

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Нижегородский государственный университет им. Н.И.Лобачевского  
Радиофизический факультет

**БЕЗУСЛОВНЫЙ ЭКСТРЕМУМ.  
ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ**

Лабораторная работа для студентов радиофизического  
факультета ННГУ

Нижний Новгород 2010г.

УДК 518.5 Безусловный экстремум. Введение в численные методы. /соч. В.В.Кулинич,М.М.Новоженов,В.И.Сумин - Нижний Новгород, изд. ННГУ им.Н.И.Лобачевского, 2001 г. -24с.

В описании лабораторной работы приведены основные сведения о некоторых наиболее важных в идейном отношении методах численного решения задач на безусловный экстремум. Сформулированы вопросы для самопроверки, помогающие уяснить суть изложенных методов. Задания лабораторной работы требуют от студентов самостоятельного исследования этих методов в конкретных ситуациях.

Составители:

Виктор Валентинович Кулинич  
Михаил Михайлович Новоженов  
Владимир Иосифович Сумин

Нижегородский государственный университет им.Н.И.Лобачевского  
2001 г.

Рассмотрим задачу нахождения безусловного минимума функции  $f(x)$  векторного аргумента  $x = \{x_1, x_2, \dots, x_n\}$ , заданной на всем пространстве  $R^n$ , символически записываемую в виде

$$f(x) \rightarrow \min, x \in R^n. \quad (1)$$

Напомним, что точка  $\bar{x}$  называется точкой локального минимума в задаче (1), если существует число  $\epsilon > 0$  такое, что для всех  $x$  удовлетворяющих условию  $\|x - \bar{x}\| \leq \epsilon$ , выполнено неравенство  $f(x) \geq f(\bar{x})$ ; здесь и далее  $\|x\| = \{x_1^2 + \dots + x_n^2\}^{1/2}$ . Точка  $\bar{x}$  называется точкой глобального минимума в задаче (1), если  $f(\bar{x}) \leq f(x), x \in R^n$ .

Для численного решения задачи (1) обычно строят некоторую последовательность векторов  $\{x^k\}_{k=0}^{\infty}$ , обрывая процесс построения тогда, когда появляется уверенность, что последний из построенных элементов последовательности близок к точке минимума в том или ином смысле. Например, если элемент  $x^k$  удовлетворяет неравенству  $\|x^k - \bar{x}\| < \bar{\delta}$ , где  $\bar{x}$ - точка минимума в задаче (1),  $\bar{\delta}$ - некоторое положительное число, то говорят, что  $x^k$  есть приближенное решение задачи (1) с точностью  $\bar{\delta}$  по аргументу. Если же выполняется неравенство  $|f(x^k) - \bar{f}| < \bar{\delta}$ , где  $\bar{f}$ - минимальное значение функции  $f(x)$  в  $R^n$ , то говорят, что число  $f(x^k)$  дает приближенное решение задачи (1) с точностью  $\bar{\delta}$  по функции.

Большинство процессов, используемых для приближенного решения задачи (1) можно представить как итерационные в виде

$$x^{k+1} = x^k + \alpha_k p^k, \quad (2)$$

где  $p^k$  - вектор, определяющий направление движения от точки  $x^k$  к точке  $x^{k+1}$ ,  $\alpha_k$  - числовой множитель, величина которого задает длину шага в направлении  $p^k$ . В различных итерационных процессах типа (2) для нахождения  $\alpha_k$  и  $p^k$  привлекаются различные сведения о минимизируемой функции  $f(x)$ . Процессы, использующие для этого только значения самой функции, называются процессами нулевого порядка или методами поиска. Процессы, требующие

вычисления производных  $f(x)$  до  $m$ -го порядка включительно, называют процессами  $m$ -го порядка. Если для определения  $a_k$  и  $p^k$  используется информация, полученная на  $S$  предыдущих итерациях, то итерационный процесс называется  $S$ -шаговым.

В качестве направления  $p^k$  естественно выбирать направление убывания функции  $f(x)$ . Такое направление называют направлением спуска. Метод (2) называют методом спуска, если при каждом  $k$  направление  $p^k$  - направление спуска, а число  $\alpha_k$  таково, что

$$f(x^{k+1}) = f(x^k + \alpha_k p^k) < f(x^k). \quad (3)$$

Важной характеристикой итерационного процесса является скорость сходимости итерационной последовательности. Пусть итерационная последовательность  $\{x^k\}_{k=0}^{\infty}$  сходится к  $\bar{x}$ , т.е.

$$\|x^k - \bar{x}\| \rightarrow 0, k \rightarrow \infty.$$

Говорят, что скорость сходимости последовательности

1) линейна, если существует число  $q \in (0, 1)$  такое, что

$$\|x^{k+1} - \bar{x}\| \leq q \|x^k - \bar{x}\|, k = 0, 1, 2, \dots;$$

2) сверхлинейна, если существует последовательность чисел  $\{q_k\}_{k=1}^{\infty}$ ,  $q_k \rightarrow 0, k \rightarrow \infty$  такая, что

$$\|x^{k+1} - \bar{x}\| \leq q_k \|x^k - \bar{x}\|, k = 0, 1, 2, \dots;$$

3) квадратична, если существует постоянная  $C > 0$  такая, что

$$\|x^{k+1} - \bar{x}\| \leq C \|x^k - \bar{x}\|^2, k = 0, 1, 2, \dots$$

### §1. Методы первого порядка.

Простейшими методами первого порядка являются одношаговые методы спуска, в которых на каждом шаге вектор  $p^k$  совпадает с направлением антиградиента функции  $f(x)$ :

$$p^k = -\nabla f(x^k).$$

Такие методы называют градиентными. Градиентные методы отличаются друг от друга способом выбора множителя  $\alpha_k$ .

Ниже описаны два градиентных метода (п.п.1,2) и один двухшаговый метод первого порядка (п.3).

**1. Метод наискорейшего спуска.** В этом методе  $\alpha_k$  выбирается из условия минимума функции  $f(x)$  вдоль направления  $p^k$ , т.е.

$$f(x^k + \alpha_k p^k) = \min_{\alpha > 0} f(x^k + \alpha p^k).$$

Таким образом, в методе наискорейшего спуска на каждом шаге необходимо решать задачу минимизации функции одной переменной.

**Замечание.** Точная нижняя грань функции  $f_k(\alpha) = f(x^k + \alpha p^k)$  на полуоси  $\alpha > 0$  может и не достигаться. Но даже, если она достигается, точное определение величины  $\alpha_k$  как точки глобального на полуоси минимума функции  $f_k(\alpha)$  не всегда возможно. Поэтому на практике имеет смысл заменить задачу нахождения  $\min_{\alpha > 0} f_k(\alpha)$  задачей отыскания минимума  $f_k(\alpha)$  на том или ином (достаточно большом) отрезке  $[0, a]$ . В случае, когда  $f_k(\alpha)$  унимодальна на  $[0, a]$ , для приближенного решения последней задачи применяют методы одномерного поиска, описанные в §3. См. также [4], гл.5, §1.

**2. Градиентный метод с дроблением шага.** Опишем два варианта метода. В обоих вариантах параметрами метода являются величины  $\alpha > 0$  и  $\lambda \in (0, 1)$ . В варианте Б) параметром будет также  $\epsilon \in (0, 1)$ . Параметр  $\lambda$  называется коэффициентом дробления. Значения параметров выбираются наперед; они одни и те же для всех итераций.

**А)** Выбор  $\alpha_k$  происходит следующим образом. Положив сначала  $\bar{\alpha} = \alpha$ , проверим неравенство

$$f(x^k + \bar{\alpha} p^k) < f(x^k). \quad (4)$$

Если оно выполнено, то берем  $\alpha_k = \bar{\alpha}$ . В противном случае значение  $\bar{\alpha}$  изменяем, домножив его на  $\lambda$  (дробление  $\bar{\alpha}$ ). Снова проверяем (4). И так до тех пор, пока неравенство (4) не выполнится. То значение  $\bar{\alpha}$ , при котором это произойдет впервые, и выбираем в качестве  $\alpha_k$ .

**Б)** В этом варианте для того, чтобы выбор  $\alpha_k$  гарантировал существенное убывание функции  $f(x)$  при переходе от точки  $x^k$  к точке  $x^{k+1}$ , стараются удовлетворить не неравенству (3) как в варианте А), а более сильному неравенству

$$f(x^{k+1}) = f(x^k + \alpha_k p^k) < f(x^k) - \epsilon \alpha_k \|\nabla f(x^k)\|^2.$$

Выбор  $\alpha_k$  происходит так. Положив сначала  $\bar{\alpha} = \alpha$ , проверим неравенство

$$f(x^k + \bar{\alpha} p^k) < f(x^k) - \epsilon \bar{\alpha} \|\nabla f(x^k)\|^2. \quad (5)$$

Если оно выполнено, то берем  $\alpha_k = \bar{\alpha}$ . В противном случае производим постепенное дробление  $\bar{\alpha}$ , домножая его на  $\lambda$ , до тех пор, пока не выполнится (5). То значение  $\bar{\alpha}$ , при котором это произойдет впервые, выберем в качестве  $\alpha_k$ .

**3. Метод сопряженных градиентов.** В этом методе  $\alpha_k$  выбирается, как и в методе наискорейшего спуска, из условия

$$f(x^k + \alpha_k p^k) = \min_{\alpha > 0} f(x^k + \alpha p^k)$$

(см. замечание в п.1). При этом вектор  $p^k$  зависит не только от градиента функции  $f(x^k)$ , но и от градиента в предыдущей точке  $\nabla f(x^{k-1})$  (т.е. метод является двухшаговым), и строится либо по правилу

$$\begin{aligned} \text{А) } p^k &= -\nabla f(x^k) + \beta_k p^{k-1}, \\ \beta_0 &= 0, \beta_k = \frac{(\nabla f(x^k), \nabla f(x^k) - \nabla f(x^{k-1}))}{\|\nabla f(x^{k-1})\|^2}, k = 1, 2, \dots; \end{aligned}$$

либо по правилу

$$\text{Б) } p^k = -\nabla f(x^k) + \beta_k p^{k-1},$$

$$\beta_k = \begin{cases} 0, k = 0, n, 2n, \dots; \\ \frac{(\nabla f(x^k), \nabla f(x^k) - \nabla f(x^{k-1}))}{\|\nabla f(x^{k-1})\|^2}, \\ k = 1, \dots, n-1, n+1, \dots, 2n-1, 2n+1, \dots \end{cases}$$

Напомним, что  $n$  - размерность пространства независимых переменных. Вариант Б) отличается от варианта А) тем, что содержит так называемую процедуру обновления - для каждого  $k$ , кратного  $n$ , переход из точки  $x^k$  в точку  $x^{k+1}$  выполняется как в методе наискорейшего спуска. Заметим, что переход из точки  $x^0$  в точку  $x^1$  выполняется как в методе наискорейшего спуска и в случае А) и в случае Б).

**4. Сходимость.** Сходимость любого метода зависит от свойств  $f(x)$ , выбора начальной точки  $x^0$  и параметров итерационного процесса. Полезна следующая теорема.

**Теорема 1.** ([1], с.47,83; [4], с.265). Пусть функция  $f(x)$  ограничена снизу, а ее градиент удовлетворяет условию Липшица. Если построение минимизирующей последовательности  $\{x^k\}$  производится по способу п.1, либо п.2 Б), либо п.3 Б), то какова бы ни была начальная точка  $x^0$ ,

$$\|\nabla f(x^k)\| \rightarrow 0, k \rightarrow \infty.$$

Если точка  $x^0$  такова, что множество  $K(x_0) = \{x : f(x) \leq f(x^0)\}$  ограничено, то последовательность  $\{x^k\}$  сходится к множеству  $S = \{x : \nabla f(x) = 0\}$  стационарных точек функции  $f(x)$ , т.е.

$$\inf_{x \in S} \|x - x^k\| \rightarrow 0, k \rightarrow \infty.$$

Заметим, что класс функций, указанный в теореме 1, весьма широк, и среди стационарных точек таких функций могут быть не только точки глобальных экстремумов, но и точки локальных экстремумов, а также седловые точки. Однако, как указано в [5], градиентные методы, например, "почти никогда" не сходятся к точке максимума или седловой точке. В то же время они не различают точек локального и глобального минимума и сходятся к произвольной из них (точнее см. [5], с.168).

Оценку скорости сходимости последовательности  $\{x^k\}$  для каждого из методов п.п.1 – 3 можно дать лишь на достаточно узких классах функций, предъявляя весьма жесткие требования к гладкости и выпуклости функций  $f(x)$ . Рассмотрим, например, класс дважды непрерывно дифференцируемых сильно выпуклых функций. Дважды непрерывно дифференцируемая функция  $f(x)$  называется сильно выпуклой в  $R^n$ , если существует постоянная  $l$ ,  $l > 0$ , такая, что для всех  $x \in R^n$  имеем

$$l\|y\|^2 \leq (\nabla^2 f(x)y, y), y \in R^n,$$

где  $\nabla^2 f(x)$ -матрица Гессе (гессиан) функции  $f(x)$ ,

$$\nabla^2 f(x) = \begin{pmatrix} f''_{x_1x_1}(x) & f''_{x_1x_2}(x) \dots & f''_{x_1x_n}(x) \\ \dots & \dots & \dots \\ f''_{x_nx_1}(x) & f''_{x_nx_2}(x) \dots & f''_{x_nx_n}(x) \end{pmatrix}.$$

Каждая функция указанного класса имеет единственную точку минимума - глобальную точку минимума в  $R^n$  (см. [4], гл.4, §3).

**Теорема 2.** ([2], с.55, 61). Пусть  $f(x)$  дважды непрерывно дифференцируема и существуют постоянные  $L, l, 0 < l \leq L$ , такие, что для всех  $x \in R^n$  имеем

$$l\|y\|^2 \leq (\nabla^2 f(x)y, y) \leq L\|y\|^2, y \in R^n. \quad (6)$$

Если построение последовательности  $\{x^k\}$  производится по способу п.2 Б) или п.1, то при любой начальной точке  $x^0$  последовательность  $\{x^k\}$  сходится к точке минимума  $\bar{x}$  с линейной скоростью. В случае способа п.1 постоянная  $q = (L - l)/(L + l)$ .

Если поверхности уровня минимизируемой функции имеют сложную сильно вытянутую (овражную) структуру, то направление антиградиента сильно отличается от направления на точку минимума и сходимость градиентных методов замедляется. Это явление называется овражным эффектом. Показателем овражности в окрестности точки минимума  $\bar{x}$  функции  $f(x)$  является отношение наибольшего собственного числа матрицы Гессе  $\nabla^2 f(\bar{x})$  к наименьшему (см. [5], с.28). Чем больше этот показатель, тем более вытянутым и крутым является "овраг" поверхности уровня  $f(x)$  в окрестности  $\bar{x}$  и тем медленнее сходятся в этой окрестности градиентные методы (см.[5], с.150).

Метод сопряженных градиентов - наилучший по скорости из рассмотренных методов первого порядка (см.[5],гл.3, §2). Для метода сопряженных градиентов в случае числа итераций, существенно большем размерности  $n$ , справедлив следующий результат (см.[5], с.74).

**Теорема 3.** Пусть функция  $f(x)$  трижды дифференцируемая сильно выпуклая функция. Тогда последовательность  $\{x^k\}$ , построенная по способу п.3 Б), сходится к точке минимума  $\bar{x}$  функции  $f(x)$ , причем в некоторой окрестности точки  $\bar{x}$  имеет место оценка скорости сходимости

$$\|x^{(k+1)n} - \bar{x}\| \leq C \|x^{kn} - \bar{x}\|^2, k = 1, 2, \dots$$

Для квадратичных функций  $f(x) = \frac{1}{2}(Ax, x) - (b, x)$  ( $A$  - положительно определенная симметричная матрица) метод сопряженных градиентов сходится за конечное число шагов, не превышающее числа  $n$ . При этом последовательные направления  $p^k$  удовлетворяют соотношению  $(Ap^i, p^j) = 0, i \neq j$ , т.е. являются ортогональными в метрике, задаваемой матрицей  $A$ ,  $A$  - ортогональными ( $A$  - сопряженными). Отсюда - название метода (см.[1], гл.2, §3).

О роли теорем сходимости для практических вычислений см., например, [5], с.39-43.

**5. Критерии окончания итерационных процессов.** Теорема 1 позволяет для окончания итерационного процесса пользоваться условием вида

$$\|\nabla f(x^k)\| < \delta, \tag{7}$$

где  $\delta > 0$  - некоторое заданное число. На практике в качестве кри-

терия окончания итерационных процессов часто используют также следующие неравенства.

При решении задачи (1) с заданной точностью по функции:

$$|f(x^k) - f(x^{k-1})| < \delta \quad \text{или} \quad \frac{|f(x^k) - f(x^{k-1})|}{1 + |f(x^{k-1})|} < \delta. \quad (8)$$

При решении задачи (1) с заданной точностью по аргументу:

$$\|x^k - x^{k-1}\| < \delta \quad \text{или} \quad \frac{\|x^k - x^{k-1}\|}{1 + \|x^{k-1}\|} < \delta. \quad (9)$$

Применяют и различные комбинации критериев (7) – (9).

Выбор величины  $\delta$  определяется заданной точностью  $\bar{\delta}$  (см. стр.3). Однако, правильный выбор  $\delta$  по заданной величине  $\bar{\delta}$  во многом зависит от искусства вычислителя. "К сожалению, надежных критериев окончания счета, которые гарантировали бы получение решения задачи (1) с требуемой точностью, и применимых к широкому классу задач, пока нет" ([4], с.264). Это замечание относится и к другим методам, описанным ниже.

## §2. Методы второго порядка.

Среди методов второго порядка основными являются методы, связанные с идеей локальной аппроксимации заданной функции квадратичной функцией.

**1.Метод Ньютона.** Для получения итерационной формулы этого метода используем следующие эвристические соображения (см.[5], с.36). Запишем для функции  $f(x)$  в окрестности точки  $x^k$  формулу Тейлора 2-го порядка

$$f(x) = Q_k(x) + o(\|x - x^k\|^2), \quad \|x - x^k\| \rightarrow 0;$$

$$Q_k(x) = f(x^k) + (\nabla f(x^k), (x - x^k)) + \frac{1}{2}(\nabla^2 f(x^k)(x - x^k), (x - x^k)). \quad (10)$$

В случае, когда гессиан  $\nabla^2 f(x^k)$  положительно определен, квадратичная функция  $Q_k(x)$  достигает глобального минимума в точке

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k), \quad (11)$$

( $\nabla Q_k(x^{k+1}) = 0$ ). Если точка  $x^{k+1}$  достаточно близка к  $x^k$ , то в силу (10) справедливо неравенство

$$f(x^{k+1}) < f(x^k),$$

т.е.  $x^{k+1}$  естественно взять следующим за  $x^k$  приближением к решению задачи (1). Формула (11) и есть итерационная формула метода Ньютона. Таким образом, в этом методе  $\alpha_k = 1$ ,

$$p^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k). \quad (12)$$

Практически  $p^k$  удобнее искать не по формуле (12), а решая систему линейных уравнений

$$[\nabla^2 f(x^k)]p^k = -\nabla f(x^k) \quad (13)$$

одним из прямых или итерационных методов (см. соответствующие лабораторные работы), исключая тем самым операцию обращения матрицы Гессе. Отметим, что для квадратичной функции  $f(x)$  метод Ньютона сходится за один шаг. Для достаточно гладкой функции  $f(x)$  с положительно определенной матрицей Гессе при удачном выборе начального приближения  $x^0$  итерационная последовательность  $\{x^k\}$  метода Ньютона сходится к точке минимума с квадратичной скоростью. Однако, найти удачное начальное приближение - задача довольно сложная, требующая определенного искусства. Модифицируя метод Ньютона введением переменного множителя  $\alpha_k$ , получают методы спуска, сходящиеся при любом начальном приближении  $x^0$ .

**2. Метод Ньютона-Рафсона.** В этом методе направление спуска определяется формулой (12), а множитель  $\alpha_k$ , регулирующий длину шага, можно выбирать, одним из следующих способов:

А) как и в методе наискорейшего спуска из условия минимума

$$f(x^k + \alpha_k p^k) = \min_{\alpha > 0} f(x^k + \alpha p^k)$$

(см. замечание в п.1 §1);

Б) так, чтобы выполнялось неравенство

$$f(x^k + \alpha_k p^k) \leq f(x^k) + \epsilon \alpha_k (\nabla f(x^k), p^k),$$

где  $\epsilon \in (0, 1/2)$ -наперед заданная постоянная, одна и та же для всех итераций. Алгоритм нахождения множителя  $\alpha_k$  здесь такой же, как и в градиентном методе с дроблением шага. Начальное значение  $\alpha = 1$ .

**Теорема 4.** (см.[2],гл.2, §2, п.2). Пусть функция  $f(x)$  дважды непрерывно дифференцируема и выполняется (6), а вторые производные  $f$  удовлетворяют условию Липшица. Тогда итерационная

последовательность метода Ньютона-Рафсона сходится к точке минимума с квадратичной скоростью, какова бы ни была начальная точка  $x^0$ . Если условие Липшица для вторых производных не выполняется, то скорость сходимости сверхлинейна.

Для окончания итерационных процессов в методах второго порядка используют те же критерии, что и в методах первого порядка.

К основным недостаткам методов второго порядка следует отнести необходимость вычисления вторых производных, а также то, что сходимость заведомо гарантируется лишь при положительной определенности гессиана функции  $f(x)$ . Важность последнего положения демонстрируется в [1] на с.59-61.

### §3. Методы нулевого порядка.

В задачах безусловной минимизации методы нулевого порядка сходятся, как правило, хуже методов, использующих информацию о производных. Однако этот недостаток зачастую перекрывается тем достоинством методов поиска, что они требуют гораздо меньше времени на подготовку задачи к решению (исключаются, например, такие трудные операции, как вычисление производных).

**1. Метод покоординатного спуска.** Для построения минимизирующей последовательности используется формула (2). При этом вектор  $p^k$  определяется по правилу (циклический покоординатный спуск):

$$p^k = e_{k-[k/n]n+1}, k = 0, 1, 2, \dots,$$

где  $[t]$ -целая часть числа  $t$ ,  $e_j = \{0, \dots, 0, 1, 0, \dots, 0\}$  (единица стоит на  $j$ -ом месте),  $j = 1, \dots, n$ . Число  $\alpha_k \in (-\infty, \infty)$  можно определять, например, следующим способом

$$f(x^k + \alpha_k p^k) = \min_{-\infty < \alpha < \infty} f(x^k + \alpha p^k).$$

Здесь можно сделать замечание, аналогичное замечанию из п.1, §1.

Метод покоординатного спуска очень прост, но не очень эффективен. Проблемы могут возникнуть, когда линии уровня сильно вытянуты, т.е. для овражных функций. В подобной ситуации поиск быстро застревает на дне такого оврага, а если начальное приближение оказывается на оси "эллипсоида" то процесс так и останется в этой точке. Хорошие результаты получаются в тех случаях, когда целевая функция представляет собой выпуклую функцию.

### 2. Метод Хука-Дживса.

В данном алгоритме предлагается логически простая стратегия поиска, в которой используются априорные сведения о топологии функции и, в то же время, отвергается уже устаревшая информация об этой функции. В интерпретации Вуда алгоритм включает два основных этапа: а) исследующий поиск вокруг базисной точки  $x^k$ ; б) поиск по "образцу" т.е. в направлении, выбранном для минимизации.

В первую очередь задается начальная точка поиска  $x^k$  и начальное приращение (шаг)  $h$ . После этого начинается исследующий поиск.

**Исследующий поиск.** Делаем пробный шаг по первой переменной  $x_1$  с заданным шагом  $h > 0$  т.е. вычисляем значение функции в точке  $(x_1^k + h, x_2^k, \dots, x_n^k)$  и если  $f(x_1^k + h, x_2^k, \dots, x_n^k) > f(x_1^k, x_2^k, \dots, x_n^k)$ , тогда точку  $(x_1^k, x_2^k, \dots, x_n^k)$  оставляем без изменений и делаем пробный шаг в противоположном направлении. Если  $f(x_1^k - h, x_2^k, \dots, x_n^k) > f(x_1^k, x_2^k, \dots, x_n^k)$ , тогда точку  $(x_1^k, x_2^k, \dots, x_n^k)$  оставляем без изменений, иначе точку  $(x_1^k, x_2^k, \dots, x_n^k)$  заменяем на точку  $(x_1^k + h, x_2^k, \dots, x_n^k)$  или  $(x_1^k - h, x_2^k, \dots, x_n^k)$  в зависимости от того, где значение функции меньше исходного. Из вновь полученной точки делаем пробные шаги по оставшимся координатам, используя тот же самый алгоритм.

Если в процессе исследующего поиска не удастся сделать ни одного удачного пробного шага, тогда необходимо скорректировать (уменьшить) шаг  $h$ . После чего вновь переходим к исследующему поиску. Если в процессе исследующего поиска сделан хотя бы один удачный пробный шаг, то переходим к поиску по образцу.

**Поиск по образцу.** После исследующего поиска мы получаем точку  $\bar{x}^k$ . Направление  $p^k = \bar{x}^k - x^k$  определяет направление, в котором функция уменьшается. Поэтому проводим минимизацию функции в указанном направлении, решая задачу

А)

$$f(x^k + \alpha_k p^k) = \min_{\alpha > 0} f(x^k + \alpha p^k).$$

Если удастся сделать удачный шаг в поиске по "образцу", то в результате находим новое приближение  $x^{k+1} = x^k + \alpha_k p^k$

Из точки  $x^{k+1}$  начинаем новый исследующий поиск и т.д.

Возможны модификации алгоритма:

Б) в процессе исследующего поиска ищется минимум по каждой переменной;

В) в процессе поиска по образцу ищется не минимум функции, а

просто делается шаг в заданном найденном направлении с фиксированным значением параметра  $\alpha_k$ , так чтобы

$$f(x^k + \alpha_k p^k) < f(x^k).$$

### 3. Метод Розенброка.

Идея метода заключается в том, что выбирается система ортогональных направлений  $p_1^k, p_2^k, \dots, p_n^k$ , в каждом из которых последовательно ищется минимальное значение, после чего система направлений поворачивается так, чтобы одна из осей совпала с направлением полного перемещения, а остальные были ортогональны между собой. Алгоритм Розенброка состоит из двух этапов:

**Покоординатный спуск.** Пусть  $x^k$  - вектор  $k$ -приближения и  $p_1^k, p_2^k, \dots, p_n^k$  - система ортогональных направлений. На первой итерации это может быть ортонормированная система координат. Начиная с заданного  $x^k$  последовательно осуществляем минимизацию функции  $f(x)$  в направлениях, соответствующих  $p_1^k, p_2^k, \dots, p_n^k$ , находя последовательные приближения:

$$\begin{aligned} x_1^{k+1} &= x_1^k + \lambda_1^k p_1^k, & \lambda_1^k &= \arg \min_{-\infty < \alpha < \infty} f(x^k + \alpha p_1^k), \\ & & \dots & \\ x_n^{k+1} &= x_{n-1}^{k+1} + \lambda_n^k p_{n-1}^k, & \lambda_n^k &= \arg \min_{-\infty < \alpha < \infty} f(x^k + \alpha p_n^k), \end{aligned}$$

**Поворот ортогональных направлений.** А) Ортогональные направления поиска  $p_1^k, p_2^k, \dots, p_n^k$  поворачивают так, чтобы они оказались вытянутыми вдоль "оврага" ("хребта").

Для этого с помощью найденных  $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$  строим систему векторов  $A_1^k, A_2^k, \dots, A_n^k$

$$A_i^k = \sum_{j=i}^n \lambda_j^k p_j^k, \quad i = \overline{1, n},$$

которую затем с помощью процедуры Грама-Шмидта ортогонализируем:

$$\begin{aligned} p_1^{k+1} &= \frac{A_1^k}{\|A_1^k\|} \\ B_i^k &= A_i^k - \sum_{j=1}^{i-1} [(A_i^k)' p_j^{k+1}] p_j^{k+1}, & p_i^{k+1} &= \frac{B_i^k}{\|B_i^k\|}, & i &= \overline{2, n}. \end{aligned}$$

Для эффективной работы алгоритма необходимо, чтобы ни один из векторов системы  $p_1^{k+1}, p_2^{k+1}, \dots, p_n^{k+1}$  не оказался равным нулевому. Для этого в алгоритме следует располагать параметры  $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$  в порядке убывания по абсолютному значению, т.е.  $|\lambda_1^k| > |\lambda_2^k| > \dots > |\lambda_n^k|$  и в построенной системе таким образом системе  $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$  отбросить  $m$  последних, если они имеются, нулевых чисел  $\lambda_m^k, \lambda_{m+1}^k, \dots, \lambda_n^k$ .

При этом положить

$$p_m^{k+1} = p_m^k, \dots, p_n^{k+1} = p_n^k$$

Б) Так как вектора  $B_{j+1}^k$  и  $\|B_{j+1}^k\|$  пропорциональны  $\lambda_j^k$  при условии

$$\sum_{i=1}^n |\lambda_i^k|^2 \neq 0$$

и, следовательно, при вычислении вектора

$$p_j^{k+1} = \frac{B_j^k}{\|B_j^k\|}$$

величины  $\lambda_j^k$  сокращаются и тогда вектора  $e_i^{k+1}$  могут быть найдены по следующему алгоритму

$$p_1^{k+1} = \frac{A_1^k}{\|A_1^k\|},$$

$$p_i^{k+1} = \frac{A_i^k \|A_{i-1}^k\|^2 - A_{i-1}^k \|A_i^k\|^2}{\|A_i^k\| \|A_{i-1}^k\| \sqrt{|\|A_i^k\|^2 - \|A_{i-1}^k\|^2|}}, \quad i = \overline{2, n}.$$

#### 4. Симплексный метод Нелдера-Мида или поиск по деформируемому многограннику.

В основу метода положено построение последовательности симплексов (многогранников),  $(n+1)$  вершин, которого и являются точкой  $x^k = x_{ij}^k, j = \overline{1, n}, i = \overline{1, n+1}$  (всего  $(n+1)$ -вершин в  $n$ -векторном пространстве. Координаты вершин  $x_{ij}^{k+1}, j = \overline{1, n}, i = \overline{1, n+1}$  совпадают с координатами вершин  $x_{ij}^k, j = \overline{1, n}, i = \overline{1, n+1}$  кроме вершины  $i = h$ , вершина  $x_{hj}^k, j = \overline{1, n}$  является наихудшей в смысле максимума минимизируемой функции. Вершина  $x_{hj}^k, j = \overline{1, n}$  определяется по специальным правилам, в результате которых симплексы деформируются в зависимости от поведения линий уровня минимизируемой функции, вытягиваясь вдоль оврагов, изменяя направление вдоль изгибов и сжимаясь в окрестности минимума.

Вычисления заканчиваются, когда значение минимизируемой функции в вершинах симплекса мало отличаются от значения в центре тяжести симплекса, либо симплекс стянулся в "точку".

Опишем  $(k + 1)$  шаг метода (построение нового симплекса).

Пусть

$$f(x_h^k) = \max_{i=\overline{1, n+1}} f(x_{ij}^k, j = \overline{1, n})$$

максимальное значение минимизируемой функции в вершинах симплекса, а

$$f(x_m^k) = \min_{i=\overline{1, n+1}} f(x_{ij}^k, j = \overline{1, n})$$

минимальное значение минимизируемой функции в вершинах симплекса.

Далее обозначим через  $x_{n+2}^k$  вершину центра тяжести симплекса без вершины  $x_h^k$  где достигается максимальное значение минимизируемой функции.

Координаты этого центра вычисляются по формуле:

$$x_{n+2, j}^k = \frac{1}{n} \sum_{i=1}^{n+1} [x_{ij}^k - x_{hj}^k], \quad j = \overline{1, n}.$$

Начальный многогранник обычно выбирается в виде регулярного симплекса (с вершиной в начале координат).

Координаты вершин регулярного симплекса в  $n$ -мерном пространстве могут быть определены матрицей, в которой столбцы представляют собой вершины симплекса, пронумерованные от 1 до  $(n+1)$  а строки - координаты вершин,  $i = \overline{1, n}$ . Матрица имеет размерность  $n \times (n + 1)$ :

$$\begin{bmatrix} 0 & d_1 & d_2 & \dots & d_2 \\ 0 & d_2 & d_1 & \dots & d_2 \\ 0 & d_2 & d_2 & \dots & d_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & d_2 & d_2 & \dots & d_1 \end{bmatrix},$$

$$d_1 = \frac{t}{n\sqrt{2}}(\sqrt{n+1} - n + 1), \quad d_2 = \frac{t}{n\sqrt{2}}(\sqrt{n+1} - 1),$$

$t$  - расстояние между вершинами.

Процедура отыскания вершин в которых минимизируемая функция имеет лучшее значение, состоит из следующих операций: 1) *отражения*; 2) *растяжения*; 3) *сжатия*; 4) *редукции*.

**1) Отражение.** Отражение представляет собой проектирование найденной вершины  $x_h^k$  через в центр тяжести  $x_{n+2}^k$  в соответствии с соотношением:

$$x_{n+3}^k = x_{n+2}^k + \alpha x_h^k,$$

где  $\alpha \geq 1$  - коэффициент отражения.

Затем если

$$f(x_{n+3}^k) \geq f(x_h^k),$$

то переходим к операции *редукции*. Если

$$f(x_{n+3}^k) < f(x_h^k) \wedge f(x_{n+3}^k) < f(x_m^k),$$

выполняем операцию *растяжения*. В противном случае, если

$$f(x_{n+3}^k) < f(x_h^k) \wedge f(x_{n+3}^k) \geq f(x_m^k),$$

полняется операция *сжатия*.

**2) Растяжение.** Операция заключается в следующем. Если

$$f(x_{n+3}^k) \leq f(x_m^k),$$

вектор  $x_{n+3}^k - x_{n+2}^k$  растягивается в соответствии с соотношением

$$x_{n+4}^k = x_{n+2}^k + \gamma(x_{n+3}^k - x_{n+2}^k)$$

$\gamma$ - коэффициент растяжения.

Если

$$f(x_{n+4}^k) \leq f(x_m^k)$$

то  $x_m^k$  заменяется на  $x_{n+4}^k$  и процедура продолжается с операции *отражения*. В противном случае  $x_m^k$  заменяется на  $x_{n+3}^k$  и переходим к операции *отражения*.

**3) Сжатие.** Если

$$f(x_{n+3}^k) > f(x_i^k)$$

для всех  $i$ ,  $i \neq h$ , то вектор  $x_h^k - x_{n+2}^k$

сжимается в соответствии с формулой

$$x_{n+5}^k = x_{n+2}^k + \beta(x_h^k - x_{n+2}^k)$$

где  $0 < \beta < 1$  - коэффициент сжатия. После этого, вершина  $x_h^k$  заменяется на вершину  $x_{n+5}^k$ , и переходим к следующему шагу, т.е. построению нового симплекса

**4) Редукция.** Если

$$f(x_{n+3}^k) > f(x_h^k),$$

то все векторы

$$x_i^k - x_m^k, \quad i = \overline{1, n+1}$$

уменьшаются в два раза с отсчетом от вершины  $x_m^k$  по формуле

$$x_i^k = x_m^k + 0.5(x_i^k - x_m^k), \quad i = \overline{1, n+1}$$

и осуществляется переход к следующему шагу алгоритма.

Если ни одна из операций (*растяжения, сжатия, редукции*) не выполняются, то производят следующее *отражение*. Если эти операции не выполняются  $N$ -заданное число раз, то поиск прекращают.

В качестве критерия остановки могут быть взяты те же правила, что и в остальных алгоритмах. Можно также использовать критерий остановки следующего вида:

$$\frac{1}{n+1} \sum_{i=1}^{n+1} |f(x_i^k) - f(x_{n+2}^k)|^{\frac{1}{2}} < \epsilon.$$

Выбор коэффициентов  $\alpha, \beta, \gamma$  обычно осуществляется эмпирически. После того, как многогранник подходящим образом промасштабирован, его размеры должны поддерживаться неизменными пока изменения в топологии задачи не потребуют многогранника другой формы. Чаще всего рекомендуют  $\alpha = 1$ ,  $\beta \in [0.4, 0.6]$ ,  $\gamma \in [2, 3]$ .

#### **5.Метод Пауэлла.**

Этот метод использует свойство квадратичной функции, заключающееся в том, что любая прямая, которая проходит через точку минимума функции, пересекает под равными углами касательные к поверхностям равного уровня функции в точках пересечения.

Переход из точки  $x^k$  в точку  $x^{k+1}$  осуществляется по следующему алгоритму:

1) Выполняют  $n$ -одномерных поисков вдоль направлений  $p_i^k, i = \overline{1, n}$  (за первоначальные направления принимают направления осей координат), причем каждый следующий поиск производится из точки, полученной из на предыдущем направлениию:

$$x_1^{k+1} = x_1^k + \lambda_1 p_1^k, \quad \lambda_1 = \arg \min_{\alpha} f(x^k + \alpha p_1^k),$$

$$x_i^{k+1} = x_{i-1}^{k+1} + \lambda_i p_i^k, \quad \lambda_i = \arg \min_{\alpha} f(x_i^{k+1} + \alpha p_i^k), \quad i = \overline{2, n}.$$

Фактически осуществляется покоординатный поиск по сопряженным направлениям.

2) Производится выбор новых направлений, построение которых базируется на следующей теореме.

**Теорема:** Если при начальной точке  $x^0$  поиска в направлении вектора  $p$  минимум функции  $f(x)$  находится к точке  $x^a$ , а при начальной точке  $x^1 \neq x^0$  поиск минимума функции  $f(x)$  в том же направлении  $p$  приводит к точке  $x^b$ , то при  $f(x^b) < f(x^a)$  направление  $x^b - x^a$  сопряжено с направлением поиска  $p$ .

**Определение** В общем случае система  $n$  линейно независимых направлений  $p_1, p_2, \dots, p_n$  называется сопряженной по отношению к некоторой положительно определенной матрице  $H$ , если

$$p_i H p_j = 0, \quad 1 \leq i \neq j \leq n$$

Поиск, осуществляемый на первом этапе, может привести к линейно зависимым направлениям, если, например, в одном из направлений  $p_i^k$  не удастся найти меньшего значения функции. Вследствие этого 2 направления могут стать коллинеарными. Отсюда вытекает, что в системе сопряженных направлений не следует заменять старое направление на новое, если после такой замены направления нового набора становятся линейно зависимыми.

Для такой проверки из точки  $x_n^{k+1}$  делается дополнительный шаг в направлении  $x_n^{k+1} - x^k$  и выполняется проверка. Если

$$f(2x_n^{k+1} - x^k) \geq f(x^k),$$

система направлений не изменяется и переход к точке  $x^{k+1} = x_n^{k+1}$  завершен, в противном случае система направлений изменяется

$$p_i^{k+1} = p_i^k, \quad i = \overline{1, n-1}, \quad p_n^{k+1} = x_n^{k+1} - x^k,$$

и окончательно осуществляют переход к точке  $x^{k+1}$  по правилу

$$x^{k+1} = x_n^{k+1} + \lambda_n p_n^k, \quad \lambda_n = \arg \min_{\alpha} f(x_n^{k+1} + \alpha p_n^{k+1}).$$

Таким образом, в результате выполнения рассмотренной процедуры осуществляется поочередная замена принятых вначале направлений поиска. В итоге после  $n$  шагов они окажутся взаимно сопряженными.

**6. Одномерный поиск.** Большинство методов поиска для функций нескольких переменных, как и метод п.1, сводится по существу к решению ряда задач поиска минимума функций одной переменной (одномерный поиск). К таким задачам приводят, как мы видели в §1, §2, и методы первого и второго порядков. Ниже описаны два эффективных метода одномерного поиска для унимодальной функции. Функция  $f(x)$ , определенная и непрерывная на отрезке  $[a, b]$ , называется унимодальной на этом отрезке, если существуют числа  $\alpha, \beta, a \leq \alpha \leq \beta \leq b$ , такие, что

- 1)  $f(x)$  строго монотонно убывает на  $[a, \alpha]$  (если  $a < \alpha$ );
- 2)  $f(x)$  строго монотонно возрастает на  $[\beta, b]$  (если  $\beta < b$ );
- 3)  $f(x)$  постоянна на  $[\alpha, \beta]$ .

В частности, если  $\alpha = \beta$ , то  $f(x)$  называется строго унимодальной на  $[a, b]$  (см.[4], с.13). Очевидно, что  $[\alpha, \beta]$  - множество точек минимума функции  $f(x)$  на отрезке  $[a, b]$ .

Методы поиска для унимодальных функций основаны на следующем их свойстве. Пусть  $f(x)$  унимодальна на  $[a, b]$ , а  $[\alpha, \beta]$ - отрезок из приведенного выше определения унимодальности. Назовем отрезком неопределенности (в задаче минимизации  $f(x)$  на  $[a, b]$ ) любой принадлежащий  $[a, b]$  отрезок, имеющий непустое пересечение с неизвестным отрезком  $[\alpha, \beta]$  точек минимума функции  $f(x)$ . Вычислив значения  $f(x^1)$  и  $f(x^2)$  в любой паре точек  $x^1, x^2 \in [a, b], x^1 < x^2$ , можно найти отрезок неопределенности более узкий, чем первоначальный отрезок  $[a, b]$ . Действительно, если  $f(x^1) < f(x^2)$ , то это будет отрезок  $[a, x^2]$ , если  $f(x^1) > f(x^2)$ , то таким отрезком будет отрезок  $[x^1, b]$ , а в случае  $f(x^1) = f(x^2)$  - отрезок  $[x^1, x^2]$ .

Используя последовательно указанное свойство, можно постепенно сужать отрезок неопределенности. Правило сужения определяется законом выбора точек  $x^1, x^2$ , в которых должна быть вычислена функция  $f(x)$ . Метод поиска тем эффективней, чем быстрее убывает при возрастании числа  $m$  отношение  $E_m = \Delta_m / \Delta_0$ , где  $\Delta_0$ -длина

первоначального отрезка неопределенности, а  $\Delta_m$  - длина отрезка неопределенности, полученного после  $m$  вычислений функции.

Критерием окончания процесса поиска может служить достижение отрезком неопределенности некоторой наперед заданной длины  $\delta$ . При этом за точку минимума приближенно принимается средняя точка последнего отрезка неопределенности.

**А) Метод равномерного поиска.**

На интервале  $[a, b]$  задается  $N+1$ -равномерно расположенных точек  $x_i = a + i \frac{b-a}{N}$ ,  $i = \overline{0, N}$  в которых вычисляются значения минимизируемой функции. Ищется точка

$$x^k = \arg \min_{i=\overline{0, N}} f(x^i),$$

и если длина интервала  $[x^{k-1}, x^{k+1}]$  меньше заданного  $\epsilon$  то за точку минимума может быть взята точка  $x^k$ , в противном случае интервал  $[x^{k-1}, x^{k+1}]$  делится равномерно.

Показатель эффективности метода равен  $E_m = \frac{2}{N+1}$

**Б) Метод половинного деления.** В этом методе точки  $x^1$  и  $x^2$  на каждом отрезке неопределенности  $[a, b]$  выбираются по правилу

$$x^1 = \frac{a+b}{2} - \frac{\epsilon}{2}, x^2 = \frac{a+b}{2} + \frac{\epsilon}{2},$$

где  $\epsilon > 0$  - некоторое достаточно малое наперед выбранное число. Показатель эффективности метода равен  $E_m = 2^{-m/2} + (1 - 2^{-m/2})\epsilon(b-a)^{-1}$ .

**В) Метод золотого сечения.** Поиск с помощью метода золотого сечения основан на разбиении отрезка неопределенности на две части, известном как "золотое сечение". При этом отношение длины всего отрезка к большей части равно отношению большей части к меньшей и равно числу  $\tau = 2^{-1}(1 + \sqrt{5}) \simeq 1.6118$  ( $\tau$ -корень уравнения  $\tau^2 = 1 + \tau$ ). В методе золотого сечения точки  $x^1$  и  $x^2$  на каждом отрезке неопределенности  $[a, b]$  выбираются по правилу

$$x^1 = b - (b-a)/\tau, x^2 = a + (b-a)/\tau.$$

Показатель эффективности метода равен  $E_m = \tau^{1-m}$ .

**Г) Метод Фибоначчи.** В методе Фибоначчи точки  $x^1$  и  $x^2$  на каждом отрезке неопределенности  $[a, b]$  выбираются по правилу

$$x^1 = a + \frac{F_n}{F_{n+2}}(b - a), \quad x^2 = \frac{F_{n+1}}{F_{n+2}}(b - a) = a + b - x^1.$$

Здесь  $\{F_n\}_{n=1}^\infty$ -последовательность чисел, определяемая рекуррентным соотношением

$$F_{n+2} = F_{n+1} + F_n, \quad F_1 = F_2 = 1$$

Нетрудно видеть, что алгоритм метода Фибоначчи очень похож на алгоритм метода золотого сечения. Интервал неопределенности сокращается точно так же, как в методе золотого сечения И на новой итерации вычисляется только одна новая точка и значение функции в ней и показатель эффективности метода равен  $E_m = F_m^{-1}$ .

Заметим, что с ростом  $n$  из-за того, что  $\frac{F_n}{F_{n+2}}$  - есть бесконечная десятичная дробь, возможно "искажение" метода и потеря интервала с точкой минимума (вследствие погрешностей вычислений). Следует также отметить, что при практическом применении метод золотого сечения по эффективности, скорости сходимости и точности получаемого решения практически не уступает методу Фибоначчи.

#### Д) Метод квадратичной интерполяции.

Здесь задаются пробные три пробные точки  $x^1 = (a + b)/2$ ,  $x^2$  и  $x^3$  так, чтобы они находились как можно ближе к точке минимума. Вычисляются значения функции в этих точках  $f(x^1)$ ,  $f(x^2)$ ,  $f(x^3)$ , затем строится квадратичный интерполяционный многочлен по трем точкам и его минимум принимается за точку  $x^1$ . Точка минимума квадратичного интерполяционного многочлена находится по формуле

$$x^* = \frac{1}{2} \frac{((x^2)^2 - (x^3)^2)f(x^1) + ((x^3)^2 - (x^1)^2)f(x^2) + ((x^1)^2 - (x^2)^2)f(x^3)}{(x^2 - x^3)f(x^1) + (x^3 - x^1)f(x^2) + (x^1 - x^2)f(x^3)}.$$

Для нахождения точки  $x^2$  задается шаг  $h > 0$  в положительном направлении от точки  $x^1$  и если  $f(x^1) > f(x^2)$ , то  $x^3 = x^1 + 2h$ , иначе  $x^3 = x^1 - h$ .

Если знаменатель в формуле для нахождения минимума квадратичного интерполяционного многочлена равен нулю, т.е. все три точки лежат на одной прямой рекомендуется выбрать за  $x^*$  точку в которой минимален вектор  $f(x^1)$ ,  $f(x^2)$ ,  $f(x^3)$ .

Критерии окончания в этом процессе либо интервал  $[x^1, x^*]$  мал, либо функция на интервале  $[x^1, x^*]$  меняется незначительно.

**Е) Эвристический алгоритм Свенна нахождения интервала унимодальности .**

Для выбранной исходной точки  $x^0$  на основе правила исключения строится относительно широкий интервал, содержащий точку оптимума. Используется эвристический подход в котором  $x^{k+1}$  пробная точка определяется по рекуррентной формуле

$$x^{k+1} = x^k + 2^k h, \quad k = 0, 1, 2, \dots,$$

где

$x^0$  - произвольно выбранная начальная точка;

$h$  - шаг поиска, знак которого может меняться на противоположный.

Знак  $h$  определяется путем сравнения значений  $f(x^0)$ ,  $f(x^0 + |h|)$  и  $f(x^0 - |h|)$ . Если  $f(x^0 - |h|) \geq f(x^0) \geq f(x^0 + |h|)$ , то согласно предположению об унимодальности, точка минимума должна располагаться правее точки  $x_0$  и величина  $h$  выбирается положительной. Если  $f(x^0 - |h|) < f(x^0) < f(x^0 + |h|)$ , то величину  $h$  следует выбирать отрицательной. Если  $f(x^0 - |h|) \geq f(x^0) \leq f(x^0 + |h|)$ , то точка минимума лежит между  $x^0 - |h|$  и  $x^0 + |h|$  и поиск граничных точек завершен. Случай, когда  $f(x^0 - |h|) \leq f(x^0) \geq f(x^0 + |h|)$ , противоречит предположению об унимодальности. Выполнение этого условия говорит о том, что функция не является унимодальной.

### Вопросы для самопроверки.

1. Каков угол между направлением спуска  $p^k$  и  $p^{k+1}$  в методе наискорейшего спуска?

2. Проведите два шага минимизации методом наискорейшего спуска, начиная с точки  $x^0 = \{1, 1\}$ , для функции  $f(x_1, x_2) = x_1^2 + 2x_2^2$ .

3. Пусть  $\{x^k\}$ -итерационная последовательность некоторого метода спуска для функции  $f(x)$ ,  $x \in R^n$ . Ломаную в  $R^n$ , соединяющую последовательно точки  $x^0, x^1, \dots, x^n, \dots$ , называют траекторией поиска. Начертите на плоскости  $(x_1, x_2)$  траекторию поиска предыдущей задачи.

4. Задача минимизации функции  $f(x_1, x_2) = x_1^2 + 2x_2^2$  решается методом п.2 Б) §1, причем  $x^0 = \{1, 1\}$ ,  $\epsilon = 0.5$ ,  $\alpha = 1$ ,  $\lambda = 0.5$ .

Сколько дроблений будет произведено для определения множителей  $\alpha_0$  и  $\alpha_1$ ? Начертите траекторию поиска.

5. Замедлится или ускорится поиск минимума в предыдущей задаче, если взять:

а)  $\epsilon = 0.5, \alpha = 1, \lambda = 0.1$ ;

б)  $\epsilon = 0.5, \alpha = 0.5, \lambda = 0.5$ ;

в)  $\epsilon = 0.9, \alpha = 1, \lambda = 0.5$ ;

г)  $\epsilon = 0.1, \alpha = 1, \lambda = 0.5$  ?

6. Определите начальное направление спуска по методу Ньютона для функции  $f(x_1, x_2) = x_1^2 + 2x_2^2$  из точки  $x^0 = \{1, 1\}$ . Найдите первое приближение. Сделайте то же самое для произвольной точки  $x^0$ .

7. Найдите первые три приближения к точке минимума функции  $f(x_1, x_2) = x_1^2 + x_2^4$  по методу Ньютона, взяв  $x^0 = \{1, 1\}$ . Сколько итераций потребуется, чтобы приблизиться к точке минимума на расстояние не большее, чем 0.01?

8. Найдите первое приближение к минимуму функции  $f(x_1, x_2) = x_1^2 + x_2^4$  по методу Ньютона-Рафсона, взяв  $x^0 = \{1, 1\}$  (модификация А)).

9. Найдите несколько первых итераций и постройте траекторию поиска минимума функции  $f(x_1, x_2) = x_1^2 + x_2^4$  по методу Ньютона-Рафсона (модификация Б)), взяв  $x^0 = \{1, 1\}, \epsilon = 1/3, \lambda = 1/2$ .

10. Проведите два шага минимизации методом сопряженных градиентов, начиная с точки  $x^0 = \{1, 1\}$ , для функции  $f(x_1, x_2) = x_1^2 + 2x_2^2$ . Постройте траекторию поиска.

11. Выполните задание пункта 10 для функции  $f(x_1, x_2) = x_1^2 + x_2^4$ .

12. Проведите два шага минимизации методом покоординатного спуска, начиная с точки  $x^0 = \{1, 1\}$ , для функции  $f(x_1, x_2) = x_1^2 + 2x_2^2$ . Постройте траекторию поиска.

13. Сколько вычислений функции нужно сделать, чтобы уменьшить длину первоначального отрезка неопределенности  $[0, 1]$  в 10 раз; а) методом половинного деления для  $\epsilon = 0.01, \epsilon = 0.05$ ; б) методом золотого сечения ?

14. Вычислите показатель овражности в точке минимума для функции  $f(x_1, x_2) = x_1^2 + 2x_2^2$  (точка минимума -  $\{0, 0\}$ ) и для функции Розенброка  $f(x_1, x_2) = 100[x_2 - x_1^2]^2 + (1 - x_1)^2$  (точка минимума -  $\{1, 1\}$ ).

15. Вычислите показатель овражности в точке минимума для

функции  $f(x_1, x_2) = 2^{-1}(x_1^2 + Lx_2^2)$ , где  $L > 1$  (точка минимума -  $\{0, 0\}$ ). Взяв за начальную точку  $\{1, 1/L\}$ , найдите итерационную последовательность метода наискорейшего спуска. Проверьте, что расстояние элементов этой последовательности до точки минимума убывает как геометрическая прогрессия со знаменателем  $(L - 1)/(L + 1)$ . Налицо проявление овражного эффекта - чем больше число  $L$  и показатель овражности, тем медленнее сходимость. Этот пример показывает также, что, вообще говоря, метод наискорейшего спуска сходится не быстрее, чем с линейной скоростью.

### Задания лабораторной работы.

В лабораторной работе требуется по указанию преподавателя выполнить некоторые из следующих заданий.

1. Для заданной функции провести теоретическое исследование применимости заданного метода минимизации.

2. Для заданной функции  $n$  переменных при заданном критерии окончания и заданной точности  $\bar{\delta}$  решить численно указанным методом задачу безусловной минимизации:

а) составить программу решения задачи минимизации для произвольной функции  $n$  переменных (алгоритм оформить в виде подпрограммы);

б) провести отладку программы на какой-либо "простой" функции (например,  $f(x_1, x_2) = x_1^2 + 2x_2^2$ );

в) с помощью одной из программ, приведенных в приложении, нарисовать картинку линий уровня минимизируемой функции в окрестности предполагаемой точки экстремума (если  $n = 2$ );

г) используя результаты расчетов на ЭВМ, начертить траекторию поиска, наложив ее на рисунок линий уровня (если  $n = 2$ );

д) проанализировать полученные результаты.

3. Исследовать с помощью ЭВМ зависимость числа итерации, необходимых для численного решения задачи минимизации заданной функции указанным методом, от:

а) выбора начальной точки;

б) параметров метода;

в) выбора критерия окончания;

г) величины  $\delta$ .

5. Сравнить поведение различных методов при численном решении одной и той же задачи минимизации.

## Требования к программам.

Подпрограмму, реализующую алгоритм минимизации, необходимо сделать независимой от конкретного вида минимизируемой функции. Этого можно достичь выносом вычисления целевой функции и производных от нее в отдельные подпрограммы. Например, так, как это сделано ниже.

### ВЫЧИСЛЕНИЕ ЦЕЛЕВОЙ ФУНКЦИИ

```
real function  $f(x, n)$ 
real  $x(n)$ 
 $f = \dots$ 
return
end
```

### ВЫЧИСЛЕНИЕ ГРАДИЕНТА

```
subroutine grad( $gr, x, n$ )
real  $x(n), gr(n)$ 
 $gr(1) = \dots$ 
 $gr(2) = \dots$ 
 $\dots$ 
return
end
```

Здесь многоточием заменены операторы, отвечающие за вычисление функции ( $f = \dots$ ) и частных производных ( $gr(1) = \dots, gr(2) = \dots$ ). Теперь во всех случаях, когда необходимо вычислить, например, значение функции, достаточно в программе в соответствующем месте вставить оператор  $y = f(x, n)$ . При необходимости сменить функцию, достаточно заменить в подпрограмме  $f$  и grad операторы, отмеченные многоточием, оставив без изменений "скелет" подпрограммы.

## Список тестовых функций.

функция $f(x, y)$	точка минимума	начальная точка
1. $(2/3)x^3 + (1/3)y^3 - x^2y + xy^2 - 5x$	{1.925, 0.8}	{3, 2}
2. $100(y - x^2)^2 + (1 - x)^2$	{1, 1}	{-1.2, 1}
3. $(y - x^2)^2 + (1 - x)^2$	{1, 1}	{0.5, 0.5}
4. $100(y - x^3)^2 + (1 - x)^2$	{1, 1}	{-1.2, 1}
5. $(y - x^2)^2 + 100(1 - x)^2$	{1, 1}	{-1.2, 1}
6. $(1.5 - x(1 - y))^2 + (2.25 - x(1 - y)^2)^2 + (2.625 - x(1 - y^3))^2$	{3, 0.5}	{2.0}
7. $(x^2 + y - 11)^2 + (x + y^2 - 7)^2$	а) {3, 2} б) {3.58, -1.85} в) {-3.78, -3.29} г) {-2.81, 3.13}	а) {1, 1} б) {-2, 1} в) {-2, 4} г) {-4, 3}

## ПРИЛОЖЕНИЕ

Приведем описание программы PICT, предназначенной для распечатки на экране линий уровня функций двух переменных  $f(x, y)$ . В начале программа PICT по заданному прямоугольнику  $\Pi = [x^{\min}, x^{\max}] * [y^{\min}, y^{\max}]$  и заданному размеру рисунка  $fh * fu$  ( $fh$  - количество символов по горизонтали;  $fu$  - количество символов по вертикали) определяет шаги по координатам  $x$  и  $y$  по формулам

$$h_x = (x^{\max} - x^{\min}) / (fh - 1), h_y = (y^{\max} - y^{\min}) / (fu - 1).$$

Результатом выполнения программы PICT будет отображение на экране шести линий уровня, соответствующих значениям

$$f_j = f_{\min} + (j - 1)(f_{\max} - f_{\min}) / 5, j = 1, 2, 3, 4, 5, 6.$$

Здесь  $f_{\max} = \max f(x, y)$ ,  $f_{\min} = \min f(x, y)$ , где максимум и минимум берутся по "сеточному прямоугольнику"  $\Pi_h$ , образованному узлами сетки с шагами  $h_x$  и  $h_y$ ,  $\Pi_h \subset \Pi$ . На экран также выводятся значения  $f_j$  и символы, с помощью которых "рисуются" соответствующие линии уровня.

Обращение к программе:

call PICT(fnc, k, fh, fu, xm, xn, ym, yn)

Входные параметры:

fnc - имя внешней подпрограммы, с помощью которой вычисляются значения  $f(x, y)$ . Подпрограмма-функция fnc должна иметь

структуру, аналогичную той, которая приведена в рекомендациях по оформлению программ. Ее необходимо описать оператором EXTERNAL;

$k$  - параметр целого типа, задающий количество строк между соседними выпечиваемыми координатными линиями, параллельными оси  $x$ ;

$fh$  - переменная целого типа, задающая число символов в строке. Если ввести значение  $fh$ , лежащее вне полуинтервала  $[0,117)$ , то программа скорректирует его, положив  $fh = 61$ ;

$fu$  - переменная целого типа, задающее число строк в рисунке. Если  $fh < 1$ , то происходит коррекция  $fu = 61$ ;

$xm, xn$  - соответственно левая и правая границы  $\Pi$ , переменные вещественного типа;

$ym, yn$  - соответственно нижняя и верхняя границы  $\Pi$ , переменные вещественного типа.

Выходные параметры: отсутствуют.

Требуемые подпрограммы: fnc.

Текст программы приводится ниже.

**Замечание.** Программа на самом деле "рисует" не линии уровня, соответствующие значению функции  $f_j$ , а некоторую "полоску" состоящую из точек  $\{x, y\}$  сеточного прямоугольника  $\Pi_h$ , удовлетворяющих условию  $f_j - w < f(x, y) < f_j + w$ ,  $w$  - внутренний параметр программы,  $w = (f_{\max} - f_{\min})/40$ .

```
external f
integer fh, fu
fh = 31
fu = 41
k = 5
xm = -4
xn = 4
ym = -2
yn = 2
call pict(f, k, fh, fu, xm, xn, ym, yn)
end
real function f(x, n)
real x(n)
f = 0.25 * x(1) ** 2 + x(2) ** 2
return
```

```

end
subroutine pict(fnc,f, fh, fu,xmin,xmax,ymn,ymx)
integer fh, fu, f, d, d1
real x(2), fl(9)
real*8aa(2), ab, ac,ad(2)
character*1 fmt(16)
logical*1cl(117), c(6),blk,pnt
equivalence (fmt(1), aa(1)), (x(1), xx), (x(2), y),
* (fmx,fl(1)), (fmx,fl(6))
data aa/'(g14.5,' , 117a1)'/
1      ,ab/'(g14.5,' / ,ac/'(14x,' /
2 ,ad/'(7x, 10(' , f10.2))'/
data c/' +', ' x', ' y', ' z', ' g', ' *' /,blk,pnt/' , ' /
if(fh.le.1.or.fh.gt.117)fh = 61
if(fu.le.1)fu = 61
hx = (xmax-xmin)/(fh - 1)
hy = (ymx-ymn)/(fu - 1)
xx =xmin
y =ymn
fmx=fnc(x, 2)
fmx=fmx
fmx=fmx
y =ymn-hy
do 5j = 1, fu
y = y + hy
xx =xmin-hx
do 4i = 1, fh
xx = xx + hx
fl(3) =fnc(x, 2)
fmx=amin1(fmx,fl(3))
4 fmx=amax1(fmx,fl(3))
5 continue
werr=(fmx-fmx)/5.
do 15i = 2, 5
15    fl(i) =fmx+(i - 1)*werr
werr=werr/8.
write(*, 90)(c(i), fl(i), i = 1, 6),xmin,xmax,hx,ymn,ymx,hy
do 16i = 1, fh
16    cl(i) =blk

```

```

y = ymx + hy
do 50i = 1, fu
y = y - hy
if(mod(i, f).ne.1) go to 20
aa(1) = ab
do 18k = 1, fh
18    cl(k) = pnt
go to 22
20 do 21k = 1, fh
21cl(k) = blk
aa(1) = ac
22    xx = xmin - hx
do 40j = 1, fh
xx = xx + hx
if(mod(j, 10).eq.1)cl(j) = pnt
do 30k = 1, 6
if(abs(fnc(x, 2) - fl(k)).ge.werr) go to 30
cl(j) = c(k)
30 continue
40 continue
if (aa(1).eq.ab) write(*,fmt)y, (cl(k), k = 1, fh)
if(aa(1).eq.ac) write(*,fmt)(cl(k), k = 1, fh)
50 continue
aa(1) = ad(1)
aa(2) = ad(2)
j = 0
xmin = xmin - hx
do 55i = 1, fh
xmin = xmin + hx
if(mod(i, 10).ne.1) go to 55
j = j + 1
fl(j) = xmin
55 continue
write(*,fmt)(fl(i), i = 1, j)
return
90 format(' обозначения'//6(2x, a1, 2x, g12.5//) /
12x, ' xmin=' , g10.3, 5x, ' xmax=' , g10.3, 5x, ' hx =' , g10.3 /
22x, ' ymin=' , g10.3, 5x, ' ymax=' , g10.3, 5x, ' hy =' , g10.3//)

```

end

На следующей странице на рис.1 приводится картинка линий уровня функции

$$f(x, y) = 0.25x^2 + y^2$$

в прямоугольнике  $\Pi = [-4, 4] \times [-2, 2]$ , полученная при помощи программы PIST. В верхней части распечатки дана таблица значений функции  $f(x, y)$  (на каждой из шести линий уровня) и символов, которыми изображается соответствующая линия уровня. Ниже напечатаны значения

$$x^{\min}, x^{\max}, y^{\min}, y^{\max}, hx, hy.$$

Распечатка получена при следующих значениях входных параметров:

$$fh = 31, fu = 41, k = 5, xm = -4, xn = 4, ym = -2, yn = 2.$$

Заметим, что изображения линий уровня несколько "размыты" из-за того, что программа на самом деле "рисует" не линии уровня, соответствующие значению функции  $f$ , а некоторую "полоску" состоящую из точек сеточного прямоугольника  $\Pi_h$ , каждая из которых удовлетворяет условию

$$|f(x, y) - f_j| < w.$$

Значение  $w$ , являющееся внутренним параметром программы, выбрано равным  $w = (f_{\max} - f_{\min})/40$ .

```

обозначения:
+ 0.17615E-11
x 1.6000
y 3.2000
z 4.8000
g 6.4000
* 8.0000
xmin=-4.00      xmax= 4.00      hx= 0.267
ymin=-2.00      ymax= 2.00      hy= 0.100

2.0000  *..g....zz.....zz....g..*
.      z .      . z      .
. g z . yyyuy . z g .
.g z yyy yyy z g.
g z yu yu z g
1.5000  g..z...yy.....yy...z..g
. z y . xxxxx . y z .
.z y .xxxxxxxxx. y z.
.z y xx xx y z.
1.0000  ....y....xx.....xx....y...z
z y xx. .xx y z
z y xx . .xx y z
0.50000 ..y...x.....x...y..
. y x . +++ . x y .
. y x . +++++ . x y .
. y xx . ++++++ . xx y .
0.12442E-05..y..xx.....+++++.....xx..y..
. y xx . ++++++ . xx y .
. y x . +++++ . x y .
. y x . +++ . x y .
-0.50000 ..y...x.....x...y..
z y xx . .xx y z
z y xx. .xx y z
-1.00000 z...y....xx.....xx....y...z
.z y xx xx y z.
.z y .xxxxxxxxx. y z.
. z y . xxxxx . y z .
-1.5000  g..z...yy.....yy...z..g
.g z yyy yyy z g.
. g z . yyyyyyy . z g .
.      z .      . z      .
-2.0000  *..g....zz.....zz....g..*
-4.00    -1.33    1.33    4.00

```

Рис. 1: Линии уровня функции  $f(x, y) = 0.25x^2 + y^2$ , нарисованные на экране программой PICT

Для отображения линий уровня функции

$$f(x, y) = 0.25x^2 + y^2$$

можно использовать и интегрированные системы, например, MATLAB. Здесь достаточно выполнить операторы

```
clear
[x,y]=meshgrid([-4:0.1:4],[-2:0.1:2]);
f=0.25* x.^2+ y.^2;
c=contour(x,y,f);
clabel(c)
xlabel('x')
ylabel('y').
```

В результате будем иметь на экране картину, приведенную на рис.2

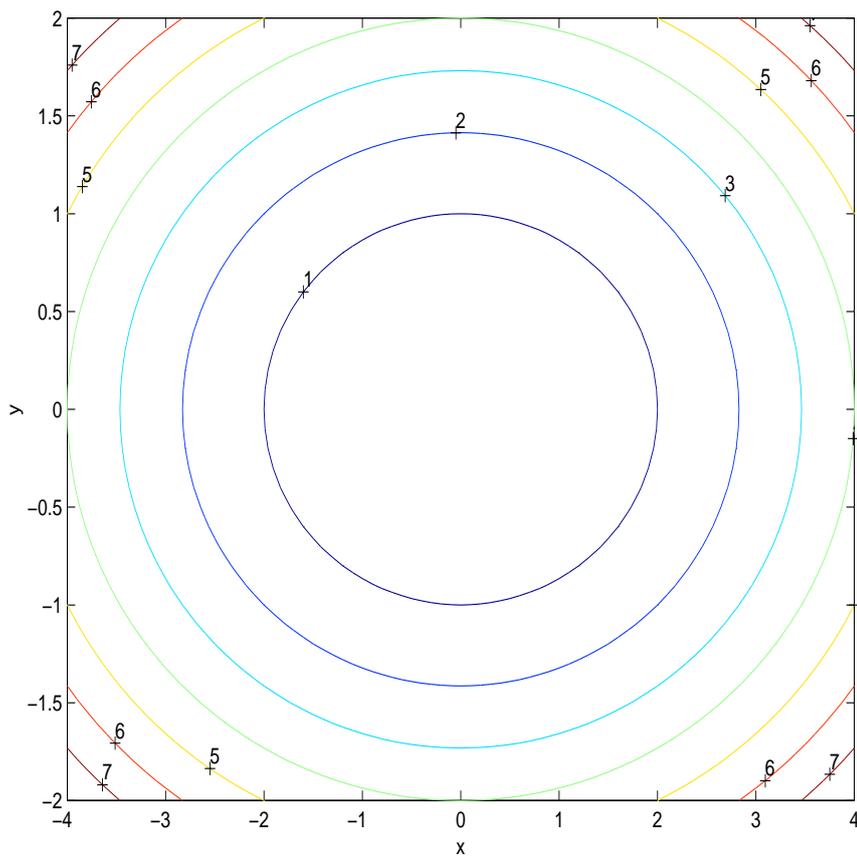


Рис. 2: Линии уровня функции  $f(x, y) = 0.25x^2 + y^2$

## ЛИТЕРАТУРА

1. Моисеев Н.Н., Иванюков Ю.П., Столярова Е.М. Методы оптимизации. М.: Наука, 1978.
2. Пшеничный Б.П., Данилин Ю.М. Численные методы в экстремальных задачах. М.: Наука, 1975.
3. Химмельблау Д. Прикладное нелинейное программирование. М.: Мир, 1975.
4. Васильев Ф.П. Численные методы решения экстремальных задач. М.: Наука, 1988.
5. Поляк Б.Т. Введение в оптимизацию. М.: Наука, 1983.

# БЕЗУСЛОВНЫЙ ЭКСТРЕМУМ ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ

Составители:

Кулинич

Виктор Валентинович

Новоженов

Михаил Михайлович

Сумин

Владимир Иосифович

Подписано к печати . Формат 60x84 1/16.

Печать офсетная. Бумага оберточная. Усл.печ.л. 1.7.

Тираж 500 экз. Заказ . Бесплатно.

Нижегородский государственный университет им.Н.И.Лобачевского.

603600 ГСП-20, Н.Новгород, просп.Гагарина, 23.

Типография ННГУ. 603600, Н.Новгород, ул.Б.Покровская, 37.