{M}\|\xi^k\|^2 \geqslant \sigma^2. \tag{3}$$

Тогда при сделанных выше предположениях для любого метода (1)

$$\mathbf{M}\|x^k - x^*\|^2 \geqslant 1/(a + kb), \quad a = 1/\mathbf{M}\|x^0 - x^*\|^2, \quad b = L^2/\sigma^2. \tag{4}$$

Подчеркнем, что в этой теореме, в отличие от любых теорем сходимости, приводившихся ранее, даются оценки скорости сходимости не сверху, а снизу. Теорема относится к любому способу выбора γ_k — в частности, и такому, для которого сходимость не имеет места.

Доказательство. Оценим условное математическое ожидание $\mathbf{M}(\|x^{k+1} - x^*\|^2 \mid x^k)$:

$$\mathbf{M}(\|x^{k+1} - x^*\|^2 \mid x^k) = \|x^k - x^* - \gamma_k R(x^k)\|^2 + \gamma_k^2 \mathbf{M}\|\xi^k\|^2,$$

$$\|x^k - x^* - \gamma_k R(x^k)\| \geqslant (\|x^k - x^*\| - \gamma_k \|R(x^k)\|)_+ \geqslant$$

$$\geqslant (\|x^k - x^*\| - \gamma_k L\|x^k - x^*\|)_+,$$

$$\mathbf{M}(\|x^{k+1} - x^*\|^2 \mid x^k) \geqslant (1 - \gamma_k L)_+^2 \|x^k - x^*\|^2 + \gamma_k^2 \sigma^2.$$

Отсюда

$$\mathbf{M}\|x^{k+1}-x^*\|^2 \geqslant (1-\gamma_k L)_+^2 \mathbf{M}\|x^k-x^*\|^2+\gamma_k^2\sigma^2.$$

Стоящая справа кусочно-квадратичная функция достигает минимума по γ_k при $\gamma_k^* = L\mathbf{M}\|x^k-x^*\|^2/(L^2\mathbf{M}\|x^k-x^*\|^2+\sigma^2)$. Отсюда получаем

$$\mathbf{M}\|x^{k+1}-x^*\|^2 \geqslant (1-\gamma_k^* L)_+^2\, \mathbf{M}\|x^k-x^*\|^2+(\gamma_k^*)^2\sigma^2 = \\ = \sigma^2\mathbf{M}\|x^k-x^*\|^2/(L^2\mathbf{M}\|x^k-x^*\|^2+\sigma^2),$$

или, обозначая $u_k = 1/(\mathbf{M}\|x^k-x^*\|^2)$, $u_{k+1} \leqslant L^2/\sigma^2 + u_k$. Таким образом, $u_k \leqslant u_0 + kL^2/\sigma^2$, т. е

$$\mathbf{M}\|x^k-x^*\|^2 \geqslant [1/\mathbf{M}\|x^0-x^*\|^2 + kL^2/\sigma^2]^{-1}.$$

Из теоремы 1 следует, что любой метод вида (1) при сделанных выше предположениях не может сходиться быстрее $1/(a+bk)$, или асимптотически — быстрее $O(1/k)$.

Приведем некоторые примеры использования этого результата. Рассмотрим *градиентный метод* минимизации $f(x)$:

$$x^{k+1} = x^k - \gamma_k s^k, \quad s^k = \nabla f(x^k) + \xi^k \tag{5}$$

при абсолютных случайных помехах:

$$\mathbf{M}\xi^k = 0, \quad \mathbf{M}\|\xi^k\|^2 \geqslant \sigma^2 \tag{6}$$

(обратите внимание, что здесь знак неравенства для дисперсии помех изменен на обратный по сравнению с 9.2). Предположим, что $f(x)$ имеет точку минимума x^*, а градиент $\nabla f(x)$ удовлетворяет условию Липшица с константой L. Тогда мы находимся в условиях применимости теоремы 1, и из нее следует, что при любом выборе γ_k для метода (5) справедлива оценка

$$\mathbf{M}\|x^k-x^*\|^2 \geqslant (1/\mathbf{M}\|x^0-x^*\|^2 + kL^2/\sigma^2)^{-1}. \tag{7}$$

Иначе говоря, никакой вариант градиентного метода при наличии абсолютных случайных помех не может сходиться быстрее $O(1/k)$ (точнее, $\mathbf{M}\|x^k-x^*\|^2 \geqslant \sigma^2/(L^2k) + o(1/k)$). Заметим, что для градиентного метода с $\gamma_k = \gamma/k$ было $\mathbf{M}\|x^k-x\|^2 = O(1/k)$, т. е. он асимптотически оптимален по порядку скорости сходимости. Более точно вопрос об оптимальности градиентного метода будет исследован далее.

Рассмотрим теперь *метод Ньютона* при наличии помех. Будем считать, что матрица $[\nabla^2 f(x^k)]^{-1}$ вычисляется точно, а градиент содержит аддитивную случайную помеху ξ^k. В этом случае метод Ньютона (модифицированный за счет введения параметра, задающего длину шага) принимает вид

$$x^{k+1} = x^k - \gamma_k [\nabla^2 f(x^k)]^{-1} (\nabla f(x^k) + \xi^k). \quad (8)$$

Относительно помех ξ^k будем считать, что они независимы и

$$\mathbf{M}\xi^k = 0, \quad \mathbf{M}\|\xi^k\|^2 \geqslant \sigma^2. \quad (9)$$

Можно показать, что в условиях теоремы о сходимости «невозмущенного» метода Ньютона детерминированная часть процесса (8) (т. е. $R(x^k) = [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$) в окрестности решения удовлетворяет условию Липшица, а случайная часть имеет дисперсию, ограниченную снизу. Таким образом, метод (8) также не может сходиться быстрее, чем со скоростью *O(1/k)*. Иначе говоря, наличие случайных помех уничтожает преимущества быстро сходящихся методов минимизации.

Приведем результат, аналогичный теореме 1, но применительно к относительным помехам.

Теорема 2. *Пусть выполнены, предположения, сформулированные в начале параграфа, и для всех k*

$$\mathbf{M}\|\xi^k\|^2 \geqslant \tau \|x^k - x^*\|^2. \quad (10)$$

Тогда для любого метода (1)

$$\mathbf{M}\|x^k - x^*\|^2 \geqslant \mathbf{M}\|x^0 - x^*\|^2 q^k, \quad q = \tau/(L^2 + \tau). \quad (11)$$

В качестве первого примера использования теоремы 2 рассмотрим *градиентный метод* при случайных относительных помехах. Пусть *f(x)* дифференцируема, существует точка минимума *x**, $\nabla f(x)$ удовлетворяет условию Липшица с константой *L*, а помеха в определении градиента независима при различных *k* и удовлетворяет условиям $\mathbf{M}\xi^k = 0$, $\mathbf{M}\|\xi^k\|^2 \geqslant \tau \|x^k - x^*\|^2$. Тогда в методе (5) при любых γ_k выполняется неравенство (11). Иными словами, градиентный метод при случайных относительных помехах не может сходиться быстрее, чем со скоростью геометрической прогрессии.

Вторым примером может служить метод *случайного поиска.* Пусть *f(x)* — квадратичная функция:

$$f(x) = (Ax, x)/2 - (b, x), \quad lI \leqslant A \leqslant LI, \quad l > 0. \quad (12)$$

Рассмотрим метод

$$x^{k+1} = x^k - (\gamma_k/(2\alpha))(f(x^k + \alpha h^k) - f(x^k - \alpha h^k))\, h^k, \quad (13)$$

где h^k — случайный равномерно распределенный на единичной сфере вектор, $\alpha>0$ — фиксированная длина пробного шага. Метод может быть записан в виде

$$x^{k+1} = x^k - \gamma_k h^k (h^k)^T \nabla f(x^k) = x^k - \gamma_k s^k,$$
$$s^k = h^k (h^k)^T \nabla f(x^k).$$

Используя результат упражнения 1, получаем

$$R(x^k) = \mathbf{M}s^k = \frac{1}{n}\nabla f(x^k),$$

$$\mathbf{M}\|\xi^k\|^2 = \mathbf{M}\|s^k - R(x^k)\|^2 = \frac{n-1}{n^2}\|\nabla f(x^k)\|^2 \geqslant \frac{n-1}{n^2} l^2 \|x^k - x^*\|^2.$$

Из теоремы 2 следует, что при любом способе выбора γ_k метод случайного поиска не может сходиться быстрее, чем геометрическая прогрессия со знаменателем

$$q = (n-1)l^2/(L^2 + (n-1)l^2). \tag{14}$$

В частности, для $f(x) = \|x\|^2/2$, $x \in \mathbf{R}^n$, метод случайного поиска сходится не быстрее прогрессии со знаменателем $(n—1)/n$.

Теорему 2 можно несколько уточнить для случая, когда $R(x)$ линейна, а для помехи известна оценка снизу не только для дисперсии, но и для матрицы ковариаций. Расматривается метод

$$x^{k+1} = x^k - \Gamma_k(A(x^k - x^*) + \xi^k), \tag{15}$$

где ξ^k независимы, x^0 случайный вектор, A^{-1} существует и

$$\mathbf{M}\xi^k = 0, \quad \mathbf{M}\xi^k(\xi^k)^T \geqslant B > 0, \quad \mathbf{M}(x^0 - x^*)(x^0 - x^*)^T > 0, \tag{16}$$

а Γ_k — детерминированные матрицы $n \times n$.

Теорема 3. *В методе* (15) *при любых* Γ_k *справедлива оценка*

$$\mathbf{M}(x^k - x^*)(x^k - x^*)^T \geqslant \left[(\mathbf{M}(x^0 - x^*)(x^0 - x^*)^T)^{-1} + kA^TB^{-1}A\right]^{-1} = \frac{1}{k}A^{-1}B(A^T)^{-1} + o\left(\frac{1}{k}\right).$$

В качестве приложения рассмотрим *обобщение градиентного метода* минимизации квадратичной функции

$$f(x) = (Ax, x)/2 - (b, x), \quad A \geqslant lI > 0$$

при наличии помех:

$$x^{k+1} = x^k - \Gamma_k(\nabla f(x^k) + \xi^k), \quad \mathbf{M}\xi^k = 0, \quad \mathbf{M}\xi^k(\xi^k)^T = \sigma^2 I. \tag{18}$$

Применяя теорему 3, получаем, что при любых Γ_k

$$\mathbf{M}(x^k - x^*)(x^k - x^*)^T \geqslant \left(U_0^{-1} + \frac{k}{\sigma^2}A^2\right)^{-1} = \frac{\sigma^2}{k}A^{-2} + o\left(\frac{1}{k}\right), \tag{19}$$

$$U_0 = \mathbf{M}(x^0 - x^*)(x^0 - x^*)^T,$$

$$\|\mathbf{M}(x^k - x^*)(x^k - x^*)^T\| \geqslant \frac{\sigma^2}{kl^2} + o\left(\frac{1}{k}\right), \tag{20}$$

причем равенство в (19), (20) достигается при

$$\Gamma_k = \left(kA + \sigma^2 A^{-1}U_0^{-1}\right)^{-1} = k^{-1}A^{-1} + o(1/k). \tag{21}$$

Сопоставляя (20) с оценкой (13) п. 9.2 для градиентного метода, получаем, что при данных условиях выбор $\gamma_k = 1/kl$ в градиентном методе является асимптотически оптимальным.

2. Оптимальные алгоритмы. До сих пор мы ограничивались довольно узким классом алгоритмов — линейными рекуррентными.

Однако вопрос об оптимальности можно решать для гораздо более общего класса процедур. В ряде случаев можно установить потенциальные возможности любых (не обязательно рекуррентных или линейных) методов минимизации при наличии случайных помех. Основным инструментом здесь является известное в статистике неравенство Крамера — Рао (информационное неравенство).

Пусть функция *f(x)* квадратична:

$$f(x) = (Ax,\ x)/2 - (b,\ x), \qquad A > 0, \tag{22}$$

а ее градиент вычисляется со случайной помехой ξ. Предположим, что помехи ξ независимы и одинаково распределены (раньше мы такого предположения не делали). Пусть уже вычислены значения $r^1 = \nabla f(x^1) + \xi^1,\ \ldots,\ r^k = \nabla f(x^k) + \xi^k$ в некоторых точках $x^1,\ldots,\ x^k$. Наконец, пусть матрицы A и A^{-1} известны. Тогда $x^i - x^* = A^{-1}r^i - A^{-1}\xi^i,\ i = 1,\ \ldots,\ k$. Обозначим $z^i = x^i - A^{-1}r^i$, $\eta^i = -A^{-1}\xi^i$. Тогда $z^i = x^* + \eta^i$. Величины z^i известны (так как x^i, r^i и A^{-1} известны), а величины η^i независимы и одинаково распределены (ибо такими являются ξ^i). Таким образом, задача свелась к следующей. Заданы векторы $z^i = x^* + \eta^i$, где η^i — реализации независимой, одинаково распределенной случайной величины. Требуется по ним оценить x^*.

Это — классическая задача оценки параметров, рассматриваемая в статистике. Для нее справедливо *неравенство Крамера — Рао,* утверждающее, что если η^i имеют плотность $p_\eta(z)$, эта плотность регулярна (т. е. справедливо равенство $\int \nabla p_\eta(z)\,dz = 0$) и существует *фишеровская информационная матрица*

$$J = \int \frac{\nabla p_\eta(z)\,\nabla^T p_\eta(z)}{p_\eta(z)}\,dz, \qquad 0 < J < \infty, \tag{23}$$

то для любой несмещенной оценки $\hat{x}^k$ вектора x^* по измерениям z^i, $i = 1, \ldots, k$, имеет место неравенство

$$\mathbf{M}\,(\hat{x}^k - x^*)(\hat{x}^k - x^*)^T \geqslant k^{-1}J^{-1}. \tag{24}$$

Иными словами, существует нижняя граница точности произвольных несмещенных оценок. Используя (24), приходим к следующему результату.

Теорема 4. *Пусть помехи ξ^i имеют плотность p(z), причем p(z) регулярна и*

$$J = \int \frac{\nabla p \nabla^T p}{p}\,dz$$

существует, $0 < J < \infty$. *Тогда для любой несмещенной оценки* $\hat{x}^k$ *точки минимума* x^* *функции* (22), *построенной по измерениям* $r^i = \nabla f(x^i) + \xi^i$, $i = 1, \ldots, k$, *в* k *точках, справедливо неравенство*

$$\mathbf{M}(\hat{x}^k - x^*)(\hat{x}^k - x^*)^T \geqslant k^{-1}A^{-1}JA^{-1}. \tag{25}$$

Важно, что сюда не входят точки измерения $x^1, \ldots, x^k$. Таким образом, при любом способе выбора k точек измерения градиента нельзя найти минимум с точностью, большей чем задаваемая неравенством (25).

Остается построить метод, для которого указанная нижняя граница достигается. Если ограничиться линейными алгоритмами

$$x^{k+1} = x^k - \gamma_k H(\nabla f(x^k) + \xi^k), \tag{26}$$

где $H > 0$ — некоторая матрица, то получаем, что асимптотически оптимальный выбор γ_k и H таков:

$$\gamma_k = 1/k, \quad H = A^{-1}, \tag{27}$$

при этом

$$\mathbf{M}(x^k - x^*)(x^k - x^*)^T \leqslant k^{-1}A^{-1}BA^{-1} + o(k^{-1}), \quad B = \mathbf{M}\xi\xi^T. \tag{28}$$

Отсюда получаем, что если ξ^i распределены нормально, то правая часть (25) совпадает с правой частью (28). Таким образом, для случая нормальных помех алгоритм (26), (27) является *асимптотически оптимальным* (не только среди линейных или рекуррентных алгоритмов). Для других распределений помехи алгоритм (26), (27), вообще говоря, не оптимален. Более того, можно показать, что правая часть (25) строго меньше правой части (28) для любого распределения, отличного от нормального. В этом случае оптимальный алгоритм можно получить, введя нелинейность в итерационный процесс

$$x^{k+1} = x^k - \gamma_k \varphi(\nabla f(x^k) + \xi^k), \tag{29}$$

где функция φ: $\mathbf{R}^n \to \mathbf{R}^n$ и γ_k выбираются следующим образом:

$$\varphi(z) = J^{-1}A^{-1}\nabla \ln p(z), \quad \gamma_k = 1/k. \tag{30}$$

Для нормальных помех метод (29), (30) переходит в (26), (27).

Можно показать, что при определенных условиях на $p(z)$ распределение величины $\sqrt{k}(x^k - x^*)$ для метода (29), (30) стремится к нормальному со средним 0 и матрицей ковариаций $A^{-1}JA^{-1}$. Сопоставляя это с правой частью (25), получаем, что метод (29), (30) является *асимптотически оптимальным.*

Практическая реализация метода (29), (30) затруднительна, так как в нем нужно знать матрицу A^{-1}, а также плотность распределения помехи. Мы не будем останавливаться на способах преодоления этих трудностей. Здесь более важен принципиальный факт — возможность

построения асимптотически оптимального алгоритма решения задачи минимизации при наличии случайных помех, причем этот алгоритм оказывается рекуррентным.

Подчеркнем еще, что все выводы здесь носили асимптотический характер. Оптимальный алгоритм для конечных k в случае нормальных помех дается выражением (21). Видно, что на начальных шагах $\left(k \ll \sigma^2 A^{-2} U_0^{-1}\right)$ Γ_k примерно постоянно: $\Gamma_k \approx \sigma^2 U_0 A$, а для больших k Γ_k убывает как k^{-1}: $\Gamma_k = k^{-1} A^{-1} + o(k^{-1})$.

Отметим также, что оптимальные алгоритмы предполагают точное знание закона распределения помехи и неустойчивы к отклонению истинного распределения от предполагаемого. Существуют способы преодоления этой трудности (так называемые *робастные алгоритмы* минимизации).

9.6. Псевдоградиентный метод с возмущением на входе для нестационарной задачи безусловной оптимизации

Проблема оптимизации того или иного функционала встает во многих практических приложениях. Хотя иногда экстремальные значения можно найти аналитически, зачастую инженерные системы имеют дело с неизвестным функционалом, значение которого или его градиента можно вычислять в задаваемых точках. Также встречаются задачи, в которых оптимизируемый функционал может изменяться во времени и сама точка экстремума может дрейфовать. В таком случае постановки задачи могут отличаться в зависимости от цели оптимизации и информации доступной для измерения. Обычно рассматривают два варианта поведения дрейфа функции: когда есть некоторый асимптотический функционал, к которому другие сходятся со временем или когда такого функционала нет. Мы рассмотрим более сложный второй случай.

Задачи оптимизации можно рассматривать в постановках с дискретным и непрерывным временем. Здесь мы ограничимся рассмотрением моделей первого типа. Пусть $f(x,n)$ - функционал, который необходимо минимизировать в момент времени n ($n \in N$). Б. Т. Поляк для решения подобных проблем детально рассматривал методы Ньютона и градиентный, которые применимы в случае дважды дифференцируемого функционала при условии $l < \nabla^2 f_k(x) < L$. Оба

метода полагаются на возможность прямого измерения градиента функционала в произвольной точке

В реальном мире измерения всегда подразумевают наличие помех. Иногда алгоритмы точно решающие проблему на бумаге не дают состоятельных оценок точки экстремума на практике. Устойчивость алгоритма к помехам очень важна практически во всех инженерных приложениях.

Для решения задач в условиях помех в пятидесятые годы прошлого столетия появляются методы стохастической аппроксимации Роббинса-Монро и Кифера-Вольфовица. Общий подход для поиска экстремума, используемый в алгоритмах стохастической аппроксимации, может быть формализован следующим образом

$$\hat{\theta}_{n+1} = \hat{\theta}_n - \alpha_n \hat{g}_n(\hat{\theta}_n). \qquad (1)$$

где $\{\hat{\theta}_n\}$ — генерируемая алгоритмом последовательность оценок точки экстремума, $\hat{g}_n$ — псевдоградиент (заменяющий градиент из метода Ньютона), который "в среднем" должен совпадать с градиентом и близок к нулю, когда его аргумент стремится к точке экстремума. Важными свойствами алгоритмов записанных в форме (1) являются простота и рекуррентность, в силу которых они стали активно применяться в разных областях науки и техники.

Алгоритмы стохастической аппроксимации с одним или двумя измерениями на каждой итерации с пробным одновременным возмущением на входе появились в работах различных исследователей конце 80-х, начале 90-х гг. XX в. В англоязычной литературе они получили название Simultaneous Perturbation Stochastic Approximation (SPSA). Эти алгоритмы известны состоятельностью оценок при почти произвольных помехах наблюдения, которые должны быть только как-то ограничены и независимы на каждой итерации от пробного случайного возмущения на входе. Более того, количество измерений делающихся на одной итерации составляет всего 1 или 2 вне завиемости от размерности d пространства состояний, что позволяет существенно повысить скорость сходимости в многомерном случае ($d >> 1$), так как у алгоритмов оценивающих градиент через конечную разность количество измерений на каждом шаге составляет $2d$.

Алгоритмы стохастической аппроксимации первоначально обосновывались в условиях, неподразумевающих нестационарность функционала. Существует версия алгоритма градиентного спуска для нестационарного случая и доказана ее сходимость в некотором смысле. Было предложено использовать для оптимизации нестационарных функционалов алгоритмы типа стохастической аппроксимации с

пробным одновременным возмущением на входе, которые могли бы быть более эффективными, так как они полагаются только на одно или два измерения на каждом шаге, а значит, способны быстрее адаптироваться к изменениям функционала. Кроме того, они более устойчивы к помехам.

Здесь мы рассматриваем применение алгоритма стохастической аппроксимации с пробным одновременным возмущением на входе для задачи оптимизации нестационарного функционала. Ниже будет рассмотрена постановка задачи для оптимизации существенно более общая, чем в некоторых работах, в силу того что в ней минимизируемая функция должна быть только один раз дифференцируема и не предполагается возможность прямого измерения градиента, а помехи наблюдения могут быть почти произвольными. Далее будет сформулированы алгоритм и теорема о среднеквадратичной стабилизируемости оценок, приводится ее доказательство. В заключение для иллюстрации приводится результат выполнения данного алгоритма для отслеживания дрейфа точки на плоскости.

1. Постановка задачи

Рассмотрим задачу минимизации нестационарного функционала среднего риска:

$$f(x,n) = E_w\{F(x,w,n)\} \to \min_x, \tag{2}$$

где $x \in \mathbf{R}^d$, $w \in \mathbf{R}^p$, $n \in \mathbf{N}$, $E_w\{\ \}$ — математическое ожидание относительно σ-алгебры, порождаемой случайными величинами *w*.

Требуется оценить θ_n — точку минимума функции $f(x, n)$, изменяющеюся с течением времени:

$$\theta_n = argmin_x f(x,n).$$

Пусть на каждой итерации *n* мы можем измерять значение:

$$y_n = F(x_n, w_n, n) + v_n, \tag{3}$$

где x_n — точка, и которой производится наблюдение, w_n — стучайные величины, выражающие неконтролируемую неопределенность, v_n — искажения в наблюдениях.

Время является дискретным и определяется номером шага (итерации) *n*.

Для характеристики поведения оценок точек минимума нестационарного функционала введем два определения.

Определение 1. *Последовательность оценок* $\hat{\theta}_n$ *точек минимума* θ_n стабилизируется в среднеквадратичном смысле, *если существует такое* $C > 0$, *что*

$$E\|\hat{\theta}_n - \theta_n\|^2 \le C,$$

где математическое ожидание $E\{\cdot\}$ *берется по всем неопределенностям, возникающим при наблюдениях, а также по случайным величинам, генерируемым при построении оценки.*

Определение 2. *Число* $\bar{L}$ *называется* ассимптотически эффективной границей среднеквадратичных невязок оценивания, *если для последовательности оценок* $\{\hat{\theta}_n\}$ *точек минимума* θ_n *для любого* $\varepsilon > 0$ *существует такое* $N \in \mathbf{N}$, *что для всех* $n > N$

$$E\|\hat{\theta}_n - \theta_n\|^2 \le \bar{L} + \varepsilon < \infty.$$

Далее будем рассматривать задачу о построении последовательности оценок $\{\hat{\theta}_n\}$ дчя задачи (2), удовлетворяющих определениям 1 или 2. при следующих условиях.

Будем считать, что дрейф минимума ограничен по норме следующим образом:

$$(\mathbf{A}) \quad \|\theta_n - \theta_{n-1}\| \le A.$$

Функции $f(\cdot,n)$ сильно выпуклыми по первому аргументу для каждого n:

$$(\mathbf{B}) \quad \langle \nabla f(x,n), x - \theta_n \rangle \ge \mu \|x - \theta_n\|^2$$

Градиент $\nabla F(\cdot,w,n)$ удовлетворяет условию Липшица с константой B, $\forall\, n$, $\forall\, w$:

$$(\mathbf{C}) \quad \|\nabla F(x,w,n) - \nabla F(y,w,n)\| \le B\|x - y\|$$

Средний модуль разности значений функции $F(x, \cdot, n)$ в точке в моменты n и $n + 1$ ограничен следующим образом:

$$(\mathbf{D}) \quad E_{w_1,w_2}|F(x,w_1,n+1) - F(x,w_2,n)| \le C\|x - \theta_n\| + D;$$

(**E**) Функции $F(\cdot,w,n)$ и $\nabla_x F(\cdot,w,n)$ равномерно ограничены:

$$F(x,w,n) \le \Phi(x,n) < \infty, \quad \forall w \in \mathbf{R}^p,$$
$$\nabla_x F(x,w,n) \le \Psi(x,n) < \infty, \quad \forall w \in \mathbf{R}^p$$

(**F**) Для помехи наблюдения v_n выполнены условия

$$|v_{2n} - v_{2n-1}| \le \sigma_v.$$

либо, если они предствляют собой последовательность случайных величин, то

$$E\{|v_{2n} - v_{2n-1}|^2\} \le \sigma_v^2.$$

Заметим, чго

1). Последнему условию удовлетворяют детерминированные, но ограниченные последовательности $\{ v_n \}$.

2). Ограничение типа (**A**) включает как дрейф типа случайных блужданий, так и направленный дрейф в определенную сторону. В конкретных задачах, блуждание типа броуновского движения может быть описано и без введение нестационарности в функционал среднего риска. Необходимость введения нестационарности появляется при наличии как случайной, так и детерминированной составляющей ограничения на дрейф. Например, можно рассматривать такое ограничение:

$$(\mathbf{A}') \quad \theta_n \le A_1\theta_{n-1} + A_2 + \xi_n,$$

где ξ_n является случайной величиной.

Мы ограничимся условием ограниченности дрейфа по норме типа (**A**). Среднеквадратичная стабилизируемость оценок алгоритма поиска минимума в условиях (**A**) означает применимость его к широкому классу различных задач.

2. Алгоритм

Зададим последовательность пробных одновременных возмущений $\{\Delta_n\}$, подаваемых на вход, как независимую последовательность бернуллиевских векторов, у которых каждая компонента принимает значения $\pm 1/\sqrt{d}$ с вероятностями $\frac{1}{2}$.

Выберем некоторый начальный вектор $\hat{\theta}_0 \in \mathbb{R}^d$. Будем оценивать последовательность точек минимума $\{ \theta_n \}$ последовательностью $\{ \hat{\theta}_n \}$, определяемой алгоритмом стохастической оптимизации с пробным одновременным возмущением на входе, который имеет следующий вид:

$$\begin{cases} x_{2n} = \hat{\theta}_{2n-2} + \beta\Delta_n,\ x_{2n-1} = \hat{\theta}_{2n-2} - \beta\Delta_n, \\ y_n = F(x_n, w_n, n) + v_n, \\ \hat{\theta}_{2n} = \hat{\theta}_{2n-2} - \frac{\alpha}{2\beta}\Delta_n(y_{2n} - y_{2n-1}), \\ \hat{\theta}_{2n-1} = \hat{\theta}_{2n-2}. \end{cases}$$

(**G**) Будем считать, что случайные величины Δ_n (рандомизация алгоритма) независимы от $\hat{\theta}_k$ помех w_k и $\hat{\theta}_0$, а также от v_k, если они предполагаются случайной природы, k= 1,2,....2n.

3. Среднеквадратичная стабилизация оценок алгоритма

Обозначим $H = 2\alpha\beta B + 2\alpha A + 2A + \frac{\alpha^2}{2\beta^2}(2B\beta C + CD)$. Определим константу K и параметр $\delta > 0$ из условия

$$K = 1 - 2\alpha\mu + C^2\frac{\alpha^2}{4\beta^2} + \delta H < 1.$$

Пусть

$$L = 2A^2 + 2A\alpha\beta B + \alpha^2 B^2 + \frac{\alpha^2}{4\beta}4BD + \frac{D^2\alpha^2}{4\beta^2} + \frac{\alpha^2}{4\beta^2}\sigma_v^2 + \frac{H}{4\delta}.$$

Теорема 1. *Пусть выполнены условия* (**A**)—(**G**) *на функции f и F, а так же* θ_n, $\hat{\theta}_n$, v_n, w_n, y_n *и* Δ_n, *величины* α *и* $\beta > 0$:

$$0 < 2\alpha\mu - C^2\frac{\alpha^2}{4\beta^2}$$

*Тогда оценки алгоритма (**A**) стабилизируются в среднеквадратичном смысле и справедлива оценка*

$$E\{\|\theta_n - \hat{\theta}_n\|^2\} \le K^n\|\theta_0 - \hat{\theta}_0\|^2 + \frac{L(1 - K^n)}{1 - K}. \qquad (5)$$

Заметим, что, в частности, в Теореме 1 устанавливается ассимптотически эффективная граница среднеквадратичных невязок оценивания $L = L/(1 — K)$.

Условия (**A**)-(**C**),(**E**)-(**G**) являются стандартными для доказательства состоятельности оценок алгоритмов стохастической аппроксимации с возмущением на входе. Ранее факт среднеквадратичной стабилизации оценок алгоритма (4) был доказан при более жестких ограничениях

Доказательство. Обозначим

$$dreif_n = \theta_{2n} - \theta_{2n-2},\ step_n = \frac{\alpha}{2\beta}(y_{2n} - y_{2n-1})\Delta_n,$$

$$err_{n-1} = \hat{\theta}_{2n-2} - \theta_{2n-2},$$

В силу алгоритма (4) и условия (**A**) для квадрата нормы разности $\|\hat{\theta}_{2n} - \theta_{2n}\|$ имеем оценку

$$\|err_n\|^2 \le \|err_{n-1}\|^2 + \|dreif_n\|^2 + \|step_n\|^2 +$$

$$+2\langle dreif_n, step_n\rangle - 2\langle dreif_n, err_{n-1}\rangle - 2\langle step_n, err_{n-1}\rangle \le$$

$$\le \|err_{n-1}\|^2 + A^2 + \|step_n\|^2 + 2\langle step_n, dreif_n\rangle - \tag{6}$$

$$-2 < dreif_n, err_{n-1}\rangle - 2\langle err_{n-1}, step_n\rangle.$$

1. В силу модели наблюдения (3) для последнего слагаемого имеем

$$-\langle err_{n-1}, step_n\rangle = -\langle err_{n-1}, \frac{\alpha}{2\beta}\Delta_n(F(\hat{\theta}_{2n-2} + \beta\Delta_n, w_{2n}, 2n) -$$

$$-F(\hat{\theta}_{2n-2} - \beta\Delta_n, w_{2n-1}, 2n-1) + v_{2n} - v_{2n-1})\rangle.$$

Обозначим $E_n\{\cdot\}$ условное математическое ожидание относительно σ-алгебры, порождаемой случайными величинами $\theta_1 \dots, \theta_{n-1}$, $\hat{\theta}_1 \dots, \hat{\theta}_{n-1}$, Применив к последней формуле $E_n\{\cdot\}$, используя $E_n\{\Delta_n(v_{2n} - v_{2n-1})\} = 0$, добавив и отняв $F(\hat{\theta}_{2n-2}, w_{2n}, 2n)$ и $F(\hat{\theta}_{2n-2}, w_{2n-1}, 2n-1)$, получаем

$$E_n\{-\langle err_{n-1}, step_n\rangle\} = -\langle err_{n-1}, \frac{\alpha}{2\beta}E_n\{\Delta_n(F(\hat{\theta}_{2n-2} + \beta\Delta_n, w_{2n}, 2n) - F(\hat{\theta}_{2n-2}, w_{2n}, 2n))\}\rangle - \langle err_{n-1}, -\frac{\alpha}{2\beta}E_n\{\Delta_n(F(\hat{\theta}_{2n-2} - \beta\Delta_n, w_{2n-1}, 2n-1) - F(\hat{\theta}_{2n-2}, w_{2n-1}, 2n-1))\}\rangle - \langle err_{n-1}, \frac{\alpha}{2\beta}E_n\{\Delta_n(F(\hat{\theta}_{2n-2}, w_{2n}, 2n) - F(\hat{\theta}_{2n-2}, w_{2n-1}, 2n-1))\}\rangle. \tag{7}$$

Рассмотрим разность по знаком $E_n\{\cdot\}$ в первом слагаемом (7). Учитывая разложение для $F(\hat{\theta}_{2n-2} + \beta\Delta_n, w_{2n}, 2n)$ по формуле Тейлора, последовательно выводим

$$E_n\{\Delta_n(F(\hat{\theta}_{2n-2} + \beta\Delta_n, w_{2n}, 2n) - F(\hat{\theta}_{2n-2}, w_{2n}, 2n))\} =$$

$$= E_n\{\Delta_n\langle\nabla F(\hat{\theta}_{2n-2} + \gamma_1\beta\Delta_n, w_{2n}, 2n), \beta\Delta_n\rangle\} =$$

$$= E_n\{\Delta_n\langle\nabla F(\hat{\theta}_{2n-2}, w_{2n}, 2n), \beta\Delta_n\rangle\} +$$

$$+E_n\{\Delta_n\langle\nabla F(\hat{\theta}_{2n-2}, w_{2n}, 2n) - \nabla F(\hat{\theta}_{2n-2} + \gamma_1\beta\Delta_n, w_{2n}, 2n), \beta\Delta_n\rangle\}.$$

при $\gamma_1 \in (0.1)$. В итоге для первого слагаемого в (7), применив (**B**), (**C**), (**E**). получаем

$$-\langle err_{n-1}, \tfrac{\alpha}{2\beta}.E_n\{\Delta_n(F(\hat{\theta}_{2n-2}+\beta\Delta_n.w_{2n},2n)-$$
$$-F(\hat{\theta}_{2n-2},w_{2n},2n))\}\rangle \le \tfrac{\alpha}{2}(-\langle err_{n-1}.\nabla f(\hat{\theta}_{2n-2}.2n)\rangle+$$
$$+B\beta\|err_{n-1}\| \le \tfrac{1}{2}(-\alpha\mu\|err_{n-1}\|^2+\alpha\beta B\|err_{n-1}\|).$$

Аналогичное соотношение получается и для второго слагаемого в (7) Для третьего слагаемого, в силу независимостиь пробного возмущения Δ_n от w_{2n} и w_{2n-1}.получаем:

$$-\langle err_{n-1}.\frac{\alpha}{2\beta}E_n\{\Delta_n(F(\hat{\theta}_{2n-2}.w_{2n},2n)-F(\hat{\theta}_{2n-2}.w_{2n-1},2n-1))\}\rangle =$$

$$=\langle err_{n-1}.\frac{\alpha}{2\beta}E_n\{\Delta_n(F(\hat{\theta}_{2n-2},w_{2n},2n)-F(\hat{\theta}_{2n-2}.w_{2n-1},2n-1))\}=0.$$

В итоге, $-E_n\{\langle err_{n-1},step_n\rangle\} \le -\alpha\mu\|err_{n-1}\|^2+\alpha\beta B\|err_{n-1}\|$.

2. Рассмотрим $E_n\{\langle step_n.dreif_n\rangle\}$. Используя разложение $step_n$ на слагаемые из предыдущею пункта, можно получить

$$E_n\{\langle step_n.dreif_n\rangle\}=\alpha\langle\nabla f(\hat{\theta}_{2n-2}.2n),dreif_n\rangle+$$
$$+A\alpha\beta B\le\alpha A\|err_{n-1}\|+A\alpha\beta B$$

3. Рассмотрим $E_n\|step_n\|^2$. Разложив как в п.1. с использованием свойств (**C**) и (**D**) получаем:

$$E_n\|F(\hat{\theta}_{2n-2}+\beta\Delta_n,w_{2n}.2n)-F(\hat{\theta}_{2n-2}.w_{2n}.2n)\|\le B\beta;$$
$$E_n\|-F(\hat{\theta}_{2n-2}-\beta\Delta_n.w_{2n-2}.2n-2)+F(\hat{\theta}_{2n-2}.w_{2n-1},2n-1)\|\le B\beta;$$
$$E_n\|F(\hat{\theta}_{2n-2},w_{2n},2n)-F(\hat{\theta}_{2n-2},w_{2n-1},2n-1)\|\le C\|err_{n-1}\|+D$$

В итоге имеем:

$$E_n\|step_n\|^2\le\frac{\alpha^2}{4\beta^2}(4B^2\beta^2+4B\beta(C\|err_{n-1}\|+D)+C^2\|err_{n-1}\|^2+$$
$$+2CD\|err_{n-1}\|+D^2+\sigma_v^2)\le\|err_{n-1}\|^2C^2\frac{\alpha^2}{4\beta^2}+$$
$$+\|err_{n-1}\|\frac{\alpha^2}{4\beta^2}(4B\beta C+2CD)+\alpha^2B^2+\frac{\alpha^2}{4\beta}4BD+\frac{D^2\alpha^2}{4\beta^2}+\frac{\alpha^2}{4\beta^2}\sigma_v^2.$$

Суммируя полученные выше оценки, учитывая вид H, выводим

$$E_n\{\|err_n\|^2\}\le\|err_{n-1}\|^2(1-2\alpha\mu+C^2\frac{\alpha^2}{4\beta^2})+$$
$$+\|err_{n-1}\|H+2A^2+2A\alpha\beta B+\alpha^2B^2+\frac{\alpha^2}{4\beta}4BD+\frac{D^2\alpha^2}{4\beta^2}+\frac{\alpha^2}{4\beta^2}\sigma_v^2.$$

В силу неравенства $2aH \le Ha^2\delta + H/\delta,\ \forall\delta > 0$.получаем

$$E_n\{\|err_n\|^2\} \le (1 - 2\alpha\mu + C^2\frac{\alpha^2}{4\beta^2} + \delta H)\|err_{n-1}\|^2 +$$

$$+2A^2 + 2A\alpha\beta B + \alpha^2 B^2 + \frac{\alpha^2}{4\beta}4BD + \frac{D^2\alpha^2}{4\beta^2} + \frac{\alpha^2}{4\beta^2}\sigma_v^2 + \frac{H}{4\delta}.$$

Выберем $0 < \delta < \dfrac{2\alpha\mu - C^2\dfrac{\alpha^2}{4\beta^2}}{H}$. При таком выборе коэффициентов мыполучаем $E_n\{\|err_n\|^2\} \le K\|err_{n-1}\|^2 + L$. Перейдя к безусловному математическому ожиданию выводим (5).

Пример

Простое практическое приложения алгоритма стохастической аппроксимации с пробным одновременным возмущением в условиях нестационарного функционала описанного выше это оценка координат движущейся точки в многомерном пространстве, когда единственное доступное измерение на каждом шаге это расстояние до нее, измеряемое с помехой. Как доказано выше, алгоритм (4) будет сходиться при условии ограниченности нормы дрейфа экстремума.

Численный пример, рассматриваемый в данном разделе, иллюстрирует решение именно этой задачи. Рассмотрим одномерный случай, когда модель дрейфа точки описывается формулой $\theta_n = \theta_{n-1} + \zeta$, где $\zeta \in B(-1,1)$ (ζ принимает значение 1 или -1 с вероятностью 0.5). Тогда будем рассматривать функцию $F(x,w,n) = f(x, n) = (x - \theta_n)^2$, которая определяет квадрат расстояния до точки. Очевидно данная функция удовлетворяет условию теоремы. Измерения на каждом шаге производятся с дополнительным шумом $y_n = f(x_n,n) + v_n$, где $v_n \in (-1,1)$. Помеха v_n генерировалась по закону $v_{2i} = 1 - (i \bmod 3)$ и для четных шагов и $v_{2i-1} = 1 - 3*(i \bmod 7)$ для нечетных. В этом случае параметры функции $A = 1$, $B = 2$, $C = 1$, $D = 1/3$, $\mu = 2$. Тогда $H = 4\alpha\beta + 2\alpha + 2 + \frac{\alpha^2}{2\beta^2}(4\beta + \frac{1}{3})$.

$K = 1 - 4\alpha + \frac{\alpha^2}{4\beta^2} + \delta H$. $L = 2 + 4\alpha\beta + 4\alpha^2 + \frac{8}{3}\frac{\alpha^2}{\beta} + \frac{37}{36}\frac{\alpha^2}{\beta^2} + \frac{H}{4\delta}$.

В эксперименте были выбраны $\alpha = 1/12$ и $\beta = 1/3$. При этом $H \approx 2{,}30$. Выбрав $\delta = 0{,}08$, получаем $K \approx 0{,}86$, $L \approx 9{,}43$ и $\overline{L} \approx 69{,}91$. Точка оптимума дрейфует, как показано на рис. 1 слева. Ошибка оценивания и асимптотическая граница показаны на рис. 1 справа.

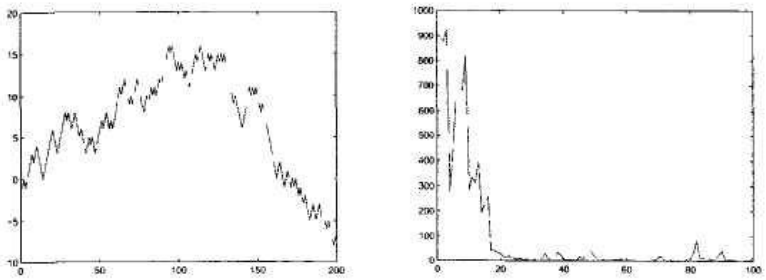


Рис. 1: Экстремум θ_n (слева) и норма ошибки оценивания (справа).

При дальнейших исследованиях следовало бы получить эффективную верхнюю границу для последовательности оценок получаемых при помощи алгоритма. Также интересно было бы усилить данный алгоритм, исиолыуя идеи полиномиальной аппроксимации дрейфа. Это бы существенно расширило условия сходимости, позволяя отказаться от равномерной ограниченности дрейфа и заменить это на полиномиальную ограниченность, которая существенно более слабая. Также следует рассмотреть версию алгоритма, когда последовательность оптимизируемых функций сходится в себе, в этом случае при убывающем шаге предложенный алгоритм будег находить точное решение, в силу того, что отклонение функции от предельной можно рассматривать как внешнюю неопределенность.

10. Стратегия оптимизационного исследования

Задача, к которой можно применить оптимизационные методы, должна включать критерий эффективности, ряд независимых переменных, а также ограничения в виде равенств и неравенств, которые и образуют модель рассматриваемой системы. Описание и построение модели реальной системы - важнейший этап оптимизационного исследования, так как он определяет практическую ценность получаемого решения и возможность его реализации.

10.1. Построение модели

Процесс оптимизации с использованием модели можно рассматривать как метод отыскания оптимального решения для реальной системы без непосредственного экспериментирования с самой системой.

«Прямой путь», ведущий к оптимальному решению, заменяется «обходным», включающим построение и оптимизацию модели, а также преобразование полученных результатов в практически реализуемую форму. При формировании модели следует учитывать только важнейшие характеристики системы. Необходимо также сформулировать логически обоснованные допущения, выбрать форму представления модели, уровень её детализации и метод реализации на ЭВМ. **Ни одну из моделей независимо от степени её детализации и сложности нельзя считать единственно «правильной».** Модели можно упорядочить по степени адекватности описания поведения реальной системы в представляющей интерес области эксплуатации. **Единственным критерием оценки модели может служить лишь достоверность полученных на модели прогнозов поведения реальной системы**.

При разработке модели стремятся к тому, что иногда называют «принципом оптимальной неточности»: **модель должна быть настолько детализирована, насколько это необходимо для целей исследования, для которого её создали.** Существует единственный надёжный способ создания модели с оптимальным уровнем неточности, а именно **метод постепенного совершенствования модели и методов оптимизации**. Начав с самой простой модели, её последовательно доводят до такого уровня, когда точность полученного значения оптимума соответствует точности используемой в модели информации. Для того, чтобы получить результаты в заданные сроки и не проводить постепенного совершенствования модели, обычно подгоняют модель под оптимизационные методы,

наиболее развитые к данному времени или освоенные специалистом, проводящим работу, или же использованные в предыдущем исследовании. При разработке модели следует также учитывать возможности и ограничения оптимизационных программ. Например, нельзя решить задачу НЛП размерности, соответствующей максимальной размерности решаемых задач ЛП.

В **оптимизационных исследованиях** обычно используются модели трёх основных типов: ***1) аналитические модели; 2) модели поверхности отклика; 3) имитационные модели.***

Модель первого типа включает уравнения материального и энергетического баланса, соотношения между проектными техническими характеристиками и уравнения, описывающие физические свойства; они образуют **системы уравнений или неравенств**. Функции в уравнениях должны принимать **вещественные значения**, которые можно вычислить для выбранных значений независимых переменных. Уравнения могут содержать интегральные или дифференциальные операторы, но на практике их лучше **аппроксимировать или заменять квадратурными формулами.** Поскольку такие модели описывают поведение системы на уровне основных технических принципов, они обычно достоверны для более широких условий работы системы, чем модели поверхности отклика.

В **модели поверхности отклика** вся система или входящие в неё части состоят из ***аппроксимирующих уравнений выбранного вида***, коэффициенты которых определяются на основе прямо или косвенно полученной информации о работе системы. Модели такого типа используются в тех случаях, когда **отклик системы непредсказуем или слишком сложен, что делает невозможным создание детализированной модели исходя из технических принципов.** Поскольку переменные взаимозависимы, модели поверхностей отклика обычно надёжны только в ограниченной области значений переменных системы. Их преимуществом является упрощённая структура.

В моделях третьего типа основные уравнения, описывающие поведение системы, группируются в отдельные модули или подпрограммы. Они описывают работу отдельных частей оборудования или реакцию системы на изменение её состояния. Каждый из этих модулей независим от других и содержит внутренние вычислительные процедуры. **Имитационные модели обычно**

используются в тех случаях, когда трудно решать уравнения с неявно заданными переменными, когда от состояния системы зависит выбор алгоритма вычислительной процедуры или соответствующих уравнений, когда в модель приходится вводить случайные возмущения. Модели этого типа обычно сложнее моделей двух описанных выше типов и, как правило, при их использовании нужны значительно большие вычислительные мощности.

Выбор типа модели определяется качеством имеющейся информации о системе, степенью понимания того, что происходит с системой и зависит от сложности самой системы.

10.2. Реализация модели

Модель для оптимизационного исследования можно записать в явном виде, а затем запрограммировать для вычисления значений функций и производных. Модель также можно генерировать с помощью ЭВМ. В случае задач линейного программирования можно генерировать матрицы, вместо того, чтобы вводить их вручную. В конкретных задачах, когда возникают связанные между собой подсистемы регулярных структур различного вида, эффективным является использование генераторов уравнений. При записи всей модели им идентифицируются только подсистемы, входящие в модель и их взаимные связи. Использование генераторов уравнений оправдано при проведении ряда исследований, даёт возможность представить модели в стандартном виде, позволяет сделать удобную документацию и сводит к минимуму ошибки и пропуски при кодировании модели.

В случае моделей поверхности отклика можно непосредственно использовать систему уравнений или её отдельные компоненты для получения информации, на основе которой можно вывести аппроксимирующие уравнения с зависимыми и независимыми переменными. Часто более сложные модели компонент системы используются для того, чтобы автоматически получить модели поверхностей отклика для последующей оптимизации.

Имитационные или аналитические модели можно сразу записывать в виде программ или воспользоваться библиотеками имитационных программ. При построении модели системы можно использовать метод блочного моделирования.

При решении большей части технических прикладных задач используются разработанные самими исследователями аналитические модели или специальные имитационные модели. Автоматическое генерирование аналитических моделей обычно используется только для моделей линейного и (или) частично целочисленного программирования. Модели поверхности отклика чаще всего используются совместно со сложными имитационными моделями, чтобы избежать непосредственной оптимизации имитационных моделей.

После того, как модель построена и выбран способ её представления, следует **подготовить задачу для решения с помощью подходящего оптимизационного алгоритма**. Подготовка задачи к решению включает три этапа:

1) модификация модели с целью преодоления вычислительных трудностей;

2) преобразование модели для повышения эффективности решения;

3) анализ модели с целью нахождения возможных признаков решения задачи.

10.3. Преодоление вычислительных трудностей.

Подобные трудности, приводящие к преждевременному прерыванию счёта, обычно вызываются четырьмя основными причинами: **плохим масштабированием, несоответствием программ для вычисления значений функций и программ для вычислений производных, недифференцируемостью входящих в модель функций, неправильным заданием области определения значений аргументов функций.** При тщательном анализе можно выявить эти ситуации и исключить их путём простой модификации модели.

В результате масштабирования осуществляется переход к относительным значениям величин, используемых в оптимизационной модели. В идеальном случае все переменные модели масштабируются таким образом, чтобы их значения находились в интервале 0.1 - 10. В этом случае векторы направления поиска и векторы возмущений квазиньютоновского метода имеют приемлемые значения.

Масштабирование можно провести путём замены переменных задачи новыми, умноженными на соответствующие коэффициенты. Таким же образом по оценкам ограничений в приближенном решении исследуется чувствительность ограничений к изменениям значений переменных. Масштабирование путём умножения ограничений на соответствующие масштабные коэффициенты позволяет сохранить их значения и значения компонент градиентов функций ограничений в интервале 0.1 - 10.

Повышение эффективности решения

Несоответствие между значениями функций в модели и значениями их производных может оказаться незамеченным, но эта ошибка может увести алгоритм поиска в ложном направлении. Простейший способ проверки соответствия значений функции и градиента состоит в вычислении разностей значений функции и сравнении полученных величин с величинами, определёнными путём вычисления производных на основе аналитического задания градиентов. Для того, чтобы исключить подобные ошибки, во многих случаях вычисляют значения градиентов по разности значений функций. Однако использование представленных в аналитическом виде градиентов позволяет повысить эффективность решения задачи, особенно в том случае, когда предусмотрено сохранение их значений для повторно встречающихся наборов переменных.

Наиболее часто недифференцируемость функций в модели возникает в двух случаях: 1) условные операторы приводят к различным выражениям; 2) работа некоторых блоков модели зависит от значений выбранных переменных или функций, а также минимаксных операторов (min, max). **Минимаксные операторы можно заменить системой неравенств.** Если в модели много условных выражений, целесообразно не применять оптимизационные алгоритмы, в которых используются значения градиентов функций.

Для предотвращения неконтролируемых выходов значений аргументов функций за пределы допустимой области вводятся дополнительные ограничения, и, по возможности, устраняются все операции деления на переменные, чтобы исключить особые точки функций и их производных.

10.4. Анализ модели с целью нахождения возможных признаков решения задачи

Сложность решения нелинейных задач **экспоненциально возрастает с увеличением количества переменных или ограничений в виде равенств или неравенств**. На стадии подготовки задачи к решению целесообразно модифицировать модель с целью **уменьшения количества ограничений, особенно нелинейных, и количества переменных**. Модели можно улучшить с помощью преобразования функций и переменных, исключения лишних ограничений, а также используя метод последовательной подстановки.

Под преобразованием функции понимается любое алгебраическое преобразование функции или объединение данной функции с какой-либо другой. Обычно проводятся преобразования, позволяющие заменить нелинейные ограничения линейными, а равенства - неравенствами. При замене равенства парой неравенств противоположных знаков, реальная возможность по упрощению вычислений возникает только тогда, когда в точке оптимума имеет существенное значение только одно из них, а второе отбрасывается. Преобразование переменных в ряде случаев позволяет повысить эффективность решения задачи, однако может вызвать осложнения, заключающиеся в появлении дополнительных локальных оптимумов, вырождении выпуклости и ослаблении сходимости.

Другим средством упрощения решения является исключение из задачи избыточных ограничений. Избыточным называется ограничение, которое не используется при определении границ допустимой области значений переменных. Хотя избыточные ограничения легко распознать, в общем случае неизвестно ни одной процедуры для их идентификации.

Размерность и число ограничений в виде равенств можно существенно сократить, решая явно или неявно некоторые из них и используя полученные решения для исключения переменных. Процедура сводится к выбору множества независимых переменных и определению такого порядка решения ограничений в виде равенств относительно зависимых переменных, при котором потребуется минимально возможное число итераций. Обычно остаётся ряд ограничений, которые не удаётся непосредственно решить относительно одной или большего числа независимых переменных. В

таком случае возможны два подхода: ***когда необходимо получить значения функций задачи, эти ограничения решаются итеративно относительно зависимых переменных, или же оставшиеся уравнения явно учитываются как ограничения в виде равенств, а остающиеся зависимые переменные считаются независимыми***.

Специфика задачи, оказывающая влияние на процесс решения, может включать: ***выпуклость, неограниченность области допустимых значений, единственность решения, существование допустимого решения.***

Доказательство выпуклости обычно требует громоздких выкладок, однако легко найти элементы задачи, делающие её невыпуклой. ***Если в задаче есть хотя бы одно нелинейное ограничение в виде равенства, то она невыпукла.*** Если таких нет, следует проверить выпуклость нелинейных ограничений в виде неравенств. ***Только убедившись в выпуклости системы ограничений, имеет смысл проверить выпуклость целевой функции.*** Если доказано, что задача является выпуклой, это существенно повышает вероятность существования единственного минимума, а также позволяет применять более широкий класс алгоритмов оптимизации.

Утверждение, что задача ограничена, означает, что все допустимые решения со значениями целевой функции можно заключить в ***конечный гиперкуб***. В технических приложениях всегда стремятся получить конечные оптимальные значения переменных. Случаев неограниченности оптимальных значений переменных можно избежать, введя разумные ограничения сверху и снизу на все переменные задачи. Однако, следует убедиться в необходимости такого шага.

Несмотря на то, что выпуклость гарантирует существование глобального оптимума, она не обеспечивает единственности решения. С другой стороны, если у задачи более одного локального минимума, то она всегда невыпуклая, но одной невыпуклости недостаточно для существования нескольких локальных минимумов. Поэтому необходим анализ задачи для определения возможности существования неединственного решения или нескольких локальных минимумов.

На последнем этапе анализа задачи до начала оптимизационных расчётов необходимо проверить ***наличие допустимых решений***. Независимо от того, необходимо это или нет для выбранного оптимизационного алгоритма, ***всегда целесообразно найти начальное допустимое решение.*** При этом можно пользоваться методом случайного поиска, безусловной минимизацией штрафных функций и последовательной минимизацией невязок ограничений.

10.5. Методы поиска и оценки решений

Методы поиска решений

При проведении оптимизационных расчётов можно использовать ряд различных методов в зависимости от вида модели, её свойств и структуры. Непосредственная оптимизация с помощью подходящего метода НЛП применима во всех случаях, однако для некоторых задач полезно воспользоваться другими приёмами, как, например, методом последовательной оптимизации, когда решается ряд подзадач, или двухэтапным методом, в котором используются промежуточные приближенные модели. В тех случаях, когда предполагается существование множества локальных минимальных решений, следует использовать такой метод, который приводит к глобальному минимуму.

С помощью аналитических моделей, а также моделей поверхности отклика решения получаются либо непосредственно, либо методом последовательной минимизации. При непосредственной оптимизации выясняют, подходит ли структура задачи для специальных оптимизационных методов, или же следует пользоваться общими алгоритмами НЛП. Специальные методы предпочтительнее, особенно если задачу приходится решать много раз. Если же задача решается только один раз, применение общего метода НЛП может оказаться предпочтительнее с точки зрения общей экономии рабочего времени.

Метод последовательной оптимизации заключается в том, что решение задачи получается в результате решения последовательных подзадач с ограничениями. Основная идея метода состоит в том, чтобы найти решение сложной задачи путём разделения переменных на две группы. В одну группу объединяются переменные, значения которых трудно определить, а в другую - переменные, значения которых

сравнительно легко вычислить. Обе подзадачи решаются раздельно, при этом проводятся координирующие вычисления для их связи.

Оптимизация имитационных моделей проводится непосредственно или с помощью различных двухэтапных методов. При непосредственной оптимизации имитационная модель используется как программа для расчётов выпуска продукции и вычисления значений ограничений. Если выполняется условие, что выходные параметры имитационной модели непрерывно дифференцируемы по входным параметрам, то применим любой градиентный алгоритм безусловной и условной оптимизации. В противном случае нужно использовать прямые методы, такие как метод комплексов или метод случайного поиска. При использовании прямых методов оптимизации в имитационных моделях часто встречаются три случая, которые могут затруднить проведение вычислений и привести к повторению итераций:

1) наличие неявных ограничений для зависимых (внутренних) переменных;

2) наличие подразумеваемых ограничений, которые приняты при построении модели;

3) наличие вычислительных процедур, которые используются при имитации.

Если в результате чего-либо из вышеперечисленного оптимизационная задача в окончательном виде оказывается слишком сложной для прямых методов оптимизации, применяют различные виды двухэтапных методов. При этом с помощью имитационной модели получают модель поверхности отклика в независимых переменных, для которой используется подходящий оптимизационный метод. Процесс решения повторяется, причём каждый раз используется поверхность отклика, модифицированная в соответствии с полученным предшествующим оптимизационным решением, до тех пор, пока разность между двумя последовательными решениями не станет достаточно малой. Двухэтапные методы отличаются прежде всего по виду используемых аппроксимирующих функций, по уровню детализации создаваемой модели поверхности отклика и по применяемым оптимизационным методам.

Для надёжной оптимизации моделей, которые могут иметь несколько локальных минимумов, следует воспользоваться несколькими методами решения задачи, чтобы найти глобальный минимум. **Известные методы поиска глобального минимума делятся на детерминированные и стохастические, которые в свою очередь могут быть эвристическими или строго обоснованными.** Простейший метод состоит в проведении ряда оптимизационных расчётов при различных начальных условиях. Иногда этот метод называется методом с несколькими начальными точками. В нём начальные точки выбираются из определённой решётки или же генерируются случайным образом. Оба этих метода эвристические и не дают полной уверенности в результате. Теоретически обоснованные методы глобальной оптимизации разработаны только для задач со специальной структурой.

Оценка решения

Самая важная часть оптимизационного исследования заключается в обосновании правильности полученного решения и анализе его чувствительности. Наиболее важным является не само решение, а ***информация о состоянии системы в окрестности решения***, что позволяет глубже понять её основные свойства. Важнейшими результатами исследования являются ответы на такие вопросы, как, например: Какие ограничения активны в полученном решении? Что составляет основную часть стоимости? Какова чувствительность решения к изменениям значений параметров? Активные ограничения указывают на ограниченные возможности системы или на то, что из-за проектных соображений систему усовершенствовать нельзя. По величине стоимости находят тот блок системы, параметры которого должны быть улучшены. Чувствительность решения к изменению значений параметров указывает на то, какие оценки параметров следует улучшить для того, чтобы безошибочно найти оптимальное решение.

Считается, что **решение, полученное в результате оптимизационных расчётов, обосновано, если ему соответствует некоторое реализуемое состояние рассматриваемой системы и оно является её оптимумом**. Поскольку вся информация имеет ограниченную точность, следует **проверять, не выходит ли полученное решение за границы достоверности модели**. Если это

обнаружено, в модель необходимо ввести дополнительные ограничения и повторить оптимизационные расчёты.

После того, как показано, что решение реализуемо, следует установить оптимальность полученного решения на качественном уровне, оценивая его техническую взаимосвязь с совокупностью полученных параметров системы. В противном случае оптимальность решения принимается как результат применения математики и вычислительной техники.

Реализующий эту процедуру подход подразумевает использование упрощённых вспомогательных моделей с целью выявления основных причин, влияющих на решение. Общая методология такова:

1) упростить модель так, чтобы можно было использовать простые алгебраические методы;

2) получить из вспомогательной модели оптимальное решение как функцию главных переменных моделей;

3) с помощью вспомогательной модели построить ряд прогнозов и проверить их на полной модели;

4) если оптимизационные расчёты подтверждают тренды, полученные из вспомогательной модели, то успех в объяснении свойств модели достигнут.

Всё это способствует уменьшению разрыва между оптимумом системы и оптимумом модели.

Целями же второго этапа оценки результатов решения, анализа чувствительности, являются следующие:

1. Отыскание параметров, оказывающих наибольшее влияние на оптимальное решение. Если такие параметры существуют, то, возможно, следует рассмотреть вопрос о коррекции соответствующих свойств системы.

2. Уточнение данных о дополнениях или модификации системы с целью улучшения показателей её работы.

3. Определения влияния на систему вариаций неточно заданных параметров. Анализ чувствительности показывает, стоит ли тратить средства для определения более точных значений некоторых параметров.

4. Выяснение возможной реакции системы на неуправляемые внешние воздействия.

Анализ чувствительности проводится двумя способами: с помощью множителей Лагранжа или методом параметрического исследования. В случае линейного программирования легко получить информацию о чувствительности системы по коэффициентам целевой функции, не проводя повторного расчёта оптимального решения. В других случаях применяются указанные выше методы. Множители Лагранжа дают полезную информацию о чувствительности целевой функции к различным ограничениям, но они не характеризуют её чувствительность к изменениям отдельных параметров. В связи с этим желательно провести серию других расчётов чувствительности модели, в которых изменяют некоторые параметры.

Проведение оптимизационного исследования нельзя свести только к расчётам по искусно составленной программе. В него входит тщательное изучение многих аспектов самой прикладной задачи, выбранной для неё модели и алгоритмов, используемых для вычислений.

Алгоритмы безусловной минимизации функций многих переменных можно сравнивать и исследовать как с теоретической, так и с экспериментальной точек зрения.

Первый подход может быть реализован полностью только для весьма ограниченного класса задач, например, для сильно выпуклых квадратичных функций. При этом возможен широкий спектр результатов от получения бесконечной минимизирующей последовательности в методе циклического покоординатного спуска до сходимости не более чем за n итераций в методе сопряженных направлений.

Мощным инструментом теоретического исследования алгоритмов являются теоремы о сходимости методов. Однако, как правило, формулировки таких теорем абстрактны, при их доказательстве используется аппарат современного функционального анализа. Кроме того, зачастую непросто установить связь полученных математических

результатов с практикой вычислений. Дело в том, что условия теорем труднопроверяемы в конкретных задачах, сам факт сходимости мало что дает, а оценки скорости сходимости неточны и неэффективны. При реализации алгоритмов также возникает много дополнительных обстоятельств, строгий учет которых невозможен (ошибки округления, приближенное решение различных вспомогательных задач и т.д.) и которые могут сильно повлиять на ход процесса.

Поэтому на практике часто сравнение алгоритмов проводят с помощью вычислительных экспериментов при решении так называемых специальных тестовых задач. Эти задачи могут быть как с малым, так и с большим числом переменных, иметь различный вид нелинейности. Они могут быть составлены специально и возникать из практических приложений, например задача минимизации суммы квадратов, решение систем нелинейных уравнений и т.п.

Приложения

Приложение 1

Метод решения задача Коши

1.1 Постановка задачи

При решении многих задач естествознания в качестве математической модели используется *задача Коши* для обыкновенных дифференциальных уравнений. Например задачи динамики системы взаимодействующих тел (в модели материальных точек), задачи химической кинетики, электрических цепей. Ряд важных уравнений в частных производных в случаях, допускающих разделение переменных, приводит к задачам для обыкновенных дифференциальных уравнений — это, как правило, краевые задачи (задачи о собственных колебаниях упругих балок и пластин, определения спектра собственных значений энергии частицы в сферически-симметричных полях и многие другие).

Мы ограничимся рассмотрением лишь задачи Коши. Полученная в общем случае задача для ОДУ (обыкновенных дифференциальных уравнений) с помощью замены переменных сводится *к нормальной системе* дифференциальных уравнений. *Задача КОШИ* для последней формулируется так:

Определить дифференцируемую функцию $u(x)$, для которой

$$\frac{du}{dx} = f(x, u) \tag{1}$$

и выполнено начальное условие

$$u(x_0) = u_0. \tag{2}$$

Здесь x_0, u_0 - заданные величины: $u = \{ u_1, u_2, ..., u_N\}$ - искомая вектор-функпия; $f(x,u) = \{f_1(x,\vec{u}), \ldots, f_N(x,\vec{u})\}$ — вектор правых частей. Относительно задачи (1-2) будем предполагать выполненными достаточные условия существования на отрезке $|x - x_0| < a$ решения $u(x)$ задачи (1)-(2).

Эйлеру принадлежит идея и рассмотрение простейшего численного метода, основанного на возможности получить разложение по формуле Тейлора для искомого решения $u(x)$ в окрестности точки x_n

$$u_{n+1} = u(x_{n+1}) = u_n + h_n u'_n + \frac{1}{2}h_n^2 u''_n + \cdots + \frac{h_n^s}{(s)!}u_n^{(s)} + O(h_n^{(s+1)}), \tag{3}$$

где $h_n = x_{n+1} - x_n$. При этом необходимые производные функции $u(x)$ можно найти дифференцируя в силу уравнения (1) функцию $f(x,u(x))$ нужное число раз

$$u' = f(x,u); \quad u'' = \frac{d}{dx}f(x,u(x)) = f_x + f_u \cdot \underbrace{u_x}_{\equiv f(x,u)} = f_x + f f_u, \quad \text{и т.д.} \tag{4}$$

Однако использовать разложение (3) с большим числом членов невыгодно: и из-за громоздкости формул (4), и из-за того, что, как правило, правая часть в (1) известна лишь приближённо и её явное численное дифференцирование нежелательно.

1.2 Метод Рунге-Кутта

Идея Рунге метода Рунге-Кутта состоит в том, чтобы используя метод неопределённых коэффициентов аппроксимировать с тем же порядком точности $O(h^s_n)$ многочлен Тейлора в формуле (3). Представим приращение функции $u(x)$ в точке x_n в виде

$$\Delta u(x_n) = u(x_{n+1}) - u(x_n) = h_n \left(\underbrace{u'_n + \frac{1}{2} h_n u''_n + \cdots + \frac{h_n^{s-1}}{(s)!} u_n^{(s)}}_{P_{s-1}(h_n)} + O(h_n^{(s)}) \right).$$

Обозначим текущий шаг $h_n \equiv h$. Речь идёт об аппроксимации многочлена

$$P_{s-1}(h) = u'_n + \frac{1}{2} h u''_n + \cdots + \frac{h^{s-1}}{(s)!} u_n^{(s)}$$

с порядком $O(h^s)$. Ограничимся рассмотрением простейшего случая $s=2$. Тогда у многочлена первого порядка

$$P_1(h) = u'_n + \frac{h}{2} u''_n = f(x_n, u_n) + \frac{h}{2} \frac{d}{dx} f(x, u)\big|_{(x_n, u_n)}$$

необходимо со вторым порядком аппроксимировать производную u''_n.

Пусть $y(x)$ - приближенная функция, дающая такую аппроксимацию. Для аппроксимации производной df/dx мы используем разностное отношение $[f(\tilde{x}, \tilde{y}) - f(x, y)]/\Delta x$ с неопределенными пока x, y. В таком случае приращение функции y имеет вид

$$\Delta y_n = y_{n+1} - y_n = h \{ \beta f(x_n, y_n) + \alpha f(x_n + \gamma h, y_n + \delta h) \}.$$

Здесь α, β, γ и δ — параметры, значения которых нужно определить. Разложим полученное приращение Δy_n в ряд по степеням h, получим

$$y_{n+1} = y_n + h(\alpha + \beta) f(x_n, y_n) + \alpha h^2 (\gamma f_x + \delta f_u)\big|_{(x_n, y_n)} + O(h^3). \quad (*)$$

Выберем параметры α, β, γ и δ так, чтобы разложение для функции y с тем же порядком аппроксимировало разложение истинного решения u. Для этого приравнивая коэффициешы в главных порядках по h полученной формулы (*) и формулы (3), найдём

$$\alpha + \beta = 1, \quad \alpha\gamma = \frac{1}{2}, \quad \alpha\beta = \frac{1}{2} f(x_n, y_n).$$

Выражая все параметры через α, получим однопараметрическое семейство двучленных схем Рунге-Кутта второго порядка точности

$$y_{n+1} = y_n + h \left[(1 - \alpha) f(x_n, y_n) + \alpha f\left(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha} f_n \right) \right], \quad (5)$$

где $0 < \alpha \leq 1$.

Замечания:

1) Выбрать параметр α так, чтобы схема (5) давала бы аппроксимацию третьего порядка невозможной.

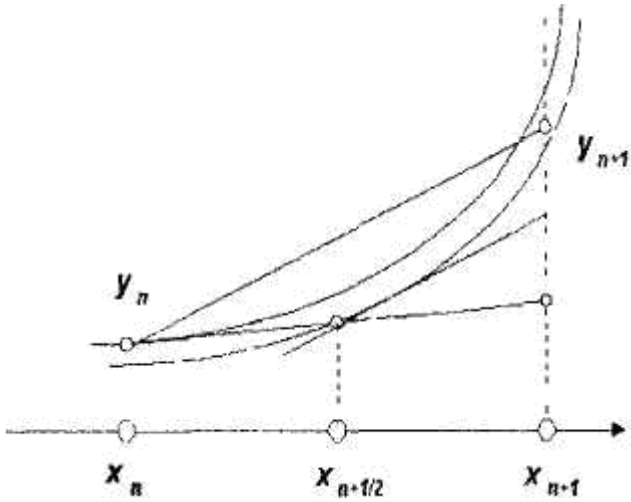
2) Приведем без доказательства теорему. *Если f(x,u) непрерывна и ограничена вместе со своими вторыми производными, то решение, полученное по схеме* (5), *равномерно сходится к точному решению с погрешностью O*(max h^2_n), *т.е. двучленная схема Рунге-Кутта имеет второй порядок точности.*

3) Формула (5) используется на практике обычно либо при α = 1, либо при α = 1/2. При α = 1 схема имеет особенно простой вид

$$y_{n+1} = y_n + \frac{h}{2} f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}h f_n\right). \quad (6)$$

Поясним её смысл. Сначала, вычислив наклон интегральной кривой уравнения (1) $f_n = f(x_n,y_n)$, делаем половинный шаг но схеме ломанных, т.е. по касательной данного наклона, и находим

$$y_{n+1/2} = y_n + \frac{1}{2}h f_n.$$



Затем в найденной точке определяем наклон интегральной кривой

$$y'_{n+1/2} = f(x_{n+1/2}, y_{n+1/2}).$$

По этому наклону определяем приращение функции на целом шаге

$$y_{n+1} = y_n + h y'_{n+1/2}.$$

Схемы подобною типа называют "предиктор-корректор".

Задача. *Дать аналогичную интерпретацию случаю схемы с* $\alpha = 1/2$.

Метод Рунге-Кутта позволяет строить схемы различною порядка гочиосли. При аппроксимации многочлена Тейлора второго порядка

$$P_2(h) = u'_n + \frac{h}{2}u''_n + \frac{h^2}{3!}u'''_n$$

с точностью $O(h^3)$ получают наиболее употребительную схему четвёртого порядка точности (точнее семейство четырёхчленных схем указанного порядка точности)

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = f(x_n, y_n), \quad k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \tag{7}$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \quad k_4 = f(x_n + h, y_n + hk_3).$$

Схемы Рунге-Кутта обладают важными достоинствами: все они имеют хорошую точность; они являются явными; допускают расчет с переменным шагом; легко обобщаются на случай систем дифференциальных уравнений. Имеено эти свойства особенно ценны при расчетах на ЭВМ.

Рекомендации:

1) Если правая часть дифференциального уравнения (1) ограничена, вместе со своими производными до четвёртого порядка, то схема (7) дает хорошие результаты благодаря малому коэффициенту в остаточном члене и быстрому возрастанию точности схемы при уменьшении шага. Если же указанных производных у правой части нет, то не худшую точность имеют схемы и меньшего порядка точности (5).

2) Шаг сетки при расчетах следуем выбирать настолько малым, чмобы обеспечить требуемую точность расчета. Других, ограничительных условий на шаг схемы в методе Рунге-Купа нет.

3) Выражения остаточных членов для формул Рунге-Купа достаточно громоздки, поэтому трудно получить априорную оценку точности метода, однако, проводя расчеты на сгущающихся сетках, можно дать апостериорную оценку точности по методу Рунге.

Приложение 2

ЭЛЕМЕНТЫ ТЕОРИИ РАЗНОСТНЫХ СХЕМ

1. Метод конечных разностей в прикладных задачах

1.1 Общая постановка задачи

Универсальным методом приближённого решения, применимым для широкого круга задач математической физики, является метод конечных разностей. Как правило задачи матаматической физики представляют собой системы нелинейных уравнений в частных производных, рассматриваемых в некоторой *t*-цилиндрической обласхи *D*:

$$D = \{(x, y, z; t) : (x, y, z) \in G, t \in [t_0, T]\} = \bar{G} \times [t_0, T].$$

При этом естественным образом выделяется "эволюционный" характер переменной *t.* Решение интересующей нас задачи подчинено в *D* дополнительным требованиям:

1) условия при $t = t_0$ (на гиперплоскости $t = t_0$) называются *начальными условиями;*

2) условия на границе $\partial D \equiv \gamma$ области *D* — *краевыми* или *граничными* условиями.

Задача с *начальными условиями* - задача в неограниченной области *D* называется задачей Коши; в отличии от *краевой* или *смешанной краевой* задачи.

Удобна общая постановка задачи, не связанная с выделением одной из переменных. Пусть $(x_1, \ldots, x_p) \equiv x \in D : \partial D = \Gamma$. Тогда для интересующей нас функции $u(x)$ имеем задачу:

$$\begin{aligned} A[u(x)] &= f(x), \quad x \in D \\ R[u(x)] &= \mu(x), \quad x \in \Gamma, \end{aligned} \tag{1-2}$$

где A и R дифференциальные операторы задачи и краевых условий. Относительно задачи (1-2) будем предполагать что она поставлена корректно, то есть операторы A и R; область D и её границы Γ таковы, что при выборе соответствующих классов функций и правых частей в уравнениях (1) и (2) решение существует, единственно и непрерывно зависит от начальных данных (и коэффициентов уравнения, то есть соответствующих операторов задачи (1-2)

С точки зрения приложений нас, естественно, будет интересовать случай, когда оператор A - линейный дифференциальный оператор в частных производных второго порядка (согласно обычной классификации уравнений это - эллиптическое, гиперболическое или параболическое уравнение). Хотя, конечно, задача (1-2) может быть и другой природы.

1.2. Разностная схема

Введём в области $D = D + \Gamma$ сетку $\Omega_h = x_{i \in I}$ состоящую из множества внутренних узлов ω_h и множества граничных узлов Γ_h:

$$\Omega_h = \{x_i\}_I = \omega_h \bigcup \Gamma_h.$$

Мы пока абстрагируемся от способа конкретного получения сетки Ω_h, в области D; смысла параметра, "h" в соответствующих сетках, контролирующего как пространственные, так и временные размеры сетки; особенностей получения сетки Γ_h на границе области Ω_h; оставим эти вопросы до рассмотрения конкретных задач.

Далее, рассмотрим сеточные функции $y(x) \equiv y_h(x)$, $x \in \Omega_h$ дискретного переменного $\{x_i\}$ и с их помощью построим приближенное решение задачи (1-2). Для этого относительно $y_h(x)$ сформулируем "разносную задачу", обычно "заменяя" операторы исходной задачи A и R их сеточными аналогами A_h и R_h. Тогда на сеточном шаблоне $\Omega_h = \omega_h \bigcup \Gamma_h$ имеем

$$\begin{aligned} A_h y_h(x) &= \varphi_h(x), \quad x \in \omega_h \\ R_h y_h(x) &= \chi_h(x), \quad x \in \Gamma_h, \end{aligned} \tag{3-4}$$

Задачу (3)-(4) назовём *разностной схемой* для задачи (1)-(2). Обычно это алгебраическая система уравнений относительно $y_i(x) \equiv y_h(x_i)$.

При переходе от исходной задачи (1)-(2) к её разностному аналогу (3)-(4) особенно важны три группы вопросов:

- существование, единственность и алгоритм построения разностного решения y_h;
- при каких условиях разностное решение $y_h(x_i)$ стремится к точному решению $u(x)$ и какова при этом скорость сходимости;
- из каких соображений и как конкретно выбирать сетку Ω_h и строить разносшую схему: A_h, R_h и φ_h, χ_h в задаче (3)-(4).

2. Основные понятия и теоремы теории разностных схем

2.1. Невязка разностной схемы.

При построении разностного уравнения задачи

$$A[u] = f \quad \Rightarrow \quad A_h y = \varphi_h$$

мы получили задачу, которой точное решение $u(x)$, как правило, не удовлетворяет (мы подразумеваем простейшую схему проектирования $u(x)$ на сетку Ω_h $\{u(x_i)\}$). Сеточную функцию

$$\psi_h = \varphi_h - A_h u$$

называют *невязкой* сеточною уравнения (3). Её удобно представить на решении $u(x)$ в виде

$$\psi_h = (Au - f)_h - (A_h u - \varphi_h) \quad \text{на } \omega_h. \qquad (5)$$

Аналогично определяются невязки граничных условий

$$\eta_h(x) = (Ru - \mu)_h - (R_h u - \chi_h) \quad \text{на } \Gamma_h. \qquad (5')$$

Как правило невязки $\psi_h(x)$ и $\eta_h(x)$ оценивают по параметру h через разложение в ряд Тейлора в предположении достаточной гладкости соответствующего решения $u(x)$ для получения представления невязки с остаточным членом вида $O(h^n)$.

2.2. Аппроксимация разностной схемы

Разностная схема (3)-(4) *аппроксимирует задачу* (1)-(2)*, если имеет место:*

$$\| \psi_h(x) \|_{\varphi_h} \to 0, \quad \| \eta_h(x) \|_{\chi_h} \to 0 \quad \text{при } h \to 0 \qquad (6)$$

То есть соответствующие невязки стремяться к нулю при $h \to 0$. *Аппроксимация задачи* (1)-(2) *имеет порядок k, если*

$$\| \psi_h(x) \|_{\varphi_h} = O(h^k); \quad \| \eta_h(x) \|_{\chi_h} = O(h^k), \quad h \to 0. \qquad (6')$$

В этих определениях нормы вычисляются для сеточных функций на ω_h, и Γ_h, но в своих функциональных пространствах (соответствующих правых частей). Вопрос о выборе норм отложим до рассмотрения частных задач. Обычно это сеточные аналоги чебышевской нормы в C или гильбертовой нормы в L_2.

Замечания:

Само решение задачи (1)-(2), как правило, неизвестно и использовать его для получения невязок ψ_h и η_h затруднительно. Поэтому берут достаточно широкий класс функций ν и требуют аппроксимации порядка k задачи (1)- (2) $\forall v \in \nu$, т.е.

$$\| (Av - f)_h - (A_h v - \varphi_h) \|_{\varphi_h} = O(h^k), \quad h \to 0.$$

При этом на решении $v \equiv u(x)$ задачи (1)-(2) аппроксимация будет не хуже, чем порядка k (а может быть и лучше).

Как правило схема (3)-(4) по различным переменным имеет различные порядки аппроксимации, например, невязка уравнения

$$\| \psi_h \|_{\varphi_h} = O(\tau^p + h^k). \quad \text{при } \tau \to 0, \quad h \to 0.$$

Такая аппроксимация называется *абсолютной* в отличии от *условной* аппроксимации в случае, когда, например

$$\| \psi_h \|_{\varphi_h} = O(\tau^p + h^k + \frac{\tau^r}{h^s}), \quad \tau \to 0, \quad h \to 0, \quad \frac{\tau^r}{h^s} \to 0.$$

При *условной аппроксимации* разностное уравнение может аппроксимировать различные дифференциальньге задачи.

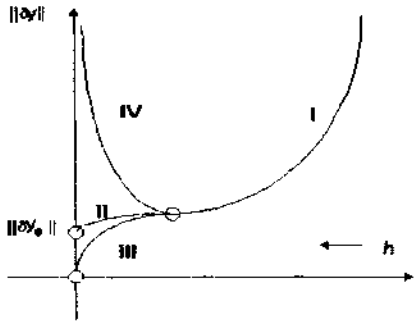
2.3. Устойчивость разностной схемы

Отсутсвие устойчивости разностной схемы характеризуется тем, что малые ошибки, допущенные на каком-либо этапе вычисления, в дальнейшем сильно возрастают и делают непригодным результат

расчёта (или вообще невозможным сам расчет). Обычно устойчивость разностной схемы оценивают по погрешности входных данных, поскольку погрешность аппроксимации, в силу определения (6), при $h \to 0$ стремится к нулю. Выделим в структуре погрешности эти слагаемые:

$$\delta y_h = \delta y_h^{in} + \delta y_h^{app.}.$$

Типичный график зависимости погрешности сеточного решения от величины шага таков:



I. При уменьшении шага сначала погрешность всех схем убывает, так как существенно уменьшается погрешность аппроксимации.

II. Для устойчивых схем погрешность сеточного решения будет стремиться к конечной величине, связанной с ошибкой входных данных. Если при $h \to 0$ ошибка входных данных исчезает, то - это случай III. То есть устойчивая схема в этом случае позволяет получить сколь угодно высокую точность расчёта.

Если же схема неустойчива (IV), то при $h \to 0$ погрешность $||\delta y_h||$ возрастает (ибо растёт объём неустойчивых вычислений). Погрешность $||\delta y_h||$ будет иметь ненулевой минимум и уже невозможно получить сколь угодно высокую точность расчсета.

Как правило погрешности *входных данных* и *аппроксимации* имеют степенной характер зависимости от $h \Rightarrow h^{\alpha}$, а неустойчивость приводит к возрастанию погрешности решения по экспоненциальному закону $\sim b^{a/h1}$ и при $h \to 0$ расчёт теряет смысл. Напомним

Разностная ехгма (3-4) устойчива по входным данным φ *и* χ, *если решение разностной схимы непрерывно зависит от входных данных и эта зависимость равномерна относительно шага сетки* h, *то есть* $\forall\, \varepsilon > 0\ \exists\, \delta\,(\varepsilon) > 0$ *(*δ *не зависит от* h*) такое, что*

$$\begin{array}{l}\forall \chi_1, \chi_2: \\ \forall \varphi_1, \varphi_2:\end{array} \quad \begin{array}{l}\| \chi_1 - \chi_2 \| < \delta(\varepsilon) \\ \| \varphi_1 - \varphi_2 \| < \delta(\varepsilon)\end{array} \quad \Rightarrow \quad \| y_1(x) - y_2(x) \|_{y_h} < \varepsilon. \quad (7)$$

Для линейных схем разностное решение линейно зависит от входных данных (в силу линейности обратного оператора) и $\delta(\varepsilon) = C\varepsilon$. Тогда

$$\|y_1 - y_2\| \le C_1 \|\varphi_1 - \varphi_2\|_{\varphi_h} + C_2 \|\chi_1 - \chi_2\|_{\chi_h}. \quad (7')$$

Замечания:

На устойчивость разностной схемы влияет не только аппроксимация уравнений (1) (то есть оператора *A),* но, и особенно, краевых условий (2).

Если переменных в задаче несколько, то рассматривают безусловную и условную устойчивость.

Входное значение $\chi_h(x)$ на гиперплоскости $t = t_0$ выделяют особо, и соответствующая устойчивость называется устойчивостью по начальным условиям. Тут важна особая роль *t.* Мы ограничимся рассмотрением разностных схем, в которых сеточная функция рассматривается на двух временных слоях t_m; t_{m+1}, т.е. $y \equiv y_h(x; t_m)$ и $\hat{y} \equiv y_h(x; t_{m+1})$. Общий вид такой схемы:

$$B_h \frac{\hat{y} - y}{\tau} + A_h y = \varphi_h.$$

Для такой схемы решение смешанной задачи Коши (с краевыми условиями) на некотором слое t^* можно рассматривать как начальное условие для всех последующих слоев по *t.*

Определение: *Двуслойная схема называется равномерно устойчивой по начальным данным, если при постановке начальных данных на любом слое* t^*, $(t_0 \le t^* < t < T)$ *она по ним устойчива, причём эта устойчивость равномерна по* t^*.

Для линейных разностных схем это означает, что $\exists C > 0$ не зависящее t^* и h и

$$\| y_1(t) - y_2(t) \|_{y_h} \le C \| y_1(t^*) - y_2(t^*) \|, \quad t_0 \le t^* < t < T \quad (7'')$$

где $y_1(x; t)$, $y_2(x; t)$ — решение разностной задачи с одинаковой правой частью $A_h y = \varphi_h$, но различными начальными данными $\chi_{1,2}|_{t^*}$.

Из равномерной устойчивости (7") следует (7') (но не наоборот).

Теорема 1. (достаточный признак равномерной устойчивости):

Пусть $y_1(x; t)$ *и* $y_2(x; t)$ *решения разностной задачи* $A_h y = \varphi_h$, *с одинаковой правой частью, отвечающие различным начальным*

условиям $\chi_{1,2}|_{t^*=t0}$. *Тогда для равномерной устойчивости* $\{A_h; R_h\}$ *по начальным данным достаточно, чтобы для всех слоев по t имело место*

$$\| \hat{y}_1 - \hat{y}_2 \|_{y_h} \le (1 + C\tau) \| y_1 - y_2 \|_{y_h}, \quad C \ge 0 \qquad (8)$$

Доказательство: Если на некотором слое t_* в решении содержится ошибка δy, то при переходе на следующий слой она возрастает не больше чем в $(1 + C\tau) \le e^{C\tau}$ раз. При достижении слоя T за $\dfrac{T - t^*}{\tau}$ шатов ошибка возрастает не более, чем в $e^{C(T-t^*)}$ раз, то есть не более чем в $e^{C(T-t_0)}$ раз. Следовательно

$$\| \delta y \| \le A \| \delta y(t_0) \| .$$

Эта оценка равномерна по t^* и h.

Фактический рост погрешности не более чем в $(1 + C\tau)^{\frac{t-t_0}{\tau}}$ раз.

Теорема 2. (признак устойчивосги двуслойной разностной схемы но правой части):

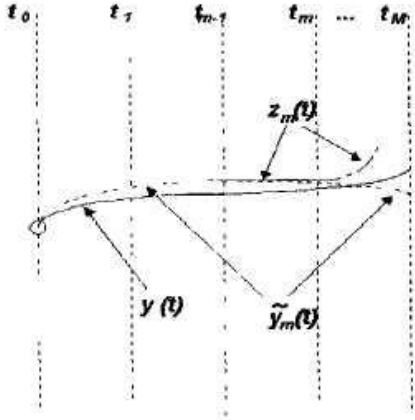
Пусть двуслойная разностная схема $A_h y = \varphi_h$ *равномерно устойчива по начальным данным и такова, что если два её решения* $A_h y_{1,2} = \varphi_{1,2}$ *на некотором слое* t_m *равны* $y_1(x;t_m) = y_2(x;t_m)$, *то на следующем слое* t_{m+1} *выплнено соотношении*

$$\| \hat{y}_1 - \hat{y}_2 \| \le C\tau \| \varphi_1 - \varphi_2 \|, \quad C > 0,$$

C - const (не зависит от h), в таком случае разностная схема устойчива по правой части φ_h .

Доказательство: Итак, пусть возмущение связанно только с правой частью φ. Тогда пусть $y(x;t)$ - решение невозмущённой разностной задачи $A_h y = \varphi$; $\tilde{y}(x; t)$ — решение возмущённой разностной задачи $Au\ \tilde{y} = \tilde{\varphi}$, причём $y(t_0) = \tilde{y}\ (t_0)$ (ибо нас интересует только возмущение правой части).

Введём в рассмотрение последовательность сеточных функций $\{z_m(x;t)\}_{m=i,2,...,}$ определенных при $t \ge t_{m-1}$ из условий:



$$\begin{cases} z_m(t_{m-1}) = z_{m-1}(t_{m-1}) \\ A_h z_m = \begin{cases} \tilde{\varphi} & , \quad t_{m-1} < t \le t_m \\ \varphi & , \quad t > t_m \end{cases} \\ z_1(t_0) = y(t_0) = \tilde{y}(t_0) = z_0(t_0) \end{cases}$$

На каждом из слоев $t \in [t_{m-1}, t_m]$ решение возмущённой задачи $\tilde{y}(t)$ совпадает с соогвегсгвующей функцией $z_m(t)$ поскольку в точку t_{m-1} начальное условие принесено функцией z_{m-1}, удовлетворяющей возмущенному уравнению на соответствующем отрезке t. Аналогично на предыдущем слое и так далее, пока мы не попадём в начальную по t точку. В точке $t = t_{m-1}$ и $\tilde{y}$ и z_{m-1} имеют то же начальное» условие и на интервале (t_{m-1}, t_m) удовлетворяют возмущенной задаче $A_h(\bullet) = \tilde{\varphi}$.

Далее, при $t \in (t_m, t_{m+1})$, функции $z_{m+1}(t)$ и $z_m(t)$ совпадают в точке t_m и удовлетворяют различным уравнениям. Тогда:

1) $\| z_{m+1}(t_{m+1}) - z_m(t_{m+1}) \| \le C\tau \| \varphi - \hat{\varphi} \|_{\varphi}$.

2) В силу равномерной устойчивости нашей задачи по начальным данным при $t \ge t_{m+1}$ функции $z_{m+1}(t)$ и $z_m(t)$ удовлетворяют одному уравнению по разностным начальным условиям. В таком случае на последнем временном слое t_M получим:

$$\| z_{m+1}(t_M) - z_m(t_M) \| \le C_2 \| z_{m+1}(t_{m+1}) - z_m(t_{m+1}) \| \le C_2 C\tau \| \varphi - \hat{\varphi} \| .$$

Откуда:

$$\| z_M(t_M) - z_0(t_M) \| \le \| z_M - z_{M-1} \| + \| z_{M-1} - z_{M-2} \| + \ldots + \| z_1 - z_0 \| \le$$

$$\le MC_2C\tau \| \varphi - \tilde{\varphi} \| = A(T - t_0) \| \varphi - \tilde{\varphi} \| .$$

Таким образом, имеет место устойчивость разностной схемы по правым частям.

Замечание: Сформулируем без доказательства достаточные условия устойчивости двуслойной разностной схемы

$$B\frac{\hat{y} - y}{\tau} + Ay = \varphi.$$

ЕСЛИ A и B > 0, *при мм* $B \ge \frac{\tau A}{2}$ >0, *то*

$$\| \hat{y} \|_A \le \| y \|_A,$$

то есть схема устойчива в А-энергетической норме по начальным данным.

2.4. Сходимость разностной схемы

Решая сеточную задачу (3)-(4) нас естественно интересует близость сеточного решения *y(x)* к решению *u(x)* задачи (1)-(2). *Разностное решение y(x) сходится к решению u(x), если*

$$\| y(x) - u(x) \|_h \to 0 \quad \text{при } h \to 0. \qquad (10)$$

Разностное решение имеет порядок точности k, если

$$\| y(x) - u(x) \| = O(h^k), \quad h \to 0. \qquad (10')$$

(или обладает сходимостью порядка k).

Напомним ещё раз, что мы рассматриваем лишь корректные разностные схемы (3)-(4), то есть решение разностной схемы существует и единственно при любых входных данных φ и χ из заданных классов функций и схема устойчива по входным данным (её решение непрерывно от них зависит).

Теорема 3: *Если решение задачи (1)-(2) и* [f,μ] *существует, разностная схема (3)-(4) корректна и аппроксимирует задачу (1)-(2), то разностное решение y[(*φ, χ*] сходится к точному:*

$$\lim_{h \to 0} \| y_h - u \| = 0.$$

("Аппроксимация + Устойчивость $\Rightarrow$ *Сходимость").*

Доказательство: Запишем невязку разностной схемы (3) (4).

$$\begin{array}{l}\psi_h = (Au - f)_h - (A_h u - \varphi_h) = \varphi_h - A_h u \\ \eta_h = (Ru - \mu) - (R_h u - \chi_h) = \chi_h - R_h u\end{array} \quad \Leftrightarrow \quad \begin{array}{l}A_h u = \varphi_h - \psi_h \\ R_h u = \chi_h - \eta_h.\end{array} \quad (*)$$

Функция *u(x)* удовлетворяет задаче *(*)* — возмущённой задаче (3)-(1). Так как схема устойчива, то $\forall\ \varepsilon > 0\ \exists\ \delta(\varepsilon) > 0$

$\| \psi_h \|_{\varphi_h} < \delta(\varepsilon), \quad \|\eta_h\|_{\chi_h} < \delta(\varepsilon) \quad \Rightarrow \quad \|y\text{-}u\|_{y_h} < \varepsilon.$

В силу аппроксимации $\forall\ \delta > 0,\ \exists\ h_0,\ \forall\ h < h_0$ имеет место

$$\| \psi_h \|_{\varphi} < \delta, \quad \| \eta \|_{\chi} < \delta.$$

Таким обраючг: $\forall \varepsilon > 0.\ \exists h_0(\delta(\varepsilon)),\ \forall h < h_0$ имеем

$$\| y_h - u \|_{y_h} < \varepsilon,$$

то есть $y \to u$ при $h \to 0$ **.**

Замечания:

Если какое-либо данное нам условие аппроксимировано точно, то устойчивость по ним можно не требовать, так как они не вносят погрешности в решение (кроме ошибок округления, тогда УСТОЙЧИВОСТЬ ПО этим данным нужна).

Для условной аппроксимации (или устойчивости) сходимость тоже носит условный характер.

Для линейных разносхных схем имеет место:$_h$

Теорема 4. *Пусть выполнены условия Теоремы 1, схема A_h, R_h линейна и имеет порядок аппроксимации k, то схема (3)-(4) сходится и её точность (сходимость) не ниже порядка k (порядка аппроксимации).*

Доказательство: Рассмотрим погрешность разностного решения

$$z(x) = y(x) — u(x).$$

Мы получили для решения исходной задачи разностную схему, возмущённую невязками

$$\begin{cases} A_h u = \varphi - \psi, & x \in \omega_h \\ R_h u = \lambda - \eta, & x \in \Gamma_h \end{cases}$$

Вычитая эти уравнения из соответствующих уравнений (3)-(4), найдём:

$$\begin{cases} A_h z = \psi \\ R_h z = \eta \end{cases} \qquad (**)$$

Схема (**) устойчива, то есть

$$\| z \|_{y_h} \le C_1 \| \psi \|_{\varphi} + C_2 \| \eta \|_{\chi}.$$

Но. поскольку исходная схема (3)-(4) обладает аппроксимацией порядка *к,* то

$$\| z \|_{y_h} \le C_1 \alpha h^k + C_2 \beta h^k = C h^k.$$

Фактическая сходимость может иметь более высокий порядок.

3. Разностные схемы для одномерного уравнения теплопроводности

3.1 Постановка задачи. Разностная схема

Рассмотрим задачу о распространении тепла на отрезке в случае простейших краевых условий 1-го рода (условий Дирихле)

$$u_t = a^2 u_{xx} + f(x,t), \qquad 0 < x < l, \quad t>0$$

начальные условия

$$u(x,0) = \mu_1(x) \equiv \mu(x) \tag{11}$$

однородные краевые условия

$$u(0,t) = \mu_2(t) \equiv 0; \quad u(l,t) = \mu_3(t) \equiv 0, \quad t \ge 0.$$

а) **Конечно-разностная аппроксимация простейших дифференциальных операторов первого порядка**.

Введем в области $D=[0, l] \times [0, T]$ сетку $\Omega \equiv \omega_h \times \omega_\tau$, где

$$\omega_h = \left\{ \begin{array}{l} 0 = x_0 < x_1 < \quad < x_N = l \\ x_i = x_0 + nh, \quad h = \frac{x_N - x_0}{N} \end{array} \right\} \quad \text{и} \quad \omega_\tau = \left\{ \begin{array}{l} 0 = t_0 < t_1 < \quad < t_M = l \\ t_m = t_0 + m\tau, \quad \tau = \frac{l - t_0}{M} \end{array} \right\}$$

Рассмотрим сеточную функцию $y(x_n, t_m)=y^m{}_n=y$ на сетке $\Omega \equiv \omega_{h\tau}$. Построим сеточные аналоги простейших дифференциальных операторов первого порядка:

$$\hat{l}u = u'(x) = \frac{du}{dx} \quad \Rightarrow \quad \begin{array}{ll} l_x y = y_{x,i} = \frac{y_{i+1} - y_i}{h}, & \text{производная вперед} \\ l_x y = y_{\bar{x},i} = \frac{y_i - y_{i-1}}{h}, & \text{производная назад} \\ l_{\mathring{x}} y = y_{\mathring{x},i} = \frac{y_{i+1} - y_{i-1}}{2h}, & \text{центральная производная} \end{array} \tag{12}$$

Их аппроксимация $L_h u$ — $(Lu)_h$ имеет следующий порядок:

Для производной вперед l_x

$$\frac{u_{i+1}-u_i}{h}-u'(x_i)=\frac{u(x_i+h)-u_i}{h}-u'(x_i)=$$

$$=\frac{u(x_i)+u'(x_i)h+O(h^2)-u_i}{h}-u'(x_i)=O(h).$$

т .е. обладает аппроксимацией 1- го порядка.

Аналогично $l_{\bar{x}}$

$$\frac{u_i-u_{i-1}}{h}-u'(x_i)=\frac{u(x_i)-u(x_i-h)}{h}-u'(x_i)=$$

$$=\frac{u_i-[u(x_i)-u'(x_i)h+O(h^2)]}{h}-u'(x_i)=O(h)$$

Центральная производная h_{0_x} имеет повышенный порядок аппроксимации

$$\frac{u_{i+1}-u_{i-1}}{2h}-u'(x_i)=\frac{u(x_i+h)-u(x_i-h)}{2h}-u'(x_i)=$$

$$=\frac{u(x_i)+u'(x_i)h+h^2/2\,u''(x_i)+O(h^3)-[u(x_i)-u'(x_i)h+h^2/2\,u''(x_i)+O(h^3)]}{2h}-$$

$$u'(x_i)=O(h^2)$$

b) **Конечно-разностная аппроксимация простейших дифференциальных операторов второго порядка.**

Определим вторую разностную производную для узла x_i (рекурентно):

$$\hat{L}=\frac{d^2}{dx^2}=\frac{d}{dx}\left(\frac{d}{dx}\right)\quad\Rightarrow\quad y_{\bar{x}x,i}=(y_{\bar{x}})_{x,i}=\frac{1}{h}(y_{\bar{x},i+1}-y_{\bar{x},i})=$$

$$=\frac{1}{h}\left(\frac{y_{i+1}-y_i}{h}-\frac{y_i-y_{i-1}}{h}\right)=\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}=y_{x\bar{x},i}. \qquad (13)$$

Получим её порядок аппроксимации

$$L_h u - (Lu)_h = u_{\bar{x}x,i} - \left(u''\right)_i = \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2} - u''(x_i) =$$

$$= \frac{1}{h^2}\left(h^2 u''(x_i) + O\left(h^4\right)\right) - u''(x_i) = O\left(h^2\right).$$

Аналогично мы можем построить аппроксимации и более сложных производных.

Разностная схема. После аппроксимации простейших дифференциальных операторов, вернемся к уравнению (11.1).

Используя так называемый метод *разностной аппроксимации,* мы можем каждый из дифференциальных операторов задачи (11) аппроксимировать соответствующим разностным оператором (12), (13). Производная вперед по *t* для *(n,m)*-го узла

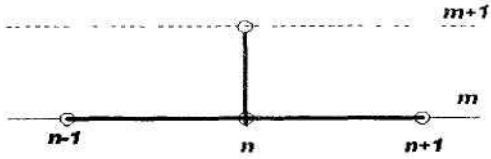
$$y_{t;n,m} = \frac{y_n^{m+1} - y_n^m}{\tau} = \frac{\hat{y}_n - y_n}{\tau} = \frac{\hat{y} - y}{\tau}.$$

Это выражение рассматривается относительно текущего узла x_n на двух слоях по *t.*

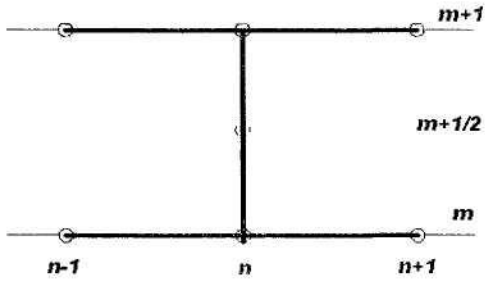
Пространственные производные второго порядка аппроксимируются разностным оператором

$$y_{\bar{x}x;n,m} = \frac{1}{h^2}\left(y_{n+1}^m - 2y_n^m + y_{n-1}^m\right) = \Lambda_h y_n^m \equiv \Lambda_h y.$$

При построении такой разностной аппроксимации на $\omega_{h,\tau}$ мы использовали шаблон из четырех узлов.



Относительно *(m* + 1)-го временного слоя схема получилась *явной* - с *(m* + 1)-го временного слоя используется только одно значение сеточной функции. В дальнейшем мы покажем, что простешая явная схема не является наилучшей в смысле аппроксимации и, особенно, устойчивости. Поэтому сразу же рассмотрим однопараметрическое семейство схем на шеститочечном шаблоне:



$$\frac{y_n^{m+1}-y_n^m}{\tau}=\alpha^2\Lambda_{\overline{x}x}\left\{\sigma y_n^{m+1}+(1-\sigma)y_n^m\right\}+\varphi_n^m.$$

где вес $\sigma\in[0;1]$.

При σ= 0 получается чисто *явная* схема, при σ= 1 - чисто *неявная* схема. При аппроксимации правой части $f(x,\ t)\Rightarrow\varphi^m{}_n$ мы использовали, так называемый, *метод непрерывных коэффициентов* в простейшей его форме, когда подбирается всего один коэффициент $\varphi^m{}_n$ (без дополнительного сложного шаблона). Итак, получаем разностную задачу:

$$\begin{cases}\frac{1}{\tau}(y_n^{m+1}-y_n^m)=\alpha^2\Lambda_h\left\{\sigma\hat{y}+(1-\sigma)y\right\}+\varphi_n^m;\quad (x_n,t_m)\in\omega_{h,\tau}\\ y_n^0=\chi_n;\\ y_0^m=y_N^m=0=\chi_{0,N}^m\end{cases}\qquad(14)$$

Уравнениие (14.1) записано относительно внутренних узлов *(n,m)* сетки $\overline{\Omega}$. При аппроксимации начальных и краевых условий мы также использовали метод неопределенных коэффициентов. Теперь изучим свойства построенной разностной схемы.

3.2. Порядок аппроксимации разностной схемы (14)

Напомним еще раз, что для определения порядка аппроксимации разностной схемы (14), нужно точное решение (11) подставить в эту схему и, в предположении достаточной гладкости решения *u(x,t)*, определить порядки невязок ψ и η по *h* и τ.

Одновременно с этим, мы проследим идею метода неопределенных коэффициентов, выбираемых из соображений обеспечения

максимального порядка аппроксимации (на примере построения φ^m_n и частично χ_n).

Введем в рассмотрение промежуточный слой по t :

$$\bar{t} = t_m + \frac{\tau}{2} = t_{m+\frac{1}{2}} = m\tau + \frac{\tau}{2}.$$

Тогда

а) временна́я часть:

$$\frac{1}{2\frac{\tau}{2}}\left(u_n^{m+1} - u_n^m\right) = u_{\underset{t;n,m+\frac{1}{2}}{0}} = u_t'\left(x_n, t_{m+\frac{1}{2}}\right) + O\left(\tau^2\right);$$

б) пространственная часть:

$$\sigma\,\hat{\bar{u}} + (1-\sigma)\,u = \sigma u\left(x_n, t_{m+\frac{1}{2}} + \frac{\tau}{2}\right) + (1-\sigma)\,u\left(x_n, t_{m+\frac{1}{2}} - \frac{\tau}{2}\right) =$$

$$= \sigma\left\{u\left(x_n, t_{m+\frac{1}{2}}\right) + \frac{\tau}{2}\overline{u}_t + \frac{1}{2!}\left(\frac{\tau}{2}\right)^2 \overline{u}_{tt} + O\left(\tau^3\right)\right\} +$$

$$+ (1-\sigma)\left\{u\left(x_n, t_{m+\frac{1}{2}}\right) - \frac{\tau}{2}\overline{u}_t + \frac{1}{2!}\left(\frac{\tau}{2}\right)^2 \overline{u}_{tt} + O\left(\tau^3\right)\right\} =$$

$$= \overline{u} + \tau\left(\sigma - \frac{1}{2}\right)\overline{u_t} + O\left(\tau^2\right).$$

Здесь чертой сверху обозначено значение функции в точке $(x_n;t_{m+1/2})$. Следовательно,

$$\Lambda_h\left[\sigma\,\hat{\bar{u}} + (1-\sigma)\,u\right] = \Lambda_h\left[\overline{u} + \tau\left(\sigma - \frac{1}{2}\right)\overline{u}_t + O\left(\tau^2\right)\right] =$$

$$= \overline{u}_{xx} + \left(\sigma - \frac{1}{2}\right)\tau\overline{u}_{txx} + O\left(\tau^2 + h^2\right).$$

Таким образом подстановка u (x, t) в разностное уравнениие (14.1) дает

$$\underline{u_t\left(x_n, \bar{t}\right)} + O\left(\tau^2\right) = \underline{a^2 u_{xx}\left(x_n, \bar{t}\right)} + a^2\tau\left(\sigma - \frac{1}{2}\right)u_{txx}\left(x_n, \bar{t}\right) + \varphi_n^m + O\left(\tau^2 + h^2\right).$$

В силу задачи (11) подчеркнутые члены анулируются, если в уравнении есть слагаемое $f(x_n, \bar{t})$. Таким образом, если мы хотим обеспечить аппроксимацию задачи (11), необходимо:

$$\varphi_n^m = f\left(x_n, \bar{t}\right) = f\left(x_n, t_{m+\frac{1}{2}}\right).$$

Тогда:

1) при $\sigma \neq 1\backslash 2$ мы получаем аппроксимацию уравнения (11.1) с порядком $O(\tau^2 + h^2)$;

2) при $\sigma = 1\backslash 2$ мы получаем повышенный порядок аппроксимации $O(\tau^2 + h^2)$ (обратим внимание на наличие симметрии в сеточном шаблоне).

3) Аппроксимация начальных условий в этой задаче тривиальна:

$$\chi^0_n = \mu(x_n, t_0)$$

чтобы не вносить дополнительной погрешности ($\eta_1 \equiv 0$) .

3.3. Устойчивость разностной схемы (14)

Напомним еще раз: *линейная схема (14) называется устойчивой по входным данным (по правой части и начальным условиям), если при достаточно малых h и τ существуют C_1, C_2 (не зависящие от h и τ), такие что,*

$$\|\delta y\| \leq C_1 \|\delta \varphi\| + C_2 \|\delta \chi\|,$$

то есть, решение непрерывно зависит от правой части и начальных условий.

Устойчивость разностной схемы, а следовательно и её сходимость при наличии аппроксимации, мы покажем в равномерной (чебышевской) метрике:

$$\|y\|_l = \max_{n,m} |y_n^m|$$

(сеточный аналог равномерной по *t* и *x* метрики).

Введем норму сеточного решения на *m*-ом слое:

$$\|y^m\| = \max_n |y_n^m|.$$

В силу Теоремы 1 (о достаточном условии равномерной устойчивости линейных разностных схем по начальным условиям) и Теоремы 2 (достаточного условия устойчивости линейной разностной схемы по правой части), нам достаточно показать, что, если существуют $C_1 \geq 0$ и $C_2 > 0$ и

$$\|y^{m+1}\| \leq (1 + \tau C_1) \|y^m\| + \tau C_2 \|\varphi\|, \qquad \textbf{(*)}$$

то схема устойчива по входным данным.

Ограничимся исследованием устойчивости в двух предельных случаях: чисто неявной (σ = 1) и чисто явной (σ = 0) схем.

а) Устойчивость чисто неявной схемы (σ=1): Рассмотрим разностное уравнениие (14.1):

$$\frac{1}{\tau}(\hat{y}-y)=a^2\Lambda[\hat{y}]+\varphi_n^m=\frac{a^2}{h^2}\left(y_{n+1}^{m+1}-2y_n^{m+1}+y_{n-1}^{m+1}\right)+\varphi_n^m.$$

Обозначим $\gamma=\frac{\tau a^2}{h^2}$, тогда

$$y_n^{m+1}-y_n^m=\gamma\left(y_{n+1}^{m+1}-2y_n^{m+1}+y_{n-1}^{m+1}\right)+\tau\varphi_n^m \Longleftrightarrow$$

$$y_n^{m+1}=y_n^m-\gamma\left(2y_n^{m+1}-y_{n+1}^{m+1}-y_{n-1}^{m+1}\right)+\tau\varphi_n^m.$$

Покажем, что в этом случае (σ = 1) достаточное уетовие устойчивости (*) выполнено. Найдем на слое $(m+1)$ тот узел k_0, в котором y_n^{m+1} принимает наибольшее значение:

$$\max_n y_n^{m+1}=y_{k_0}^{m+1}\geq y_n^{m+1},\quad \forall n.$$

Тогда

$$2y_{k_0}^{m+1}-y_{k_0+1}^{m+1}-y_{k_0-1}^{m+1}\geq 0.$$

Поэтому

$$y_{k_0}^{m+1}\leq y_{k_0}^m+\tau\varphi_{k_0}^m\leq\max_n y_n^m+\tau\max_{n,m}\varphi_n^m. \quad (**)$$

С другой стороны, найдем на слое $(m+1)$ узел t_0 где y_n^{m+1} принимает минимальное значение:

$$\min_n y_n^{m+1}=y_{t_0}^{m+1}\leq y_n^{m+1},\forall n.$$

Тогда

$$2y_{t_0}^{m+1}-y_{t_0+1}^{m+1}-y_{t_0-1}^{m+1}\leq 0$$

и

$$y_{t_0}^{m+1}\geq y_{t_0}^m+\tau\varphi_{t_0}^m\geq\min_n y_n^m+\tau\min_{n,m}\varphi_n^m. \quad (***)$$

Объединяя (**) и (***), найдем:

$$\left\|y^{m+1}\right\|=\max_n|y_n^{m+1}|\leq\left\|y^m\right\|+\tau\left\|\varphi\right\|,$$

что совпадает с условием (*) при $C_1 = 0$, $C_2 = 1$. Таким образом, неявная схема ($\sigma = 1$) *безусловно* устойчива по входным данным (при любых τ и h).

б) Устойчивость чисто явной схемы ($\sigma = 0$):
Для чисго явной схемы уравнение (14.1) имеет вид

$$\frac{1}{\tau}\left(y_n^{m+1} - y_n^m\right) = \alpha^2 \Lambda\left[y_n^m\right] + \varphi_n^m.$$

Откуда

$$y_n^{m+1} = y_n^m + \gamma\left(y_{n+1}^m - 2y^m + y_{n-1}^m\right) + \tau\varphi_n^m = (1-2\gamma)\, y_n^m + \gamma y_{n-1}^m + \gamma y_{n+1}^m + \tau\varphi_n^m$$

Пусть $(1 — 2\gamma) > 0$, то есть $0 < \gamma < \frac{1}{2}$, гогда

$$\left|y_n^{m+1}\right| = \left|(1-2\gamma)\, y_n^m + \gamma y_{n-1}^m + \gamma y_{n+1}^m + \tau\varphi_n^m\right| \leq$$

$$\leq (1-2\gamma)\left|y_n^m\right| + \gamma\left|y_{n+1}^m\right| + \gamma\left|y_{n-1}^m\right| + \tau\left|\varphi_n^m\right|, \quad \forall n$$

Тем самым

$$\left\| y^{m+1}\right\| \leq \left\|y^m\right\| + \tau\|\varphi\|, \quad C_1 = 0, C_2 = 1$$

Итак при

$$\gamma = \frac{\tau a^2}{h^2} < \frac{1}{2} \qquad (15)$$

явная схема устойчива. Это условие накладывает жесткие ограничения на временной шаг сетки:

$$\tau < \frac{h^2}{2a^2} \qquad (15^*)$$

Покажем, что при $\gamma > \frac{1}{2}$ явная схема *неустойчива* в чебышевской норме. Для этого достаточно показать, что, однажды возникнув, ошибка в решении будет при дальнейших вычислениях неограниченно возрастать. Рассмотрим однородною задачу (без правой части) Соответствующие возмущения - это возмущения начальных условий на данном слое. Схема при этом имеет вид

$$y_n^{m+1} = (1 - 2\gamma)\, y_n^m + \gamma y^{m}_{n-1} + \gamma y_{n+1}^m$$

Пусть на m- ом слое возникла ошибка δy^m_n, тогда

$$\tilde{y}_n^m = y_n^m + \delta y_n^m$$

и, поскольку $\tilde{y}_n^m$ - это решение той же схемы,

$$\tilde{y}_n^{m+1} = y_n^{m+1} + \delta y_n^{m+1} = (1-2\gamma)(y_n^m + \delta y_n^m) + \gamma \tilde{y}_{n+1}^m + \gamma \tilde{y}_{n-1}^m,$$

то в силу линейности нашей задачи, получаем уравнение для ошибки

$$\delta y_n^{m+1} = \delta y_n^m (1-2\gamma) + \gamma \delta y_{n+1}^m + \gamma \delta y_n^m{}_1$$

Предположим, что ошибка является быстро осциллирующей функцией и имеет вид

$$\delta y^m = (-1)^n \varepsilon, \quad \varepsilon > 0,$$

где ε - некоторое достаточно малое число, тогда

$$\delta y_n^{m+1} = (1-2\gamma)(-1)^n \varepsilon + \gamma(-1)^{n+1}\varepsilon + \gamma(-1)^{n-1}\varepsilon = (-1)^n \varepsilon (1-4\gamma)$$

Но, так как $\gamma > 1/2$, то $4\gamma > 2$ и

$$\delta y_n^{m+1} = (-1)^{n+1}\varepsilon(4\gamma - 1).$$

Следовательно через *k* временных слоев

$$\left|\delta y_n^{m+k}\right| = \varepsilon (4\gamma - 1)^k \to \infty, \quad k \to \infty.$$

Уменьшение шага τ (при $\gamma > 1/2$) не спасает, ибо при фиксированном *T* растет обьем неустойчивых вычислений (за счет числа шагов), следовательно и ошибка. Значит явная схема σ =0 при

$$\gamma = \frac{\tau a^2}{h^2} > \frac{1}{2}$$

— неустойчива.

Замечания:

1) В силу устойчивости наших схем, мы показали, что $\|y^{m+1}\| \le \|y^m\| + \tau \|\varphi\|$. Это неравенство доказывает принцип максимума для наших схем: Пусть φ = 0 тогда

$$\|y^{m+1}\| \le \|y^m\| \le \ldots \le \|y^0\| = \|\chi\|$$

таким образом, во внутренних точках *t* и *x* норма решения не превосходит норму начальных условий.

2) В сеточном аналоге нормы L_2 методом гармоник (далее) можно показать, чю схема (14) устойчива при

$$\sigma \geq \frac{1}{2} - \frac{h^2}{4\tau a^2}. \qquad (15')$$

В частности

a) σ = 1\2 - безусловно устойчивая схема.

b) Схема с σ = 0 устойчива при условии

$$\frac{h^2}{4\tau a^2} \geq \frac{1}{2} \quad \Leftrightarrow \quad \frac{\tau a^2}{h^2} \leq \frac{1}{2} \Leftrightarrow \quad \tau \leq \frac{h^2}{2a^2}.$$

3) Можно показать, что в *C* схема (14) устойчива по входным данным при

$$\sigma \geq \frac{1}{2} - \frac{h^2}{2\tau a^2}. \qquad (15'')$$

В частности схема с σ= 0 устойчива при условии

$$\tau \leq \frac{h^2}{a^2}.$$

3.4 Сходимость разностной схемы (14)

Рассмотрим погрешность сеточного решения

$$z_n^m = y_n^m - u_n^m$$

$u^m_n = u\,(x_n, t_m)$ при простейшем способе проектирования $u(x, t)$ на сетку Ω.

Мы показали, что при наличии аппроксимации и устойчивости разностной схемы она обладает сходимостью, и порядок точности схемы (14) не ниже её порядка аппроксимации. В нашем случае имеет место равномерная сходимость либо сходимость в среднем (в той же метрике, где есть и устойчивость). Поэтому для погрешности сеточного решения имеем оценки

a) $\sigma = \frac{1}{2}$:

$$\|z\|_c = O\left(h^2 + \tau^2\right)$$
$$u(x,t) \in C^{(4)}[0,l] \times C^{(3)}[0,T]. \qquad (16)$$

b) $\sigma \neq \frac{1}{2}$:

$$\|z\|_c = O\left(\tau + h^2\right)$$
$$u(x,t) \in C^{(4)}[0,l] \times C^{(2)}[0,T]$$

При этом для обеспечения соответствующей аппроксимации, решение задачи (11) должно обладать указанной гладкостью.

3.5 Алгоритмы численного решения задачи (14). Прогонка

Сделаем краткое замечание относительно способов решения задачи (11).

а) В случае явной схемы (σ = 0). Лгорит м очевиден и определяется написанной явной формулой:

$$\begin{cases} y_n^{m+1} = (1-2\gamma)\, y_n^m + \gamma y_{n-1}^m + \gamma y_{n+1}^m + \tau\varphi_n^m; \quad 1 \le n \le N-1 \\ y_0^{m+1} = y_N^{m+1} = 0; \\ y_n^0 = \backslash_n \end{cases} \qquad (14^*)$$

Напомним, что $\gamma < 1/2$.

б) Для неявной схемы (σ = 1). Решение на *(m* + 1)-ом временном слое находим из формул

$$\hat{y}_n = y_n - \gamma\,(2\hat{y}_n - \hat{y}_{n+1} - \hat{y}_{n-1}) + \tau\varphi_n^m,$$

что приводит к алгебраической системе

$$(1+2\gamma)\,\hat{y}_n + \gamma\hat{y}_{n-1} + \gamma\hat{y}_{n+1} = y_n + \tau\varphi_n^m \Leftrightarrow$$

$$\begin{cases} A_n\hat{y}_{n-1} + B_n\hat{y}_n + C_n\hat{y}_{n+1} = F_n; \\ \hat{y}_0 = \hat{y}_N = 0. \end{cases} \qquad (14^{**})$$

Это СЛАУ с трехдиагональной матрицей, имеющей диагональное преобладание $B_n \ge A_n + C_n$. В таком случае решение $\hat{y}_n$ существует и единственно. Решение дается формулами прогонки. Вычисления устойчивы. Общий объем вычислений при переходе на *(m* + 1)-ый слой *O(*9*N)* действий и требуется всего *O*(3*N*) ячеек памяти для хранения матрицы СЛАУ.

Замечания: Мы рассмотрели однопараметрическое семейство схем (14) для одномерного уравнения теплопроводности.

Явная схема *(*σ = 0) алгоритмически наиболее проста, но требует выполнения жестких условий устойчивости $\tau < \dfrac{h^2}{2a^2}$, поэтому используется редко.

Широкое применение имеет схема σ = 1\2, повышенной точности $O(h^2 + \tau^2)$ - безусловно устойчивая схема.

Схемы с σ = 1\2, σ = 1 особенно эффективны для уравнениий с переменными коэффициентами или для квазилинейных уравнениий.

4. Разностные схемы для одномерного уравнения колебаний

4.1 Постановка задачи. Разностная схема "крест"

Рассмотрим задачу для уравнения колебаний на отрезке с краевыми **условиями** 1-го рода (задача Дирихле)

$$u_{tt} = a^2 u_{xx} + f(x,t), \qquad 0 < x < l, \quad t > 0$$

начальные условия

$$u(x,0) = \mu_1(x), \quad u_t(x,0) = \mu_2(x), \quad t = 0, \quad x \in [0,l] \tag{17}$$

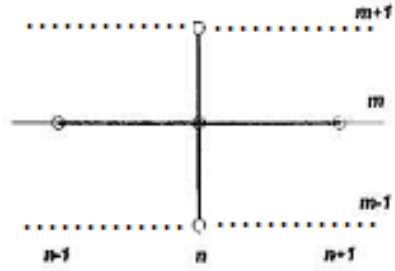
краевые условия 1-го рода

$$u(0,t) = \mu_3(t) \equiv 0; \quad u(l,t) = \mu_4(t) \equiv 0, \quad t \geq 0$$

Введём обозначения

$$y = y_n^m; \quad \hat{y} = y_n^{m+1}; \quad \check{y} = y_n^{m-1};$$

и, используя метод разностной аппроксимации, построим схему «крест» для одномерного уравнения теплопроводности.



Разностная аппроксимация самого уравнения;

$$\frac{1}{\tau^2}(\hat{y} - 2y + \check{y}) = \frac{a^2}{h^2}(y_{n+1} - 2y_n + y_{n-1}) + \varphi_n^m. \tag{18.1}$$

При аппроксимации правой части мы использовали метод *неопределенных коэффициентов.*

Начальное условие для функции *u(x)* аппроксимируется точно

$$y(x_n, 0) = y_n^0 = \chi_{1n} = \mu_1(x_n) \Leftrightarrow \eta_1 \equiv 0.$$

Аппроксимация краевых условий также не вносит дополнительных погрешностей $\eta_3 \equiv 0$ и $\eta_4 \equiv 0$

$$y_0^m = \chi_3^m = \mu_3(t_m) = 0; \quad y_N^m = \chi_4^m = \mu_4(t_m) = 0.$$

При аппроксимации начального условия для производной

$$y_t(x_n, 0) = \frac{y_n^1 - y_n^0}{\tau} = \chi_{2n}$$

порядок аппроксимации зависит от способа построения сеточной функции χ_2. Простейшая аппроксимация

$$\chi_{2n} = \mu_2(x_n) \implies \eta_2 \equiv O(\tau).$$

Если использовать само уравнение, то можно получить более аккуратную аппроксемацию начального условия

$$\frac{y_n^1 - y_n^0}{\tau} \Rightarrow \frac{u(x_n, \tau) - u(x_n, 0)}{\tau} = u_t'(x_n, 0) + \frac{\tau}{2} u_{tt}''(x_n, 0) + O(\tau^2)$$

Допустим:

$$u_{tt}(x_n, 0) = a^2 u_{xx}(x_n, 0) + f(x_n, 0) = a^2 \mu_{1xx}(x_n) + f_n^0.$$

При этом можно использовать аппроксимацию порядка $O(h^2)$ для $\mu_{1xx}(x_n)$. Таким образом

$$\frac{y_n^1 - y_n^0}{\tau} = \mu_2(x_n) + \frac{\tau}{2}\left(a^2 \mu_{1xx}(x_n, 0) + f_n^0\right); \quad \eta_2 = O(\tau^2).$$

Теперь запишем разностную схему для исходной задачи (17)

$$\frac{\hat{y} - 2y + \check{y}}{\tau^2} = a^2 \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + \varphi_n^m$$

краевые условия

$$y_0^m = \chi_3^m = \mu_3(t_m) \equiv 0; \quad y_n^m = \chi_4^m = \mu_4(t_m) \equiv 0 \qquad (18)$$

начальные условия

$$y_n^0 = \chi_1(x_n) = \mu_1(x_n)$$

$$\frac{y_n^1 - y_n^0}{\tau} = \chi_2(x_n) = \begin{cases} \mu_2(x_n) & \Rightarrow \quad \eta_2 = O(\tau) \\ \mu_2(x_n) + \frac{\tau}{2}\left(a^2\mu_{1xx}(x_n) + f(x_n, 0)\right) & \Rightarrow \quad \eta_2 = O(\tau^2). \end{cases}$$

Это *явная схема* относительно $\widehat{y} \equiv y_n^{m+1}$. После того, как найдено $\{y^1{}_n\}$ из начального условия далее расчётные формулы просты.

4.2 Порядок аппроксимации разностной схемы (18)

Сам принцип построения разностной схемы (18) позволяет утверждать, что:

1) $\varphi_n^m = f(x_n, t_m)$ — необходимое условие для аппроксимации;

2) порядок аппроксимации (18.1) есть $O(\tau^2+h^2)$ в силу симметрии полученных разностных формул;

3) с учетом (18.2) $\Rightarrow$ общий порядок аппроксимации схемы $O(\tau^2+h^2)$.

4.3. Устойчивость разностной схемы (18)

Для доказательства устойчивости схемы (18) используем *метод разделения переменных* (поскольку коэффициенты схемы постоянны или их можно "заморозить" на данном временном слое) или *метод гармоник*. Этим методом доказывался устойчивость разностной схемы в сеточном аналоге $\mathcal{L}_2$ ('"в среднем"').

На каждом временном слое сеточная функция по $\{x_n\}$ может быть разложена по собственным сеточным функциям сеточного оператора Лапласа $\Lambda_{\bar{x}x}$ это "косинусы" и "синусы" от $\left(\frac{\pi k}{l} x, \right)$ для k-ой функции. Поведение гармоник на различных слоях по t характеризуется множителями роста гармоники ρ_k, т. е. рассматривается устойчивость решения вида

$$y_n^m = y(x_n, t_m) = (\rho_k)^m e^{ik\tau_n}, \qquad k = 0; \pm 1; \pm 2; \ldots$$

Имеет место теорема

Теорема 5. *Двуслойная разностная схема с постоянными коэффициентами устойчива в среднем по начальным данным , если* $\forall k$ *(т. е. для любой гармоники) множитель роста удовлетворяет условию*

$$|\rho_k| \le 1+C\tau;\ C \ge 0 - const. \qquad (*)$$

Ограничимся замечаниями:

1) Фактически *const* $C \ge 0$ не должна быть очень большой На практике условие (*) проверяют для $C = 0$, т. е. $|\rho_k| \le 1$

2) Условие (*) в некотором смысле и необходимо, т.е если существует гармоника k_0 для которой (*) не выполняйся, то схема неустойчива.

Теперь вернемся к нашей задаче (18). Пусть $\varphi_n^m \equiv 0$ $y_n^m = e^{ikhn}$ - начнем с этого слоя. Тогда

$$\hat{y} = \rho_k e^{ikhn} = \rho_k y; \quad \check{y} \Rightarrow y = \rho_k \check{y}$$

Однородное уравнение (18.1) даст

$$\left(\rho_k - 2 + \frac{1}{\rho_k}\right) = \underbrace{\frac{\tau^2 a^2}{h^2}}_{\gamma^2}\left(e^{ikh} - 2 + e^{-ikh}\right) = \gamma^2(2\cos kh - 2) = -4\gamma^2 \sin^2\frac{kh}{2}$$

Множители роста k -ой гармоники p_k удовлетворяют уравнению

$$\rho_k^2 + 2\rho_k\left(1 - 2\gamma^2 \sin^2\frac{kh}{2}\right) + 1 = 0. \qquad (**)$$

По теореме Виетта $(\rho_k)_1(\rho_k)_2=1$ и требование устойчивости $|(\rho_k)_{1,2}| \le 1$ выполнено, если только

$$\left|(\rho_k)_{1,2}\right| = 1$$

Значит $(\rho_k)_1$ и $(\rho_k)_2$ - комплексно-сопряженные числа. Это в свою очередь возможно лишь при отрицатетьном дискриминанте уравнения $(^{**})$ $D < 0$.

Итак

$$\left(1 - 2\gamma^2 \sin^2\frac{kh}{2}\right)^2 - 1 < 0 \quad \Leftrightarrow \quad \left|1 - 2\gamma^2 \sin^2\frac{kh}{2}\right| < 1$$

Это условие относительно γ (точнее τ и h) и оно заведомо верно $\forall k$, если $\gamma^2 < 1$, т.с.

$$\frac{\tau a}{h} < 1 \quad \text{-- условие Куранта.} \qquad (19)$$

Замечания:

1) Схема "крест'" устойчива в среднем по начальным данным при дополнительном условии $\tau a/h < 1$.

2) При условии (19) схема "крест" устойчива по правой части;

3) При условии (19) схема "крест" устойчива по начальным данным и правой части в равномерной сеточной норме (в *C).*

4.4. Сходимость схемы "крест"

Установленный нами порядок аппроксимации и устойчивость схемы (18) позволяет утверждать наличие сходимости схемы (в соответствующей метрике) с точностью не ниже порядка аппроксимации. Итак

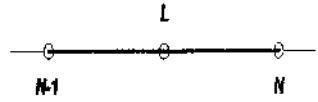
$$\|z\| = \|y - u\| = \begin{cases} O(\tau + h^2) \\ O(\tau^2 + h^2) \end{cases} \quad \text{при} \quad \frac{\tau a}{h} < 1 \Leftrightarrow \tau < \frac{h}{a}.$$

Сходимость указанных порядков возможна лишь для решений, обладающих достаточной гладкостью, чтобы обеспечить аппроксимацию этих порядков. Достаточно

$$u(x,t) \in C^{(4)}[0,l] \times C^{(4)}[0,T].$$

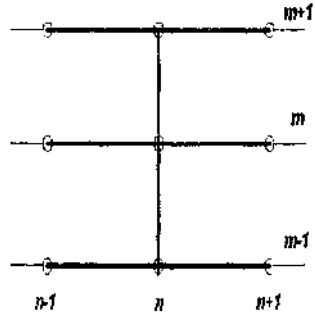
Замечания к п. 4:

1) При аппроксимации краевых условий 2-го рода, например $u_x(l, t) = \mu_4(t)$, удобно сетку по x строить так. чтобы точка $x=l$ оказалась бы между узлами сетки, тогда



$$\frac{y_N^m - y_{N-1}^m}{h} = \mu_4(t_m) \quad \Rightarrow \quad \eta_4 = O(h^2).$$

2) Не представляет труда построить для одномерного уравнения колебаний неявную 9-ти точечную схему с весами.



В шаблоне использованы три временных слоя. Основное уравнение схемы

$$\frac{1}{\tau^2}(\hat{y}-2y+\check{y})=a^2\Lambda_{xx}[\sigma\hat{y}+(1-2\sigma)y+\sigma\check{y}]+\varphi_n^m,$$

где 0 ≤σ≤1\2.

5. Многомерные разностные схемы для уравнения теплопроводности

Рассмотрим задачу о распределении тепла в прямоугольной области:

$$\begin{cases} u_t = a^2(u_{x_1x_1}+u_{x_2x_2})+f(x_1,x_2,t), & \begin{matrix}0<x_1<l_1\\ 0<x_2<l_2\\ 0<t\end{matrix} \\ u\,|_\Gamma = \mu_\Gamma(t)\ \text{(задача Дирихле)} \\ u\,|_{t=0} = \mu(x_1,x_2) \end{cases} \qquad (20)$$

Будем предполагать, что задача (20) корректна и входные данные обеспечивают нужную гладкость решения.

5.1. Разностная схема

Обобщим на задачу (20) схемы п.3. Рассмотрим в $\bar{D}$ равномерную сетку:

$$\overline{\omega}_{h\ h_2\ \tau} = \left\{ \begin{array}{r} x_{1n} = nh_1;\ n = \overline{1,N};\ h_1 = \dfrac{l_1}{N} \\ (x_{1n}, x_{2k}, t_m):\ x_{2k} = kh_2;\ k = \overline{1,K};\ h_2 = \dfrac{l_2}{K} \\ t_m = m\tau;\ m = \overline{1,M};\ \tau = \dfrac{T}{M} \end{array} \right\}.$$

Граничные условия аппроксимируются в этом случае точно:

$$\chi_\Gamma^m = \mu_\Gamma(t_m); \quad \eta_\Gamma = 0; \quad \begin{array}{l}\text{на каждой} \\ \text{стороне прямоуголь-} \\ \text{ника,}\end{array}$$

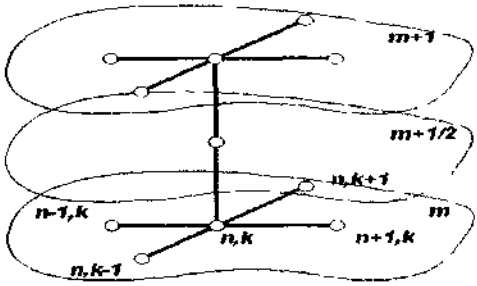
поскольку точки сетки естественным образом задают границу области $\overline{D}$.

Пусть $y_{n,k}^m = y$: $\widehat{y} = y_{n\,k}^{m+1}$. Составим двуслойную схему с весами. Аппроксимируем оператор Лапласа $\Delta_2 \Rightarrow$

$$\Lambda = \Lambda_1 + \Lambda_2, \qquad \text{где}$$

$$\Lambda_1 y = \Lambda_{\overline{x}_1 x_1} y = \frac{1}{h_1^2}(y_{n+1,k} - 2y_{n,k} + y_{n-1,k})$$

$$\Lambda_2 y = \Lambda_{\overline{x}_2 x_2} y = \frac{1}{h_2^2}(y_{n,k+1} - 2y_{n,k} + y_{n,k-1}).$$



Эти операторы аппроксимируют

$$\frac{\partial^2}{\partial x_1^2} \text{ и } \frac{\partial^2}{\partial x_2^2}$$

со вторым порядком по пространственным переменным. Сеточный оператор ($\Lambda_1 + \Lambda_2$) аппроксимирует оператор Лапласа Δu в узле *(n,k)* с невязкой $O(h_1^2 + h_2^2)$.

Тогда основное уравнение задачи (20) аппроксимируется разностным уравнением

$$\frac{1}{\tau}(\hat{y}_{n,k} - y_{n,k}) = a^2(\Lambda_1 + \Lambda_2)[\sigma\hat{y}_{n,k} + (1-\sigma)y_{n,k}] + \varphi_{n,k}^m \quad (21.1)$$

Существенный недостаток схемы (21) в *многомерном* случае связан с тем. что как чисто явная схема σ = 0, как и неявная σ≠0 схемы приводят к неэффективным численным алгоритмам для построения решения на слое *T*. Если из соображений аппроксимации $h_1 \sim h_2$: $N \sim K$, то оценка числа арифметических действий для явной σ = 0 схемы для построения решения на последнем слое *T* есть $O(N^4)$. Действительно, для перехода на следующий временной слой решается явная система уравнений с числом неизвестных $O(NK) \sim O(N^2)$. При этом требования устойчивости схемы ограничивают временной шаг $\tau \sim \left(\frac{1}{h^2}\right)^{-1} \sim h^2 \sim N^{-2}$. Что и приводит к общей оценке числа арифметических действий $O(N^4)$.

Для неявный схем σ≠0 положение ещё чуже. Ограничиваясь абсолютно устойчивым вариантом схем при $\sigma \geq \frac{1}{2}$, на каждом временном слое приходится решать СЛАУ с N^2 уравнений при ширине ленты порядка $O(2N)$. Метод исключения Гаусса требует $O(N^6)$ с учётом ленточной структуры матрицы - $O(N^4)$ действий. Требование аппроксимации даёт $O(N)$ шагов по времени. Итого — $O(N^5)$! Неявная схема менее выгодна в этом случае!

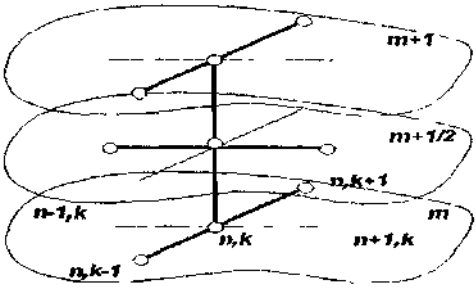
Поэтому предпочтение отдают абсолютно устойчивым ($\tau \sim h$), экономичным разностным схемам, в которых при переходе на очередной временной слой совершаемся всего $O(N^2)$ действий.

6. Продольно-поперечная разностная схема для уравнения теплопроводности. Экономичные разностные схемы

Введем промежуточный по *t* слой $(m+\frac{1}{2})$ и рассмотрим разностную схему

$$\frac{\overline{y}_{n,k}-y_{n,k}}{\tau/2}=a^2\Lambda_1\overline{y}_{n,k}+a^2\Lambda_2 y_{n,k}+\overline{f}_{n,k}$$

$$\frac{\widehat{y}_{n,k}-\overline{y}_{n,k}}{\tau/2}=a^2\Lambda_1\overline{y}+a^2\Lambda_2\widehat{y}+\overline{f}_{n,k}. \qquad \textbf{(23)}$$



Обсудим построение решения уравнения (23) на $(m+1)$ слое:

1) Уравнение (23.1) позволяет найти $\overline{y}_{n,k}$ по неявной схеме относительно x_1 и по явной схеме относительно $x_2 \Rightarrow$. Решается система с 3-х диагональной матрицей относительно переменной x_1 эффективным методом прогонки по x_1 при каждом k (k - раз прогонка с $O(N)$ действий $\Rightarrow O(NK)$ действий).

2) Уравнение (23.2) позволяет найти $\hat{y}_{n,k}$ по неявной схеме относительно x_2 и по явной схеме относительно $x_1 \Rightarrow$ прогонка по x_2 при каждом $n \Rightarrow O(N\ K)$ действии $\Rightarrow$ итого $O(2N\ K) \sim O(N^2)$ действии.

3) Диагональные коэффициенты в соответствующих матрицах на каждом шаге преобладают — тем самым решение существует, единственно и вычисления по формулам прогонки устойчивы.

4) Общее число действий при переходе на $(m+1)$- ый временной слой $O(30\ N^2)$ действий.

Другие достоинства схемы:

6.1 Устойчивость продольно-поперечной схемы

Воспользуемся методом гармоник. Рассмотрим

$$y_{n,k} = \exp(ix_{1n}p + ix_{2k}q); \quad \overline{y} = p'_{p,q}\, y; \quad \widehat{y} = p''_{p,q}\, \overline{y}$$

(свои множители роста на каждом полуслое). Тогда (23.1):

$$p'_{p,q} - 1 = \frac{\tau a^2}{2h_1^2}\left(-4\sin^2\frac{ph_1}{2}\right)p'_{p,q} + \frac{\tau a^2}{2h_2^2}\left(-4\sin^2\frac{qh_2}{2}\right),$$

т.е.

$$\left.\begin{array}{l} p'_{p,q} = \dfrac{1 - \frac{2\tau a^2}{h_2^2}\sin^2\frac{qh_2}{2}}{1 + \frac{2\tau a^2}{h_1^2}\sin^2\frac{ph_1}{2}} \\ \text{Аналогично} \\ p''_{p,q} = \dfrac{1 - \frac{2\tau a^2}{h_1^2}\sin^2\frac{ph_1}{2}}{1 + \frac{2\tau a^2}{h_2^2}\sin^2\frac{qh_2}{2}} \end{array}\right\} \quad \Rightarrow \quad |p_{pq}| = |p'_{p,q}\, p''_{p,q}| \le 1$$

всегда! $\forall\, p$ и q. Таким образом схема (23) безусловно (абсолютно) устойчива $\mathcal{L}_2$ по начальным данным (и по правой части тоже).

Для рассмотренной схемы имеет место абсолютная устойчивость в C но начальным условиям и по правой част.

Осталось установить аппроксимацию.

6.2 Аппроксимация продольно-поперечной схемы

Исключим из (23) слой $\overline{y}_{n,k}$. Для этого вычтем уравнения (1)-(2), найдём:

$$2\frac{\overline{y}_{nk}}{\tau/2} - \frac{\widehat{y}_{nk} + y_{nk}}{\tau/2} = -a^2\Lambda_2(\widehat{y} - y). \qquad \text{т.е.}$$

$$\overline{y}_{nk} = \frac{\widehat{y}_{nk} + y}{2} - \frac{\tau a^2}{4}\Lambda_2(\widehat{y} - y); \qquad \textbf{(*)}$$

Складывая уравнения (1) - (2), найдем:

$$\frac{\widehat{y}_{nk} - y_{nk}}{\tau/2} = a^2\Lambda_1(2\overline{y}_{nk}) + a^2\Lambda_2(\widehat{y}_{nk} + y_{nk}) + 2\overline{f}_{nk}.$$

Откуда, с учетом (*), получим

$$\frac{\widehat{y}_{nk}-y_{nk}}{\tau}=a^2\Lambda_1\left(\frac{\widehat{y}_{nk}+y}{2}\right)-\frac{\tau a^2}{4}\Lambda_1\Lambda_2(\widehat{y}-y)+a^2\Lambda_2\frac{\widehat{y}_{nk}+y_{nk}}{2}+\overline{f}_{nk}=$$

$$=a^2(\Lambda_1+\Lambda_2)\frac{\widehat{y}_{nk}+y_{nk}}{2}-\underbrace{\frac{\tau a^2}{4}\Lambda_1\Lambda_2\overbrace{(\widehat{y}-y)}^{O(\tau)}}_{O(\tau^2)}+\overline{f}_{nk}.$$

Итак, это почти симметричная схема с $\sigma_1 = \sigma_2 = \frac{1}{2}$, тем самым — схема обладает аппроксимацией при условии $\overline{f}_{nk}=f\left(x_{1n},x_{2n},t_{m+\frac{1}{2}}\right)$ и порядок аппроксимации

$$\psi=O\left(\tau^2+h_1^2+h_2^2\right).$$

Схема (23) безусловно устойчива и обладает повышенной аппроксимацией, следовательно она сходится в указанной прямоугольной области на равномерной сетке и обладает точностью не хуже, чем

$$\|y-u\|=O\left(\tau^2+h_1^2+h_2^2\right).$$

Замечания:

1) Схема обладает той же сходимостью в *C*.

2) Для обеспечения указанного порядка точности разностной схемы грс-бусчся, чтобы решения исходной задачи обладали гладкостью не хуже, чем

$$u(x_1,x_2,t)\in C^{(5)}([0;l_1]\times[0;l_2])\bigcup C^{(5)}[0;T].$$

Приложение 3

Downloads page

ALGLIB User Guide - Одномерная и многомерная оптимизация - L-BFGS алгоритм минимизации функции многих переменных

L-BFGS алгоритм минимизации функции многих переменных

Об алгоритме

Квази-Ньютоновские методы: принцип работы

Классический метод Ньютона ипользует гессиан функции. Шаг метода определяется, как произведение матрицы, обратной к гессиану, на градиент функции. Если функция является положительно определенной квадратичной формой, то за один шаг данного метода мы окажемся в её минимуме. В случае знаконеопределенной квадратичной формы, у которой нет минимума, мы сойдемся к седловой точке или к максимуму. Одним словом, метод ищет стационарную точку квадратичной формы.

На практике обычно приходится иметь дело с функциями, не являющимися квадратичными формами. Если такая функция гладкая, то в окрестностях минимума она достаточно хорошо описывается квадратичной формой, чтобы метод Ньютона сошелся к минимуму. Но с тем же успехом он может сойтись к оказавшемся рядом максимуму, совершив шаг в направлении возрастания функции вместо шага, уменьшающего значение функции.

Квази-Ньютоновские методы решают эту проблему следующим образом: вместо гессиана используется его положительно определенная аппроксимация. Если гессиан положительно определен, то мы совершаем шаг по методу Ньютона. Если гессиан знаконеопределен, то перед совершением шага по методу Ньютона мы модифицируем гессиан так, чтобы он был положительно определен.

Смысл данного подхода в том, что шаг всегда совершается в направлении убывания функции. В случае, если гессиан положительно определен, мы используем его для построения квадратичной аппроксимации поверхности, что должно ускорить сходимость. Если гессиан знаконеопределен, то мы просто движемся в направлении убывания функции.

Выше было сказано, что мы совершаем шаг по методу Ньютона. На самом деле это не совсем так - таким образом мы лишь определяем направление, в котором будет совершаться шаг. Некоторые модификации квази-Ньютоновских методов проводят вдоль указанной прямой точный линейный поиск минимума, но доказано, что

достаточно добиться лишь существенного уменьшения значения функции, а искать точный минимум не обязательно. Данный алгоритм сначала пытается совершить шаг по методу Ньютона, а если он не приводит к уменьшению значения функции, то ищется шаг в том же направлении, меньший по величине и уменьшающий значение минимизируемой функции.

L-BFGS схема обновления гессиана

Гессиан функции доступен далеко не всегда, гораздо чаще мы можем вычислить только градиент функции. Поэтому используют следующую схему работы: на основе N последовательных вычислений градиента строится гессиан функции и совершается квази-Ньютоновский шаг. Существует специальная формула, позволяющая итеративно получать аппроксимацию гессиана, причем на каждом шаге аппроксимирующая матрица остается положительно определенной. В данном алгоритме используется BFGS-схема обновления, названная по первым буквам имен Broyden-Fletcher-Goldfarb-Shanno (если быть точным, то эта формула строит не сам гессиан, а обратную к нему матрицу; таким образом, не надо тратить время на её обращение).

Буква L в названии схемы происходит от слов "limited memory". В случае больших размерностей объем памяти порядка N^2, требуемый для хранения гессиана, оказывается слишком большой нагрузкой, также как и затраты машинного времени на его обработку. Поэтому вместо использования N значений градиента для построения гессиана можно обойтись меньшим числом значений, позволяющим использовать объем памяти порядка $N \cdot M$. Обычно на практике M выбирают в промежутке от 3 до 7, в сложных случаях можно увеличить эту константу до 20. Разумеется, в результате такой экономии мы получим не сам гессиан, а лишь его аппроксимацию. С одной стороны, при этом замедляется сходимость. С другой, скорость работы может даже вырасти. На первый взгляд парадоксальное, это утверждение не содержит противоречий: сходимость измеряется числом итераций алгоритма, в то время, как скорость работы - числом тактов процессора, потраченных на вычисления.

Вообще-то говоря, изначально этот метод разрабатывался для оптимизации функций очень большого числа аргументов (сотни и тысячи), поскольку именно в этом случае увеличение числа итераций

из-за пониженной точности аппроксимации гессиана полностью окупается снижением накладных расходов на обновление гессиана, однако нет причин, по котором этот метод нельзя применять для задач малой размерности. Основным его достоинством является масштабируемость, поскольку он обеспечивает высокое быстродействие на задачах большой размерности, при этом позволяя решать и задачи малой размерности.

Разностные схемы и аналитический градиент

Если известен градиент функции, то алгоритму требуется намного меньше итераций для сходимости, чем методам, не использующим информацию о градиенте. Одно значение градиента в плане информативности эквивалентно N значениям функции, так что такое различие в быстродействии вполне объяснимо. Вместе с тем, многое зависит от того, как именно вычисляется градиент. Если градиент вычисляется по разностной схеме, то уменьшение числа итераций будет компенсировано пропорциональным ростом их трудоемкости из-за использования разностной схемы. Если градиент известен в аналитической форме и эффективно вычисляется, то L-BFGS алгоритм будет значительно быстрее.

> **Замечание**
> Не вычисляйте градиент функции на основе двухточечной разностной формулы - она недостаточно точна. В ряде случаев алгоритм просто не сможет работать, и завершится с сообщение об ошибке. Используйте хотя бы четырехточечную формулу.

ALGLIB User Guide - Одномерная и многомерная оптимизация - Метод Левенберга-Марквардта

Метод Левенберга-Марквардта

Метод Левенберга-Марквардта – хороший выбор, если вам требуется минимизировать функцию вида $F=f_1^2(x_1,\ldots,x_n)+\ldots$

$+f_m{}^2(x_1,..,x_n)$. Алгоритм удачно сочетает в себе метод наискорейшего спуска (т.е. минимизации вдоль градиента) и метод Ньютона (т.е. использование квадратичной модели для ускорения поиска минимума функции). От метода наискорейшего спуска алгоритм позаимствовал стабильность работы, от метода Ньютона – ускоренную сходимость в окрестностях минимума. Ниже приведено обсуждение стандартной реализации алгоритма Левенберга-Марквардта, её недостатков, и улучшенной версии алгоритма, входящей в пакет ALGLIB. Перед чтением этого раздела рекомендуется ознакомиться с описанием алгоритма в Википедии или в Numerical Recipes. Далее мы предполагаем, что читающий понимает общие принципы работы алгоритма Левенберга-Марквардта.

Применение метода Левенберга-Марквардта

Солвер для нелинейного МНК

Наиболее часто встречающееся применение метода Левенберга-Марквардта - решение задач нелинейной регрессии. В принципе, ничто не мешает вам использовать для этого интерфейс, представляемый субпакетом minlm, и рассмотренный ниже. Однако в пакете ALGLIB существует специальный интерфейс для решения таких задач, входящий в состав субпакета lsfit. Использование специализированного интерфейса обычно более удобно, чем работа с методом оптимизации напрямую.

Оптимизатор функции, представленной в виде суммы квадратов

Вторым, также часто встречающимся применением метода Левенберга-Марквардта является оптимизация функций, которые могут быть представлены в виде суммы квадратов:

$$F(x_0,..,x_{N-1}) = \sum_{i=0}^{M-1} f_i^2(x_0,..,x_{N-1})$$

Хотя минимум такой $F(x)$ может быть найден с использованием алгоритмов для функций общего вида (нелинейный CG или L-BFGS),

метод Левенберга-Марквардта позволяет использовать знание внутренней структуры для более быстрой сходимости к минимуму.

Оптимизатор функции обшего вида

Последним, менее известным применением метода Левенберга-Марквардта является оптимизация функций *общего* вида, т.е. функций, которые не разлагаются на сумму квадратов более простых функций. Метод Левенберга-Марквардта имеет смысл применять, если нам доступен Гессиан функции $F(x)$ и мы хотим использовать его для оптимизации.

Начало работы с методом Левенберга-Марквардта

Выбор режима оптимизации

В зависимости от того, какая информация о функции доступна, алгоритм может использоваться в следующих вариантах:

- **V** (function **v**ector). Функция $F(x)$ представлена, как сумма квадратов. Доступен только вектор функций f. Якобиан вычисляется с использованием численного дифференцирования и метода секущих.
- **VJ** (**v**ector+**J**acobian). Функция $F(x)$ представлена, как сумма квадратов. Доступны вектор функций f и Якобиан J.
- **FGH** (**f**unction+**g**radient+**H**essian). Функция $F(x)$ имеет общий вид. Нам доступны значение $F(x)$, градиент G и Гессиан H.

Буквы в названии схемы являются суффиксом, который дописывается к имени подпрограммы `minlmcreate`, использующейся для создания оптимизатора. Так, пользователям ALGLIB доступны следующие подпрограммы: minlmcreatev, minlmcreatevj, minlmcreatefgh.

> **Замечание**
> Также доступен дополнительный вариант алгоритма, который можно использовать для задач с разреженным Якобианом - **VGJ** (**v**ector+**g**radient+**J**acobian). В этом

режиме алгоритм использует вектор функций, Якобиан, а также градиент функции $F(x)$, равный произведению $f^T J$. Этот режим имеет смысл использовать в сочетании со второй стратегией ускорения сходимости (см. ниже).

Какую же схему следует выбрать? Для быстрого старта мы рекомендуем начать со схемы **V** (подпрограмма minlmcreatev), потому что она наиболее проста в использовании. От вас требуется только вектор функций f, и не требуется Якобиан. Вы просто пишете код, вычисляющий значение функции, а пакет ALGLIB берет на себя вопросы, связанные с численным дифференцированием.

Следующий шаг. Итак, вы убедились, что пакет ALGLIB (и ваш код для вычисления функции) работают нормально. Как мы уже говорили, оптимизация без использования Якобиана очень проста в реализации, но не очень эффективна. Кроме того, численное дифференцирование не позволяет найти минимум с точностью, существенно превышающей шаг дифференцирования. Если вам требуется хорошее быстродействие (или высокая точность), то имеет смысл реализовать вычисление аналитического Якобиана и перейти к схеме **VJ.**

Замечание
Если вы осуществляете оптимизацию функции общего вида (с использованием Гессиана), то вам придется сразу начинать со схемы **FGH** и реализовать всё - функцию, градиент, Гессиан.

Выбор критериев остановки

Пакет ALGLIB предлагает пользователям четыре критерия остановки:

- после снижения градиента $F(x)$ до заданной величины
- после совершения достаточно малого шага
- после достаточно малого изменения функции на последнем шаге
- по достижению предельного числа итераций

Вы можете установить один или несколько критериев в различных сочетаниях с использованием функции minlmsetcond. После того, как

алгоритм завершит свою работу, вы можете проанализировать код завершения и определить, какой именно критерий сработал.

Мы настоятельно рекомендуем использовать первый критерий - малое значение градиента $F(x)$. Этот критерий гарантирует, что алгоритм остановится только в достаточно хорошей точке, независимо от того, насколько медленно или быстро мы к ней приближаемся. Критерии, связанные с изменением шага или функции, менее надежны, так как в некоторых случаях алгоритм может совершать небольшие шаги даже вдали от минимума (например, так иногда бывает при оптимизации без использования Якобиана).

> **Замечание**
> В общем случае нельзя гарантировать, что сработает именно тот критерий остановки, который вы установили. Например, алгоритм может сделать шаг, который приведет нас точно в минимум функции, и тогда сработает критерий, связанный с нулевым значением градиента - независимо от того, какие критерии были установлены. Возможны и другие ситуации, когда срабатывает не тот критерий, который вы установили (например, из-за погрешностей операций с плавточкой).

Запуск алгоритма и получение результатов

После того, как объект-оптимизатор создан и настроен, вы можете запустить процесс оптимизации путем вызова функции minlmoptimize. Аргументами функции являются оптимизатор и callbacks, вычисляющие оптимизируемую функцию/градиент. Результат работы может быть получен при помощи вызова minlmresults.

Примеры

ALGLIB Reference Manual содержит ряд примеров, посвященных оптимизации с использованием алгоритма Левенберга-Марквардта:

- пример minlm_d_v, который демонстрирует оптимизацию без использования производных

- пример minlm_d_vj, который демонстрирует оптимизацию с использованием Якобиана
- пример minlm_d_fgh, который демонстрирует оптимизацию по схеме **FGH**
- пример minlm_d_restarts, который демонстрирует использование быстрых рестартов

В этих примерах рассмотрено несколько наиболее типичных способов использования оптимизатора. Вы можете скопировать код примера в свою среду разработки, запустить его, проанализировать результаты, попробовать внести свои изменения. Мы рекомендуем ознакомиться с этими примерами перед тем, как вы начнете писать свой код, использующий ALGLIB.

Улучшая быстродействие

Быстрый перезапуск

Если вы последовательно решаете ряд задач с одними и теми же характеристиками (размерность, параметры оптимизатора), то вы можете создавать новый объект-оптимизатор каждый раз, когда вы приступаете к решению новой задачи. Однако создание оптимизатора - трудоемкий процесс, в котором активно используется динамическое выделение памяти. Более эффективным решением является использование функции minlmrestartfrom, которая позволяет перезапустить уже созданный оптимизатор с новой позиции без повторного выделения памяти.

Ускорение сходимости

Оригинальный алгоритм Левенберга-Марквардта предполагает построение квадратичной модели функции и совершение шага, после чего модель отбрасывается и мы строим новую квадратичную модель. Именно такой алгоритм используется по умолчанию в схемах **VJ** и **FGH**. Однако построение квадратичной модели с нуля может быть очень трудоемким процессом, что приводит нас к первой стратегии ускорения сходимости.

Первая стратегия ускорения сходимости состоит в том, что после совершения шага мы не вычисляем Якобиан заново, а обновляем его по методу секущих, используя значения функций (не производных) в новой точке. Обновленный Якобиан менее точен, и качество следующего шага будет меньше, но он все же приведет к уменьшению значения функции. В итоге мы совершаем больше шагов (и решаем больше систем линейных уравнений), но меньшее количество раз вычисляем Якобиан. Очевидно, что такая стратегия хороша, если стоимость вычисления Якобиана размером MxN высока - существенно выше, чем стоимость разложения Холецкого матрицы размером NxN.

Эта стратегия включается вызовом `minlmsetacctype(state,1)` и может быть использована вместе с любой схемой оптимизации, включающей использование вектора значений функции (**V**, **VJ**, **VGJ**). Она включена по умолчанию при использовании схемы **V**. В этом случае Якобиан вычисляется с использованием численного дифференцирования, что является трудоемкой процедурой, и использование первой стратегии всегда оправдано. В прочих случаях эта стратегия должна быть явно включена функцией minlmsetacctype.

Вторая стратегия ускорения сходимости диаметрально противоположна первой. Вспомним, что стоимость шага по методу Левенберга-Марквардта складывается из двух составляющих: вычисления Якобиана и решения системы линейных уравнений (трудоемкость $O(M \cdot N^2)$). Первая стратегия разработана для случая, когда вычисление Якобиана является дорогостоящей операцией - существенно более дорогой, чем решение системы линейных уравнений. Вторая стратегия разработана для противоположной ситуации - вычисление Якобиана является дешевой операцией с трудоемкостью $O(N \cdot M)$. В этом случае естественным является минимизировать количество систем линейных уравнений, которые нам надо решать, и повторно использовать квадратичную модель, даже если это приведет к дополнительным вычислениям Якобиана.

Для того, чтобы достичь этой цели, мы чередуем итерации Левенберга-Марквардта и предобусловленного L-BFGS алгоритма. В качестве предобуславливателя мы используем квадратичную модель, построенную на предыдущем шаге. В качестве целевой функции - $F(x)$. Градиент $F(x)$, необходимый для работы L-BFGS алгоритма, мы получаем одним из двух способов:

- через вычисление произведения $2 \cdot f^T \cdot J$. Этот способ не требует дополнительной информации (кроме вектора функции и Якобиана).
- через запрос градиента у пользователя. Этот способ имеет смысл использовать, если Якобиан разрежен и произведение $2 \cdot f^T \cdot J$ может быть вычислено более эффективно, чем через формирование матрицы J и умножение на вектор f.

Эта стратегия включается вызовом `minlmsetacctype(state,2)` и может быть использована вместе с любой схемой оптимизации, при которой доступны Якобиан или градиент (**VJ**, **VGJ**, **FGH**).

ALGLIB User Guide - Одномерная и многомерная оптимизация –

Метод активных множеств

ASA-алгоритм

Метод активных множеств (ASA) - это общее название семейства алгоритмов для решения задачи оптимизации с ограничениями вида $g_i(x) \geq 0$. Название метода происходит от используемой классификации ограничений, в соответствии с которой они делятся на активные и неактивные в текущей точке. Ограничение неактивно, если $g_i(x) > 0$. Если же $g_i(x)=0$, то ограничение может быть как неактивным, так и активным (в зависимости от выбора множества активных ограничений).

Наиболее общая формулировка метода активных множеств включает две чередующиеся стадии. На первой стадии активные ограничения интерпретируются, как ограничения вида равенства, после чего решается (приближенно) задача оптимизации со смешанными ограничениями (равенства и неравенства). На второй стадии принимается решение об активации или деактивации ограничений (обычно в зависимости от знака множителей Лагранжа). Неформально говоря, текущая точка путешествует по множеству допустимых x, "прилипая" к границам и "отлипая" от них.

Основным достоинством метода является простота его реализации для задачи с ограничениями вида $a_i \le x_i \le b_i$. Активация ограничений состоит в "замораживании" компонент x, что позволяет использовать практически любой алгоритм оптимизации без ограничений. Итерации метода могут быть очень дешевыми, т.к. отсутствует необходимость строить сложные квадратичные модели функции и ограничений.

Реализация в пакете ALGLIB

В пакете ALGLIB реализована незначительная модификация алгоритма, описанного в 'A new active set algorithm for box constrained optimization' (William W. Hager and Hongchao Zhang). Этот алгоритм чередует итерации нелинейного метода сопряженных градиентов и метода проекции градиента. Первый алгоритм позволяет добиться хорошей сходимости после того, как найдено подходящее множество ограничений. Второй алгоритм используется для активации или деактивации ограничений и позволяет активировать за одну итерацию сразу несколько ограничений. Метод обладает глобальной сходимостью при условии, что $grad(f)$ непрерывен по Липшицу на множестве $L = \{ x : f(x) \le f(x_0) \}$. Одним из достоинств является сравнительно низкая стоимость итераций, умеренно отличающаяся от стоимости итераций метода сопряженных градиентов без ограничений.

Быстродействие

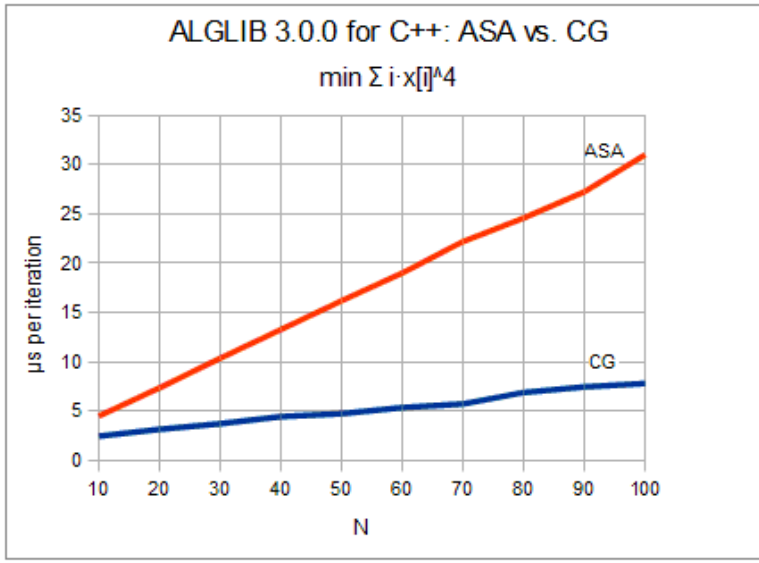
ASA против CG на задачах без ограничений

В этом эксперименте мы сравним стоимость итерации ASA со стоимостью итерации классического метода сопряженных градиентов. Стоимость итерации ASA складывается из двух составляющих: стоимости лежащего в основе метода сопряженных градиентов и накладных расходов, связанных с обработкой ограничений. Для того, чтобы выделить составляющую, связанную с собственно обработкой ограничений, мы решим с помощью обоих алгоритмов следующую задачу:

- минимизируемая функция: $f(x) = x0^4 + 2 \cdot x1^4 + ... + (n+1) \cdot xn^4$.

- стартовая точка: $xs = [10, ..., 10]$.
- ограничения: $-100 \le x_i \le +100$, т.е. в минимуме все ограничения неактивны.
- размерность задачи n: в диапазоне 10...100 с шагом 10.
- алгоритмы: CG и ASA

Для тестирования использовался компьютер с процессором Intel Core 2, тактовой частотой 2.4 GHz. По итогам тестирования были получены следующие результаты:



Четерехкратный прирост длительности итерации показывает, насколько дорого обходится обработка ограничений. Но действительно ли ASA в четыре раза медленнее CG? В худшем случае - да. Однако в нашем примере была выбрана очень простая функция f, стоимость вычисления градиента которой невелика в сравнении со стоимостью итерации любого из используемых методов. В практических задачах время, требуемое для вычисления градиента f, может на порядок превосходить время работы собственно алгоритма. На этом фоне накладные расходы, связанные с обработкой ограничений, могут оказаться незаметны, и быстродействие ASA будет практически равно быстродействию CG в аналогичной задаче, но без ограничений.

Список обозначений

$\mathbf{R}^n$ — n-мерное вещественное евклидово пространство.

$\{x_1, ..., x_n\}$ — компоненты вектора $x \in \mathbf{R}^n$.

$\| \bullet \|$ — норма в $\mathbf{R}^n$: $\|x\|^2 = x^2_1 + ... + x^2_n$.

$(\bullet, \bullet)$ — скалярное произведение в $\mathbf{R}^n$: $(x, y) = x_1y_1 + ... + x_ny_n$

I — единичная матрица

A^{T} — матрица, транспонированная к A.

A^+ — псевдообратная матрица к A.

$A \geq B$ — матрицы A и B симметричны и $A - B$ неотрицательно определена

$A > B$ — матрицы A и B симметричны и $A—B$ положительно определена.

$\|A\|$ — норма матрицы A: $\|A\| = \max\limits_{\|x\|=1} \|Ax\|$.

$\rho(A)$ — спектральный радиус матрицы A .

$x \geq y$ — все компоненты вектора $x \in \mathbf{R}^n$ не меньше соответствующих компонент вектора $y \in \mathbf{R}^n$: $x_i \geq y_i$, $i = 1, ..., n$.

$\mathbf{R}^n{}_+$ — неотрицательный ортант в $\mathbf{R}^n$: $\mathbf{R}^n{}_+ = \{ x \in \mathbf{R}^n: x \geq 0\}$.

x_+ — положительная часть вектора $x \in \mathbf{R}^n$: $(x_+)_i = \max \{0, x_i\}$, $i = 1, ... n$.

$x^* = \arg\min\limits_{x \in Q} f(x)$ —любая точка глобального минимума $f(x)$ на Q: $x \in Q$, $f(x^*) = \min\limits_{x \in Q} f(x)$.

$X^* = Arg\min\limits_{x \in Q} f(x)$ — множество точек глобального минимума $f(x)$ на Q: $X^* = \{x^* = \arg\min\limits_{x \in Q} f(x)\}$.

$\nabla f(x)$, $f'(x)$ — градиент скалярной функции $f(x)$.

$\nabla g(x)$, $g'(x)$ — производная векторной функции $g(x)$, матрица Якоби.

$\nabla^2 f(x)$, $f''(x)$—матрица вторых производных, гессиан.

$L'_x(x, y)$, $L''_{xx}(x, y)$ — градиент и матрица вторых производных $L(x, y)$ по переменной x.

$df(x)$ — субградиент выпуклой функции.

$\partial_\varepsilon f(x)$ — ε-субградиенг выпуклой функции.

$f'(x; y)$ — производная функции $f(x)$ в точке x по направлению y.

$D(f)$ — область определения функции $f(x)$.

Conv Q — выпуклая оболочка множества Q.

Q — внутренность множества Q.

$\varnothing$ — пустое множество.

$P_Q(x)$ — проекция точки x на множество Q .

$\rho(x, Q)$ — расстояние от точки x до множества Q: $\rho(x,Q)=\inf\limits_{y\in Q} \| x-y \|$

$o(h(x))$ — если g: $\mathbf{R}^n\to\mathbf{R}^m$, h: $\mathbf{R}^n\to\mathbf{R}^s$ и $\|g(x)\|/\|h(x)\| \to 0$ при $\|x\|\to 0$.

$O(h(x^*))$ — если g: $\mathbf{R}^n\to\mathbf{R}^m$, h: $\mathbf{R}^n\to\mathbf{R}^S$ и найдутся $\varepsilon > 0$, α такие, что $\|g(x)\| \le \alpha \| h(x)\|$ при $\|(x)\| \le \varepsilon$, то $g(x) = O(h(x))$.

$o(u_k)$ — если последовательности $u_k \in \mathbf{R}^n$, $v_k \in \mathbf{R}^m$, $k = 1, 2,\ldots$, таковы, что $\| v_k \|/\| u_k \| \to 0$ при $k\to\infty$, то $v_k = o(u_k)$.

$O(u_k)$ — если для последовательностей $u_k \in \mathbf{R}^n$, $v_k \in \mathbf{R}^m$, $k = 1, 2,\ldots$, найдутся $\alpha > 0$, k_0 такие, что $\| v_k \| \le \alpha \| u_k \|$ при $k \ge k_0$, то $v_k = O(u_k)$.

$M\xi$ — математическое ожидание случайной величины ξ

$M(\xi | x)$ — условное математическое ожидание случайной величины ξ, зависящей от x, при фиксированном значении x.

$\forall$ — квантор общности: $\forall x \in Q$ — «для всех $x \in Q$».

Литература

1.Основная

1. Аоки М. Введение в методы оптимизации. — М.: Паука, 1977.
2. Бахвалов Н. С. Численные методы.—М.: Наука, 1973.
3. В а й н б е р г М. М. Вариационный метод и метод монотонных операторов в теории нелинейных уравнений —М.: Наука, 1972.
4. Габасов Р., Кириллова Ф. М. Методы оптимизации. — Минск: Б ГУ, 1975.
5. Демьянов В. Ф., Рубинов А М. Приближенные методы решения экстремальных задач —Л.: ЛГУ, 1968.
6. Зангвилл У. Нелинейное программирование. Единый подход --М.: Сов. Радио, 1973.
7. Зойтендейк Г. Методы возможных направлений —М.: ИЛ, 1963.
8. Карманов В. Г. Математическое программирование. — М.: Наука, 1975.
9. Кононюк А.Ю. Вища математика. К.1. — К.: КМТ, 2009.
10.Кононюк А.Ю. Вища математика. К.2. — К.: КМТ, 2009.
11.Кононюк А.Е. Дискретная математика. К.1, ч.1 — К.: Освіта України, 2010.
12.Кононюк А.Е. Дискретная математика. К.1, ч.2 — К.: Освіта України, 2010.
13.Кононюк А.Е. Дискретная математика. К.2, ч.1 — К.: Освіта України, 2011.
14.Кононюк А.Е. Дискретная математика. К.2, ч.2 — К.: Освіта України, 2011.
15.Кононюк А.Е. Дискретная математика. К.2, ч.3 — К.: Освіта України, 2011.
16.Кононюк А.Е. Дискретная математика. К.3, ч.1 — К.: Освіта України, 2011.

17.Кононюк А.Е. Дискретная математика. К.3, ч.2 — К.: Освіта України, 2011.
18. М о и с е е в Н. Н., И в а н и л о в Ю. П., Столярова Е. М. Методы оптимизации.—М.: Наука, 1978.
19. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными — М.: Мир, 1975.
20. Полак Э. Численные методы оптимизации. Единый подход — М: Мир, 1974.
21. Поляк Б. Т. Введение в оптимизацию. —М.: Наука, 1983.
22. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах.— М.: Наука, 1975.
23. Растригин Л. А. Системы экстремального управления —М.: Наука, 1974.
24. С е а Ж. Оптимизация. Теория и алгоритмы.— М.: Мир, 1973
25. Уайлд Д. Дж. Методы поиска оптимума.—М.: Науки 1967.
26. Федоренко Р. П. Приближенное решение задач оптимального управления. — М.: Наука, 1978.
27. Фиакко А., Мак-Кормик Дж. Нелинейное программирование: методы последовательной безусловной минимизации.—М: Мир 1972.
28. Хи м м е л ь б л а у Д. Прикладное нелинейное программирование М.: Мир, 1975.
29. Численные методы условной оптимизации /Под ред. Ф. Гилла, У Мюр рея. — М.: Мир, 1977.
30. Э р р о у К. Дж., ГурвицЛ.УдзаваХ. Исследования по лпигП ному и нелинейному программированию. — М.: ИЛ, 1962.

2. Дополнительная

1. *Абакаров А.Ш., Сушков Ю.А.* Статистическое исследование одного алгоритма глобальной оптимизации. — Труды ФОРА, 2004.
2. *Акулич И.Л.* Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. пец. вузов. — М.: Высшая школа, 1986.
3. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. Пер. с англ. — М.: Мир, 1985.
4. *Жиглявский А.А., Жилинкас А.Г.* Методы поиска глобального экстремума. — М.: Наука, Физматлит, 1991.
5. *Карманов В.Г.* **Математическое программирование** = Математическое программирование. — Изд-во физ.-мат. литературы, 2004.

6. *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. — М.: Наука, 1970. — С. 575-576.
7. *Коршунов Ю.М., Коршунов Ю.М.* Математические основы кибернетики. — М.: Энергоатомиздат, 1972.
8. *Максимов Ю.А.,Филлиповская Е.А.* Алгоритмы решения задач нелинейного программирования. — М.: МИФИ, 1982.
9. *Максимов Ю.А.* Алгоритмы линейного и дискретного программирования. — М.: МИФИ, 1980.
10. *Огирко И. В.* Расчет и оптимизация термоупругого состояния тел с учетом геометрической и физической нелинейности : Автореф. дис. на соиск. учен. степ. д-ра физ.-мат. наук : (01.02.04) / Казан. гос. ун-т им.— Казань, 1989.
11. *Плотников А.Д.* Математическое программирование = экспресс-курс. — 2006. — С. 171. — ISBN 985-475-186-4
12. *Растригин Л.А.* Статистические методы поиска. — М.: 1968.
13. *Хемди А. Таха* Введение в исследование операций = Operations Research: An Introduction. — 8 изд.. — М.: «Вильямс», 2007. — С. 912. — ISBN 0-13-032374-8
14. Никайдо Х. Выпуклые структуры и математическая экономика. — М.: Мир, 1972
15. Кини Р. Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения.- М.: Радио и связь, 1981
16. Соболь И. М., Статников Р. Б. Выбор оптимальных параметров в задачах со многими критериями. — М.: Наука, 1981
17. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. — М.: Наука, 1982
18. Морозов В. В., Сухарев А. Г., Федоров В. В. Исследование операций в задачах и упражнениях. — М.: Высшая школа, 1986
19. Юдин Д. Б. Вычислительные методы теории принятия решений. — М.: Наука, 1989
20. Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. Лекции по теории графов. — М.: Наука, 1990
21. Штойер Р. Многокритериальная оптимизация. — М.: Радио и связь, 1992
22. Батищев Д. И., Коган Д. И. Вычислительная сложность экстремальных задач переборного типа. — Изд. ННГУ, Н. Новгород, 1994
23. Коротченко А. Г., Тихонов В. А. Методические указания (сборник задач) по курсу «Модели и методы принятия решений» — Изд. ННГУ, Н. Новгород, 2000

24. Коротченко А. Г., Бобков А. Н. Принципы оптимальности в задачах принятия решений (методическая разработка) — Изд. ННГУ, Н. Новгород, 2002
25. Батищев Д. И. Задачи и методы векторной оптимизации. — Изд. ГГУ, Горький, 1979
26. Розен В. В. Цель- оптимальность- решение: Математические модели принятия оптимальных решений. — М.: Радио и связь, 1982
27. Батищев Д. И. Методы оптимального проектирования. — М.: Радио и связь, 1984

28. Г. М. Уланов и др. Методы разработки интегрированных АСУ промышленными предприятиями. М.: Энергоатомиздат – 1983.
29. А. М. Анохин, В. А. Глотов, В.В. Павельев, А.М. Черкашин. Методы определения коэффициентов важности критериев “Автоматика и телемеханика”, №8, 1997, с3-35.
30. Таха, Хэмди А. Введение в исследование операций – М.:Мир,2001, с354-370.
31. Р. Штойер. Многокритериальная оптимизация: теория, вычисления, приложения. М.:Наука, 1982, с14-29, 146-258.
32. Многокритериальная оптимизация. Математические аспекты. М.:Наука, 1989, с116-123.
33. В.В. Подиновский, В.Д. Ногин. Парето-оптимальные решения многокритериальных задач. М.: Наука, 1982, с9-64.
34. В. В. Хоменюк. Элементы теории многокритериальной оптимизации. М.: Наука, 1983, с8-25.
35. Д.И.Батищев, С.А.Исаев, Е.К.Ремер. Эволюционно-генетический подход к решению задач невыпуклой оптимизации. /Межвузовский сборник научных трудов «Оптимизация и моделирование в автоматизированных системах», Воронеж, ВГТУ, 1998г, стр.20-28.
36. Д.И.Батищев, С.А.Исаев. Оптимизация многоэкстремальных функций с помощью генетических алгоритмов. /Межвузовский сборник научных трудов «Высокие технологии в технике, медицине и образовании», Воронеж, ВГТУ, 1997г, стр.4-17.
37. С.А.Исаев. Популярно о генетических алгоритмах. Интернет-ресурс http://bspu.ab.ru./Docs/~saisa/ga/ga-pop.html.
38. С.А.Исаев. Обоснованно о генетических алгоритмах. Интернет-ресурс http://bspu.ab.ru/Docs/~saisa/ga/text/part1.html.
39. С.А.Исаев. Решение многокритериальных задач. Интернет-ресурс http://bspu.ab.ru/Docs/~saisa/ga/idea1.html.
40. Раздел «Математика\Optimization Toolbox». Интернет-ресурс http://www.matlab.ru/optimiz/index.asp.

41. Система СИМОП для автоматизации выбора рациональных решений в комплексах САПР и АСНИ. Интернет-ресурс. http://www.software.unn.ac.ru/mo_evm/research/symop.html

42. Интегрированный пакет многокритериальной оптимизации «МАЛТИ». Интернет-ресурс http://ksu.kst.kz/emf/kafkiber.htm

43. Комплексный инженерный анализ - прочность, динамика, акустика. Интернет-ресурс http://osp.admin.tomsk.ru/ap/1998/02/31.htm

44. Программы семейства COSMOS – универсальный инструмент конечно-элементного анализа. Интернет-ресурс http://cad.com.ru/7/Info/cosmos_3.html

Научно-практическое издание

Кононюк Анатолий Ефимович

Основы теории оптимизации

Книга 2
Безусловная оптимизация

Авторская редакция

Подписано в печать 21.03.2011 г.
Формат 60x84/16.
Усл. печ. л. 16,5. Тираж 300 экз.

Издатель и изготовитель:
Издательство «Освита Украины»
04214, г. Киев, ул. Героев Днепра, 63, к. 40
Свидетельство о внесении в Государственный реестр
издателей ДК №1957 от 23.04.2009 г.
Тел./факс (044) 411-4397; 237-5992
E-mail: osvita2005@ukr.net, www.rambook.ru