МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(государственный технический университет)

Кафедра 308

Дисциплина

«Теория и методы оптимизации»

Лабораторная работа №2

«Алгоритмы прямого поиска в многомерной минимизации»

(методические указания)

Объем – 8 часов

Составил доцент, к.т.н. В.Г.Герасименко

Москва, 2002г.

Работа 2. АЛГОРИТМЫ ПРЯМОГО ПОИСКА в многомерной минимизации

1. ВВЕДЕНИЕ

Методы безусловной минимизации для многомерных задач оптимизации можно разделить на три класса в соответствии с типом информации, используемой при реализации того или иного метода:

- методы прямого поиска (или методы нулевого порядка), основанные на вычислении только значений целевой функции *f(x)*;
- градиентные методы (методы первого порядка), в которых используются точные значения первых производных *f(x)*, а также, если это необходимо, и значения самой функции *f(x)*;
- методы ньютоновского типа (или методы второго порядка), когда при минимизации используется информация о значениях производных второго порядка.

В данной лабораторной работе рассматриваются методы прямого поиска, для реализации которых требуются только значения целевой функции. В дальнейшем мы будем предполагать, что к функции *f(x)* предъявляется лишь одно требование - она должна быть только непрерывна, т.е. ее можно вычислить в любой точке области переменных. Что касается ее градиента, то он может существовать, либо нет-для прямых методов это не имеет принципиального значения.

На разработку методов прямого поиска минимума было затрачено много усилий. Можно предполагать, что одним из первых методов прямого поиска был метод *покоординатного* спуска (или *релаксационный* метод). На рис.2.1 представлен пример покоординатного поиска минимума двумерной целевой функции *f(x)*, *линии равного уровня* которой $f(x)=C_i$, i=1,2,3 представляют собой эллипсы.

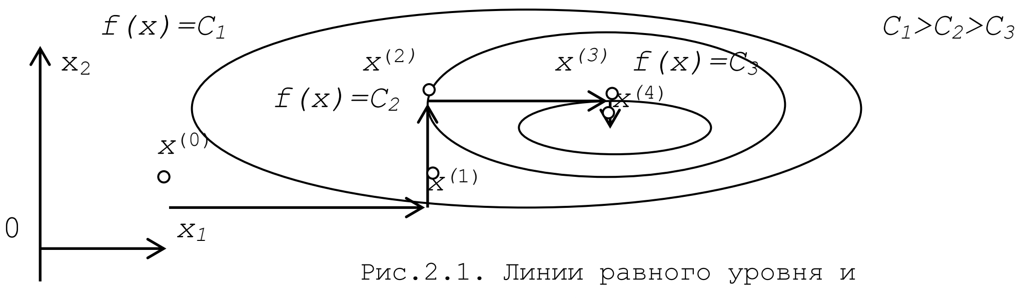


Рис.2.1. Линии равного уровня и траектория поиска методом покоординатного спуска.

Метод работает так. Из начальной точки поиска $x^{(0)}$ производится *одномерный поиск* вдоль координатной оси x_1, и находится точка $x^{(1)}$.

Затем из этой точки проводится также одномерный поиск вдоль второй координатной оси x_2, затем снова вдоль оси x_1 и т.д. В результате траектория поиска приходит в точку минимума, или в ее окрестность, что определяется точностью применяемого метода одномерного поиска.

Заметим, что в качестве алгоритмов одномерного поиска можно использовать любой из методов, описанных в *Работе 1* данного пособия. Теоретически данный метод эффективен лишь в случае достаточно гла-дких целевых функций. Кроме того *скорость его сходимости* невысока. Поэтому возникла необходимость разработки более эффективных методов прямого поиска.

С общих позиций, методы прямого поиска можно разделить на эвристические и теоретические.

Эвристические методы, как это следует из названия, реализуют процедуры поиска с помощью интуитивных представлениях о топологических свойствах *f(x)* и обеспечивают получение частных эмпирических результатов.

Теоретические методы основаны на фундаментальных математических теоремах и поэтому обладают такими операционными свойствами, как сходимость. Рассмотрим такие методы прямого поиска:

- поиск по симплексу (или S^2-метод)и его модификацию;
- метод Хука-Дживса;
- метод сопряженных направлений Пауэлла.

Первые два из перечисленных методов хотя и относятся к категории эвристических но реализуют принципиально различающиеся стратегии поиска. Метод Пауэлла основан на теории параллельных подпространств.

2. МЕТОД ПОИСКА ПО СИМПЛЕКСУ (S^2- МЕТОД)

Методы прямого поиска основаны на идее выбора некоторой начальной *базовой* точки и последующего вычисления значений *f(x)* в точках, окружающих базовую точку (рис.2.2).

Далее нужно выбрать *наилучшую точку*, т.е. точку с минимальным значением *f(x)* (пусть, например, это будет точка *5* на рис.2.2) и присвоить ей статус новой базовой точки, после чего итерация повторяется.

Заметим, что при таком симметричном расположении пробных точек, последующая итерация требует вычисления значений функции уже только в трех новых точках вместо пяти.

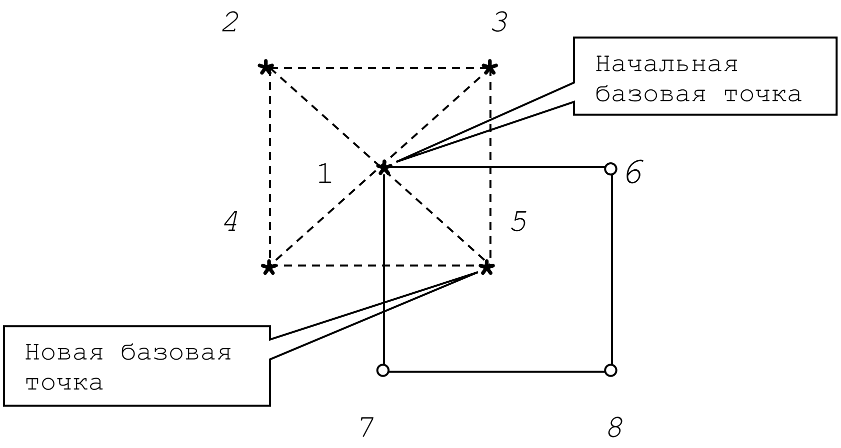


Рис.2.2. Итерации метода прямого поиска

Естественное желание минимизировать количество вычислений функции на каждой итерации привело к созданию ***симплекс-метода (или S^2-метода)***.

В этом методе в качестве множества пробных точек выбирается *минимально возможное* число точек, которое равно (n+1), где n-размерность задачи, причем их расположение определяется n-мерным регулярным *симплексом*[1]. Таким образом, для задач размерности n=2, минимальное количество пробных точек будет равно 3.

(Заметим, что указанннй метод не имеет отношения к симплекс-методу линейного программирования, а сходство названий носит случайный характер).

В S^2-алгоритме используется важное ***свойство симплексов***:

- *новый симплекс можно построить на любой грани начального симплекса путем переноса одной выбранной вершины на соответствующее расстояние вдоль прямой, проведенной через центр тяжести остальных вершин начального симплекса.*

Полученная таким образом точка является вершиной нового симплекса, а выбранная при построении вершина начального симплекса отбрасывается.

Рис.2.3 иллюстрирует процесс построения нового симплекса в двумерном случае в задаче минимизации.

---

[1] Симплекс(от лат.simplex-простой)-простейший выпуклый многогранник данного числа измерений n. Под двумерным симплексом понимают произвольный треугольник, а если это равномерный треугольник, то симплекс называется регулярным.

Предположим, что на рис.2.3,а) точка $x^{(1)}$ соответствует максимальному значению функции, т.е. она является *худшей* из трех и значит, в соответствии с алгоритмом подлежит *отражению* через центр тяжести оставшихся двух точек $x^{(2)}$ и $x^{(3)}$.

Такое отражение, а по существу перенос худшей точки $x^{(1)}$ может быть зеркальным или симметричным, когда расстояние от $x^{(1)}$ до центра тяжести (ЦТ) оставшихся равно расстоянию от ЦТ до новой точки $x^{(4)}$.

Именно такой случай иллюстрирует рис.2.3, б). Таким образом мы достаточно просто получили новый симплекс $[x^{(2)}, x^{(3)}, x^{(4)}]$.

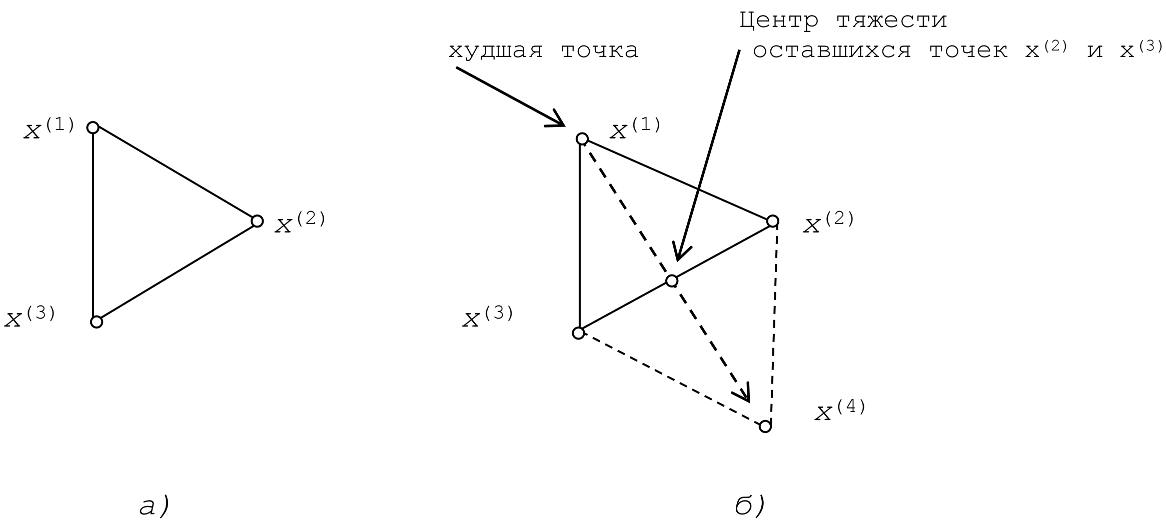


Рис.2.3. Построение нового симлекса в S^2- методе.

а)- начальный симплекс $x^{(1)}$, $x^{(2)}$, $x^{(3)}$.; б)- новый симплекс $x^{(2)}$, $x^{(3)}$, $x^{(4)}$.

Нетрудно заметить, что после формирования нового симплеса, для продолжения поиска минимума нам потребуется только одно новое вычисление целевой функции в новой точке $x^{(4)}$.

Работа алгоритма симплекс-метода начинается с построения регулярного симплекса в пространстве независимых переменных и вычисления значений целевой функции в каждой из вершин симплекса. При этом определяется вершина, которой соответствует наибольшее значение целевой функции. Затем найденная вершина отражается через центр тяжести остальных вершин симплекса в новую точку, которая используется в качестве вершины нового симплекса ипроцесс повторяется.

Формулы симплекс-алгоритма. Реализация рассматриваемого алгоритма основана на вычислениях трех типов:

(1) построение регулярного симплекса при заданных базовой точке и масштабном множителе;

(2) расчет координат отраженной точки;

(3) проверка «правила останова» при зацикливании симплекса.

Построение симплекса при известных начальной (базовой) точке $x^{(0)}$ и масштабном множителе α основано на следующей формуле

$$x^{(i)} = \begin{cases} x_j^{(0)} + \delta_1, & \text{если} \quad j \neq i, \\ x_j^{(0)} + \delta_2, & \text{если} \quad j = i, \end{cases}$$

для i и $j=1,2,3, \ldots ,N$.

Приращения δ_1 и δ_2, зависящие только от размерности задачи N и выбранного масштабного множителя α, определяются по формулам

$$\delta_1 = \left[\frac{(N+1)^{1/2} + N - 1}{N\sqrt{2}} \right] \alpha ,$$

$$\delta_2 = \left[\frac{(N+1)^{1/2} - 1}{N\sqrt{2}} \right] \alpha \ .$$

Величина масштабного множителя α выбирается исследователем, исходя из характеристик целевой функции. Например, при $\alpha=1$ ребра регулярного симплекса имеют единичную длину.

Вычисления второго типа, связанные с отражением относительно центра тяжести, также представляют несложную процедуру. Пусть $x^{(j)}$ – точка, подлежащая отражению, а центр тяжести остальных N точек расположен в точке

$$x_c = \tfrac{1}{N} \sum_{i=0, i\neq j}^{N} x^{(i)}.$$

Известно, что ***все точки прямой, проходящей через $x^{(j)}$ и x_c , задаются формулой***

$$x = x^{(j)} + \lambda (x_c - x^{(j)}) .$$

Действительно, при $\lambda=0$ получаем исходную точку $x^{(j)}$, тогда как значение $\lambda=1$ соответствует центру тяжести x_c. Проверьте, куда отразится точка при $\lambda=0.5$?

Как мы уже отмечали выше, для того, чтобы вновь построенный симплекс обладал свойством регулярности, отражение должно быть симметричным. Следовательно, новая вершина получается при $\lambda=2$. Таким образом, очевидно что:

$$x^{(j)}_{нов.}=2x_c-x^{(j)}_{предыдущ.}.$$

Теперь рассмотрим **вычисление «правила останова»**.

Итак, итерации продолжаются до тех пор, пока не начнется так называемое *циклическое движение по двум или более симплексам*. Причина циклического движения заключается в том, что либо очередным симплексом «накрывается» точка минимума, либо поиск остановился на некотором симметричном участке целевой функции. В таких ситуациях нужно использовать следующие правила.

- Если вершина, которой соответствует наибольшее значение целевой функции, построена на предыдущей итерации, то вместо нее берется вершина, которой соответствует следующее по величине значение целевой функции.
- Если некоторая вершина симплекса не исключается на протяжении более чем M итераций, то необходимо уменьшить размеры симплекса с помощью *коэффициента редукции* и построить новый симплекс, выбрав в качестве базовой ту точку, которой соответствует минимальное значение целевой функции. Исследователи из опыта предложили вычислять M по формуле

$$M = 1.65N + 0.05N^2,$$

  где N – размерность задачи, а M округляется до ближайшего целого числа. Для применения этого правила требуется установить коэффициент редукции.
- Если размеры симплекса, или разности между значениями функции в вершинах становятся достаточно малыми, то поиск можно завершить. Для применения этих правил необходимо задать величину параметра окончания поиска.

Выводы. Изложенный выше алгоритм S^2-метода имеет такие преимущества, как простоту логической структуры, невысокий уровень требований к объему памяти, устойчивость в работе даже в тех случаях, когда ошибка вычислений значений $f(x)$ велика, поскольку алгоритм оперирует наибольшими значениями функции в вершинах, а не наименьшими.

К недостаткам метода обычно относят следующие.

Не исключено возникновение трудностей, связанных с масштабированием, поскольку все координаты вершин симплекса зависят от одного и того же масштабного множителя α. Предварительное масштабирование переменных, т.е. приведение их к сравнимым между собой значениям обычно решает эту проблему.

Алгоритм работает слишком медленно, так как полученная на предыдущих итерациях информация не используется для ускорения поиска.

Не существует простого способа расширения симплекса, не требующего пересчета значений целевой функции во всех точках образца. Таким образом, если α по какой-либо причине уменьшается (например, если встречается область с узким “оврагом” или “хребтом”), то поиск будет продолжаться с уменьшенной величиной шага.

Рассмотрим модифицированный метод поиска по симплексу (метод Нелдера-Мида), который частично устраняет перечисленные выше недостатки. Учитывая, что вообще то не видно веских оснований для сохранения свойства *регулярности симплекса* в процессе поиска, то при отражении худшей точки существует возможность как его растяжения, так и сжатия.[2]

При расчетах по методу Нелдера и Мида используются вершины симплекса $x^{(h)}$ (которой соответствует наибольшее значение целевой функции $f^{(h)}$), $x^{(g)}$ (которой соответствует следующее по величине значение целевой функции $f^{(g)}$) и $x^{(L)}$ (которой соответствует наименьшее значение целевой функции $f^{(L)}$).

Напомним, что отражение вершины симплекса осуществляется по прямой

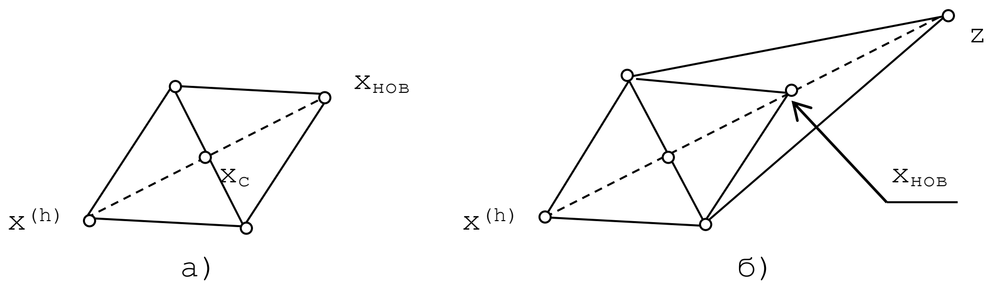
$$x = x^{(h)} + \lambda (x_c - x^{(h)}),$$

или

$$x = x^{(h)} + (1+\theta)(x_c - x^{(j)}).$$

При θ=1 имеет место так называемое *нормальное* отражение симплекса, поскольку точка $x_{нов.}$ располагается на расстоянии $|x_c - x^{(j)}|$ от точки x_c. Если $-1 \le \theta < 1$, то наблюдается *сжатие* симплекса, тогда как выбор $\theta > 1$ обеспечивает *растяжение* симплекса.

На рис. 2.4 приведены возможные варианты отражения в нерегулярном симплексе.



---

[2] По этой причине процедуру Нелдера и Мида иногда называют методом поиска по деформируемому многограннику.

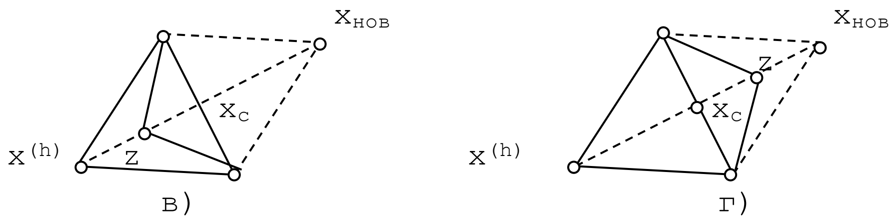


Рис. 2.4. Растяжение и сжатие симплекса

а)-нормальное отражение ($\theta=\alpha=1$), $f^{(L)}<f(x_{нов.})<f^{(g)}$;

б)-растяжение ($\theta=\gamma>1$), $f(x_{нов.})<f^{(L)}$;

в)-сжатие ($\theta=\beta<0$), $f(x_{нов.})>f^{(g)}$, $f(x_{нов.})\geq f^{(h)}$;

г)-сжатие ($\theta=\beta>0$), $f^{(g)}<f(x_{нов.})<f^{(L)}$.

Обозначим для удобства три значения параметра θ, используемые для нормального отражения, сжатия и растяжения через α, β и γ соответственно.

Реализация метода Нелдера-Мида начинается также с построения исходного симплекса, определения отражаемой точки и центра тяжести оставшихся. После нормального отражения осуществляется проверка правила останова, и если поиск не закончен, то на основе тестов окончательно выбирается один из возможных вариантов отражения. В качестве эмпирических значений параметров α, β и γ авторы рекомендуют использовать $\alpha=1$, $\beta=0.5$, $\gamma=2$.

Результаты многочисленных экспериментов показали, что этот метод является наиболее эффективным из всех методов поиска по образцу, если $N\leq 6$. Заметим также, что опыт применения метода показал, что *ориентация исходного симплекса* в отличие от его формы является одним из существенных факторов, влияющим на эффективность поиска, и, кроме того, некоторые авторы предлагают использовать несколько другие значения параметров $\alpha=2$, $\beta=0.25$, $\gamma=2.5$. Такой выбор параметров позволяет обеспечить хорошую работу алгоритма при повторении последовательных растяжений симплекса.

3. МЕТОД ХУКА-ДЖИВСА

Метод поиска Хука-Дживса стал результатом конструктивных попыток повышения эффективности прямого поиска за счет того, что

поиск, периодически проводимый вдоль направления

$$d^{(k)}=x^{(k)}-x^{(k-1)},$$

где $x^{(k)}$, $x^{(k-1)}$ соответственно последующая и предыдущая точки поиска, позволяет существенно ускорить сходимость итераций к точке минимума.

По существу процедура Хука-Дживса представляет собой комбинацию так называемого «*исследующего*» поиска с циклическим изменением переменных и «*ускоряющего*» поиска по образцу с использованием определенных эвристических правил.

Исследующий поиск ориентирован на выявление характера локального поведения целевой функции и определение направлений вдоль «оврагов».

Полученная в результате исследующего поиска информация затем используется в процессе поиска по образцу при движении по «оврагам».

Для проведения исследующего поиска необходимо задать величину шага вдоль каждого координатного направления, причем эта величина может быть разной, в зависимости от характера функции.

Поиск начинается в некоторой исходной, начальной точке, из которой делается пробный шаг. Если значение $f(x)$ в пробной точке не больше чем в исходной, то шаг поиска рассматривается как успешный, и фиксируется новая точка.

В противном случае необходимо вернуться к начальной точке и сделать шаг в противоположном направлении. После перебора всех N координат будет сформирована новая точка, которая называется *базовой* и исследующий поиск завершается.

Ускоряющий поиск по образцу заключается в реализации единственного шага из полученной базовой точки вдоль прямой, соединяющей эту точку с предыдущей базовой точкой.

Новая точка образца определяется по формуле

$$x_p^{(k+1)}=x^{(k)}+(x^{(k)}-x^{(k-1)}).$$

Как только движение по образцу не приводит к уменьшению целевой функции, точка $x_p^{(k+1)}$ фиксируется в качестве временной базовой точки и вновь проводится исследующий поиск.

Если в результате получается точка с меньшим значением функции, чем в точке $x^{(k)}$, то она рассматривается как новая базовая точка $x^{(k+1)}$. С другой стороны, если исследующий поиск неудачен, необходимо вернуться в точку $x^{(k)}$ и провести исследующий поиск с целью выявления нового направления минимизации.

Рано или поздно возникает ситуация, когда такой поиск уже не приводит к успеху. В этом случае требуется уменьшить величину исследующего шага и возобновить исследующий поиск.

Правило останова сравнивает требуемую точность с длиной исследующего шага, и прекращает поиск в случае, если длина шага не превышает точность.

Алгоритм Хука-Дживса можно представить в следующем виде.

1. Задать начальные условия и параметры поиска:

$x^{(0)}$-начальная точка;

Δ_i, $i=1,2,3,\ldots,N$-параметр исследующего шага;

$\alpha>1$ – коэффициент уменьшения шага;

$\varepsilon>0$ – параметр окончания поиска.

2. *Провести исследующий поиск.*
3. *Если исследующий поиск удачен, то перейти к 5;*
4. *Если $|\Delta x| < \varepsilon$,*

   то $x^*=x^{(k)}$; стоп.

   Иначе:

   уменьшить исследующий шаг в α раз;
   перейти к 2.
5. *Провести поиск по образцу:*

   $$x_p^{(k+1)}=x^{(k)}+(x^{(k)}-x^{(k-1)}).$$
6. *Провести исследующий поиск, используя $x_p^{(k+1)}$ в качестве базовой точки; пусть $x^{(k+1)}$ – полученная в результате точка.*
7. *Если $f(x^{(k+1)})<f(x^{(k)})$,*

   то $x^{(k-1)}=x^{(k)}$, $x^{(k)}=x^{(k+1)}$; перейти к 5.

   Иначе перейти к 4.

Несмотря на кажущуюся простоту представленного алгоритма, программная реализация метода Хука-Дживса требует определенного искусства и навыков, поэтому целесообразно представить текст бейсик-программы этого метода, которая поможет при составлении собственной программы.

Эта программа написана для минимизации квадратичной функции n=3-го порядка

$$f(x)=(x_1-2)^2+(x_2-5)^2+(x_3+2)^4.$$

Минимум этой функции находится в точке $(2.0,5.0,-2.0)^T$.

Текст программы минимизации
методом Хука-Дживса

```
10   PRINT "МЕТОД ХУКА-ДЖИВСА"
20   REM Целевая функция вычисляется в строке 2000
30   PRINT "Введите число переменных":INPUT N
40   DIM X(N),B(N),Y(N),P(N)
50   PRINT "Введите начальную точку X1,X2,… ,XN"
60   FOR I=1 TO N: INPUT X(I): NEXT I
70   PRINT "Введите длину исследующего шага": INPUT H
80   K=H: FE=0
90   FOR I=1 TO N
100  Y(I)=X(I): P(I)=X(I): B(I)=X(I): NEXT I
110  GOSUB 2000: FI=Z
120  PRINT "Начальное значение функции" Z
130  FOR I=1 TO N: PRINT X(I);"   ";:NEXT I:PRINT ""
140  PS=0: BS=0
150  REM Исследование вокруг базовой точки
180  J=1: FB=FI
200  X(J)=V(J)+K
210  GOSUB 2000
```

```
220  IF Z<FI THEN GOTO 280
230  X(J)=V(J)-K
240  GOSUB 2000
250  IF Z<FI THEN GOTO 280
260  X(J)=Y(J)
270  GOTO 290
280  Y(J)=X(J)
290  GOSUB 2000
300  FI=Z
310  PRINT “Исследующий поиск” Z
320  FOR I=1 TO N: PRINT X(I);”   “;:NEXT I:PRINT “”
330  IF J=N THEN GOTO 360
340  J=J+1
350  GOTO 200
360  IF FI<FB-1E-08 THEN GOTO 540
370  REM После оператора 360, если функция уменьшилась,
        произвести поиск по образцу
380  IF PS=1 AND BS=0 THEN GOTO 420
390  REM Но если исследование производилось вокруг точки шаблона PT,
395     и уменьшение функции не было достигнуто, то изменить базовую
        точку в операторе 420
400  REM В противном случае уменьшить длину шага в операторе 490
410  GOTO 490
420  FOR I=1 TO N:P(I)=B(I):Y(I)=B(I):X(I)=B(I):NEXT I
430  GOSUB 2000
440  FI=Z:ZB=Z
450  PRINT “Замена базисной точки”
460  FOR I=1 TO N:PRINT X(I);”   “;:NEXT I:PRINT “”
470  REM (следует за комментарием в строке 395)
         И провести исследование вокруг новой базисной точки
480  J=1:GOTO 200
490  K=K/10
500  PRINT “Уменьшить длину шага”
510  IF K<1E-08 THEN GOTO 700
520  REM Если поиск не закончен, то произвести новое
525  REM исследование вокруг новой базовой точки
530  J=1:GOTO 200
535  REM Поиск по образцу
540  FOR I=1 TO N:P(I)=2*Y(I)-B(I)
550  B(I)=Y(I):X(I)=P(I):Y(I)=X(I)
560  NEXT I
570  GOSUB 2000:FB=FI:PS=1:BS=0:FI=Z
580  PRINT “Поиск по образцу” Z
590  FOR I=1 TO N:PRINT X(I);”   “;:NEXT I:PRINT””
600  REM После этого произвести исследование вокруг
        последней точки образца
610  J=1:GOTO 200
700  PRINT “   Минимум найден”
710  FOR I=1 TO N:PRINT “X”I”=”P(I):NEXT I:PRINT “”
750  PRINT “МИНИМУМ ФУНКЦИИ РАВЕН” FB
760  PRINT “КОЛИЧЕСТВО ВЫЧИСЛЕНИЙ ФУНКЦИИ РАВНО” FE
790  END
2000 Z=(X(1)-2)^2+(X(2)-5)^2+(X(3)+2)^4
2010 FE=FE+1
2020 REM Счетчик количество вычислений функции
2030 RETURN
```

Приведем некоторые пояснения к приведенной программе.

В строке 360, где выясняется вопрос об изменении базисной точки мы избегаем уменьшения длины из-за возможного накопления ошибки (*ограниченная* точность вычислений компьютера) путем введения длины шага, равной 10^{-8} (в строке 510 можно задать вместо используемой длины шага 10^{-8} собственное значение).

Также отслеживается, где производится исследование-в базисной точке (BS=1,PS=0) или в точке образца (BS=0,PS=1). Как можно убедиться на практике, если не принимать такие меры предосторожности даже программа с удовлетворительной логикой не будет работоспособной.

Необходимо обратить внимание на организацию промежуточной печати результатов, что является необходимым атрибутом процесса отладки и тестирования программы.

Ясно, что программа будет работать быстрее, если удалить строки с операторами промежуточной печати. Подпрограмма, начинающаяся со строки 2000, вычисляет значение целевой функции.

Для начальной точки $(4.0, 5.0, 3.0)^T$ и начального шага длиной 1.0 ниже приведены промежуточные результаты.По ним можно легко проконтролировать правильность работы собственной программы.

```
МЕТОД ХУКА-ДЖИВСА
Введите число переменных              3
Введите начальную точку X1,X2,…,XN    4  -2   3
Введите длину шага                    1
Начальное значение функции  678
4     -2     3
Исследующий поиск    675
3     -2     3
Исследующий поиск    662
3     -1     3
Исследующий поиск    293
3     -2     2
Поиск по образцу     106
2      0     1
Исследующий поиск    106
2      0     1
Исследующий поиск     97
2      1     1
Исследующий поиск     32
2      1     0
Поиск по образцу       5
1      3    -2
Исследующий поиск      4
2      3    -2
Исследующий поиск      1
2      4    -2
Поиск по образцу      20
2      7    -4
Исследующий поиск     20
2      7    -4
Исследующий поиск     17
2      6    -4
Исследующий поиск      2
2      6    -3
Замена базисной точки  1
2      4    -2
Исследующий поиск      1
2      4    -2
Исследующий поиск      0
2      5    -2
Исследующий поиск      0
2      5    -2
    . . .
```

(…и, наконец, последний фрагмент распечатки)

```
Уменьшить длину шага
    Минимум найден
X1 = 2
X2 = 5
X3 =-2
МИНИМУМ ФУНКЦИИ РАВЕН 0
КОЛИЧЕСТВО ВЫЧИСЛЕНИЙ ФУНКЦИИ РАВНО 127
```

Применение одномерного поиска в алгоритме Хука-Дживса является вполне обоснованной модификацией поиска по образцу, позволяющий увеличить скорость сходимости. Основанием такой модификации является то, что если движение по образцу приводит к успеху, имеются основания для того, чтобы полностью использовать возможности движения вдоль прямой, определяемой образцом. Рис. 2.5 поясняет эту идею.

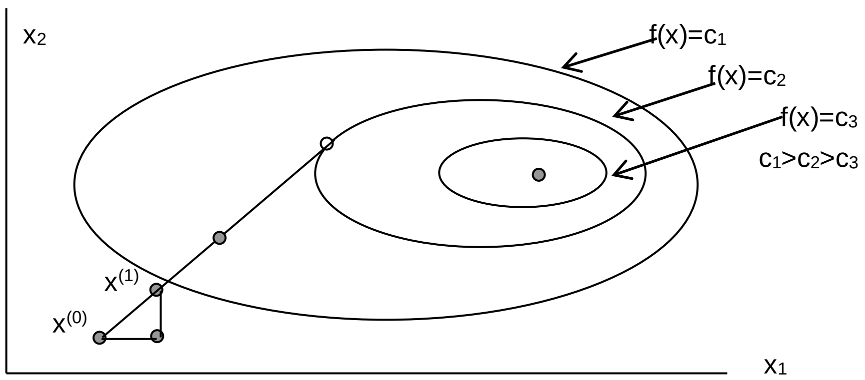


Рис. 2.5. Ускорение поиска по образцу с помощью одномерного поиска

4. Метод Пауэлла

Переходим к изучению ***метода Пауэлла***, который, как уже отмечалось выше является теоретическим методом и поэтому обладает наилучшими операционными характеристиками среди методов нулевого порядка.

При работе этого алгоритма информация, полученная на предыдущих итерациях, используется для построения векторов направления поиска, а также для устранения зацикливания последовательности координатных поисков.

Метод Пауэлла ориентирован на минимизацию квадратичных функций, хотя может быть использован для минимизации функций общего вида, но с меньшей эффективностью.

Метод опирается на следующее элементарное свойство квадратичных функций.

Свойство параллельного подпространства.

Пусть заданы квадратичная функция (или квадартичная форма) $q(x)$, две произвольные несовпадающие точки $x^{(1)}$ и $x^{(2)}$, а также некоторый вектор направления d (рис.2.5).

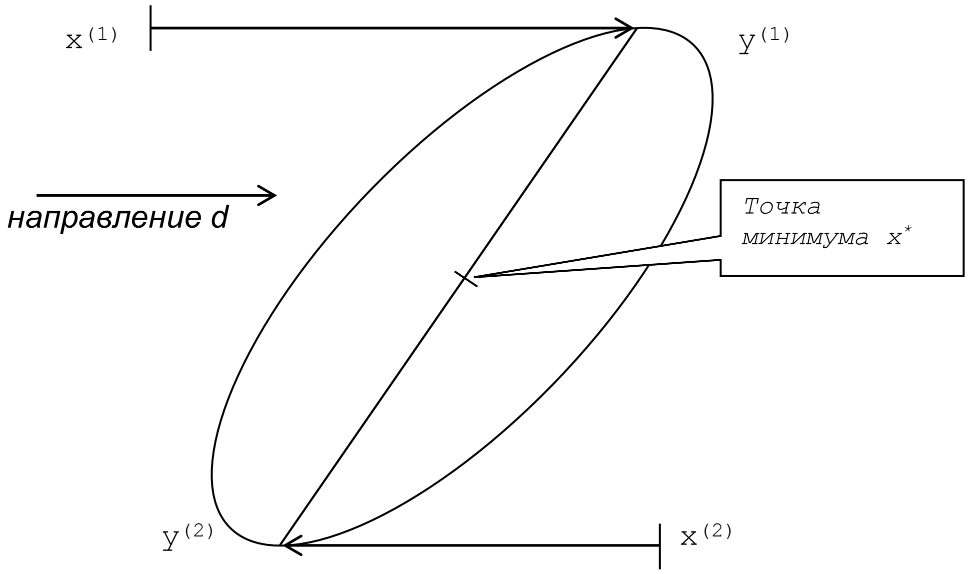


Рис. 2.5. Сопряженные направления на плоскости
(свойство параллельного подпространства)

Тогда, если точка $y^{(1)}$ минимизирует функцию $q(x)$ из $x^{(1)}$ вдоль направления d, т.е. $y^{(1)} = \min\ [q(x^{(1)}+\lambda d)]$, а точка $y^{(2)}$ минимизирует $q(x^{(2)}+\lambda d)$, то направление $(y^{(2)} - y^{(1)})$ *сопряжено* с направлением d, т.е.

$$(y^{(2)} - y^{(1)})^T \mathbf{Q} d = 0,$$

где **Q** – симметрическая матрица квадратичной функции;
T – знак транспонирования.

Доказательство свойства параллельного подпространства.

Как известно, квадратичную функцию общего вида в матричном виде можно записать так:

$$q(x) = (1/2)x^T\mathbf{Q}x + b^Tx + a.$$

Точки прямой, исходящей из точки $x^{(1)}$ в направлении d, задаются формулой $x = x^{(1)} + \lambda d$. Минимум $q(x)$ вдоль направления d определяется путем нахождения значения λ^*, при котором $\partial q / \partial \lambda = 0$.

Вычислим эту производную по правилу дифференцирования сложной функции:

$$\frac{\partial q}{\partial \lambda} = \frac{\partial q}{\partial x}\frac{\partial x}{\partial \lambda} = (b^T + x^T Q)d\ .$$

По предположению теоремы минимум достигается в точке $y^{(1)}$, следовательно,

$$[(y^{(1)})^{T}\boldsymbol{Q} + b^{T}]d = 0.$$

Аналогично, так как минимум q(x) при движении из точки $x^{(2)}$ в направлении d достигаетсчя в точке $y^{(2)}$, имеем

$$[(y^{(2)})^{T}\boldsymbol{Q} + b^{T}]d = 0.$$

Вычитая последнее уравнение из предпоследнего, получаем

$$(y^{(2)} - y^{(1)})^{T}\boldsymbol{Q}d = 0,$$

т.е. направления d и $y^{(2)}-y^{(1)}$) оказываются $\boldsymbol{Q}$–сопряженными, и свойство параллельного подпространства для квадратичных функций доказано.

Для практического использования метода Пауэлла не слишком удобно задавать две начальные точки и некоторое направление, поэтому предпочтительнее строить систему сопряженных направлений, исходя из одной начальной точки, что легко осуществить для N-мерных квадратичных функций при помощи единичных координатных векторов (или *ортов*) $e^{(1)}, e^{(2)}, \ldots, e^{(N)}$.

Конкретизируем этот вывод для случая N=2. При заданной начальной точке $x^{(0)}$ проведем одномерный поиск, например, вдоль $e^{(1)}$, получив точку $x^{(1)}$, из которой вновь проведем одномерный поиск но уже вдоль $e^{(2)}$, найдя точку $x^{(2)}$. Далее снова проведем одномерную минимизацию из $x^{(2)}$ по направлению $e^{(1)}$, получив точку $x^{(3)}$.

Можно показать, что тогда направления $(x^{(3)} - x^{(1)})$ и $e^{(1)}$ оказываются $\boldsymbol{Q}$–сопряженными,а искомый минимум q(x) определится с помощью последнего одномерного поиска вдоль направления $(x^{(3)} - x^{(1)})$.

Таким образом, в рассмотренном случае для локализации точки минимума достаточно провести ровно четыре одномерных поиска.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1. Покажите аналитически, что функция

$$f(x_1,x_2,x_3)=3x_1^2+2x_2^2+x_3^2-2x_1x_2-2x_1x_3+2x_2x_3-6x_1-4x_2-2x_3$$

является выпуклой.

5.2. Для какой-либо квадратичной двумерной функции изобразите несколько линий равного уровня. На фоне этих линий уровня покажите несколько начальных итераций поиска минимума методами:

- S-квадрат;
- Хука-Дживса.

Покажите на этом же рисунке, как изменятся указанные начальные итерации, если в методах при операциях отражения и ускоряющем поиске использовать одномерный поиск.

5.3. Найдите и классифицируйте стационарные точки функции

$$f(x_1,x_2)=2x_1^3+4x_1x_2^2-10x_1x_2+x_2^2.$$

С помощью какой-либо графической программы (например Mathcad) постройте линии уровня этой функции в окне координат

$$-3\le x_1,x_2\le 3.$$

Укажите на рисунке все найденные стационарные точки.

5.4. Предположим, что в результате поиска минимума функции

$$f(x_1,x_2)=[x_1^2+(x_2+1)^2][x_1^2+(x_2-1)^2]$$

были найдены следующие четыре точки:

$\mathbf{x}^{(1)}=(0,0)^T$, $\mathbf{x}^{(2)}=(0,1)^T$, $\mathbf{x}^{(3)}=(0,-1)^T$, $\mathbf{x}^{(4)}=(1,1)^T$.

Классифицируйте (аналитически) эти точки. Постройте линии уровня этой функции и на рисунке подтвердите полученные результаты.

5.5. Приведите к виду суммы полных квадратов q(x) следующую функцию

$$f(x)=4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1.$$

Сравните линии уровня функций f(x) и q(x).

Самостоятельно задайте начальную точку x^0 и графически реализуйте шаги минимизации *релаксационного* метода для f(x) и q(x).

5.6. **Задача.** Пусть необходимо сконструировать прямоугольный контейнер с крышкой для перевозки 400 м^3 сыпучего материала через реку. Стоимость рейса туда и обратно равна 4.2 $. Известно, что дно контейнера делается из материала стоимостью 15 $/м^2, боковые стенки-из материала стоимостью 5 $/м^2, а крышка – 25 $/м^2.

- Решите задачу минимизации полных затрат на перевозку груза.
- Постройте линии уровня целевой функции в области найденной экстремальной точки, подставив в целевую функцию какое-либо одно из найденных значений вектора решения.

6. ПРАКТИЧЕСКАЯ ЧАСТЬ

6.1. Проведите вычислительный эксперимент для изучения и сравнения характеристик следующих методов минимизации

- S-квадрат метод;
- метод Нелдера-Мида;
- метод Хука-Дживса.

Каждый из перечисленных методов минимизации нужно испытать на трех целевых функциях:

квадратичной форме f_1

$$f_1(x) = 12x_1^2 + 6x_1x_2 + 2x_2^2 - 2x_1 - x_2;$$

и функции Розенброка f_2, f_3 при двух разных значениях коэффициента нелинейности 0.5 и 100:

$$f_2(x) = 0.5(x_2 - x_1^2)^2 + (1 - x_1^2);$$

$$f_3(x) = 100(x_2 - x_1^2)^2 + (1 - x_1^2).$$

Для этого используйте прилагаемые к данной работе программы поддержки:

S2_Q, S2_R – метод S-квадрат для f_1 и f_2, f_3;

NLMD_Q, NLMD_R – метод Нелдера-Мида для f_1 и f_2, f_3;

HJ_Q, HJ_R – метод Хука-Дживса для f_1 и f_2, f_3.

Примечание. *Еще одна программа NMD_MNK потребуется в пп.6.5.*

Данные вычислений нужно отразить в табличной форме, образец которой дан в табл. 2.1.

Сравнительный вычислительный эксперимент, очевидно, нужно проводить при одинаковых исходных условиях. Поэтому для каждой задачи табл. 2.1 нужно вводить следующие исходные данные минимизации:

- точность поиска минимума Epsilon = 0.05;
- координаты начальных точек поиска x^0 берутся из табл.2.1. Варианты начальных точек задан так, что они располагаются приблизительно на одинаковом расстоянии до точки минимума;
- начальная длина стороны симплекса всегда равна 0.5, а параметры отражения, сжатия и растяжения соответственно равны 1.0, 0.5 и 2.0

Указание. *Для всех экспериментов с начальными точками под номером 1) на фоне линий уровня постройте несколько начальных шагов траекторий минимизации и конечную точку.*

Таблица 2.1. Результаты сравнения методов минимизации

Метод минимизации / Целевая функция	**Метод S-квадрат**	**Метод Нелдера-Мида**	**Метод Хука-Дживса**
Квадратичная форма $f_1(x) = 8x_1^2 + 4x_1x_2 + 5x_2^2$; *Точный минимум* $x^* = (0, 0)^T$ *Значение f в минимуме* $f(x^*) = 0$	1) Нач. точка $x^0 = (2, 2)^T$ x** = f(x**) = Nвф = 2) Нач. точка $x^0=(2.6, 1.2)^T$ …	1) Нач. точка $x^0 = (2, 2)^T$ x** = f(x**) = Nвф = 2) Нач. точка $x^0=(2.6, 1.2)^T$ …	1) Нач. точка $x^0 = (2,2)^T$ x** = f(x**) = Nвф = 2) Нач. точка $x^0=(2.6,1. 2)^T$ …
Функция Розенброка $f_2(x) = 0.5(x_2 - x_1^2)^2 + (1 - x_1^2)$; *Точный минимум* $x^* = (1, 1)^T$ *Значение f_2 в минимуме* $f_2(x^*) = 0$	1) Нач. точка $x^0=(-1.2, 1)^T$ … 2) Нач. точка $x^0=(-0.6, -0.6)^T$ …	1) Нач. точка $x^0=(-1.2, 1)^T$ … 2) Нач. точка $x^0=(-0.6, -0.6)^T$ …	1) Нач. точка $x^0=(-1.2, 1)^T$ … 2) Нач. точка $x^0=(-0.6, -0.6)^T$ …
Функция Розенброка $f_3(x) = 100(x_2 - x_1^2)^2 + (1 - x_1^2)$; *Точный минимум* $x^* = (1, 1)^T$ *Значение f_3 в минимуме* $f_3(x^*) = 0$	1) Нач. точка $x^0=(-1.2, 1)^T$ … 2) Нач. точка $x^0=(-0.6, -0.6)^T$ x** = f(x**) = Nвф =	1) Нач. точка $x^0=(-1.2, 1)^T$ … 2) Нач. точка $x^0=(-0.6, -0.6)^T$ x** = f(x**) = Nвф =	1) Нач. точка $x^0=(-1.2, 1)^T$ … 2) Нач. точка $x^0=(-0.6, -0.6)^T$ x** = f(x**) = Nвф =

Примечание. *1) Условные обозначения в табл. 2.1:*

x** - *оценка точки минимума;*

f(x**) – *значение минимума в точке оценки;*

Nвф – *число вычислений значений функции при поиске минимума.*

2) Очевидно, что для заполнения таблицы необходимо проделать

3(функции)x3(метода)x2(нач.точки) =18 экспериментов.

После заполнения табл. 2.1. сравните методы и сделайте краткие выводы по их сходимости, вычислительным затратам и т.д.

6.2. Самостоятельно напишите, отладьте и протестируйте программу минимизации функции из пп.5.5 методом сопряженных направлений Пауэлла.

В качестве исходных данных поиска предусмотрите в меню программы два варианта: поиск при двух начальных точках и поиск при одной начальной точке. Покажите соответствующие траектории поиска на фоне линий уровня.

6.3. **Задача.** Найти бункер без крышки в виде усеченного конуса (рис. 2.6) для хранения сыпучего материала минимальной стоимости объема 10 м^3, если основание изготавливается из материала стоимостью 2 \$/м^2 а остальная часть – из материала стоимостью 5\$/м^2.

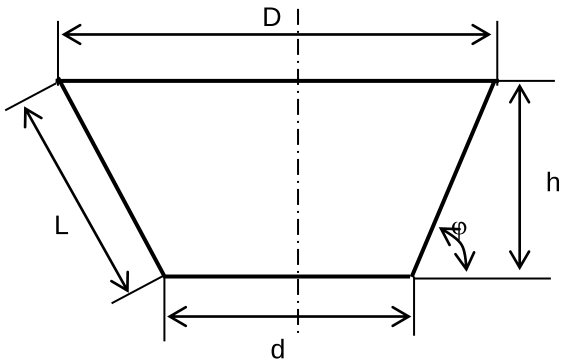


Рис. 2.6. Эскиз бункера без крышки

Указание. *Вначале найдите решение аналитически. Для этого воспользуйтесь ограничением на объем бункера для того, чтобы исключить одну переменную, и решите получаемую безусловную задачу двух переменных. Полезно описать геометрическую форму бункера с помощью различного набора конструктивных параметров.*

С помощью программы Mcad постройте линии уровня найденной двумерной целевой функции в области экстремальной точки.

Однозначное ли решение получено?

Если нет, то поясните почему?

Как нужно анализировать решение на чувствительность к какому-либо параметру? Приведите конкретные расчеты такого анализа.

6.4. **Задача.** Инженер проводит измерения некоторой физической величины y. Он получил следующую таблицу ее значений в зависимости от аргумента x:

x	1.0	2.0	3.0	4.0
y	1.05	1.25	1.55	1.59

Предположим, что из теоретических соображений инженеру известно, что функциональную связь y=f(x) можно описать следующей модельной двухпараметрической кривой

$$y_m = \frac{k_1 x}{1 + k_2 x},$$

где k_1, k_2 – неизвестные параметры.

Используя метод наименьших квадратов (МНК), сформулируйте задачу наилучшей аппроксимации экспериментальных данных y заданной модельной кривой y_{m}.

Найдите решение задачи аналитическим путем.

Сопоставьте на графике полученную модельную кривую y_{m} с экспериментальными данными.

6.5. С помощью прилагаемой к работе программы NMD_MNK найдите решение задачи предыдущего пункта. На фоне линий равного уровня функции невязки постройте начальный симплекс и несколько начальных точек траектории минимизации.

Сравните аналитическое решение пп. 6.4 и численное решения.

6.6. Решите задачу минимизации пп.6.4 с помощью метода сопряженных направлений Пауэлла.

Для этого используйте программу, составленную по пп.6.2, сделав соответствующие изменения в части вычисления целевой функции.

Проиллюстрируйте графически начальную стадию поиска минимума методом Пауэлла и покажите конечную точку траектории.

Сравните полученное решение с предыдущими.