

16. О численных методах многомерной оптимизации.

Задача многомерной безусловной оптимизации формулируется в виде:

$$\min_{x \in X} f(x),$$

где $x = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ – точка в n -мерном пространстве $X = \mathbb{R}^n$, то есть целевая функция $f(x) = f(x^{(1)}, \dots, x^{(n)})$ – функция n аргументов.

Так же как и в первой лабораторной работе мы рассматриваем задачу минимизации. Численные методы отыскания минимума, как правило, состоят в построении последовательности точек $\{x_k\}$, удовлетворяющих условию $f(x_0) > f(x_1) > \dots > f(x_n) > \dots$. 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 принято делить на классы, в зависимости от максимального порядка производных минимизируемой функции, вычисление которых предполагается. Так, методы, использующие только значения самой целевой функции, относят к методам нулевого порядка (иногда их называют также методами прямого поиска); если, кроме того, 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.

1.2. Градиентные методы. 1.2.1. Общая схема градиентного спуска.

Как известно, градиент функции в некоторой точке x_k направлен в сторону наискорейшего локального возрастания функции и перпендикулярен линии уровня (поверхности постоянного значения функции $f(x)$, проходящей через точку x_k). Вектор, противоположный градиенту $f'(x_k)$, называется антиградиентом, который направлен в сторону наискорейшего убывания функции $f(x)$. Выбирая в качестве направления спуска p_k антиградиент $-f'(x_k)$ в точке x_k , мы приходим к итерационному процессу вида:

$$x_{k+1} = x_k - \alpha_k f'(x_k), \alpha_k > 0, k=0, 1, 2, \dots$$

В координатной форме этот процесс записывается

$$x_{k+1}^{(i)} = x_k^{(i)} - \alpha_k \frac{\partial f}{\partial x^{(i)}}(x_k), i = 1, 2, \dots, n.$$

следующим образом:

Все итерационные процессы, в которых направление движения на каждом шаге совпадает с антиградиентом функции, называются градиентными методами. Они отличаются друг от друга только способом выбора шага α_k . Существует много различных способов выбора α_k , но наиболее распространены: метод с постоянным шагом, метод с дроблением шага и метод наискорейшего спуска.

1.2.2. Градиентный метод с постоянным шагом.

Основная проблема в градиентных методах – это выбор шага α_k . Достаточно малый шаг α_k обеспечивает убывание функции, то есть выполнение неравенства:

$$f(x_k - \alpha_k f'(x_k)) < f(x_k),$$

но может привести к неприемлемо большому количеству итераций, необходимых для достижения точки минимума. С другой стороны, слишком большой шаг может вызвать неожиданный рост функции (невыполнение условия убывания) либо привести к колебаниям около точки минимума. Однако проверка условия убывания на каждой итерации является довольно трудоемкой, поэтому в методе градиентного спуска с постоянным шагом задают $\alpha = \alpha_k$ постоянным и достаточно малым, чтобы можно было использовать этот шаг на любой итерации. При этом приходится мириться с возможно большим количеством итераций. Утешением является лишь то, что трудоемкость каждой итерации, в этом случае, минимальна (нужно вычислять только градиент $f'(x_k)$).

Схема алгоритма

Шаг 1.

Задаются начальное приближение x_0 , постоянный шаг α , условия останова алгоритма ε_3 . Вычисляется значение градиента $f'(x_k)$ – направление поиска. Присваивается $k=0$.

Шаг 2.

Определяется точка очередного эксперимента:

$$x_{k+1} = x_k - \alpha f'(x_k),$$

или, в координатной форме:

Шаг 3.

$$x_{k+1}^{(i)} = x_k^{(i)} - \alpha \frac{\partial f}{\partial x_i}(x_k^{(1)}, \dots, x_k^{(n)}), \text{ где } i = 1, 2, \dots, n.$$

Вычисляется значение градиента в точке x_{k+1} :

$$f'(x_{k+1}),$$

или, в координатной форме:

$$\left\{ \frac{\partial f}{\partial x_1}(x_{k+1}^{(1)}, \dots, x_{k+1}^{(n)}), \dots, \frac{\partial f}{\partial x_n}(x_{k+1}^{(1)}, \dots, x_{k+1}^{(n)}) \right\}$$

Шаг 4.

Если $\|f'(x_{k+1})\| \leq \varepsilon_3$, то поиск заканчивается, при этом:

Иначе $k=k+1$ и переходим к шагу 2. $x_{k+1}, \tilde{y} = f(x_{k+1})$.

1.2.3. Градиентный метод с дроблением шага.

В методе градиентного спуска с дроблением шага величина шага α_k выбирается так, чтобы было выполнено неравенство:

$$f(x_k - \alpha_k f'(x_k)) - f(x_k) \leq -\delta \alpha_k \|f'(x_k)\|^2,$$

где $0 < \delta < 1$ – произвольно выбранная постоянная (одна и та же для всех итераций). Это требование на выбор шага α_k более жесткое, чем условие убывания, но имеет тот же смысл: функция должна убывать от итерации к итерации. Однако при выполнении неравенства функция будет уменьшаться на гарантированную величину, определяемую правой частью неравенства.

Процесс выбора шага протекает следующим образом. Выбираем число $\alpha > 0$, одно и то же для всех итераций. На k -й итерации проверяем выполнение неравенства при $\alpha_k = \alpha$. Если оно выполнено, полагаем $\alpha_k = \alpha$ и переходим к следующей итерации. Если нет, то шаг α_k дробим, например уменьшаем каждый раз в два раза, до тех пор, пока оно не выполнится.

Схема алгоритма

Шаг 1.

Задаются x_0 , ε_3 , δ и начальное значение шага α .

Вычисляется значение градиента $f'(x_0)$ – направление поиска.

Присваивается $k=0$.

Шаг 2.

$$\alpha_k = \arg \min_{\alpha \geq 0} f(x_k - \alpha f'(x_k))$$

Проверяется условие: $f(x_k - \alpha f'(x_k)) - f(x_k) \leq -\delta \alpha \|f'(x_k)\|^2$. Если выполняется, то переходим к шагу 3, иначе дробим значение α ($\alpha = \alpha/2$) и повторяем шаг 2.

Шаг 3.

Определяется точка очередного эксперимента: $x_{k+1} = x_k - \alpha f'(x_k)$.

$$\tilde{x} = x_{k+1}, \tilde{y} = f(x_{k+1}).$$

Шаг 4.

Вычисляется значение градиента в точке x_{k+1} : $f'(x_{k+1})$.

Шаг 5.

Если $\|f'(x_{k+1})\| \leq \varepsilon_3$, то поиск заканчивается, при этом:

Иначе $k=k+1$ и переходим к шагу 2.

$$\tilde{x} = x_{k+1}, \tilde{y} = f(x_{k+1}).$$

$$x_1 = x_0 - \alpha_0 \frac{\partial f}{\partial x_1}(x_0) e_1,$$

1.2.4. Метод наискорейшего спуска.

В градиентном методе с постоянным шагом величина шага, обеспечивающая убывание функции $f(x)$ от итерации к

$$x_2 = x_0 - \alpha_0 \frac{\partial f}{\partial x^2}(x_0) e_2,$$

итерации, оказывается очень малой, что приводит к необходимости проводить большое количество итерации для достижения точки минимума. Поэтому методы спуска с переменным шагом являются более экономными. Алгоритм, на каждой итерации которого шаг α_k выбирается из условия

$$f(x_k - \alpha_k f'(x_k)) = \min_{\alpha \geq 0} f(x_k - \alpha f'(x_k)).$$

минимума функции $f(x)$ в направлении движения, то есть: называется методом наискорейшего спуска. Разумеется, этот способ выбора α_k сложнее ранее рассмотренных вариантов.

Реализация метода наискорейшего спуска предполагает решение на каждой итерации довольно трудоемкой вспомогательной задачи одномерной минимизации. Как правило, метод наискорейшего спуска, тем не менее, дает выигрыш в числе машинных операций, поскольку обеспечивает движение с самым выгодным шагом, ибо решение задачи одномерной минимизации связано с дополнительными вычислениями только самой функции $f(x)$, тогда как основное машинное время тратится на вычисление ее градиента $f'(x_k)$.

Следует иметь в виду, что одномерную минимизацию можно производить любым методом одномерной оптимизации, что порождает различные варианты метода наискорейшего спуска.

Схема алгоритма

Шаг 1.

Задаются x_0, ϵ_3 . Вычисляется градиент $f'(x_0)$, направление поиска.

Присваивается $k=0$.

Шаг 2.

Определяется точка очередного эксперимента:

$$x_{k+1} = x_k - \alpha_k f'(x_k),$$

где α_k – минимум задачи одномерной минимизации:

Шаг 3.

Вычисляется значение градиента в точке x_{k+1} :

$$f'(x_{k+1}).$$

Шаг 4.

Если $\|f'(x_{k+1})\| \leq \epsilon_3$, то поиск точки минимума заканчивается и полагается:

Иначе $k=k+1$ и переход к шагу 2.

1.3. Метод покоординатного спуска.

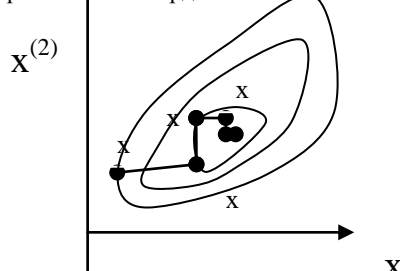
Желание уменьшить объем вычислительной работы, требуемой для осуществления одной итерации метода наискорейшего спуска, привело к созданию методов покоординатного спуска.

Пусть $x_0 = \{x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(n)}\}^T$ начальное приближение. Вычислим частную производную по первой координате и примем:

где $e_1 = \{1, 0, \dots, 0\}^T$ – единичный вектор оси $x^{(1)}$. Следующая итерация состоит в вычислении точки x_2 по формуле:

где $e_2 = \{0, 1, 0, \dots, 0\}^T$ – единичный вектор оси $x^{(2)}$ и т. д.

Таким образом, в методах координатного спуска мы спускаемся по ломанной, состоящей из отрезков прямых, параллельных координатным осям.



Спуск по всем координатам составляет одну

$$x_{kn+j} = x_{kn+j-1} - \alpha_{kn+j-1} \frac{\partial f}{\partial x^j}(x_{kn+j-1}) e_j,$$

«внешнюю» итерацию. Пусть k – номер очередной внешней итерации, а j – номер той координаты, по которой производится спуск. Тогда формула, определяющая следующее приближение к точке минимума, имеет вид:

где $k=0,1,2,\dots$; $j=1,2,\dots,n$.
В координатной форме формула выглядит так:

$$x_{kn+j}^{(i)} = x_{kn+j-1}^{(i)}, \text{ если } i \neq j,$$
$$x_{kn+j}^{(i)} = x_{kn+j-1}^{(i)} - \alpha_{kn+j-1} \frac{\partial f}{\partial x_j}(x_{kn+j-1}^{(i)}), \text{ если } i = j.$$

После $j=n$ счетчик числа внешних итераций k увеличивается на единицу, а j принимает значение равное единице.

Величина шага α_k выбирается на каждой итерации аналогично тому, как это делается в градиентных методах. Например, если $\alpha_k=\alpha$ постоянно, то имеем покоординатный спуск с постоянным шагом.

Схема алгоритма покоординатного спуска с постоянным шагом

Шаг 1.
При $k=0$ вводятся исходные данные $x_0, \varepsilon_1, \alpha$.

Шаг 2.
Осуществляется циклический по j ($j=1,2,\dots,n$) покоординатный спуск из точки x_{kn} по формуле:

$$x_{kn+j} = x_{kn+j-1} - \alpha \frac{\partial f}{\partial x_j}(x_{kn+j-1})e_j.$$

Шаг 3.
Если $\|x_{(k+1)n} - x_{kn}\| \leq \varepsilon_1$, то поиск минимума

$$\tilde{x} = x_{(k+1)n}, \tilde{y} = f(x_{(k+1)n}).$$

заканчивается, причем:
Иначе $k=k+1$ и переходим к шагу 2.

Если же шаг α_k выбирается из условия минимума

$$\varphi(\alpha) = f(x_{kn+j-1} - \alpha \frac{\partial f}{\partial x_j}(x_{kn+j-1})e_j),$$

функции:

то мы получаем аналог метода наискорейшего спуска, называемый обычно методом Гаусса – Зейделя.

Схема метода Гаусса – Зейделя

Шаг 1.
При $k=0$ вводятся исходные данные x_0, ε_1 .

Шаг 2.
Осуществляется циклический по j ($j=1,2,\dots,n$) покоординатный спуск из точки x_{kn} по формулам:

$$x_{kn+j} = x_{kn+j-1} - \alpha_{kn+j-1} \frac{\partial f}{\partial x_j}(x_{kn+j-1})e_j.$$

где α_{kn+j-1} является решением задачи одномерной минимизации функции:

Шаг 3.
Если $\|x_{(k+1)n} - x_{kn}\| \leq \varepsilon_1$, то поиск минимума

$$\tilde{x} = x_{(k+1)n}, \tilde{y} = f(x_{(k+1)n}).$$

заканчивается, причем:

$$\varphi(\alpha) = f(x_{kn+j-1} - \alpha \frac{\partial f}{\partial x_j}(x_{kn+j-1})e_j)$$

Иначе $k=k+1$ и переходим к шагу 2.

1. Метод дихотомии.

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.

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

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

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 $\pm \varepsilon$ вычисленной точки x , т.е.

$$F1 = F(x - \varepsilon),$$

$$F2 = F(x + \varepsilon).$$

3) Сравниваем $F1$ и $F2$ и отбрасываем одну из половинок отрезка $[a, b]$ (рис. 9.1).

При поиске минимума:

Если $F1 < F2$, то отбрасываем отрезок $[x, b]$, тогда $b = x$. (рис. 9.1.a)

Иначе отбрасываем отрезок $[a, x]$, тогда $a = x$. (рис. 9.1.б)

При поиске максимума:

Если $F1 < F2$, 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. $|b - a| \leq \varepsilon$

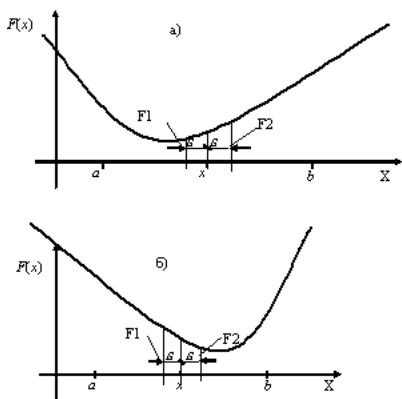


Рис. 9.1. Поиск экстремума функции $F(x)$ методом дихотомии

Схема алгоритма *метода дихотомии* представлена на рис. 9.2.

На рис. 9.2: 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.

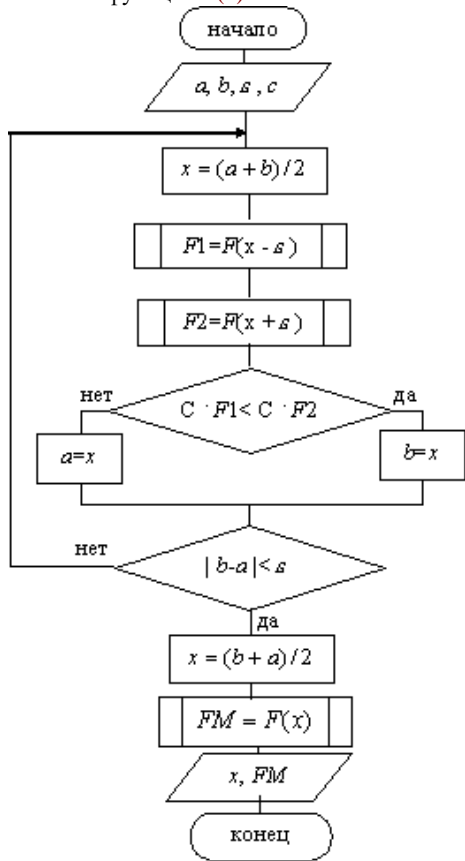


Рис. 9.2. Схема алгоритма метода дихотомии
 2. Метод Фибоначчи. Метод "золотого сечения"

Одним из методов *однопараметрической оптимизации* является *метод Фибоначчи*.

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. 9.3](#)). 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*?

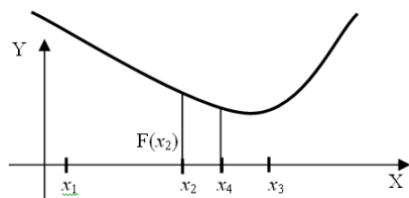


Рис. 9.3.

Положим $x_2 - x_1 = L$ и $x_3 - x_2 = R$, причем $L > R$, как показано на рис. 9.3, 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 точки. 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 $\epsilon/2$ 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, $L_{n-1} = 2L_n - \epsilon$ (рис.9.4, 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.9.4, средняя часть).

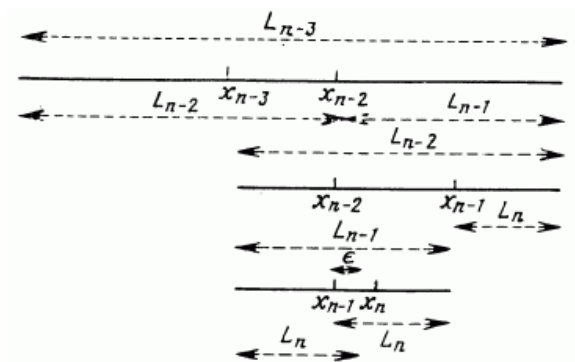


Рис. 9.4.

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. 9.4, верхняя часть)

В общем случае $L_{j-1}=L_j + L_{j+1}$ при $1 \leq j < n$.

Таким образом,

$$\begin{aligned} L_{n-1} &= 2L_n - \varepsilon, \\ L_{n-2} &= L_{n-1} + L_n = 3L_n - \varepsilon, \\ L_{n-3} &= L_{n-2} + L_{n-1} = 5L_n - 2\varepsilon, \\ L_{n-4} &= L_{n-3} + L_{n-2} = 8L_n - 3\varepsilon \quad \text{и т.д.} \end{aligned}$$

Если определить последовательность чисел Фибоначчи следующим образом: $F_0=1$, $F_1=1$, и $F_k=F_{k-1}+F_{k-2}$ для $k=2, 3, \dots$, то

$$L_{n-j} = F_{j+1}L_n - F_j\varepsilon, \quad j = 1, 2, \dots, n-1.$$

Если начальный интервал $(a;b)$ имеет длину $L = (b-a)$, то

$$\begin{aligned} L_1 &= F_n L_n - \varepsilon F_{n-2}, \quad \text{т.е.} \\ L_n &= \frac{L_1}{F_n} + \varepsilon \frac{F_{n-2}}{F_n}. \end{aligned}$$

Следовательно, произведя n вычислений функции, мы уменьшим начальный интервал неопределенности в $1/F_n$ раз по сравнению с его начальной длиной (пренебрегая ε), и это - наилучший результат.

Если поиск начал, 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:

$$\begin{aligned} L_2 &= F_{n-1}L_n - \varepsilon F_{n-3} = \\ &= F_{n-1} \frac{L_1}{F_n} + \varepsilon \frac{(F_{n-1}F_{n-2} - F_n F_{n-3})}{F_n} = \\ &= \frac{F_{n-1}}{F_n} L_1 + \frac{(-1)^n \varepsilon}{F_n}. \end{aligned}$$

После того как найдено положение первой точки, числа Фибоначчи больше не нужны. Используемое значение ε может определяться из практических

соображений. Оно должно быть меньше $L_1 \setminus F_{n+x}$, 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. 9.5](#)).

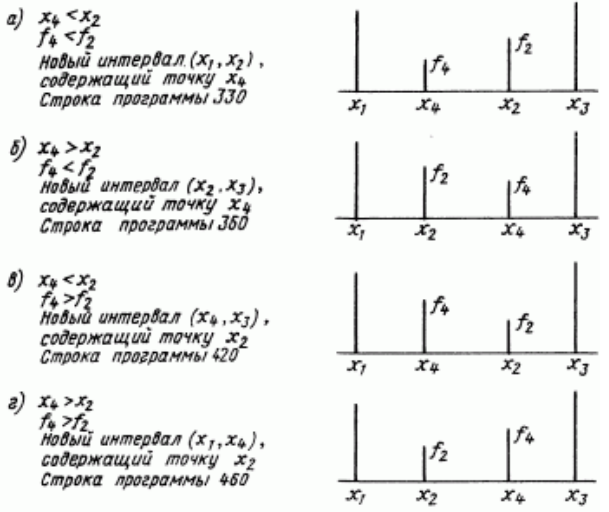


Рис. 9.5.

Следующий из методов одномерной оптимизации называется *методом "золотого сечения"*.

Не всегда можно заранее определить, сколько раз придется вычислять функцию. В *методе Фибоначчи* это нужно знать для определения L_2 , т.е. положения начальной точки (см. уравнение 2.4).

Метод "золотого сечения" почти столь же эффективен, как и *метод Фибоначчи*, однако 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 (см. уравнение 2.1), записываем

$$L_{j-1} = L_j + L_{j+1}$$

Однако если n не известно, то мы не можем использовать условие $L_{n-1} = L_n - \epsilon$. Если отношение последующих интервалов будет постоянным, т.е.

$$\frac{L_{j-1}}{L_j} = \frac{L_j}{L_{j+1}} = \frac{L_{j+1}}{L_{j+2}} = \dots = \tau,$$

то

$$\frac{L_{j-1}}{L_j} = 1 + \frac{L_{j+1}}{L_j},$$

т.е. $\mathbb{A} = \mathbb{I} + \mathbb{I} \backslash \mathbb{A}$

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.

$$L_n = \frac{L_1}{\tau^{n-1}}$$

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 $\lim_{n \rightarrow \infty} F_{n-1}/F_n = 1/n$,

то из уравнения (2.4) видно, что поиск *методом "золотого сечения"* является предельной формой поиска *методом Фибоначчи*. Название "золотое сечение" произошло от названия отношения в уравнении (2.7). Видно, что L_j .
 $_1$ делится на две части так, что отношение целого к большей части равно отношению большей части к меньшей, т.е. равно так называемому "золотому отношению".

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. 9.6](#)).

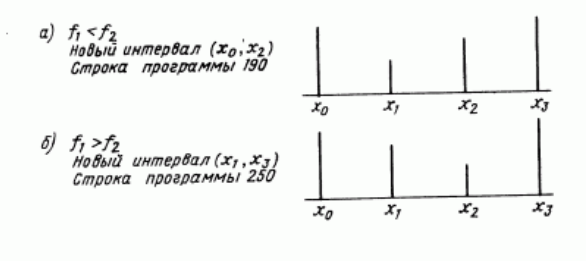
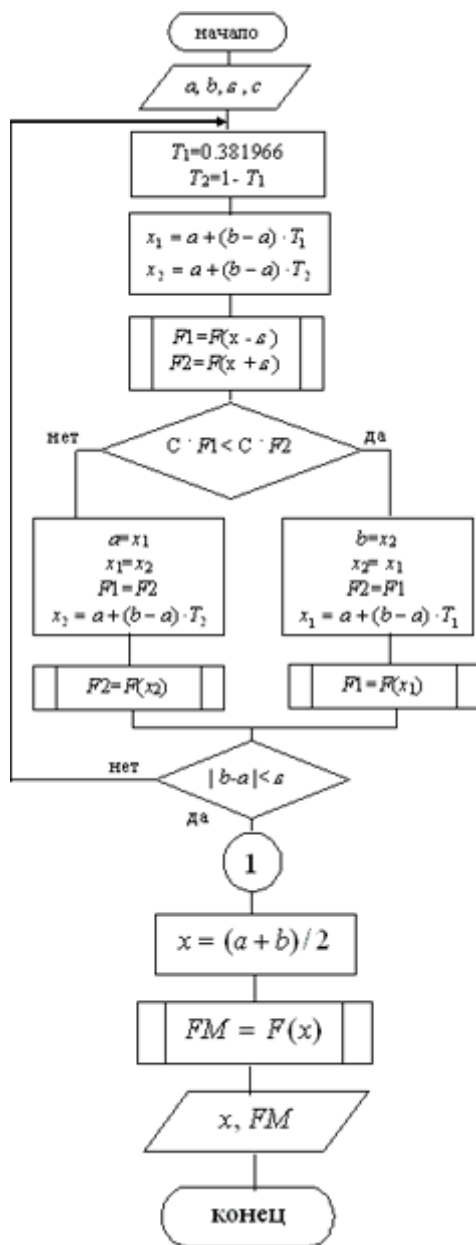


Рис. 9.6.

Метод гарантирует нахождение минимума в самых неблагоприятных условиях, однако он обладает медленной сходимостью.

Схема алгоритма *метода "золотого сечения"* представлена на [рис 9.7](#)



[увеличить изображение](#)

Рис. 9.7. 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.

3. 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 \bar{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

$$f(x) = 0,$$

где $f(x) = F'(x)$. \bar{x} - корень уравнения (3.1),

точка, то есть, координата в которой $F'(x)=0$, а

функция $F(x)$ имеет минимум (или максимум) ([рис.9.8](#)).

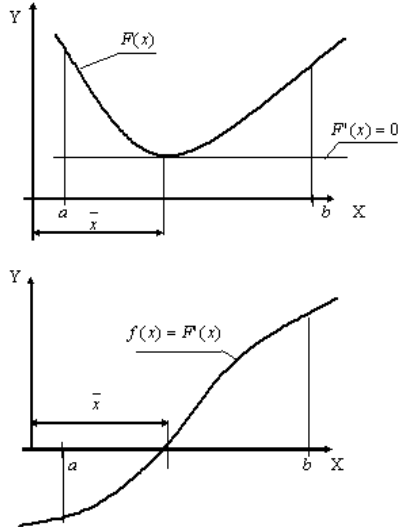


Рис. 9.8. Вогнутая функция $F(x)$ и ее производная $f(x)$.

Алгоритм *метода Ньютона* сводится к линейному представлению функции $f(x)$ и решению уравнения (3.1).

Разложим функцию $f(x)$ в ряд Тейлора:

$$f(x_{i+1}) = f(x_i) + h_i \cdot f'(x_i) + \frac{h_i^2}{2!} \cdot f''(x_i) + \frac{h_i^3}{3!} \cdot f'''(x_i) + \dots,$$

где $h_i = x_{i+1} - x_i$.

Отбросим члены ряда, содержащие h_i^2, h_i^3, \dots .

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:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{F'(x_i)}{F''(x_i)}.$$

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

$$f(x) \cdot f''(x) > 0,$$

где

$$f''(x) = F'''(x) = F3$$

Словесный 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 (3.2).
3. Итерационный процесс поиска продолжается до тех пор, пока

$$|x_{i+1} - x_i| < \varepsilon.$$

На основании (3.2) условие (3.4) можно записать как

$$\left| x_i - \frac{f(x_i)}{f'(x_i)} - x_i \right| < \varepsilon$$

В результате условие (3.4) будет иметь вид

$$\left| \frac{f(x_i)}{f'(x_i)} - x_i \right| < \varepsilon$$

В точке экстремума \bar{x} производная $F'(x)$ меняет знак.

Если в точке \bar{x} функция $F(x)$ имеет минимум, то производная $F'(x)$ в окрестности \bar{x} меняет знак с отрицательного на положительный, т.е. $F'(x)$ является возрастающей функцией, значит, $F''(x) > 0$ (рис. 9.9 а).

Если в точке \bar{x} функция $F(x)$ имеет максимум, то производная $F'(x)$ в окрестности \bar{x} меняет знак с положительного на отрицательный, т.е. $F'(x)$ является убывающей функцией, значит, $F''(x) < 0$ (рис. 9.9 б).

Следовательно, по знаку $F''(x)$ можно судить: в точке \bar{x} максимум или минимум функции $F(x)$.

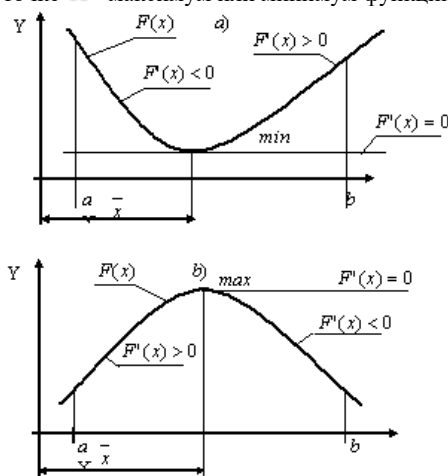


Рис. 9.9.

Если функция $F(x)$ не дифференцируема или вычисление ее производных очень сложно, то для определения производных функции $F(x)$ можно воспользоваться приближительными оценками производных с помощью разностных схем:

$$F'(x) = \frac{\Delta F}{h}; \quad F''(x) = \frac{\Delta F'}{h}; \quad \text{и т.д.}$$

Схема алгоритма метода Ньютона представлена на рис. 9.10.

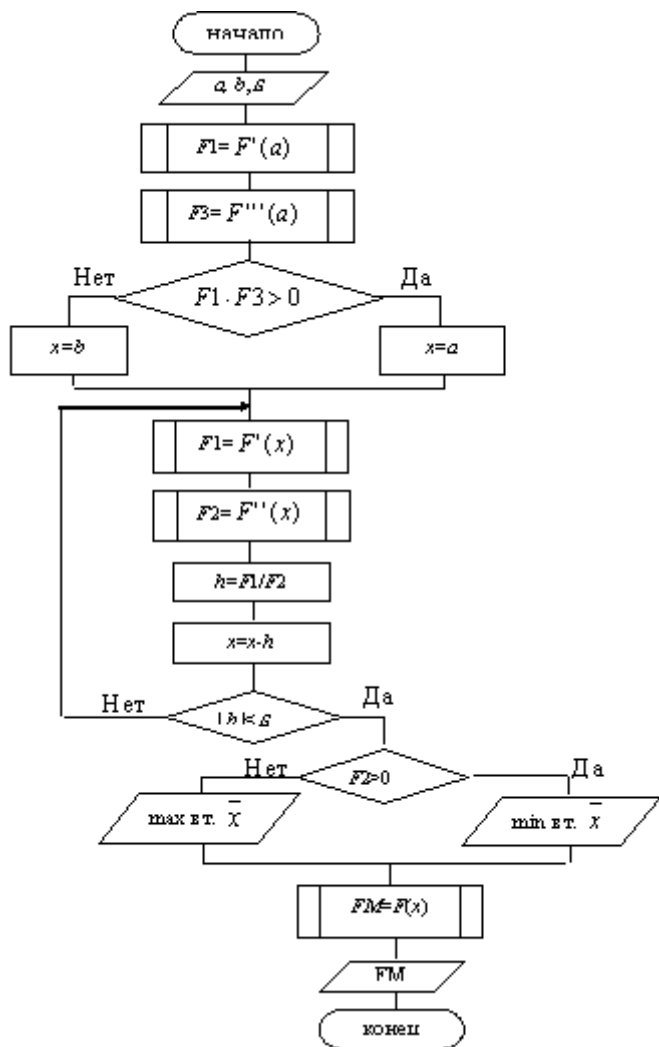


Рис. 9.10. Схема алгоритма метода Ньютона

На рис.9.10: \bar{x} - координата точки в которой функция $F(x)$ имеет минимальное (или максимальное) значение, FM - значение, функции $F(x)$ в точке \bar{x} .