**Методы оптимизации**

*Оптимизация* – выбор наилучшего решения.

Сложность или невозможность отыскания аналитического решения привело к тому, что постепенно стало ясно, что любая задача может считаться решенной, если *указан алгоритм, позволяющий численно построить приближенное решение с требуемой точностью*.

*Математическая теория оптимизации* включает в себя фундаментальные результаты и численные методы, позволяющие находить наилучший вариант из множества возможных альтернатив *без их полного перебора* и сравнения.

Итак, есть варианты решения задачи, среди которых надо найти лучший. Как оценить какой метод лучше? Надо определить критерий качества.

*Критерий качества* – функционал, действующий из множества вариантов решения задачи в множества вещественных чисел. Тогда понятие хуже - лучше тождественно больше - меньше. Один вариант лучше другого, если, например, значение функционала меньше, и неопределенность теряется.

У одной и той же задачи часто бывает возможным наличие нескольких функционалов качества. При этом нахождение их экстремума оказывается сложным, и выбраться из этой ситуации можно за счет методов многокритериальной оптимизации.

В этом курсе мы будем заниматься поиском экстремума одного функционала.

Итак, рассмотрим *функционал* *ϕ:* *X*→ *R̅*, где

*X* – множество вариантов или допустимое множество (область определения функционала);

 – расширенная вещественная прямая.

Пусть *c* ⊂ *X* – некоторое подмножество *X***.**

Задача:  называется *экстремальной задачей с ограничением* *c*. (экстремум – максимум или минимум)

**Терминология и классификация**

*C ≠ X*

*C = X*

Минимизация функции (безусловная минимизация)

Математическое программирование (условная минимизация)

*X = Rn*

*Методы оптимизации*

(исследование операций)

[более широкий термин]

Экстремальные задачи

Вариационное исчисление

Оптимальное управление

*X* – множество функций

*X* [более сложная структура]

**Этапы решения оптимизационной задачи**

Процесс принятия решения в исследовании операций представляет собой сложный процесс, который условно можно разбить на 4 этапа:

***1 этап***: Построение качественной модели рассматриваемой проблемы, т.е. выделение факторов, которые представляются наиболее важными, установление закономерностей, которым они подчиняются.

***2 этап***: Построение математической модели, включающей в себя выбор функционала *ϕ* (или целевой функции переменных), *ϕ*(*x*) → min(max), формирование ограничений (условий) в виде равенств или неравенств, например

.

Этот этап требует привлечения математических знаний и характеризуется, как правило, большим количеством переменных (*n* и *m* – велики).

***3 этап***: Решение математической задачи – выбор метода, реализация его и получение результата (применение ЭВМ, разработка программ, применение существующих СП и т.д.).

***4 этап***: Анализ полученного результата. Выясняется степень адекватности модели (результаты вычислений) и моделируемого объекта (имитационные данные).

**Примеры математических моделей**

Вообще, теория математических моделей является предметом специализированного курса и требует знаний в той области, которой принадлежит моделируемый объект. Рассмотрим традиционные примеры, иллюстрирующие применение метода математического моделирования в задачах экономического содержания.

*Задача о рационе*: Пусть имеется *n* – число продуктов питания и *m* – число питательных веществ.

Пусть  – содержание j-го вещества в единице i-го продукта;

 – минимальная (суточная) потребность (человека) в j-ом веществе;

 – стоимость единицы i-го продукта;

 – искомое количество (суточное потребление) i-го продукта.

Тогда  – общее содержание j-го питательного вещества;

 – стоимость (суточного) рациона.

|  |  |
| --- | --- |
| *Задача*:  найти min  (или тот набор, (*x*1,…,*xn*) при котором достигается минимум)  при условии, что | Типичная задача линейного программирования |

*Транспортная задача:* Требуется составить план перевозок однородного груза таким образом, чтобы общая стоимость перевозок была минимальной.

Пусть  – количество единиц груза в i-ом пункте отправления ();

 – потребность в j-ом пункте назначения () в единицах груза;

 – стоимость перевозки единицы груза из i-го пункта в j-ый;

 – планируемое количество единиц груза для перевозки из i-го пункта в j-ый.

Тогда  – общая (суммарная) стоимость перевозок;

 – количество груза, вывозимого из i-го пункта;

 – количество груза, доставляемого в j-ый пункт.

В простейшем случае должны выполняться следующие очевидные условия:

.

Таким образом, математическая формулировка транспортной задачи имеет вид:

Найти:

,

при условиях

.

Задача носит название *замкнутой транспортной модели*, а условие  является естественным условием разрешимости замкнутой транспортной задачи.

**Обозначения и определения**

*Rn* – *n*-мерное евклидово пространство.

 – вектор столбец в *Rn*;

 – вектор-строка в *Rn*;

 – скалярное произведение, *x*, *y*∈*Rn*;

 – евклидова норма вектора в *Rn*;

 – матрица,  – транспортированная матрица;

 – произведение матрицы (*m*×*n*) на вектор (*n*×1);

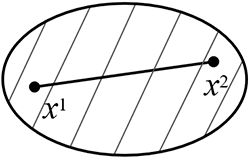
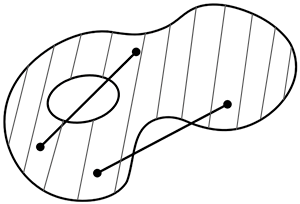
*ϕ*(*x*), *f*(*x*), *g*(*x*),… – как правило, вещественные (скалярные) функции, т.е. *ϕ*: *Rn* → *R*;

– градиент функции *ϕ* в точке *x*0 (*n*-мерный вектор);

|  |  |
| --- | --- |
|  | – матрица Гессе (матрица вторых производных) функции *ϕ* в точке *x*0; |

Т.к. , то *H*(*x*0) – есть вещественная симметричная матрица.

**Определение.** Множество *X*⊂ *Rn* называется *выпуклым*, если для ∀*x*1, *x*2∈*X*, ∀*λ*∈[0,1] . Иными словами, множество *X* выпукло, если оно вместе с любыми своими двумя точками *x*1 и *x*2 содержит соединяющий их отрезок.

* *

Выпуклое множество Невыпуклое множество

**Примеры.**

На числовой прямой *R* выпуклыми множествами являются всевозможные промежутки, т.е.:

* одноточечные множества;
* интервалы;
* полуинтервалы;
* отрезки;
* полупрямые;
* сама прямая.

В пространстве *Rn* примерами выпуклых множеств служат:

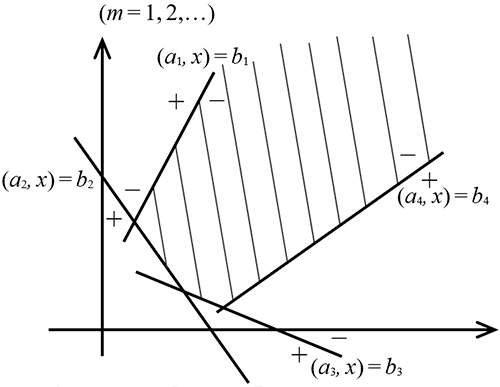
* само подпространство;
* любое его линейное подпространство;
* одноточечное множество;
* шар;
* отрезок,
* а также следующие множества:

 – прямая, проходящая через (⋅) *x*0 в направлении вектора *h*.

 – луч, выходящий из (⋅) *x*0 в направлении *h*.  – гиперпространство с нормалью *p*.

 – порождаемые ею полупространства.

Все перечисленные множества в *Rn*, кроме шара, являются частными случаями *выпуклого множества вида*:

,

где *A* – некоторая матрица размера *m*×*n* со строками ,

*b* ⊂ *Rm* – вектор.

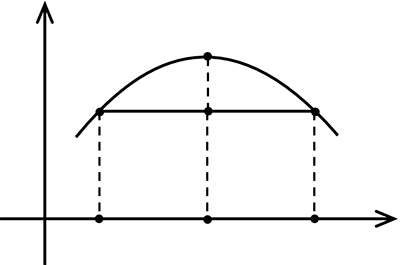
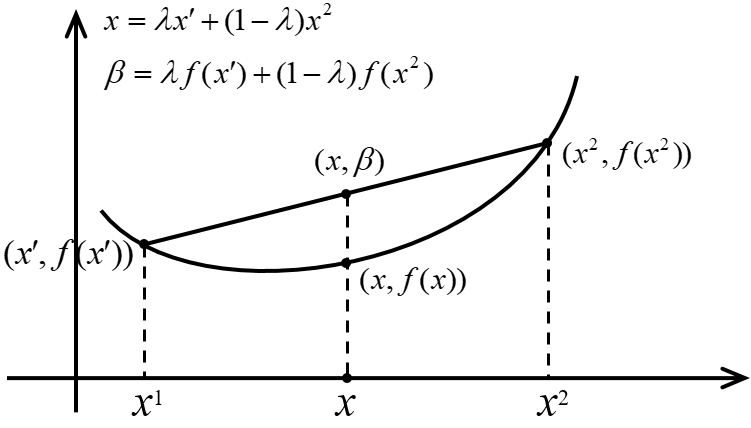
Множества такого вида называют *полиэдральными* или *полиэдрами*. Таким образом, полиэдр – множество решений некоторой системы конечного числа линейных неравенств (пересечение конечного числа полупространств).

**Определение.** Функция *f*, определенная на выпуклом множестве *X*⊂ *Rn*,называется *выпуклой на* *X*, если

,

при ∀*x*1, *x*2∈*X*, *λ*∈[0,1].

Если для любых  неравенство выполняется как строгое, то *f* называется строго выпуклой на *X*. Функция называется (строго) вогнутой, если функция –*f* (строго) выпукла.



Функцию , где *a*∈*Rn*, *b*∈*R* будем называть линейной. Ясно, что для неё исходное неравенство выполняется как равенство. Поэтому она выпукла и вогнута одновременно, но не строго.

#### Минимизация функций

Сама по себе постановка задачи оптимизации проста и естественна: заданы множество *X* и функция *ϕ*(*x*), определенная на *X*, требуется найти точку минимума или максимума функции *ϕ* на *X*.

Условимся записывать задачу на минимум в виде

, где

*ϕ* – *целевая функция*; *X* – *допустимое множество*.

Условимся также, что в дальнейшем будем рассматривать задачу на min, поскольку задача

.

Если допустимое множество *X = Rn*, то задача называется *безусловной* минимизацией, иначе, когда *X*≠ *Rn* – задача *условной* минимизации.

Отметим, что само понятие точки минимума неоднозначно и требует уточнения.

**Определение.** Точка *x*\*∈*X* называется:

1) точкой глобального минимума функции *ϕ* на множестве *X*, или глобальным решением задачи, если ;

2) точкой локального минимума *ϕ* на *X*, или локальным решением задачи, если существует

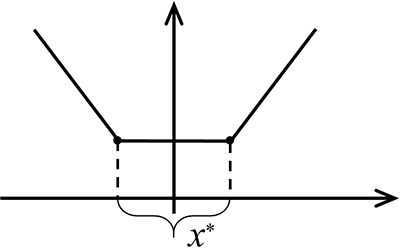
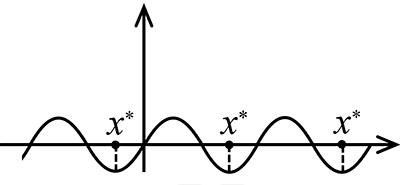


(Если  – *δ*-окрестность (⋅) *x*\* и *x*\* – локальный min, то *x*\* – глобальный min в области *X*∩ Δ).

Если неравенства в 1) и 2) выполняются как строгие, то говорят, что *x*\* – точка строгого min в глобальном или локальном смысле.

Ясно, что глобальное решение является локальным, обратное – не верно.

**Пример.** Глобальных min может быть много:



счетное множество

континуальное множество

Для записи того факта, что *x*\* является точкой глобального min функции *ϕ* на *X* используем запись:



или эквивалентная ей запись:

 – *оптимальная* точка.

Множество всех точек глобального min *ϕ* на *X* обозначим:



Таким образом,  – это произвольная точка из множества .

В дальнейшем мы часто будем прибегать к геометрической интерпретации задач оптимизации, основанной на понятии линий (или поверхностей) уровня функции *ϕ*, т.е. множеств вида:

- такое множество носит название *поверхность уровня α*.

Напомним известный факт из анализа: если функция *ϕ* дифференцируема в точке *x*, то градиент *ϕ*′(*x*) ортогонален к проходящей через *x* линии уровня *α* и направлен (если ) в сторону возрастания функции *ϕ*, т.е. поверхность *Lα* делит *Rn* на два подпространства:

 и .

*Задача* поиска оптимальной точки может быть сформулирована следующим образом: найти *α*\* = min*α* среди тех *α*, для которых *Lα*∩*X*≠ ∅. Тогда любая точка *x*∈*Lα*\* является оптимальной точкой.

Возможны два случая:

* *x\** лежит внутри *X* – рис. 1;
* *x\** лежит на границе *X* – рис. 2.

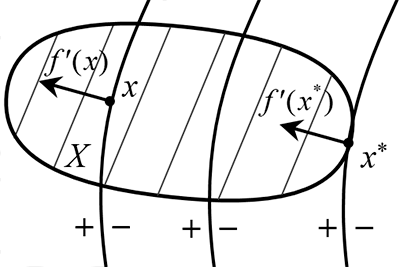
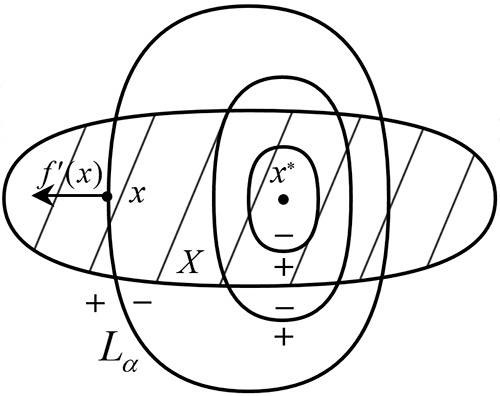


Рис.1

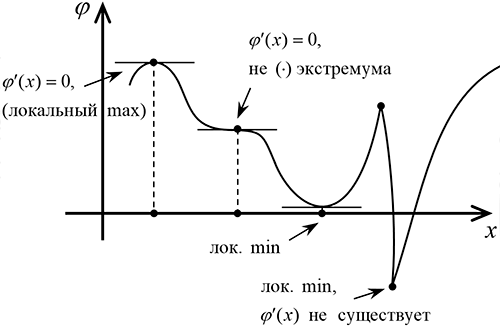
Рис.2

При изучении задач оптимизации в первую очередь возникает вопрос о существовании решения. Напомним в этой связи некоторые результаты из *математического анализа*.

**Теорема 1** (Вейерштрасса).Если *X* – компакт в *Rn* (т.е. замкнутое ограниченное множество), а *ϕ* – непрерывная функция на *X*, то существует *x*\*∈*X*: *x*\* – глобальный минимум *ϕ* на *X*, т.е. *глобальное решение задачи*  *существует*!

**Теорема 2** (необходимые условия локального минимума). Если *ϕ* – дифференцируема в точке *x*\*∈*X* и *x\** – локальный минимум, то *ϕ*′(*x*\*) = 0 (градиент равен нулю).

**Определение.** Точка  в , называется *стационарной* (обратное не верно).

******

**Теорема 3** (достаточное условие локального минимума).

Если ** дважды дифференцируема в точке *x*\*∈*Rn* и выполняется:

1) *ϕ*(*x*\*) = 0;

2) матрица Гессе *ϕ*″(*x*\*) положительно определена,

то *x\** – (строгий) локальный минимум функции **

**Определение.** Матрица *H* называется *положительно определенной*, если .

*Критерий Сильвестра*: *H* положительно определена ⇔ ее главные миноры положительны.

Приведем несколько теорем для выпуклых задач.

**Определение.** Задача минимизации  называется выпуклой, если *X* – выпуклое множество, ** – выпуклая функция на *X*.

Справедливы следующие теоремы:

**Теорема 4.** Если задача минимизации  выпукла, то любое её локальное решение является также глобальным.

**Доказательство.** Пусть *x\** – локальное решение задачи, т.е. при некотором *ε*> 0 выполняется условие:

 при ,

где  – шар радиуса *ε* > 0 с центром в *x\**.

Для любых точек *x*∈*X*: *x* ≠ *x\**, положим .

Тогда . Покажем это.

1. Пусть ,

Если  ⇒ точка 

1. Пусть 



⇒ точка 

и, следовательно,

,

т.е. *x\** – глобальное решение задачи, *ч.т.д.*

Таким образом, *для выпуклых задач* понятия *локального и глобального решений не различаются*.

Второе свойство выпуклых задач можно высказать в виде следующего общего принципа: *необходимые условия оптимальности* в том или ином классе задач минимизации при соответствующих предположениях выпуклости *оказываются и достаточным*.

**Теорема 5.** Пусть функция ** выпукла на *Rn* и дифференцируема в точке *x*\*∈*Rn*. Если *ϕ*′(*x*\*) = 0, то *x*\* – точка минимума функции на *Rn*, т.е. решение задачи минимизации .

**Доказательство.** Для ∀*x*∈*X*, *λ*∈[0,1] имеем

.

Преобразуя эту формулу и, пользуясь дифференцируемостью функции ** в точке *x\**, получаем:

;

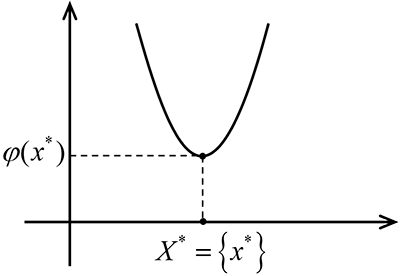
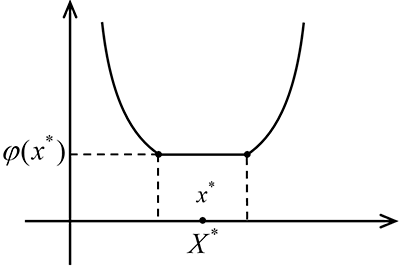
Отсюда предельным переходом при *λ*→ 0 выводим, что , *ч.т.д.* (т.е. для ∀*x*∈*X* ).

Полученные свойства выпуклых задач имеют важное значение не только в теории, но и в численных методах оптимизации. Дело в том, что большинство существующих численных методов позволяет, вообще говоря, находить лишь локальные решения, а точнее – стационарные точки задачи. Теоремы 4 и 5 говорят о том, что для выпуклой задачи *отыскание стационарной точки* автоматически означает *отыскание решения, причем глобального*.

Укажем ещё одно полезное свойство выпуклых задач.

**Теорема 6.** Пусть задача минимизации  выпукла и имеет решение.

Тогда множество её решений  выпукло. Если при этом **(*x*) строго выпукла на *X*, то решение единственно, т.е. *X \**состоит из одной точки.



**Доказательство:**

1. Пусть 

При этом

 (\*)

По определению *X\** неравенство может выполняться только как равенство, поскольку 

, т.е. *X\** – выпукло.

1. Пусть ** – строго выпукла. Если предположить, что в *X\** существуют две различные точки *x*1 и *x*2, то при *λ*∈[0,1] неравенство (\*) должно быть строгим, что невозможно, т.к. *ϕ*\*– min и получается < min.

*Трудности*:

1. В случаях, когда функция ** достаточно проста, теоремы 1-3 помогают решить задачу минимизации даже в явном виде. Однако зачастую задача поиска стационарных точек является нетривиальной. А затем – перебор стационарных точек в поисках точки локального минимума, затем – перебор локальных экстремумов в поисках глобального экстремума.
2. Для задач условной минимизации теоремы 1-3 применимы в случае, когда локальное решение *x\** – внутренняя точка допустимого множества *X*. Если же экстремум достигается в угловых точках границы множества условий, то нарушается дифференцируемость ⇒ неприменимость методов классического анализа.

Т.о., в большинстве случаев задачу min*ϕ*(*x*) приходится *решать численно с применением ЭВМ и специальных методов минимизации*.

**Безусловная минимизация функции**

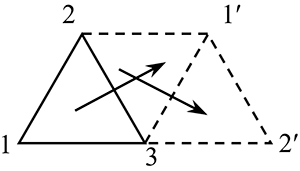
Методы оптимизации функций в *Rn* делятся на:

* локальные методы (поиск локального min, т.е. такой точки *x\**, что существует *δ* > 0, );
* нелокальные (или прямые) методы (поиск глобального min для ограничений снизу функции **(*x*), т.е. если *α\** – нижняя грань, то поиск такой точки *x*\*: *ϕ*(*x*\*) = *α*\*). Для этих методов не требуется аналитического задания функции, надо только уметь вычислять ее значение в любой точке. Обычно – для функций сложной структуры.

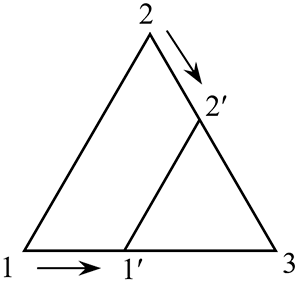
Нелокальные методы сводятся к уменьшению области, внутри которой находится оптимальная точка. Пример нелокального метода – *симплексный метод*.

**Определение.** *Симплекс* – выпуклое тело в *Rn*, состоящее из (*n*+ 1) равноудаленных точек – вершин симплекса, отрезок их соединяющий – ребро симплекса, в *R*2 – треугольник, в *R*3 – тетраэдр.

Неформальное описание симплексного метода: состоит из двух процедур – отражение и сжатие.

– *отражение*: симметричное отражение вершины с наибольшим значением **(*x*) относительно противоположной грани ["перекатывание симплекса"]. Если , то выбирается другая (*i +*1)-я вершина.

Когда зацикливание (все (*n*+ 1)-вершины перебрали), то

– *сжатие*: уменьшение размеров симплекса при сохранении вершины с наименьшим значением **(*x*), затем переход к отражению, и так далее, пока ребро симплекса не станет меньше некоторого числа: .

*Достоинства*: с большой вероятностью метод не распознает локальный минимум ("не остановится").

*Локальные* методы основаны на построении *релаксационной* последовательности {*xi*} такой, что  и .

Поэтому *релаксационные* методы называют также *методами спуска*.

**Классификация релаксационных методов**

С одной стороны,

* *одношаговые* методы:  – каждый шаг (*i*+ 1) зависит только от предыдущей точки *xi* и значения функции *ϕ*(*xi*);
* *двухшаговые* методы:  – зависимость от двух предыдущих точек;
* *и т.д.;*

С другой стороны,

* *методы нулевого порядка:* если используются только значения минимизируемой функции **(*x*);
* *методы первого порядка*: если используются только значение **(*x*) и **′(*x*);
* *методы второго порядка*: если используются значения **(*x*), **′(*x*) и **″(*x*);
* *etc;*

**Градиентные методы (методы первого порядка)**

Итак, будем рассматривать задачу:

 (безусловная минимизация),

предполагая, что функция **(*x*) непрерывно дифференцируема на *Rn*, т.е. *ϕ*(*x*)∈*C*1(*Rn*).

По определению дифференцируемой функции

, (1)

где .

Если , то при достаточно малых  главная часть приращения для ** будет определяться дифференциалом функции . Оценим величину Справедливо неравенство Коши-Буняковского:

,

причем, если *ϕ*′(*x*) ≠ 0, то правое неравенство превращается в равенство, только при *h*=−*αϕ*′(*x*), а левое только при *h*=*αϕ*′(*x*), где *α*= *const*≥ 0.

Отсюда ясно, что при *ϕ*′(*x*) ≠ 0 *направление наибыстрейшего возрастания* функции **(*x*) в точке *x* совпадает с *направлением градиента* **(*x*), а направление наибыстрейшего убывания – с направлением антиградиента –**′(*x*).

Это свойство градиента лежит в основе ряда итерационных методов минимизации функций. Один из таких – *градиентный*. Он предполагает, как, впрочем, и все остальные итерационные методы, наличие априорной точки начального приближения.

Предположим, что начальная точка *x*0 уже выбрана, тогда градиентный метод заключается в построении последовательности {*xk*} по правилу:

 (2)

*αk* – величина шага, *xk* – направление спуска.

Если , то шаг  можно выбрать так, чтобы получить релаксационную последовательность: . Действительно, подставляя (2) в (1), имеем:

,

при всех достаточно малых *αk*> 0.

Если , то *xk* – стационарная точка. В этом случае процесс (2) прекращается и проводятся дополнительные исследования поведения функции в окрестности точки *xk* для выяснения того, достигается ли в точке *xk* минимум функции **(*x*) или не достигается.

Существуют различные *способы выбора величины шага* *­­k* в методе (2). В зависимости от способа выбора *­­k* можно получить различные варианты градиентного метода.

**Метод наискорейшего спуска**

На луче , направленном по антиградиенту, введем функцию одной переменной



и определим *­­k* из условий

.

Другими словами *­­k* выбирается так, чтобы *ϕ*(*xk*+1) в заданном направлении была наименьшей для чего на любом шаге необходимо решать задачу одномерной минимизации функции *ψ* (*α*), например, с помощью .

**Пример.** Рассмотрим задачу



с начальной точкой .

Из общих соображений ясно, что *ϕ*min = 0 при 

*1-й шаг*:



Ищем



Функция *ψ*(*α*) имеет следующий вид:



Решаем уравнение , т.е.

;

.

*2-й шаг*:

;

.

Решаем уравнение  ⇒

; ⇒

.

*3-й шаг*:





Решаем уравнение  ⇒

;

, и.т.д.

*Представим решение задачи графически:*

Из графического представления можно сделать вывод, что имеет место:

|  |  |
| --- | --- |
| C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\15.png | ⇒ а) сходимость к истинной точке минимума  б) взаимная перпендикулярность градиентов |

**Свойства метода наискорейшего спуска**

1. На любом шаге направление спуска меняется на ортогональное. Действительно, *αk* ищется из условия  ⇒



1. Точка *xk*+1 лежит на луче, исходящем из точки *xk* и касательным к поверхности уровня *Lϕ*(*xk*+1). Действительно, с одной стороны, несомненно, что . С другой стороны, градиент *ϕ*′(*xk*+1) ортогонален касательной к поверхности уровня *Lϕ*(*xk*+1), поэтому по свойству 1 направление спуска касательно к поверхности *Lϕ*(*xk*+1).

*Иначе.* *ϕ*′(*xk*+1) ортогонален направлению спуска ⇒ луч, проходящий из точки *xk* – касательной к поверхности .

*Проблемы* (общие для релаксационных методов).

1. Имеет ли последовательность {*xk*} предел в смысле сходимости по норме: существует ?
2. Является ли этот предел аргументом, составляющим минимум функции *ϕ*?
3. Какова скорость сходимости  или *ϕ*(*xk*) – *ϕ*(*x*\*)?
4. Каковы вычислительные затраты.

**Исследование метода наискорейшего спуска для квадратичной функции**

Рассмотрим квадратичную функцию

,

где *A* – симметричная, положительно определенная матрица.

Можно показать, что *A* – симметричная положительно определенная матрица ⇔ *ϕ* – строго выпукла.

, т.е.  – стационарная точка.

Попробуем записать метод наискорейшего спуска для квадратичной функции.

Итак,



 (w)

,

т.к. *A* – положительно определена, и значит для нее справедливо: (*Ah*, *h*) > 0 ∀*h*∈*Rn*≠ 0.

Для определения скорости сходимости оценим отношение



Имеем:



С другой стороны,



Для простоты дальнейших изложений предположим, что матрица *A* приведена к диагональному виду (т.е. выполнено преобразование координат) так, что , где *λi* – собственные числа матрицы *A*.

* Собственные числа симметричной положительно определенной матрицы всегда положительны.
* Для симметричной матрицы существует ортогональная матрица (*TT* = *T*-1) *T* такая, что *TTAT* – диагональная матрица .

Если , то

,

Тогда

.

Если ввести обозначение , то



Это называется *геометрической скоростью сходимости* (сходимость геометрической прогрессии).

Рассмотрим величину

.

Верхний предел  называется *порядком сходимости метода*.

В нашем случае квадратичной функции

.

Поэтому



⇒ 

⇒ получили сходимость с порядком 1 или *линейную сходимость*. Бывает порядок больше 1 – *сверхлинейная сходимость*.

При исследовании метода наискорейшего спуска для квадратичной функции получили, в частности, следующие результаты:

1. 
2. 

**Определение.**

Пусть *ϕ*(*xk*)→*ϕ*(*x*\*) при *k* → ∞.

Последовательность *ϕ*(*xk*) сходится к *ϕ*(*x\**) *линейно* (с линейной скоростью, со скоростью геометрической прогрессии), если существуют такие константы *q*∈(0,1) и *k*0, что

, при *k*≥ *k*0.

Последовательность *ϕ*(*xk*) сходится к *ϕ*(*x\**) *сверхлинейно*, если

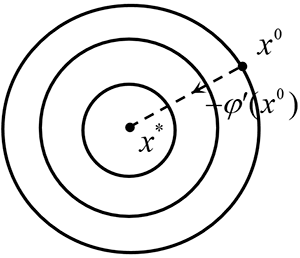
, при *k*→ ∞.

Последовательность *ϕ*(*xk*) сходится к *ϕ*(*x\**) с *квадратичной скоростью*, если существуют такие константы *c*≥ 0 и *k*0, что

, при *k* ≥ *k*0.

Вообще, порядок сходимости, равный 1, означает, что значение величины *k* убывает, в основном, по закону геометрической прогрессии. Порядок сходимости, равный 2 (квадратичная сходимость) означает, что при достаточно больших *k* *k*+1 ~ *k*2. В этом случае, если к тому же *k* – малая величина, например,  при , то *k*+1 равно , т.е. фактически удваивается число нулей после запятой.

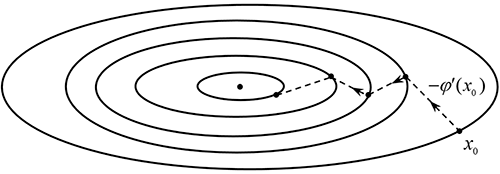
*Частные случаи*:

1. Пусть *l*= *L*, т.е. матрица *A*= ­*LI*= *lI* – пропорциональна единичной окружности (линии уровня – окружности).

Тогда:



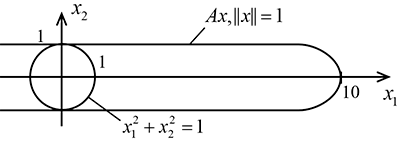
⇒ *f*(*xk*+1) = *f*(*x*\*) метод сходится за один шаг.

1. *l*≤ *L*: сходимость может быть еле заметной (*q*~ 1), а графически это означает, что линии уровня функции сильно вытянуты и функция имеет так называемый "овражный" характер. Это означает, что небольшое изменение некоторых переменных приводит к резкому изменению значений функции – эта группа переменных характеризует "склон оврага", а по остальным переменным, задающим направление "дна оврага", функция меняется незначительно.

Число  называется числом обусловленности матрицы ⇒ *cond* ≥ 1.

Матрица называется *хорошо обусловленной*, если *cond* ~ 1 и наоборот.

Вообще, число обусловленности геометрически можно трактовать как меру искажения отображения матрицей *A* единичной сферы. Действительно, *cond*(*A*) есть отношение наибольшего к наименьшим расстояниям между точками на единичной сфере после её отображения матрицей *A*. Чем больше *cond*(*A*), тем больше искажение единичной сферы при её преобразовании в эллиптическую форму – пусть *A*= *diag*(10,1).

*Вывод*: Метод наискорейшего спуска быстро сходится для хорошо обусловленных матриц и наоборот.

*Почему так много внимания уделяли квадратичной функции?*

В окрестности locmin любую функцию можно приблизить квадратичной, и всё сказанное выше про матрицу *A* будет справедливым для матрицы Гесса *H*(*x*\*), которая заменяет *A* в рассмотренном выше примере.

*Геометрически*: Линии уровня становятся замкнутыми и по мере приближения к *x*\* всё более напоминают эллипс.

*Общий случай.*

**Определение 1.** Функция ** на множестве *X**Rn* удовлетворяет *условию Липшица*, если существует . Если градиент функции ** существует, непрерывен и удовлетворяет условию Липшица, то обозначается ***C*1,1.

**Определение 2.** Функция ** называется *сильно выпуклой с параметром* , если .

**Теорема** (о сходимости метода наискорейшего спуска). Рассмотрим задачу . Пусть ***С*1,1(*Rn*) и ** – сильно выпуклая c параметром æ. Тогда при любом начальном приближении для последовательности {*xk*}, построенной по методу наискорейшего спуска, справедливы соотношения:

1) 

2) 

**Замечания.**

1. Для *квадратичной функции* :

1. постоянная Липшица *L* есть наибольшее собственное число матрицы *A*:

;

1. она сильно выпукла с параметром . Действительно,





2. Эквивалентные *ограничения сильной выпуклости*:

1. *ϕ* – сильно выпукла ⇔  – выпукла (это означает, что ** имеет "квадратичный запас" выпуклости);
2. пусть ***C*1, *ϕ* – сильно выпукла ⇔ ;
3. пусть ***С*2, ** – сильно выпукла ⇔ , ∀*x*, т.е.  [в смысле положительной определенности разности матриц]. С другой стороны, из условия Липшица , поэтому для сильно выпуклой ***С*2 существует двойная оценка матрицы Гессе: .

Покажем, что . С одной стороны, из б) имеем



С другой стороны,



⇒ , *ч.т.д.*

Выпуклость:

* .

Строгая выпуклость:

* .

Сильная выпуклость:

*  для ∀*u*,*υ*∈*Rn*

*Графическое представление дифференциальных критериев выпуклости.*

|  |  |
| --- | --- |
| 21 | График *выпуклой* функции расположен не ниже касательной плоскости , проходящей через произв. точку поверхности . |
| C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\20.png | График *строго выпуклой* функции имеет единственную общую точку с этой плоскостью. |
| C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22.png | Пусть *υ*\* – (⋅)min ⇒ . Поверхность  – это параболоид вращения с вершиной в точке .  ⇒ График *сильно выпуклой* функции расположен внутри некоторого параболоида вращения. |

**Другие градиентные методы**

Напомним, градиентные методы заключаются в построении релаксационной последовательности:



Градиентные методы различаются между собой способом выбора *αk*.

**1.** **Метод наискорейшего спуска**, который был рассмотрен выше, заключается в выборе

.

Такой способ выбора *αk* является в некотором смысле наилучшим, т.к. он обеспечивает *достижение наименьшего значения функции* вдоль заданного направления. Однако он требует решения на любом шаге *одномерной задачи минимизации*. Эти задачи решаются, как правило, приближенно с помощью численных методов, что приводит к *значительному объему вычислений*. Кроме того, метод может привести к *плохой сходимости* (овраги!).

Другим подходом для построения релаксационной последовательности является попытка определить *αk* до начала вычислений. Какие есть для этого основания?

Допустим, что можно построить оценку для *αk* такую, что для *ε*∈(0,1) выполняется неравенство



Тогда очевидно, что  и соответствующий метод минимизации будет методом спуска.

Справедливы следующие утверждения:

##### **Лемма 1.** Пусть функция C1,1(Rn) и



Тогда для *xk**Rn*, **(0,1) условие (1) выполнено при



**Лемма 2.** Пусть ** дважды дифференцируема и матрица Гессе удовлетворяет условию Липшица и



Тогда для *xk**Rn*, **(0,1) условие (1) выполняется при



**2. Градиентный метод с постоянным шагом.**

В этом методе полагается *αk*≡ *const*. При этом иногда удается добиться выполнения условия (1). Но для этого необходимо знать константы *M* и *D*, что далеко не всегда удается вычислить.

Т.о., метод прост в реализации, но есть проблемы со сходимостью.

**Пример.** Пусть **(*x*) = *x*2

**min = 0; *x\** = 0;

Тогда

 ⇔ метод сходится.

Сходимость *медленная*!

**3. Градиентный метод с убыванием длины шага.**

В ряде методов достаточно потребовать выполнения условий:

 (например, )

На интуитивном уровне объяснение следующее:

* условие сходимости ряда  накладывают, чтобы добиться достаточно быстрой сходимости последовательности *k* к нулю с целью обеспечения сходимости метода в окрестности точки экстремума *x*\*.
* условие расходимости ряда  призвано обеспечить достижение точки экстремума *x*\* даже при неудачном выборе начального приближения *x*0, т.е. при больших расстояниях от *x*0 до *x*\*.

Сходимость *медленная*!

**4. Градиентный метод с дроблением шага.**

Ещё один адаптивный способ выбора коэффициентов *αk*. Выбираются некоторые *constβ *>0 и 0< *λ*< 1 (обычно *λ*= ½). Для коэффициента *α* = ** проверяется выполнение условия . Если оно выполняется, то полагают *k*= **. Если нет, то производится дробление шага, т.е. принимается *α* = **, и т.д. до тех пор, пока не выполнится требуемое неравенство.

Процесс дробления не может продолжаться бесконечно, поскольку −*ϕ*′(*x*) – направление убывания функции. Первое *α*, при котором условие выполнено и принимается за *αk*.

Как показывает следующая лемма, с помощью описанного процесса дробления шага можно добиться выполнения неравенства. (1)

**Лемма 3.** Пусть функция *ϕ* дифференцируема на *Rn*. Тогда для  найдется такое *α*0 > 0, что при ∀*α*∈(0, *α*0] выполнено условие

.

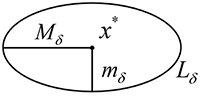
Если необходимое неравенство оказывается выполненным при начальном значении *α*= *β*, то иногда полезно увеличить шаг, взяв *α* = *μβ*, где *μ*> 1. Так можно продолжать до тех пор, пока значения функции не перестанут уменьшаться. Последнее *α*, при котором произошло уменьшение, и берется в этом случае за *αk*.

**5. Овражный метод** – эвристический двухшаговый метод минимизации овражных функций.

*Характеристика степени овражности*:

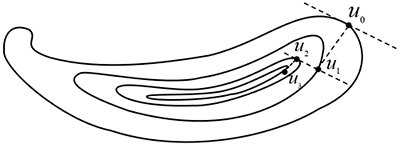
Пусть *x*\* – точка минимума, *δ*> 0.

Рассмотрим поверхность уровня .

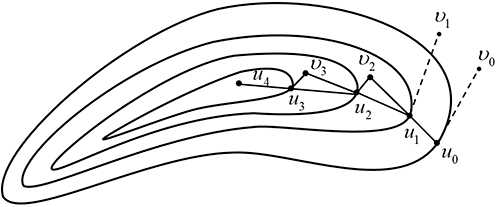
Введем 



**Определение.** Тогда  называется *числом обусловленности точки locmin* .

****Рассмотрим "овражную" функцию (вытянута вдоль некоторых направлений). Если точка лежит на склоне оврага, то направление спуска из этой точки будет почти перпендикулярно к направлению "дна оврага", и в результате приближения {*uk*}, получаемые градиентным методом, будут *поочередно находиться то на одном, то на другом "склоне оврага"*. Если "склоны оврага" достаточно круты, то такие скачки "со склона на склон" точек {*uk*} могут сильно замедлить сходимость градиентного метода.

Для ускорения сходимости можно предложить следующий эвристический прием, называемый *овражным методом*:

Пусть *υ*0, *υ*1 – две произвольные близкие точки. Совершаем из них по одному шагу методом наискорейшего спуска (или ∀ вариант градиентного метода).

Попадаем в окрестность "дна оврага". Соединяя их прямой, делаем большой шаг в полученном направлении, перемещаясь вдоль "дна оврага". Из полученной точки *υ*2, которая находится на "склоне оврага", производят спуск с помощью градиентного метода и определяют следующую точку *u*2 на "дне оврага" и т.д.

Формула метода выглядит следующим образом.



Здесь:

 определяет знак - чтобы спускаться, а не подниматься;

 - определяет направление спуска по дну оврага;

*h -* овражный шаг, выбирается эмпирически и от него многое зависит.

Если *h* – большое, то на крутых склонах точки *υk* могут *слишком далеко удаляться от "дна оврага"* ⇒ большие объемы вычислений для градиентного метода спуска в очередную точку на "дне оврага", кроме этого может произойти выброс точки *υk* из "оврага" и *правильное направление поиска будет потеряно*.

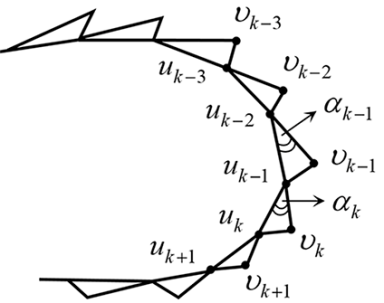
Если *h* – малое, то эффект от применения овражного метода может быть незначительным.

*Эффективность* применения овражного метода может *резко возрасти*, если величину *h* выбирать переменной, реагирующей на "повороты" оврага, с тем, чтобы:

1. быстрее проходить прямолинейные участки на "дне оврага" за счет увеличения овражного шага;
2. на крутых поворотах "оврага" избежать выброса из "оврага" за счет уменьшения овражного шага;
3. добиться min отклонения точки *υk* от дна оврага с целью уменьшения объема вычислений для градиентного метода.

Для правильной реакции на "повороты" оврага надо учитывать "кривизну" оврага.

Причем информацию о кривизне желательно получить по результатам предыдущих шагов.

Один из способов выбора шага:

,

где *k* – угол между векторами , определяемый

условием

,

*c* − *const* > 1 – параметр алгоритма.

 возрастает при возрастании кривизны ⇒ при переходе от участка с меньшей кривизной на участок с большей кривизной имеем

 и наоборот.

На участках с постоянной кривизной



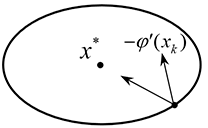
⇒ шаг остается постоянным, который был сформирован при выходе на рассматриваемый участок.

Параметр *с* регулирует чувствительность "метода к изменению кривизны (повороты)" и во многом определяет скорость движения "по оврагу".

# Методы II порядка минимизации функции

(использование вторых производных)

*Общая идея*:



направление спуска по градиентному методу

желаемое направление

(надо "довернуть" направление спуска)

Последовательность {*xk*} будем строить по формулам:

,

где *γk* – длина шага, *Hk* – матрица поворота (*n*×*m*).

Как выбрать матрицу *Hk*?

Если взять квадратичную функцию

,

то хочется сразу попасть в экстремальную точку:



⇒ в качестве "матрицы доворота" надо брать .

*В общем случае*: пусть *ϕ* – дважды дифференцируема в *Rn*, разложим *ϕ*(*x*) в ряд Тейлора в точке

.

Иначе формулу можно представить в виде:

, где  – квадратичная функция.

Пренебрегаем  и ищем .

;

Пусть *ϕ*″(*xk*) – положительно определена для ∀*xk*∈*Rn* ⇒ существует [*ϕ*″(*xk*)-1].

Решая это уравнение, получим:

 – это и есть метод Ньютона.

Квадратичная функция с положительно определенной *ϕ*″ сильно выпукла, тогда уравнение определяет единственную точку глобального минимума .

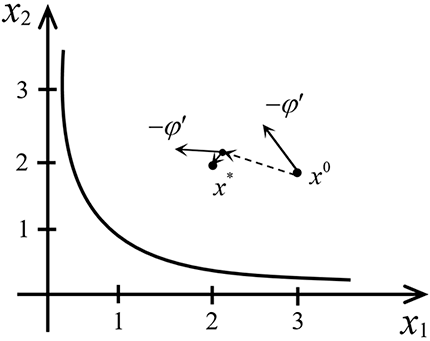
Далее рассмотрим пример использования метода Ньютона для решения задачи минимизации функции.

*Пример.*

Область существования (*ϕ*″(*x*))-1 совпадает с областью положительной определенности матрицы *ϕ*″(*x*), которую мы будем искать с помощью критерия Сильвестра.

***Критерий Сильвестра.***

Симметрическая матрица *A* положительно определена ⇔ если все её главные миноры положительны. При этом главным минором матрицы *A* называется определитель матрицы, построенной из элементов матрицы *A*, стоящих на пересечении строк и столбцов с одинаковыми номерами.

.

Возьмем 









 , и.т.д.

Можно показать, что сходимость при  будет хуже.

*Достоинства метода Ньютона:*

1. Для квадратичной функции сходится за один шаг (метод Ньютона можно рассмотреть, как градиентный метод с преобразованием координат [умножение на *H*-1] таким, что исчезает "овраг", т.е. линии уровня становятся окружностями).
2. Высокая скорость сходимости. Можно показать, что

.

Порядок сходимости:



⇒ важен выбор *q* в алгоритме. Если *q*=10-1, то за один шаг точность результата увеличивается на 2 разряда, а при линейной сходимости – на один разряд.

*Недостатки метода Ньютона:*

1. Локальная сходимость (матрица Гессе должна быть невырожденной). Начальное приближение надо выбирать в окрестности точки локального минимума.
2. Большие вычислительные затраты:

* вычисление матрицы *ϕ*″;
* необходимость обращать её.

*Общие рекомендации*: сначала применять градиентный метод, затем – метод Ньютона.

Например, существует так называемый метод Марквардта-Левенберга:

.

*При больших* *αk* (вдали от *x*\*)матрица и это фактически градиентный метод.

*При малых* *αk* (вдали от *x*\*) это метод Ньютона.

Имеет место:

**Теорема** (о сходимости метода Ньютона).

Пусть

1) *ϕ* – сильно выпукла на *Rn* с параметром æ.

2) *ϕ*∈*c*2,2, т.е. *ϕ*″ – дважды дифференцируема и *ϕ*″ удовлетворяет условию Липшица:

;

3) Начальное приближение *x*0 удовлетворяет условию

,

т.е. , где 0 < *q* < 1.

Тогда последовательность  сходится к точке минимума *x*\* с квадратичной скоростью:

 (квадратичная сходимость)

1. Несколько слов о норме матрицы:

**Определение.** Норму (*n*×*n*)-матрицы определим следующим образом:

,

Тогда

.

Поскольку *ATA* есть симметричная (*n*×*n*)-матрица, то существует ортогональная матрица  такая, что  – диагональная матрица  ⇒

, где  – наибольшее собственное значение матрицы *ATA*.

 – наименьшее собственное значение матрицы *ATA*.

Для такой нормы выполняются все три условия

1. , если *A* – ненулевая (покомпонентно);
2. ;
3. , где *α* – скаляр.

Кроме того, из определения нормы матрицы следует, что

.

Имеем также

.

2. Отметим ещё раз, что для сходимости метода Ньютона начальная точка *x*0 должна выбираться достаточной близкой к искомой точке *x\**. Это требование в теореме выражено условием 3). Действительно, сильная выпуклость *ϕ* означает:

;

⇒ чем меньше *q*, тем ближе надо выбирать точку *x*0 к *x\** и тем быстрее сходимость.

***Достоинства и недостатки градиентных методов и метода Ньютона***

|  |  |  |
| --- | --- | --- |
| **Метод** | **Достоинства** | **Недостатки** |
| Градиентный метод | 1. Глобальная сходимость, т.е. слабые требования на начальные приближения точки х0 и к *f*(*x*);  2. Относительная простота вычислений | 1. Медленная сходимость (геометрическая скорость сходимости, порядок сходимости *d*=1);  2. Необходимость вычисления длины шага. |
| Метод Ньютона | 1. Быстрая сходимость. | 1. Локальная сходимость (начальная точка должна быть близка к *х\**); 2. Большой объем вычислений (на любом шаге требуется вычислять матрицу вторых производных и обращать её). 3. Жесткие требования на саму функцию (непрерывная вторая производная). |

*Порядок применения методов*

1. На 1-м этапе – методы 1-ого порядка, т.к. они обеспечивают глобальную сходимость.
2. На 2-м этапе (когда приращения невелики) – выгодно применять методы 2-ого порядка.

Перечисленные методы (градиентные и Ньютона) являются классическими.

Можно предложить методы более высокого порядка, тогда естественно ожидать, что порядок сходимости будет равен *p*.

Как уже отмечалось, к недостаткам метода Ньютона относится:

* локальная сходимость;
* большой объем вычислений;
* жесткие требования на гладкость функции.

В силу названных причин применение классического метода Ньютона не всегда приводит к успеху.

Многочисленные модификации направлены на то, чтобы, *сохраняя* основное достоинство метода Ньютона – *быструю сходимость*, *уменьшить трудоемкость и ослабить* требования на выбор *начального приближения*.

**Метод Ньютона с регулировкой шага**

Рассмотрим метод

,

это метод Ньютона с регулировкой шага.

При *αk* ≡ 1 мы получили классический метод Ньютона.

Выбор *αk* обычно – из условия min функции, вдоль заданного направления, или методом дробления шага, обеспечивающего выполнение условия *ϕ*(*xk*+1) < *ϕ­*(*xk*).

Можно показать, что подобные методы регулировки шага *сходятся* при *любой начальной* точке *x*0∈*Rn*, причем скорость сходимости будет либо *сверхлинейна*, либо *квадратичная* в зависимости от требований, которым удовлетворяет функция *ϕ*.

Таким образом, с помощью регулировки длины шага преодолевается недостаток метода, связанный с необходимостью отыскания хорошего начального приближения.

Однако, трудоемкость вычислений при этом не исчезает.

Более перспективным в этом плане оказывается другой подход, при котором строится аппроксимация матрицы (*ϕ*″(*xk*))-1 на основе информации о значениях градиентов *ϕ*′(*xk*), *ϕ*′(*xk*+1),…

**Квазиньютоновы методы**

Пусть функция *ϕ* дважды дифференцируема. Рассмотрим метод

 (1)

*αk* – шаг, *Hk* – матрица.

1. Если *Hk* = единичная, имеем градиентный метод.
2. Если , то это метод Ньютона (с точностью до шага).
3. Если , то имеем метод, который объединяет достоинства обоих методов.

Заметим, что

.

Полагая невырожденной матрицу *ϕ*″(*xk*+1), отсюда с точностью до членов более высокого порядка малости по сравнению с  имеем:

.

Рассмотрим квадратичную функцию . Для нее

,

и приближенное равенство обращается в точное:

.

Поэтому естественно потребовать, чтобы для матрицы *Hk*+1, приближающей (*ϕ*″(*xk*+1))-1, выполнялось условие:

 (\*)

Это условие называется *квазиньютоновским*. Оно лежит в основе целого ряда методов аппроксимации (*ϕ*″)-1. Соответствующие методы минимизации, для которых на любом шаге выполняется квазиньютоновское условие, также называются квазиньютоновскими.

Пусть приближения к (*ϕ*″)-1 пересчитываются шаг от шага по формуле .

Различные квазиньютоновские методы различаются способом вычисления "добавки" Δ*Hk* таким образом, чтобы удовлетворялось соотношение (\*).

**1. Метод Дэвидона-Флетчера-Пауэлла**

Обозначим:



Метод заключается в построении релаксационной последовательности по следующему правилу:

 (1.1)

Длина шага *αk* в квазиньютоновых методах выбирается так же, как в методе наискорейшего спуска:



Как правило, начальное значение *H*0 = *I*. Вообще, если *H*0 ­– симметричная матрица, то *Hk* ­– симметричная матрица для любого *k*.

**2. Метод Бройдена-Флетчера-Шанно**

Имеем.

Если поставить задачу уточнять обратную матрицу, т.е.  тогда:

 (1.2)

(этот метод более устойчив к ошибкам округления)

Можно доказать, что для квадратичной функции , где *A* – симметричная, положительно определенная матрица, оба метода (1.1) и (1.2) при любом начальном приближении *x*0∈*Rn* генерируют одну и ту же последовательность точек , причем

,

т.е. *квазиньютоновские методы* позволяют найти min *квадратичной* функции *за n-шагов*.

Для неквадратичной функции, это не так. Однако можно показать, что при соответствующих предположениях , причем скорость сходимости сверхлинейна.

Так, например, пусть *ϕ* – дважды непрерывно дифференцируемая функция, сильно выпукла на *Rn*.

Тогда при любом начальном приближении *x*0∈*Rn* последовательность точек {*xk*}, определяемая формулами (1.1.) и (1.2), сходится к *x*\*.

Если, при этом, для всех *x*:, справедливы неравенства

,

то *xk* сходится к *x*\* сверхлинейно

.

*Замечания о квазиньютоновских методах*:

1. Это двухшаговые методы.
2. Для квадратичных функций сходятся за *n*-шагов.
3. Обладают следующими преимуществами:

* небольшая вычислительная сложность;
* более глобальная сходимость, чем в методе Ньютона;
* сверхлинейная скорость сходимости.