Московский Государственный авиационный институт им. Серго Орджоникидзе

Кафедра прикладной информатики

ЛАБОРАТОРНЫЕ РАБОТЫ

«Использование численных методов для решения оптимизационных задач»

МОСКВА 1996

Лабораторные работы «Использование численных методов для решения оптимизационных задач»

Март 1996 г.

Работа №1. РЕШЕНИЕ ЗАДАЧ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

Цель работы

Целью работы является приобретение практических навыков по использованию программ численных методов для анализа и решения задач одномерной оптимизации.

Порядок выполнения работы

1. Получить вариант задания у преподавателя.
2. Привести поставленную задачу к формальному математическому виду:

$$\text{минимизировать } f(x)$$

где x — независимая переменная, $f(x)$ — целевая функция.

3. Исходя из содержательного смысла задачи оценить интервал значений независимой переменной, в котором содержится минимум целевой функции (локализовать минимум).
4. Составить подпрограмму вычисления значений целевой функции, оформив ее в виде подпрограммы-функции (см. Указания по использованию подпрограмм). Составить основную программу построения таблицы значений целевой функции. Затем с помощью этой программы вычислить целевую функцию в 10-15 различных точках исходного интервала и убедиться в том, что минимум действительно локализован. Представить результаты вычислений в виде таблицы и построить график целевой функции (построение производить вручную).
5. Составить основную программу решения задачи методом деления интервала пополам (см. Указания по использованию подпрограмм) и найти оптимальное значение независимой переменной x^* и соответствующее значение целевой функции $f(x^*)$ с максимально возможной точностью.
6. Выполнить анализ чувствительности найденного оптимального решения к изменениям независимой переменной. Для этого, вычисляя значения целевой функции в окрестности найденного оптимума, оценить размер наибольшего интервала, в котором целевая функция отличается от своего оптимального значения не более, чем на 1%. Затем, используя полученную оценку, определить минимальную допустимую относительную точность (в %%), с которой необходимо задать оптимальное значение независимой переменной, чтобы обеспечить оптимум целевой функции с точностью не хуже 1%. Дать содержательную интерпретацию чувствительности оптимального решения с точки зрения поставленной задачи.
7. Включить в подпрограмму вычисления целевой функции счетчик числа вызовов этой подпрограммы. Определить число вычислений целевой функции, необходимых для достижения заданной относительной точности при решении задачи тремя различными численными методами одномерной оптимизации (см. Указания по использованию подпрограмм). Исходный интервал должен быть одним и тем же для всех вариантов расчета. Результаты представить в виде табл. 1.1 и сделать соответствующие выводы.

Таблица 1.1

Метод	Число вычислений $f(x)$			
	$\delta = 10^{-2}$	$\delta = 10^{-3}$	$\delta = 10^{-4}$	$\delta = 10^{-5}$
деления интервала пополам Фибоначчи «золотого сечения»				

Требования к отчету

Отчет должен содержать:

1) исходную постановку оптимизационной задачи;

2) формальную математическую постановку оптимизационной задачи с необходимыми выкладками;

3) таблицу значений и график целевой функции (10-15 точек);

4) оптимальное решение задачи, полученное методом деления интервала пополам с максимально возможной точностью;

5) анализ чувствительности решения в окрестности точки оптимума;

6) результаты сравнения эффективности численных методов одномерной оптимизации и соответствующие выводы.

Указания по использованию подпрограмм

Для одномерной минимизации унимодальных функций могут использоваться следующие численные методы, относящиеся к числу поисковых методов:

метод деления интервала пополам;

метод Фибоначчи;

метод «золотого сечения».

Метод деления интервала пополам. Подпрограмма, реализующая метод деления интервала пополам, называется *BISEC*. Вызов этой подпрограммы имеет следующий вид:

```
CALL BISEC(F, A, B, EPS)
```

Параметр *F* должен быть объявлен в вызывающей программе в операторе *EXTERNAL*. Этот параметр задает имя подпрограммы-функции, вычисляющей значение целевой функции для заданного значения независимой переменной. Соответствующая подпрограмма-функция должна быть оформлена следующим образом:

```
REAL FUNCTION F(X)
REAL X
F = значение целевой функции
RETURN
END
```

Перед вызовом подпрограммы *BISEC* вещественные (*REAL*) параметры *A* и *B* должны определять границы исходного интервала, содержащего искомый минимум целевой функции, а вещественный параметр *EPS* — требуемый размер окончательного интервала (этот параметр может рассматриваться как абсолютная точность независимой переменной). После выполнения подпрограммы *BISEC* параметры *A* и *B* будут определять границы окончательного интервала, удовлетворяющего условию $|A - B| \le EPS$. Параметр *EPS* не изменяется.

Метод Фибоначчи. Подпрограмма, реализующая метод Фибоначчи, называется *FIBON*. Вызов этой подпрограммы имеет следующий вид:

```
CALL FIBON(F, A, B, EPS)
```

Параметры подпрограммы *FIBON* полностью эквивалентны параметрам подпрограммы *BISEC* (см. выше).

Метод «золотого сечения». Подпрограмма, реализующая метод «золотого сечения», называется *GOLDS*. Вызов этой подпрограммы имеет следующий вид:

```
CALL GOLDS(F, A, B, EPS)
```

Параметры подпрограммы *GOLDS* также полностью эквивалентны параметрам подпрограммы *BISEC* (см. выше).

Перед выполнением работы файлы с исходными текстами подпрограмм *BISEC*, *FIBON* и *GOLDS* необходимо переписать с библиотечного диска на рабочий. Также рекомендуется ознакомиться с исходными текстами этих подпрограмм.

Варианты задач к работе №1

1. Капля жидкости, начальная масса которой 10^{-4} *кг*, падает под действием силы тяжести, равномерно испаряясь, причем убыль массы пропорциональна времени с коэффициентом пропорциональности $4{\cdot}10^{-5}$ *кг/с*. Через какое время после начала падения кинетическая энергия капли будет наибольшей и какова она? (Сопротивлением воздуха пренебречь.)

2. Расходы на топливо для самолета пропорциональны кубу его скорости. Известно, что при скорости 700 *км/ч* расходы на топливо составляют 30 000 *руб/ч*, остальные же расходы (не зависящие от скорости) составляют 480 000 *руб/ч*. При какой скорости самолета общая сумма расходов на 1 *км* пути будет наименьшей? Чему равна эта наименьшая общая сумма расходов?

3. Три пункта *A*, *B* и *C* расположены так, что $\angle ABC = 50°$. Из пункта *A* в пункт *B* вылетает самолет со скоростью 800 *км/ч*, а одновременно с ним из пункта *B* в пункт *C* — вертолет со скоростью 500 *км/ч*. Через какое время (после начала движения) расстояние между самолетом и вертолетом будет минимальным и каково это расстояние, если *AB* = 1800 *км*? (Кривизной поверхности Земли пренебречь.)

4. Груз весом 200 *кг*, лежащий на горизонтальной поверхности, должен быть сдвинут приложенной к нему силой, причем коэффициент трения равен 0,25. Под каким углом (в градусах) к горизонту надо приложить силу, чтобы ее величина оказалась наименьшей? Чему равна эта наименьшая сила?

5. Скорость течения воды по круглой трубе прямо пропорциональна гидравлическому радиусу *R*, вычисляемому по формуле $R = S / P$, где *S* — площадь сечения потока воды в трубе, а *P* — смоченный (подводный) периметр трубы. Степень заполнения трубы водой характеризуется центральным углом, опирающемся на горизонтальную поверхность текущей воды. При какой степени заполнения трубы (в градусах) скорость течения воды будет наибольшей?

6. На странице книги печатный текст должен занимать 150 *кв. см*. Левое и верхнее поля должны быть по 2 *см*, правое — 1 *см*, а нижнее — 2,5 *см*. Если принимать во внимание только экономию бумаги, то какими должны быть наиболее выгодные размеры страницы? Чему равно в этом случае отношение (в %%) площади текста к общей площади страницы?

7. Вблизи завода *A* по намеченной прямой к городу *B* проводится железная дорога. Под каким углом (в градусах) к проектируемой железной дороге нужно провести шоссе с завода *A*, чтобы доставка грузов из *A* в *B* была наиболее дешевой, если стоимость перевозки 1 тонно-километра по шоссе в 4,8 раза дороже, чем по железной дороге?

8. Два самолета летят в одной плоскости и прямолинейно под углом 136° с одинаковой скоростью, равной 800 *км/ч*. В некоторый момент один самолет пришел в точку пересечения линий движения, а второй не дошел до нее на 160 *км*. Через какое время расстояние между самолетами будет наименьшим и чему равно это расстояние?

9. Два коридора шириной 2,4 *м* и 1,6 *м* пересекаются под прямым углом. Какова наибольшая длина доски, которую можно перенести (горизонтально) из одного коридора в другой?

10. Предприятие может производить товар *A* с затратами 20 *руб/кг* и товар *B* с затратами 10 *руб/кг*. Отдел сбыта полагает, что можно продать 1 000 000 / ($x^{1,8}y$) кг товара *A* в день и 2 000 000 / ($xy^{2,1}$) *кг* товара *B* в день, где *x* — продажная цена товара *A* (*руб/кг*), а *y* — продажная цена товара *B* (*руб/кг*). Какова наибольшая ожидаемая прибыль, если *A* и *B* продаются по одной и той же цене? Чему равна соответствующая оптимальная продажная цена?

11. Над центром круглой площадки радиусом 20 *м* необходимо расположить источник света, чтобы он наилучшим образом освещал периметр площадки. Какова оптимальная высота расположения источника света, если степень освещенности некоторой поверхности прямо пропорциональна косинусу угла падения лучей и обратно пропорциональна квадрату расстояния до источника света?

12. Конический резервуар, радиус основания которого равен 10 *м*, а высота — 35 *м*, наполнен жидкостью. В этот резервуар погружается металлический шар. Каким должен быть радиус шара, чтобы объем жидкости, вытесненной из резервуара погруженной в жидкость частью шара, был наибольшим? Чему в этом случае равно отношение (в %%) объема вытесненной жидкости к общему объему жидкости, находившейся в резервуаре до погружения шара?

13. На возвышении 500 *м* расположено артиллерийское орудие. Под каким углом (в градусах) к горизонту необходимо произвести выстрел, чтобы снаряд пролетел наибольшее расстояние, если начальная скорость снаряда равна 1200 *км/с*? Чему равно это расстояние?

14. Балка прямоугольного сечения со свободно опертыми концами равномерно нагружена по всей длине. Стрела ее прогиба обратно пропорциональна моменту инерции сечения балки $I = xy^3 / 12$, где *x* и *y* — размеры балки. Определить размеры балки, при которых обеспечивается наименьшая стрела прогиба, если балка вырезается из круглой заготовки диаметром 0,5 *м*.

15. В неподвижную чашу, имеющую форму полушара радиусом 1 *м*, опущен стержень длиной 2,8 *м*. При каком угле наклона (в градусах) стержень будет находиться в состоянии равновесия? (Трением стержня о стенки чаши пренебречь.)

Приложение 1

Исходный текст подпрограммы *BISEC*

```
      SUBROUTINE BISEC(F,A,B,EPS)
*
*     Поиск минимума унимодальной функции методом деления интервала
*     пополам.
*
      REAL F,A,B,EPS
      EXTERNAL F
      INTEGER K
      REAL X1,X2,X3,F1,F2,F3
      PRINT 10
   10 FORMAT (///' МЕТОД ДЕЛЕНИЯ ИНТЕРВАЛА ПОПОЛАМ')
      PRINT 20,A,B,EPS
   20 FORMAT (/' A=',1PG12.5,' B=',1PG12.5,' EPS=',1PG12.5)
      PRINT 25
   25 FORMAT (/2X,'K',6X,'A',12X,'B',12X,'X',10X,'F(X)')
      K=0
      X2=0.5*(A+B)
      F2=F(X2)
*
*     Выполнен K-й шаг:
*     A,B   - границы текущего интервала;
*     X2    - середина текущего интервала;
*     F2    - значение целевой функции в точке X2.
*
   30 PRINT 40,K,A,B,X2,F2
   40 FORMAT (1X,I2,4(1X,1PG12.5))
      IF (ABS(A-B).LE.EPS) RETURN
      K=K+1
      X1=0.5*(A+X2)
      F1=F(X1)
      IF (F1.LT.F2) THEN
         B=X2
         X2=X1
         F2=F1
         GOTO 30
      ENDIF
      X3=0.5*(X2+B)
      F3=F(X3)
      IF (F3.LT.F2) THEN
         A=X2
         X2=X3
         F2=F3
         GOTO 30
      ENDIF
      A=X1
      B=X3
      GOTO 30
      END
```

Исходный текст подпрограммы *FIBON*

```
      SUBROUTINE FIBON(F,A,B,EPS)
*
*     Поиск минимума унимодальной функции методом Фибоначчи.
*
      REAL F,A,B,EPS
      EXTERNAL F
      INTEGER NFIB,IFIB/0/,K,N,KASE
      PARAMETER (NFIB=50)
      REAL FIB(0:NFIB),D,XM,X1,X2,FM,F1,F2
      SAVE IFIB,FIB
      PRINT 10
```

```
   10 FORMAT (///' МЕТОД ФИБОНАЧЧИ')
      PRINT 20,A,B,EPS
   20 FORMAT (/' A=',1PG12.5,' B=',1PG12.5,' EPS=',1PG12.5)
*
*     Вычисление чисел Фибоначчи.
*
      IF (IFIB.EQ.0) THEN
         IFIB=1
         FIB(0)=1.0
         FIB(1)=1.0
         DO 30 K=2,NFIB
            FIB(K)=FIB(K-1)+FIB(K-2)
   30    CONTINUE
      ENDIF
*
*     Определение необходимого числа шагов.
*
      D=ABS(A-B)/EPS
      DO 40 N=0,NFIB-1
         IF (FIB(N+1).GE.D) GOTO 60
   40 CONTINUE
      PRINT 50
   50 FORMAT (/' Запрошена слишком высокая точность')
      RETURN
   60 PRINT 70,N
   70 FORMAT (/' Будет выполнено ',I2,' шаг(а,ов)')
*
*     Начальный шаг.
*
      PRINT 75
   75 FORMAT (/2X,'K',6X,'A',12X,'B',12X,'X',10X,'F(X)')
      K=0
      D=(FIB(N-1)/FIB(N+1))*(B-A)
      XM=A+D
      FM=F(XM)
      KASE=0
*
*     Выполнен K-й шаг:
*     A,B   - границы текущего интервала;
*     XM    - внутренняя точка текущего интервала;
*     FM    - значение целевой функции в точке XM.
*
*     XM расположена ближе к A, если KASE равно 0;
*     XM расположена ближе к B, если KASE равно 1.
*
   80 PRINT 90,K,A,B,XM,FM
   90 FORMAT (1X,I2,4(1X,1PG12.5))
      IF (K.EQ.N) RETURN
      K=K+1
      IF (K.EQ.N) GOTO 110
*
*     Очередной (не последний) шаг.
*
      D=(FIB(N-K)/FIB(N-K+2))*(B-A)
      X1=A+D
      X2=B-D
      IF (KASE.EQ.0) THEN
         F1=FM
         F2=F(X2)
      ELSE
         F1=F(X1)
         F2=FM
      ENDIF
  100 IF (F1.LT.F2) THEN
         B=X2
         XM=X1
         FM=F1
```

```
         KASE=1
      ELSE
         A=X1
         XM=X2
         FM=F2
         KASE=0
      ENDIF
      GOTO 80
*
*     Последний шаг.
*
*     XM расположена в центре текущего интервала.
*
  110 X1=XM
      X2=XM+0.01*(B-A)
      F1=FM
      F2=F(X2)
      GOTO 100
      END
```

Исходный текст подпрограммы *GOLDS*

```
      SUBROUTINE GOLDS(F,A,B,EPS)
*
*     Поиск минимума унимодальной функции методом золотого сечения.
*
      REAL F,A,B,EPS
      EXTERNAL F
      INTEGER K,KASE
      REAL C,XM,X1,X2,FM,F1,F2
      PRINT 10
   10 FORMAT (///' МЕТОД ЗОЛОТОГО СЕЧЕНИЯ')
      PRINT 20,A,B,EPS
   20 FORMAT (/' A=',1PG12.5,' B=',1PG12.5,' EPS=',1PG12.5)
      PRINT 25
   25 FORMAT (/2X,'K',6X,'A',12X,'B',12X,'X',10X,'F(X)')
      K=0
      C=0.5*(3.0-SQRT(5.0))
      XM=A+C*(B-A)
      FM=F(XM)
      KASE=0
*
*     Выполнен K-й шаг:
*     A,B   - границы текущего интервала;
*     XM    - внутренняя точка текущего интервала;
*     FM    - значение целевой функции в точке XM.
*
*     XM расположена ближе к A, если KASE равно 0;
*     XM расположена ближе к B, если KASE равно 1.
*
   30 PRINT 40,K,A,B,XM,FM
   40 FORMAT (1X,I2,4(1X,1PG12.5))
      IF (ABS(A-B).LE.EPS) RETURN
      K=K+1
      X1=A+C*(B-A)
      X2=B-C*(B-A)
      IF (KASE.EQ.0) THEN
         F1=FM
         F2=F(X2)
      ELSE
         F1=F(X1)
         F2=FM
      ENDIF
      IF (F1.LT.F2) THEN
         B=X2
         XM=X1
```

```
         FM=F1
         KASE=1
      ELSE
         A=X1
         XM=X2
         FM=F2
         KASE=0
      ENDIF
      GOTO 30
      END
```

Пример использования подпрограммы *BISEC*

```
      PROGRAM TANK1
*
*     Оптимальное проектирование цилиндрического бака методом деления
*     интервала пополам.
*
      REAL S,A,B,EPS
      EXTERNAL S
      A=1.0
      B=3.0
      EPS=0.01
      CALL BISEC(S,A,B,EPS)
      PRINT 10,A,B
   10 FORMAT (//' Окончательный интервал: A=',1PG12.5,' B=',1PG12.5)
      STOP
      END
      REAL FUNCTION S(D)
      REAL D,PI
      PI=3.14159
      S=(80.0+PI*D**3)/(4.0*D)
      RETURN
      END
```

```
МЕТОД ДЕЛЕНИЯ ИНТЕРВАЛА ПОПОЛАМ

A=  1.0000     B=  3.0000     EPS= 1.00000E-02

 K      A            B            X           F(X)
 0   1.0000       3.0000       2.0000       13.142
 1   2.0000       3.0000       2.5000       12.909
 2   2.0000       2.5000       2.2500       12.865
 3   2.2500       2.5000       2.3750       12.851
 4   2.2500       2.3750       2.3125       12.849
 5   2.3125       2.3750       2.3438       12.848
 6   2.3125       2.3438       2.3281       12.848
 7   2.3281       2.3438       2.3359       12.847
 8   2.3320       2.3398       2.3359       12.847

Окончательный интервал: A=  2.3320     B=  2.3398
Stop - Program terminated.
```

Пример использования подпрограммы *FIBON*

```
      PROGRAM TANK2
*
*     Оптимальное проектирование цилиндрического бака методом Фибоначчи.
*
      REAL S,A,B,EPS
      EXTERNAL S
      A=1.0
      B=3.0
      EPS=0.01
      CALL FIBON(S,A,B,EPS)
      PRINT 10,A,B
```

```
   10 FORMAT (//' Окончательный интервал: A=',1PG12.5,' B=',1PG12.5)
      STOP
      END
      REAL FUNCTION S(D)
      REAL D,PI
      PI=3.14159
      S=(80.0+PI*D**3)/(4.0*D)
      RETURN
      END
```

```
МЕТОД ФИБОНАЧЧИ

A=  1.0000     B=  3.0000     EPS= 1.00000E-02

Будет выполнено 11 шаг(а,ов)

 K      A            B            X           F(X)
 0   1.0000       3.0000       1.7639       13.782
 1   1.7639       3.0000       2.2361       12.871
 2   1.7639       2.5279       2.2361       12.871
 3   2.0558       2.5279       2.2361       12.871
 4   2.2361       2.5279       2.3476       12.848
 5   2.2361       2.4163       2.3476       12.848
 6   2.3047       2.4163       2.3476       12.848
 7   2.3047       2.3734       2.3476       12.848
 8   2.3047       2.3476       2.3305       12.848
 9   2.3219       2.3476       2.3305       12.848
10   2.3305       2.3476       2.3391       12.848
11   2.3305       2.3392       2.3391       12.848

Окончательный интервал: A=  2.3305     B=  2.3392
Stop - Program terminated.
```

Пример использования подпрограммы *GOLDS*

```
      PROGRAM TANK3
*
*     Оптимальное проектирование цилиндрического бака методом золотого
*     сечения.
*
      REAL S,A,B,EPS
      EXTERNAL S
      A=1.0
      B=3.0
      EPS=0.01
      CALL GOLDS(S,A,B,EPS)
      PRINT 10,A,B
   10 FORMAT (//' Окончательный интервал: A=',1PG12.5,' B=',1PG12.5)
      STOP
      END
      REAL FUNCTION S(D)
      REAL D,PI
      PI=3.14159
      S=(80.0+PI*D**3)/(4.0*D)
      RETURN
      END
```

```
МЕТОД ЗОЛОТОГО СЕЧЕНИЯ

A=  1.0000     B=  3.0000     EPS= 1.00000E-02

 K      A            B            X           F(X)
 0   1.0000       3.0000       1.7639       13.782
 1   1.7639       3.0000       2.2361       12.871
 2   1.7639       2.5279       2.2361       12.871
 3   2.0557       2.5279       2.2361       12.871
```

4	2.2361	2.5279	2.3475	12.848
5	2.2361	2.4164	2.3475	12.848
6	2.3050	2.4164	2.3475	12.848
7	2.3050	2.3738	2.3475	12.848
8	2.3050	2.3475	2.3313	12.848
9	2.3212	2.3475	2.3313	12.848
10	2.3313	2.3475	2.3375	12.847
11	2.3313	2.3413	2.3375	12.847
12	2.3313	2.3375	2.3351	12.847

```
Окончательный интервал: A=  2.3313     B=  2.3375
Stop - Program terminated.
```

Работа №2. РЕШЕНИЕ ЗАДАЧ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

Цель работы

Целью работы является приобретение практических навыков по использованию программ численных методов для анализа и решения задач безусловной оптимизации.

Порядок выполнения работы

1. Получить вариант задания у преподавателя.
2. Привести поставленную задачу к формальному математическому виду:

$$\text{минимизировать } f(x_1, x_2, \ldots, x_n),$$

где n — размерность задачи; $x_1, x_2, \ldots, x_n$ — независимые переменные; f — целевая функция.

Примечание. Во всех вариантах задания $n = 2$.

3. Исходя из содержательного смысла задачи выбрать «разумные» интервалы изменения независимых переменных, в которых может (предположительно) находиться искомое оптимальное решение. Выбрать начальные приближения независимых переменных внутри соответствующих оцененных интервалов.

4. Составить подпрограмму вычисления значений целевой функции, оформив ее в виде подпрограммы-функции (см. Указания по использованию подпрограмм). Составить основную программу решения задачи методом Нелдера-Мида (см. Указания по использованию подпрограмм) и найти оптимальные значения независимых переменных x^*_1 и x^*_2, а также соответствующее значение целевой функции $f(x^*_1, x^*_2)$ в точке оптимума (в качестве начального размера многогранника *SIZE* взять десятую часть оцененного интервала для x_1 или x_2; параметр *EPS* взять равным $10^{-3} \div 10^{-4}$). Если оптимальное решение найти не удается, выбрать другие начальные приближения независимых переменных и повторить решение задачи указанным методом.

5. Используя составленную подпрограмму вычисления значений целевой функции протабулировать целевую функцию при различных равноотстоящих значениях независимых переменных (по каждой переменной взять 8-10 точек внутри соответствующего оцененного интервала). Представить результаты табуляции в виде прямоугольной сетки значений (в каждом узле сетки должно быть отмечено соответствующее значение целевой функции), после чего нанести на эту сетку линии уровня (линии равных значений) целевой функции (всего 8-10 уровней) и отметить найденную оптимальную точку (x^*_1, x^*_2). (Все построения производить вручную на отдельном листе миллиметровой бумаги.) По характерному рисунку линий уровня в окрестности отмеченной точки убедиться, что эта точка действительно является точкой локального оптимума (минимума или максимума).

6. Используя вывод подпрограммы *NELDM* построить и нанести на сетку целевой функции траекторию поиска оптимальной точки методом Нелдера-Мида (для всех промежуточных точек указать порядковый номер соответствующего шага вычислений).

7. Составить основную программу решения задачи методом Хука-Дживса (см. Указания по использованию подпрограмм) и повторить решение задачи, используя те же начальные приближения, что и для метода Нелдера-Мида (в качестве начальных приращений переменных *DX*(1) и *DX*(2) взять десятые части оцененных интервалов для x_1 и x_2; параметр *EPS* взять равным $10^{-3} \div 10^{-4}$). Убедиться, что найденное оптимальное решение совпадает (в пределах заданной точности) с решением, полученным ранее.

8. Используя вывод подпрограммы *HOOKE* построить и нанести на сетку целевой функции траекторию поиска оптимальной точки методом Хука-Дживса (аналогично п. 6).

9. Найти частные производные целевой функции $\partial f/\partial x_1$ и $\partial f/\partial x_2$ и составить подпрограмму вычисления градиента целевой функции (см. Указания по использованию подпрограмм). Составить основную программу решения задачи методом наискорейшего спуска (см. Указания по использованию подпрограмм) и повторить решение задачи, используя те же начальные приближения, что и для метода Нелдера-Мида (параметр *EPS* взять равным $10^{-2} \div 10^{-3}$; параметр *EPS*1 выбрать так, чтобы он соответствовал относительной точности $10^{-4} \div 10^{-5}$). Если метод расходится, выбрать другие начальные приближения независимых переменных и повторить решение задачи. Сравнить найденное оптимальное решение с решениями, полученными другими методами.

10. Используя вывод подпрограммы *STEEP* построить и нанести на сетку целевой функции траекторию поиска оптимальной точки методом наискорейшего спуска (аналогично п. 6).

11. Найти вторые частные производные целевой функции $\partial^2 f/\partial x_1^2$, $\partial^2 f/\partial x_1 \partial x_2 = \partial^2 f/\partial x_2 \partial x_1$ и $\partial^2 f/\partial x_2^2$, и составить подпрограмму вычисления матрицы Гессе (см. Указания по использованию подпрограмм). Составить основную программу решения задачи методом Ньютона-Рафсона (см. Указания по использованию подпрограмм) и повторить решение задачи, используя те же начальные приближения, что и для метода Нелдера-Мида (параметр *EPS* взять равным $10^{-2} \div 10^{-3}$). Если метод расходится, выбрать другие начальные приближения независимых переменных и повторить решение задачи. Сравнить найденное оптимальное решение с решениями, полученными другими методами.

12. Используя вывод подпрограммы *NEWTR* построить и нанести на сетку целевой функции траекторию поиска оптимальной точки методом Ньютона-Рафсона (аналогично п. 6).

13. Произвести решение задачи всеми четырьмя численными методами для четырех различных вариантов начальных приближений независимых переменных. Используя вывод подпрограмм *HOOKE*, *NELDM*, *STEEP* и *NEWTR* представить результаты вычислений в виде табл. 2.1 и сделать соответствующие выводы (обратить при этом внимание на скорость сходимости различных численных методов вблизи точки оптимума).

Примечание. Одно вычисление градиента приравнивается n вычислениям целевой функции; одно вычисление матрицы Гессе приравнивается n^2 вычислениям целевой функции.

Таблица 2.1

Метод	Число вычислений $f(x_1, x_2)$			
	$x_1^0 =$ $x_2^0 =$	$x_1^0 =$ $x_2^0 =$	$x_1^0 =$ $x_2^0 =$	$x_1^0 =$ $x_2^0 =$
Хука-Дживса Нелдера-Мида наискорейшего спуска Ньютона-Рафсона				

Требования к отчету

Отчет должен содержать:

1) исходную постановку оптимизационной задачи;

2) формальную математическую постановку оптимизационной задачи с необходимыми выкладками;

3) сетку значений (8-10 равноотстоящих значений по каждой из независимых переменных) целевой функции, ее линии уровня и траектории поиска оптимального

решения четырьмя численными методами (для одного и того же варианта начальных приближений);

4) начальные приближения и результаты для всех вариантов расчета (включая и неудачные варианты);

5) результаты сравнения эффективности численных методов безусловной минимизации и соответствующие выводы.

Указания по использованию подпрограмм

Для безусловной минимизации унимодальных функций могут использоваться следующие численные методы:

метод Хука-Дживса (метод конфигураций);

метод Нелдера-Мида (метод деформируемого многогранника);

метод наискорейшего спуска;

метод Ньютона-Рафсона.

(Первые два метода относятся к поисковым, третий — к градиентным, а четвертый — к ньютоновским методам оптимизации.)

Метод Хука-Дживса. Подпрограмма, реализующая метод Хука-Дживса называется HOOKE. Вызов этой подпрограммы имеет вид:

CALL HOOKE(*N*, *F*, *X*, *DX*, *EPS*, *MAXFEV*, *IER*, *Y*)

Входные параметры подпрограммы *HOOKE* следующие:

N — размерность задачи (число независимых переменных);

F — этот параметр должен быть объявлен в вызывающей программе в операторе *EXTERNAL* и задает имя подпрограммы-функции, вычисляющей значения целевой функции по заданным значениям независимых переменных. Соответствующая подпрограмма-функция должна быть оформлена следующим образом:

REAL FUNCTION F(*X*)
REAL X(*N*)
F = значение целевой функции
RETURN
END

X — массив размерности *N*, содержащий начальные приближения независимых переменных;

DX — массив размерности *N*, содержащий начальные приращения независимых переменных (эти начальные приращения должны быть такими, чтобы охватить достаточно большую окрестность начальной точки). В процессе поиска эти приращения уменьшаются;

EPS — положительное число, определяющее условие окончания поиска. Поиск заканчивается, если условие

$$|DX(I)| \le EPS, \text{ если } |X(I)| \le 1$$

$$|DX(I)/X(I)| \le EPS, \text{ если } |X(I)| > 1$$

выполняется для всех $I = 1, 2, \ldots, N$, где $DX(I)$ — текущее приращение переменной $X(I)$;

MAXFEV — максимальное допустимое число вычислений целевой функции;

Y — вспомогательный массив размерности *N*.

Выходные параметры подпрограммы *HOOKE* следующие:

X — найденное оптимальное решение (если *IER*=0) или конечные приближения независимых переменных (если *IER*=1);

DX — конечные приращения переменных;

IER — признак завершения поиска. Если *IER*=0, то найдено оптимальное решение. Если же *IER*=1, то *MAXFEV* вычислений целевой функции оказалось недостаточно для определения оптимального решения (в последнем случае параметр *X* будет соответствовать последней достигнутой точке, уменьшающей значение целевой функции).

Метод Нелдера-Мида. Подпрограмма, реализующая метод Нелдера-Мида называется *NELDM*. Вызов этой подпрограммы имеет вид:

CALL NELDM(*N*, *F*, *X*, *SIZE*, *EPS*, *MAXFEV*, *IER*, *Y*, *FY*)

Входные параметры подпрограммы *NELDM* следующие:

N — размерность задачи (число независимых переменных);

F — имя подпрограммы-функции, вычисляющей значения целевой функции по заданным значениям независимых переменных (см. выше аналогичный параметр подпрограммы *HOOKE*);

X — массив размерности *N*, содержащий начальные приближения независимых переменных;

SIZE — размер начального многогранника (этот параметр аналогичен начальным приращениям переменных в подпрограмме *HOOKE*);

EPS — положительное целое число, определяющее условие окончания поиска. Поиск заканчивается, если выполняется условие

$$\left\{\frac{1}{N+1}\sum_{k=0}^{N}\left[f(x^k)-f(x^*)\right]^2\right\}^{1/2} \le EPS,$$

где $f(x^k)$ — значение целевой функции в k-й вершине текущего многогранника (в процессе поиска размеры многогранника уменьшаются), $f(x^*)$ — наилучшее достигнутое значение целевой функции;

MAXFEV — максимальное допустимое число вычислений целевой функции;

Y — вспомогательный массив размерности *N*(*N*+4);

FY — вспомогательный массив размерности (*N*+4).

X — найденное оптимальное решение (если *IER*=0) или конечные приближения независимых переменных (если *IER*=1);

IER — признак завершения поиска. Если *IER*=0, то найдено оптимальное решение. Если же *IER*=1, то *MAXFEV* вычислений целевой функции оказалось недостаточно для определения оптимального решения (в последнем случае параметр *X* будет соответствовать последней достигнутой точке, уменьшающей значение целевой функции).

Метод наискорейшего спуска. Подпрограмма, реализующая метод наискорейшего спуска, называется *STEEP*. Вызов этой подпрограммы имеет вид:

CALL STEEP(*N*, *F*, *GRAD*, *X*, *EPS*, *SMAX*, *EPS*1, *MAXFEV*, *IER*, *G*, *Y*)

Входные параметры подпрограммы *STEEP* следующие:

N — размерность задачи (число независимых переменных);

F — имя подпрограммы-функции, вычисляющей значения целевой функции по заданным значениям независимых переменных (см. выше аналогичный параметр подпрограммы *HOOKE*);

GRAD — этот параметр должен быть объявлен в вызывающей программе в операторе *EXTERNAL* и задает имя подпрограммы, вычисляющей компоненты градиента целевой функции при заданных значениях независимых переменных. Соответствующая подпрограмма должна быть оформлена следующим образом:

```
SUBROUTINE GRAD(X, FX, G)
REAL X(N), FX, G(N)
G(1) = ∂f/∂x1
G(2) = ∂f/∂x2
. . . . . . . . . .
G(N) = ∂f/∂xn
RETURN
END
```

X — массив размерности *N*, содержащий начальные приближения независимых переменных;

EPS — положительное число, определяющее условие окончания поиска. Поиск заканчивается, если выполняется условие

$$\| \operatorname{grad} f(x) \| \le EPS,$$

где $\| \cdot \|$ — сферическая норма (длина) вектора;

SMAX — максимальная допустимая величина шага в направлении наискорейшего спуска;

*EPS*1 — положительное число, определяющее абсолютную точность локализации наилучшей точки вдоль направления наискорейшего спуска методом «золотого сечения»;

MAXFEV — максимальное допустимое число вычислений целевой функции;

G — вспомогательный массив размерности *N*;

Y — вспомогательный массив размерности *N*.

Выходные параметры подпрограммы *STEEP* следующие:

X — найденное оптимальное решение (если *IER*=0) или конечные приближения независимых переменных (если *IER*=1);

IER — признак завершения поиска. Если *IER*=0, то найдено оптимальное решение. Если же *IER*=1, то *MAXFEV* вычислений целевой функции оказалось недостаточно для определения оптимального решения (в последнем случае параметр *X* будет соответствовать последней достигнутой точке).

Метод Ньютона-Рафсона. Подпрограмма, реализующая метод Ньютона-Рафсона, называется NEWTR. Вызов этой подпрограммы имеет вид:

```
CALL NEWTR(N, F, GRAD, HESS, X, EPS, MAXFEV, IER, G, H)
```

Входные параметры подпрограммы *NEWTR* следующие:

N — размерность задачи (число независимых переменных);

F — имя подпрограммы-функции, вычисляющей значения целевой функции по заданным значениям независимых переменных (см. выше аналогичный параметр подпрограммы *HOOKE*);

GRAD — имя подпрограммы, вычисляющей компоненты градиента целевой функции при заданных значениях независимых переменных (см. выше аналогичный параметр подпрограммы *STEEP*);

HESS — этот параметр должен быть объявлен в вызывающей программе в операторе *EXTERNAL* и задает имя подпрограммы, вычисляющей компоненты матрицы Гессе (матрицы вторых производных) целевой функции при заданных значениях независимых переменных. Соответствующая подпрограмма должна быть оформлена следующим образом:

```
SUBROUTINE HESS(X, FX, H)
REAL X(N), FX, H((N*N+N)/2)
H(1) = ∂²f/∂x1²
```

$H(2) = \partial^2 f / \partial x_2 \partial x_1$
$H(3) = \partial^2 f / \partial x_2^2$
$H(4) = \partial^2 f / \partial x_3 \partial x_1$
$H(5) = \partial^2 f / \partial x_3 \partial x_2$
$H(6) = \partial^2 f / \partial x_3^2$
. . . .
RETURN
END

(поскольку матрица Гессе является симметричной, то в подпрограмме *HESS* вычисляется только верхний треугольник этой матрицы);

X — массив размерности *N*, содержащий начальные приближения независимых переменных;

EPS — положительное число, определяющее условие окончания поиска. Поиск заканчивается, если выполняется условие

$$\| \text{grad}\, f(x) \| \leq EPS,$$

где $\| \cdot \|$ — сферическая норма (длина) вектора;

MAXFEV — максимальное допустимое число вычислений целевой функции;

G — вспомогательный массив размерности *N*;

H — вспомогательный массив размерности $(N^2 + N)/2$.

Выходные параметры подпрограммы *NEWTR* следующие:

X — найденное оптимальное решение (если *IER*=0) или конечные приближения независимых переменных (если *IER*=1 или *IER*=2);

IER — признак завершения поиска. Если *IER*=0, то найдено оптимальное решение. Если *IER*=1, то *MAXFEV* вычислений целевой функции оказалось недостаточно для определения оптимального решения. Если же *IER*=2, то матрица Гессе не является положительно определенной (в текущей точке). (В случаях *IER*=1 или *IER*=2 параметр *X* будет соответствовать последней достигнутой точке).

Перед выполнением работы файлы с исходными текстами подпрограмм *HOOKE*, *NELDM*, *STEEP* и *NEWTR* необходимо переписать с библиотечного диска на рабочий. Также рекомендуется ознакомиться с исходными текстами этих подпрограмм.

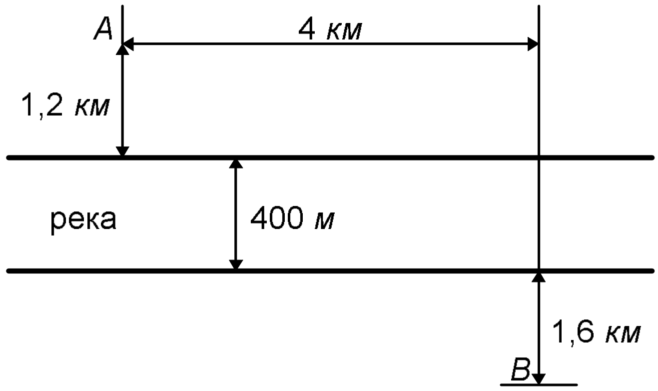
Варианты задач к работе №2

1. Требуется переправить 4000 $м^3$ сыпучего материала через большую реку. Для перевозки груза необходимо изготовить контейнер с квадратным дном. Известно, что стоимость каждого рейса на противоположный берег реки и обратно равна 1420 *руб*; стоимость материалов для изготовления дна контейнера равна 2000 $руб/м^2$, боковых стенок контейнера — 500 $руб/м^2$, крышки контейнера — 1500 $руб/м^2$. Какие размеры должен иметь контейнер, чтобы полные затраты на перевозку груза были минимальными?

2. Расстояния между пунктами А, В и С равны 15 *км*, 17 *км* и 20 *км*. В каком месте должна быть расположена ретрансляционная станция, чтобы сумма расстояний от этой станции до указанных пунктов была наименьшей?

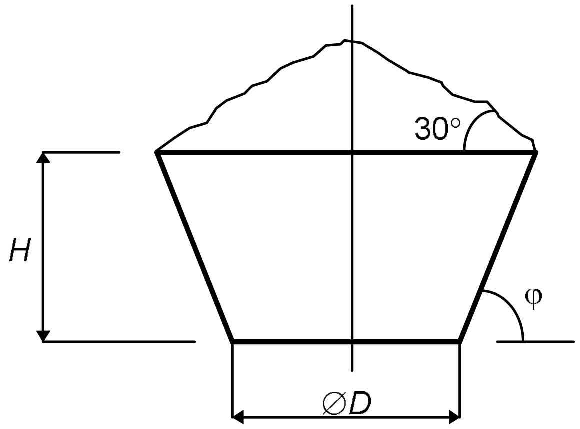
3. Предприятие может производить товар *А* с затратами 20 *руб/кг* и товар *В* с затратами 10 *руб/кг*. Отдел сбыта полагает, что каждый день можно продавать $2 \cdot 10^6/(x^2y)$ *кг* товара *А* и $3 \cdot 10^6/(xy^2)$ *кг* товара *В*, где *x* — продажная цена товара *А* в рублях за килограмм, *y* — продажная цена товара *В* в рублях за килограмм. Определить *x* и *y*, которые могут обеспечить максимальную прибыль.

4. Два лагеря A и B расположены следующим образом:



Какой путь должен выбрать гонец, чтобы попасть из лагеря A в лагерь B за кратчайшее время, если он может идти со скоростью 4 *км/ч* и плыть со скоростью 0,8 *км/ч*? (Течением реки пренебречь.)

5. Конический бункер для хранения зерна имеет следующий вид:



Основание бункера изготавливается из деревянных плит стоимостью 800 *руб/м*^{2}, а остальная часть бункера — из листового металла стоимостью 1300 *руб/м*^{2}. Какие параметры H, D и φ должен иметь бункер объемом 500 *м*^{3}, чтобы его стоимость была минимальной? (Используйте ограничение на объем бункера, чтобы исключить одну переменную из целевой функции.)

6. Инженер-исследователь полагает, что связь между переменными x и y можно выразить двухпараметрической функцией

$$y(x) = k_1 x \,/\, (1 + k_2 x).$$

Найти k_1 и k_2 в соответствии с критерием наименьших квадратов

$$Q = \sum_i \left[y(x_i) - y_i \right]^2$$

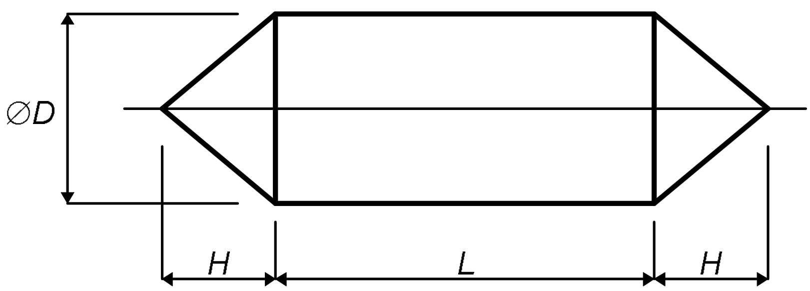
на основе следующих экспериментальных данных:

x_i	1,00	2,00	3,00	4,00
y_i	1,05	1,25	1,55	1,59

7. Расстояния между равными положительными зарядами равны 3 *см*, 4 *см* и 6 *см*. В каком месте между этими зарядами надо расположить пробный отрицательный заряд, чтобы действующая на него сила была наименьшей?

8. Рабочие станции A, B и C расположены так, что расстояния между ними равны 30 *м*, 40 *м* и 45 *м*. В каком месте должен быть расположен файл-сервер, чтобы общая длина оптоволоконного кабеля, соединяющего файл-сервер с указанными рабочими станциями, была как можно меньше?

9. Топливный бак имеет следующий вид:



Какие параметры D, H и L должен иметь этот бак, чтобы площадь его поверхности была равна 100 *м*^{2} при максимально возможной вместимости? (Используйте ограничение на площадь поверхности бака, чтобы исключить одну переменную из целевой функции.)

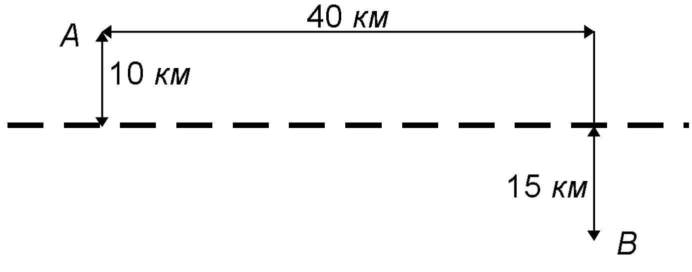
10. Если воздух (показатель адиабаты $\gamma = 1{,}4$) сжимается от давления p_0 до давления $p > p_0$, то работа сжатия 1 моля воздуха равна

$$A = RT_0 \frac{\gamma}{\gamma - 1}\left[\left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}} - 1\right],$$

где R — газовая постоянная, T_0 — температура воздуха до сжатия. Требуется сконструировать трехступенчатый компрессор с двумя промежуточными холодильниками, охлаждающими воздух снова до температуры T_0, для сжатия воздуха от 1 *атм* до 50 *атм*. Каковы должны быть давления p_1 и p_2 после первой и второй ступени соответственно, чтобы общие затраты энергии на сжатие 1 моля воздуха были как можно меньше?

11. Для транспортировки 10 000 *м*^{3} сыпучих отходов через реку необходимо изготовить цилиндрический контейнер. Известно, что стоимость каждого рейса на противоположный берег реки и обратно равна 1420 *руб*; стоимость материалов для изготовления дна контейнера равна 1800 *руб*/*м*^{2}, боковой стенки — 450 *руб*/*м*^{2}, крышки контейнера — 1200 *руб*/*м*^{2}. Какие размеры должен иметь контейнер, чтобы полные затраты на транспортировку были минимальными?

12. Вблизи двух заводов A и B проходит железная дорога:



Каким образом надо расположить подъездные пути (шоссе), соединяющие заводы A и B с железной дорогой, чтобы доставка грузов из A в B была наиболее дешевой, если стоимость перевозки 1 тонно-километра по шоссе в 4,8 раза дороже, чем по железной дороге?

13. Цилиндрический резервуар для хранения токсичной жидкости имеет следующий вид:

Какие параметры D, L и R должен иметь этот резервуар, чтобы площадь его поверхности была равна 600 $м^2$ при максимально возможной вместимости? (Используйте ограничение на площадь поверхности резервуара, чтобы исключить одну переменную из целевой функции.)

14. Полуэмпирическая формула зависимости плотности воздуха ρ от высоты h имеет следующий вид:

$$\rho(h) = k_1 e^{-k_2 h}.$$

Найти значения параметров k_1 и k_2 в соответствии с критерием наименьших квадратов

$$Q = \sum_i \left[\rho(h_i) - \rho_i\right]^2$$

на основе следующих экспериментальных данных:

h_i, *м*	0	5000	10000	15000	20000
ρ_i, *êã/ì*^{3}	1,25	0,75	0,42	0,20	0,09

15. В задаче, связанной с принятием решения, желательно минимизировать ожидаемый риск, определяемый следующим образом:

$$\mathring{A}\{\text{ðèñê}\} = e^{-(xy-1)^2} + \frac{(y^2 - 0{,}5)^2}{0{,}132}.$$

Найти минимальный ожидаемый риск и соответствующие значения параметров x и y.

Приложение 2

Исходный текст подпрограммы *HOOKE*

```
      SUBROUTINE HOOKE(N,F,X,DX,EPS,MAXFEV,IER,Y)
*
*     Минимизация функции нескольких переменных методом Хука-Дживса.
*
      INTEGER N,MAXFEV,IER
      REAL F,X(N),DX(N),EPS,Y(N)
      EXTERNAL F
      INTEGER I,K,NFEV
      REAL ALFA,BETA,FX,FY,YI,TEMP
      PARAMETER (ALFA=1.0,BETA=0.5)
      PRINT 10
   10 FORMAT (///' МЕТОД ХУКА-ДЖИВСА')
      PRINT 20,N,EPS,MAXFEV
   20 FORMAT (/' N=',I2,'   EPS=',1PG12.5,'   MAXFEV=',I6)
      PRINT 30,(I,X(I),I,DX(I),I=1,N)
   30 FORMAT (/' Начальные значения переменных и приращений:'/
     1   (' X(',I2,')=',1PG12.5,' DX(',I2,')=',1PG12.5))
*
*     Начальный шаг.
*
      K=0
      NFEV=1
      FX=F(X)
      PRINT 40,K,NFEV
   40 FORMAT (/' Шаг ',I4,': целевая функция вычислена ',I6,' раз(а)')
      PRINT 50,(I,X(I),I=1,N)
   50 FORMAT (4(' X(',I2,')=',1PG12.5:))
      PRINT 60,FX
   60 FORMAT (' Значение целевой функции:',1PG12.5)
*
*     Исследующий поиск вокруг точки X.
*
  100 IF (NFEV.GT.MAXFEV) GOTO 300
      DO 110 I=1,N
         Y(I)=X(I)
  110 CONTINUE
      FY=FX
      DO 120 I=1,N
         YI=Y(I)
         Y(I)=YI+DX(I)
         NFEV=NFEV+1
         TEMP=F(Y)
         IF (FY.GT.TEMP) THEN
            FY=TEMP
            GOTO 120
         ENDIF
         Y(I)=YI-DX(I)
         NFEV=NFEV+1
         TEMP=F(Y)
         IF (FY.GT.TEMP) THEN
            FY=TEMP
            GOTO 120
         ENDIF
         Y(I)=YI
  120 CONTINUE
      IF (FY.LT.FX) GOTO 200
*
*     Нет улучшения целевой функции вокруг точки X - проверка достижения
*     требуемой точности.
*
      DO 130 I=1,N
         TEMP=DX(I)
```

```
         IF (ABS(X(I)).GT.1.0) TEMP=TEMP/X(I)
         IF (ABS(TEMP).GT.EPS) GOTO 150
  130 CONTINUE
      PRINT 140
  140 FORMAT (/' Достигнута требуемая точность')
      IER=0
      RETURN
*
*     Уменьшение приращений переменных и повторение исследующего поиска
*     вокруг точки X.
*
  150 DO 160 I=1,N
         DX(I)=BETA*DX(I)
  160 CONTINUE
      PRINT 170
  170 FORMAT (/' Уменьшены приращения переменных')
      GOTO 100
*
*     Выполнен K-й шаг: F(Y) < F(X).
*
*     Точка X становится старой базовой точкой X';
*     точка Y становится новой  базовой точкой X.
*
*     Вычисление Y = X + ALFA*(X-X')   (поиск по образцу).
*
  200 K=K+1
      DO 210 I=1,N
         TEMP=X(I)
         X(I)=Y(I)
         Y(I)=X(I)+ALFA*(X(I)-TEMP)
  210 CONTINUE
      FX=FY
      PRINT 40,K,NFEV
      PRINT 50,(I,X(I),I=1,N)
      PRINT 60,FX
*
*     Исследующий поиск вокруг точки Y.
*
      IF (NFEV.GT.MAXFEV) GOTO 300
      NFEV=NFEV+1
      FY=F(Y)
      DO 220 I=1,N
         YI=Y(I)
         Y(I)=YI+DX(I)
         NFEV=NFEV+1
         TEMP=F(Y)
         IF (FY.GT.TEMP) THEN
            FY=TEMP
            GOTO 220
         ENDIF
         Y(I)=YI-DX(I)
         NFEV=NFEV+1
         TEMP=F(Y)
         IF (FY.GT.TEMP) THEN
            FY=TEMP
            GOTO 220
         ENDIF
         Y(I)=YI
  220 CONTINUE
      IF (FY.LT.FX) GOTO 200
*
*     Нет улучшения целевой функции вокруг точки Y по сравнению с
*     базовой точкой X - нужен исследующий поиск вокруг точки X.
*
      GOTO 100
*
*     Число вычислений целевой функции превысило допустимый максимум.
```

```
*
  300 PRINT 310
  310 FORMAT (/' Число вычислений целевой функции > MAXFEV')
      IER=1
      RETURN
      END
```

Исходный текст подпрограммы *NELDM*

```
      SUBROUTINE NELDM(N,F,X,SIZE,EPS,MAXFEV,IER,Y,FY)
*
*     Минимизация функции нескольких переменных методом Нелдера-Мида.
*
      INTEGER N,MAXFEV,IER
      REAL F,X(N),SIZE,EPS,Y(N,N+4),FY(N+4)
      EXTERNAL F
      INTEGER N1,N2,N3,N4,I,J,K,NFEV,JL,JH,JS
      REAL ALFA,BETA,GAMA,FX,VN,D1,D2,TEMP
      PARAMETER (ALFA=1.0,BETA=0.5,GAMA=2.0)
      PRINT 10
   10 FORMAT (///' МЕТОД НЕЛДЕРА-МИДА')
      PRINT 20,N,SIZE,EPS,MAXFEV
   20 FORMAT (/' N=',I2,'   SIZE=',1PG12.5,'   EPS=',1PG12.5,
     1   '   MAXFEV=',I6)
      PRINT 30,(I,X(I),I=1,N)
   30 FORMAT (/' Начальные значения переменных:'/
     1   (' X(',I2,')=',1PG12.5))
      N1=N+1
      N2=N+2
      N3=N+3
      N4=N+4
*
*     Начальный шаг.
*
      K=0
      NFEV=1
      FX=F(X)
      PRINT 40,K,NFEV
   40 FORMAT (/' Шаг: ',I4,': целевая функция вычислена ',I6,' раз(а)')
      PRINT 50,(I,X(I),I=1,N)
   50 FORMAT (4(' X(',I2,')=',1PG12.5:))
      PRINT 60,FX
   60 FORMAT (' Значение целевой функции:',1PG12.5)
*
*     Вычисление координат вершин начального многогранника и значений
*     целевой функции в этих вершинах.
*
      VN=N
      D1=SIZE*((SQRT(VN+1.0)+VN-1.0)/(SQRT(2.0)*VN))
      D2=SIZE*((SQRT(VN+1.0)-1.0)/(SQRT(2.0)*VN))
      DO 80 I=1,N
         Y(I,1)=X(I)
         DO 70 J=2,N1
            IF (J.EQ.I+1) THEN
               Y(I,J)=X(I)+D1
            ELSE
               Y(I,J)=X(I)+D2
            ENDIF
   70    CONTINUE
   80 CONTINUE
      FY(1)=FX
      DO 90 J=2,N1
         NFEV=NFEV+1
         FY(J)=F(Y(1,J))
   90 CONTINUE
*
```

```
*     Начало основного цикла поиска.
*
  100 IF (NFEV.LE.MAXFEV) GOTO 120
      PRINT 110
  110 FORMAT (/' Число вычислений целевой функции > MAXFEV')
      IER=1
      RETURN
  120 K=K+1
*
*     JL = номер вершины с наименьшим значением целевой функции;
*     JH = номер вершины с наибольшим значением целевой функции;
*     JS = номер вершины со вторым по величине (после JH-й вершины)
*          значением целевой функции.
*
      JL=1
      JH=1
      DO 130 J=2,N1
         IF (FY(JL).GT.FY(J)) JL=J
         IF (FY(JH).LT.FY(J)) JH=J
  130 CONTINUE
      JS=1
      IF (JH.EQ.1) JS=2
      DO 140 J=1,N1
         IF (J.EQ.JH) GOTO 140
         IF (FY(JS).LT.FY(J)) JS=J
  140 CONTINUE
      DO 150 I=1,N
         X(I)=Y(I,JL)
  150 CONTINUE
      FX=FY(JL)
      PRINT 40,K,NFEV
      PRINT 50,(I,X(I),I=1,N)
      PRINT 60,FX
*
*     N2 = центр тяжести всех вершин, исключая JH-ю вершину.
*
      DO 170 I=1,N
         TEMP=0.0
         DO 160 J=1,N1
            IF (J.NE.JH) TEMP=TEMP+Y(I,J)
  160    CONTINUE
         Y(I,N2)=TEMP/VN
  170 CONTINUE
      NFEV=NFEV+1
      FY(N2)=F(Y(1,N2))
*
*     Проверка условия окончания поиска.
*
      TEMP=0.0
      DO 180 J=1,N1
         TEMP=TEMP+(FY(N2)-FY(J))**2
  180 CONTINUE
      IF (SQRT(TEMP/(VN+1.0)).GT.EPS) GOTO 200
      PRINT 190
  190 FORMAT (/' Достигнута требуемая точность')
      IER=0
      RETURN
*
*     N3 = образ JH-й вершины, отраженный относительно центра тяжести.
*
  200 DO 210 I=1,N
         Y(I,N3)=Y(I,N2)+ALFA*(Y(I,N2)-Y(I,JH))
  210 CONTINUE
      NFEV=NFEV+1
      FY(N3)=F(Y(1,N3))
      IF (FY(N3).LT.FY(JL)) GOTO 300
      IF (FY(N3).GT.FY(JS)) GOTO 400
```

```
*
*     Замена JH-й вершины на более подходящую точку.
*
  220 DO 230 I=1,N
         Y(I,JH)=Y(I,N3)
  230 CONTINUE
      FY(JH)=FY(N3)
      GOTO 100
  240 DO 250 I=1,N
         Y(I,JH)=Y(I,N4)
  250 CONTINUE
      FY(JH)=FY(N4)
      GOTO 100
*
*     FY(N3) < FY(JL): растяжение.
*
  300 DO 310 I=1,N
         Y(I,N4)=Y(I,N2)+GAMA*(Y(I,N3)-Y(I,N2))
  310 CONTINUE
      NFEV=NFEV+1
      FY(N4)=F(Y(1,N4))
      IF (FY(N4).LT.FY(JL)) GOTO 240
      GOTO 220
*
*     FY(N3) > FY(JS): сжатие.
*
  400 IF (FY(N3).GT.FY(JH)) GOTO 420
      DO 410 I=1,N
         Y(I,JH)=Y(I,N3)
  410 CONTINUE
  420 DO 430 I=1,N
         Y(I,N4)=Y(I,N2)+BETA*(Y(I,JH)-Y(I,N2))
  430 CONTINUE
      NFEV=NFEV+1
      FY(N4)=F(Y(1,N4))
      IF (FY(N4).LT.FY(JH)) GOTO 240
*
*     Редукция многогранника относительно JL-й вершины.
*
      DO 450 I=1,N
         DO 440 J=1,N1
            Y(I,J)=Y(I,JL)+0.5*(Y(I,J)-Y(I,JL))
  440    CONTINUE
  450 CONTINUE
      DO 460 J=1,N1
         IF (J.EQ.JL) GOTO 460
         NFEV=NFEV+1
         FY(J)=F(Y(1,J))
  460 CONTINUE
      GOTO 100
      END
```

Исходный текст подпрограммы *STEEP*

```
      SUBROUTINE STEEP(N,F,GRAD,X,EPS,SMAX,EPS1,MAXFEV,IER,G,Y)
*
*     Минимизация функции нескольких переменных методом наискорейшего
*     спуска.
*
      INTEGER N,MAXFEV,IER
      REAL F,X(N),EPS,SMAX,EPS1,G(N),Y(N)
      EXTERNAL F,GRAD
      INTEGER I,K,NFEV,KASE
      REAL FX,GN,A,B,C,SM,S1,S2,FM,F1,F2
      PRINT 10
   10 FORMAT (///' МЕТОД НАИСКОРЕЙШЕГО СПУСКА')
```

```
      PRINT 20,N,EPS,SMAX,EPS1,MAXFEV
   20 FORMAT (/' N='I2,' EPS=',1PG12.5,' SMAX=',1PG12.5,' EPS1=',
     1   1PG12.5,' MAXFEV=',I6)
      PRINT 30,(I,X(I),I=1,N)
   30 FORMAT (/' Начальные значения переменных:'/
     1   (' X(',I2,')=',1PG12.5))
      K=0
      NFEV=1
      FX=F(X)
*
*     Выполнен K-й шаг:
*     X      - базовая точка;
*     FX     - значение целевой функции в базовой точке.
*
  100 NFEV=NFEV+N
      CALL GRAD(X,FX,G)
      GN=0.0
      DO 110 I=1,N
         GN=GN+G(I)**2
  110 CONTINUE
      GN=SQRT(GN)
      PRINT 120,K,NFEV
  120 FORMAT (/' Шаг ',I4,': целевая функция вычислена ',I6,' раз(а)')
      PRINT 130,(I,X(I),I=1,N)
  130 FORMAT (4(' X(',I2,')=',1PG12.5:))
      PRINT 140,(I,G(I),I=1,N)
  140 FORMAT (4(' G(',I2,')=',1PG12.5:))
      PRINT 150,FX,GN
  150 FORMAT (' Значение целевой функции:',1PG12.5,
     1   '   Норма градиента:',1PG12.5)
      IF (GN.GT.EPS) GOTO 170
      PRINT 160
  160 FORMAT (/' Найдена стационарная точка')
      IER=0
      RETURN
  170 IF (NFEV.LE.MAXFEV) GOTO 200
      PRINT 180
  180 FORMAT (/' Число вычислений целевой функции > MAXFEV')
      IER=1
      RETURN
*
*     Минимизация целевой функции вдоль направления наискорейшего спуска
*     методом золотого сечения.
*
  200 K=K+1
      C=0.5*(3.0-SQRT(5.0))
      A=0.0
      B=SMAX
      SM=C*SMAX
      DO 210 I=1,N
         Y(I)=X(I)-SM*(G(I)/GN)
  210 CONTINUE
      NFEV=NFEV+1
      FM=F(Y)
      KASE=0
  220 IF (ABS(A-B).LE.EPS1) THEN
         DO 230 I=1,N
            X(I)=Y(I)
  230    CONTINUE
         FX=FM
         GOTO 100
      ENDIF
      S1=A+C*(B-A)
      S2=B-C*(B-A)
      IF (KASE.EQ.0) THEN
         F1=FM
         DO 240 I=1,N
```

```
            Y(I)=X(I)-S2*(G(I)/GN)
  240    CONTINUE
         NFEV=NFEV+1
         F2=F(Y)
      ELSE
         DO 250 I=1,N
            Y(I)=X(I)-S1*(G(I)/GN)
  250    CONTINUE
         NFEV=NFEV+1
         F1=F(Y)
         F2=FM
      ENDIF
      IF (F1.LT.F2) THEN
         B=S2
         SM=S1
         FM=F1
         KASE=1
      ELSE
         A=S1
         SM=S2
         FM=F2
         KASE=0
      ENDIF
      GOTO 220
      END
```

Исходный текст подпрограммы *NEWTR*

```
      SUBROUTINE NEWTR(N,F,GRAD,HESS,X,EPS,MAXFEV,IER,G,H)
*
*     Минимизация функции нескольких переменных методом Ньютона-Рафсона.
*
      INTEGER N,MAXFEV,IER
      REAL F,X(N),EPS,G(N),H((N*(N+1))/2)
      EXTERNAL F,GRAD,HESS
      INTEGER I,K,NFEV
      REAL FX,GN
      PRINT 10
   10 FORMAT (///' МЕТОД НЬЮТОНА-РАФСОНА')
      PRINT 20,N,EPS,MAXFEV
   20 FORMAT (/' N=',I2,'   EPS=',1PG12.5,'   MAXFEV=',I6)
      PRINT 30,(I,X(I),I=1,N)
   30 FORMAT (/' Начальные значения переменных:'/
     1   (' X(',I2,')=',1PG12.5))
      K=0
      NFEV=1
      FX=F(X)
*
*     Выполнен K-й шаг:
*     X     - базовая точка;
*     FX    - значение целевой функции в базовой точке.
*
  100 NFEV=NFEV+N
      CALL GRAD(X,FX,G)
      GN=0.0
      DO 110 I=1,N
         GN=GN+G(I)**2
  110 CONTINUE
      GN=SQRT(GN)
      PRINT 120,K,NFEV
  120 FORMAT (/' Шаг ',I4,': целевая функция вычислена ',I6,' раз(а)')
      PRINT 130,(I,X(I),I=1,N)
  130 FORMAT (4(' X(',I2,')=',1PG12.5:))
      PRINT 140,(I,G(I),I=1,N)
  140 FORMAT (4(' G(',I2,')=',1PG12.5:))
      PRINT 150,FX,GN
```

```
  150 FORMAT (' Значение целевой функции:',1PG12.5,
     1    '    Норма градиента:',1PG12.5)
      IF (GN.GT.EPS) GOTO 170
      PRINT 160
  160 FORMAT (/' Найдена стационарная точка')
      IER=0
      RETURN
  170 IF (NFEV.LE.MAXFEV) GOTO 200
      PRINT 180
  180 FORMAT (/' Число вычислений целевой функции > MAXFEV')
      IER=1
      RETURN
  200 K=K+1
      NFEV=NFEV+N*N
      CALL HESS(X,FX,H)
      CALL CHOLS(N,H,G,IER)
      IF (IER.EQ.0) GOTO 220
      PRINT 210
  210 FORMAT (/' Матрица Гессе не является положительно определенной')
      IER=2
      RETURN
  220 DO 230 I=1,N
         X(I)=X(I)-G(I)
  230 CONTINUE
      NFEV=NFEV+1
      FX=F(X)
      GOTO 100
      END
      SUBROUTINE CHOLS(N,A,B,IER)
*
*     Решение системы уравнений AX = B с положительно определенной
*     симметрической матрицей коэффициентов.
*
      INTEGER N,IER
      REAL A((N*(N+1))/2),B(N)
      INTEGER I,J,K,P,Q,R
      REAL X
      P=0
      DO 30 I=1,N
         Q=P+1
         R=0
         DO 20 J=1,I
            X=A(P+1)
            DO 10 K=Q,P
               R=R+1
               X=X-A(K)*A(R)
   10       CONTINUE
            R=R+1
            P=P+1
            IF (I.EQ.J) THEN
               IF (X.LE.0.0) THEN
                  IER=1
                  RETURN
               ENDIF
               A(P)=1.0/SQRT(X)
            ELSE
               A(P)=X*A(R)
            ENDIF
   20    CONTINUE
   30 CONTINUE
      P=1
      DO 50 I=1,N
         Q=I-1
         X=B(I)
         DO 40 J=1,Q
            X=X-A(P)*B(J)
            P=P+1
```

```
   40    CONTINUE
         B(I)=X*A(P)
         P=P+1
   50 CONTINUE
      DO 70 I=N,1,-1
         P=P-1
         Q=I+1
         R=P
         X=B(I)
         DO 60 J=N,Q,-1
            X=X-A(R)*B(J)
            R=R-J+1
   60    CONTINUE
         B(I)=X*A(R)
   70 CONTINUE
      IER=0
      RETURN
      END
```

Пример использования подпрограммы *HOOKE*

```
      PROGRAM TANK4
*
*     Проектирование цилиндрического бака для минимизации общих затрат
*     методом Хука-Дживса.
*
      INTEGER N,MAXFEV,IER
      PARAMETER (N=2)
      REAL SIGMA,X(N),DX(N),EPS,Y(N)
      EXTERNAL SIGMA
      X(1)=6.0
      X(2)=6.0
      DX(1)=1.0
      DX(2)=1.0
      EPS=0.01
      MAXFEV=1000
      CALL HOOKE(N,SIGMA,X,DX,EPS,MAXFEV,IER,Y)
      PRINT 10,IER,X(1),X(2),SIGMA(X)
   10 FORMAT (//' IER= ',I1//
     1   ' H=',1PG12.5,' D=',1PG12.5,' SIGMA=',1PG12.5)
      STOP
      END
      REAL FUNCTION SIGMA(X)
      REAL X(2),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      SIGMA=80000.0/(PI*D**2*H)+10.0*PI*D*H+2.5*PI*D**2
      RETURN
      END
```

```
МЕТОД ХУКА-ДЖИВСА

N= 2   EPS= 1.00000E-02   MAXFEV=  1000

Начальные значения переменных и приращений:
X( 1)=  6.0000     DX( 1)=  1.0000
X( 2)=  6.0000     DX( 2)=  1.0000

Шаг    0: целевая функция вычислена      1 раз(а)
X( 1)=  6.0000     X( 2)=  6.0000
Значение целевой функции:  1531.6

Шаг    1: целевая функция вычислена      5 раз(а)
X( 1)=  5.0000     X( 2)=  5.0000
Значение целевой функции:  1185.5
```

```
Шаг    2: целевая функция вычислена     10 раз(а)
X( 1)=  4.0000     X( 2)=  4.0000
Значение целевой функции:  1026.2

Уменьшены приращения переменных

Шаг    3: целевая функция вычислена     20 раз(а)
X( 1)=  3.5000     X( 2)=  4.5000
Значение целевой функции:  1013.1

Шаг    4: целевая функция вычислена     25 раз(а)
X( 1)=  2.5000     X( 2)=  5.0000
Значение целевой функции:  996.49

Уменьшены приращения переменных

Уменьшены приращения переменных

Уменьшены приращения переменных

Шаг    5: целевая функция вычислена     44 раз(а)
X( 1)=  2.5625     X( 2)=  5.0000
Значение целевой функции:  996.37

Уменьшены приращения переменных

Шаг    6: целевая функция вычислена     56 раз(а)
X( 1)=  2.5313     X( 2)=  5.0313
Значение целевой функции:  996.33

Уменьшены приращения переменных

Шаг    7: целевая функция вычислена     68 раз(а)
X( 1)=  2.5156     X( 2)=  5.0469
Значение целевой функции:  996.33

Достигнута требуемая точность

IER= 0

H=  2.5156     D=  5.0469     SIGMA=  996.33
Stop - Program terminated.
```

Пример использования подпрограммы *NELDM*

```
      PROGRAM TANK5
*
*     Проектирование цилиндрического бака для минимизации общих затрат
*     методом Нелдера-Мида.
*
      INTEGER N,MAXFEV,IER
      PARAMETER (N=2)
      REAL SIGMA,X(N),SIZE,EPS,Y(N,N+4),FY(N+4)
      EXTERNAL SIGMA
      X(1)=6.0
      X(2)=6.0
      SIZE=1.0
      EPS=0.01
      MAXFEV=1000
      CALL NELDM(N,SIGMA,X,SIZE,EPS,MAXFEV,IER,Y,FY)
      PRINT 10,IER,X(1),X(2),SIGMA(X)
   10 FORMAT (//' IER= ',I1//
     1   ' H=',1PG12.5,' D=',1PG12.5,' SIGMA=',1PG12.5)
      STOP
      END
      REAL FUNCTION SIGMA(X)
```

```
      REAL X(2),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      SIGMA=80000.0/(PI*D**2*H)+10.0*PI*D*H+2.5*PI*D**2
      RETURN
      END
```

```
МЕТОД НЕЛДЕРА-МИДА

N= 2   SIZE=  1.0000       EPS= 1.00000E-02   MAXFEV=  1000

Начальные значения переменных:
X( 1)=  6.0000
X( 2)=  6.0000

Шаг:    0: целевая функция вычислена      1 раз(а)
X( 1)=  6.0000     X( 2)=  6.0000
Значение целевой функции:  1531.6

Шаг:    1: целевая функция вычислена      3 раз(а)
X( 1)=  6.0000     X( 2)=  6.0000
Значение целевой функции:  1531.6

Шаг:    2: целевая функция вычислена      6 раз(а)
X( 1)=  6.9312     X( 2)=  4.4564
Значение целевой функции:  1311.4

Шаг:    3: целевая функция вычислена      9 раз(а)
X( 1)=  5.4650     X( 2)=  3.1669
Значение целевой функции:  1087.1

Шаг:    4: целевая функция вычислена     12 раз(а)
X( 1)=  5.4650     X( 2)=  3.1669
Значение целевой функции:  1087.1

Шаг:    5: целевая функция вычислена     15 раз(а)
X( 1)=  4.6328     X( 2)=  3.6164
Значение целевой функции:  1049.3
```

(шаги с 6 по 29 не показаны)

```
Шаг:   30: целевая функция вычислена     83 раз(а)
X( 1)=  2.5075     X( 2)=  5.0466
Значение целевой функции:  996.33

Достигнута требуемая точность

IER= 0

H=  2.5075     D=  5.0466     SIGMA=  996.33
Stop - Program terminated.
```

Пример использования подпрограммы *STEEP*

```
      PROGRAM TANK6
*
*     Проектирование цилиндрического бака для минимизации общих затрат
*     методом наискорейшего спуска.
*
      INTEGER N,MAXFEV,IER
      PARAMETER (N=2)
      REAL SIGMA,X(N),EPS,SMAX,EPS1,G(N),Y(N)
      EXTERNAL SIGMA,GRAD
      X(1)=6.0
      X(2)=6.0
```

```
      EPS=0.5
      SMAX=1.0
      EPS1=0.01
      MAXFEV=1000
      CALL STEEP(N,SIGMA,GRAD,X,EPS,SMAX,EPS1,MAXFEV,IER,G,Y)
      PRINT 10,IER,X(1),X(2),SIGMA(X)
   10 FORMAT (//' IER= ',I1//
     1   ' H=',1PG12.5,' D=',1PG12.5,' SIGMA=',1PG12.5)
      STOP
      END
      REAL FUNCTION SIGMA(X)
      REAL X(2),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      SIGMA=80000.0/(PI*D**2*H)+10.0*PI*D*H+2.5*PI*D**2
      RETURN
      END
      SUBROUTINE GRAD(X,FX,G)
      REAL X(2),FX,G(2),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      G(1)=-80000.0/(PI*D**2*H**2)+10.0*PI*D
      G(2)=-160000.0/(PI*D**3*H)+10.0*PI*H+5.0*PI*D
      RETURN
      END
```

```
МЕТОД НАИСКОРЕЙШЕГО СПУСКА

N= 2 EPS=  .50000     SMAX=  1.0000     EPS1= 1.00000E-02 MAXFEV=  1000

Начальные значения переменных:
X( 1)=  6.0000
X( 2)=  6.0000

Шаг    0: целевая функция вычислена      3 раз(а)
X( 1)=  6.0000     X( 2)=  6.0000
G( 1)=  168.85     G( 2)=  243.45
Значение целевой функции:  1531.6       Норма градиента:  296.27

Шаг    1: целевая функция вычислена     16 раз(а)
X( 1)=  5.4330     X( 2)=  5.1824
G( 1)=  130.69     G( 2)=  184.74
Значение целевой функции:  1270.0       Норма градиента:  226.29

Шаг    2: целевая функция вычислена     29 раз(а)
X( 1)=  4.8583     X( 2)=  4.3702
G( 1)=  80.802     G( 2)=  95.674
Значение целевой функции:  1091.5       Норма градиента:  125.23

Шаг    3: целевая функция вычислена     42 раз(а)
X( 1)=  4.3549     X( 2)=  3.7741
G( 1)=  24.301     G( 2)= -21.449
Значение целевой функции:  1038.7       Норма градиента:  32.413

Шаг    4: целевая функция вычислена     55 раз(а)
X( 1)=  3.6090     X( 2)=  4.4325
G( 1)=  39.739     G( 2)=  20.958
Значение целевой функции:  1016.0       Норма градиента:  44.927

Шаг    5: целевая функция вычислена     68 раз(а)
X( 1)=  3.3365     X( 2)=  4.2888
G( 1)=  10.374     G( 2)= -21.310
Значение целевой функции:  1008.9       Норма градиента:  23.701
```

(шаги с 6 по 16 не показаны)

```
Шаг   17: целевая функция вычислена    224 раз(а)
X( 1)=  2.5387     X( 2)=  5.0066
G( 1)= -.33826     G( 2)= -1.4570
Значение целевой функции:  996.34       Норма градиента:  1.4958

Шаг   18: целевая функция вычислена    237 раз(а)
X( 1)=  2.5399     X( 2)=  5.0115
G( 1)=  .26400     G( 2)= -.80518
Значение целевой функции:  996.33       Норма градиента:  .84736

Шаг   19: целевая функция вычислена    250 раз(а)
X( 1)=  2.5348     X( 2)=  5.0269
G( 1)=  1.0891     G( 2)=  .42750
Значение целевой функции:  996.33       Норма градиента:  1.1700

Шаг   20: целевая функция вычислена    263 раз(а)
X( 1)=  2.5301     X( 2)=  5.0251
G( 1)=  .33600     G( 2)= -.21443
Значение целевой функции:  996.33       Норма градиента:  .39860

Найдена стационарная точка

IER= 0

H=  2.5301     D=  5.0251     SIGMA=  996.33
Stop - Program terminated.
```

Пример использования подпрограммы *NEWTR*

```
      PROGRAM TANK7
*
*     Проектирование цилиндрического бака для минимизации общих затрат
*     методом Ньютона-Рафсона.
*
      INTEGER N,MAXFEV,IER
      PARAMETER (N=2)
      REAL SIGMA,X(N),EPS,G(N),H((N*(N+1))/2)
      EXTERNAL SIGMA,GRAD,HESS
      X(1)=2.0
      X(2)=2.0
      EPS=0.5
      MAXFEV=1000
      CALL NEWTR(N,SIGMA,GRAD,HESS,X,EPS,MAXFEV,IER,G,H)
      PRINT 10,IER,X(1),X(2),SIGMA(X)
   10 FORMAT (//' IER= ',I1//
     1   ' H=',1PG12.5,' D=',1PG12.5,' SIGMA=',1PG12.5)
      STOP
      END
      REAL FUNCTION SIGMA(X)
      REAL X(2),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      SIGMA=80000.0/(PI*D**2*H)+10.0*PI*D*H+2.5*PI*D**2
      RETURN
      END
      SUBROUTINE GRAD(X,FX,G)
      REAL X(2),FX,G(2),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      G(1)=-80000.0/(PI*D**2*H**2)+10.0*PI*D
      G(2)=-160000.0/(PI*D**3*H)+10.0*PI*H+5.0*PI*D
      RETURN
      END
      SUBROUTINE HESS(X,FX,HH)
```

```
      REAL X(2),FX,HH(3),H,D,PI
      H=X(1)
      D=X(2)
      PI=3.14159
      HH(1)=160000.0/(PI*D**2*H**3)
      HH(2)=160000.0/(PI*D**3*H**2)+10.0*PI
      HH(3)=480000.0/(PI*D**4*H)+5.0*PI
      RETURN
      END
```

```
МЕТОД НЬЮТОНА-РАФСОНА

N= 2   EPS=  .50000       MAXFEV=  1000

Начальные значения переменных:
X( 1)=  2.0000
X( 2)=  2.0000

Шаг    0: целевая функция вычислена      3 раз(а)
X( 1)=  2.0000     X( 2)=  2.0000
G( 1)= -1528.7     G( 2)= -3088.9
Значение целевой функции:  3340.2       Норма градиента:  3446.4

Шаг    1: целевая функция вычислена     10 раз(а)
X( 1)=  2.4629     X( 2)=  2.4880
G( 1)= -600.03     G( 2)= -1226.3
Значение целевой функции:  1911.4       Норма градиента:  1365.2

Шаг    2: целевая функция вычислена     17 раз(а)
X( 1)=  2.9452     X( 2)=  3.0680
G( 1)= -215.51     G( 2)= -458.10
Значение целевой функции:  1276.4       Норма градиента:  506.26

Шаг    3: целевая функция вычислена     24 раз(а)
X( 1)=  3.2503     X( 2)=  3.7108
G( 1)= -58.473     G( 2)= -146.26
Значение целевой функции:  1056.0       Норма градиента:  157.51

Шаг    4: целевая функция вычислена     31 раз(а)
X( 1)=  3.0135     X( 2)=  4.3785
G( 1)= -8.7086     G( 2)= -37.883
Значение целевой функции:  1005.9       Норма градиента:  38.871

Шаг    5: целевая функция вычислена     38 раз(а)
X( 1)=  2.5611     X( 2)=  4.9143
G( 1)= -6.3686     G( 2)= -9.9046
Значение целевой функции:  996.79       Норма градиента:  11.775

Шаг    6: целевая функция вычислена     45 раз(а)
X( 1)=  2.5229     X( 2)=  5.0296
G( 1)= -.14458     G( 2)= -.39984
Значение целевой функции:  996.32       Норма градиента:  .42518

Найдена стационарная точка

IER= 0

H=  2.5229     D=  5.0296     SIGMA=  996.32
Stop - Program terminated.
```

Работа №3. РЕШЕНИЕ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Цель работы

Целью работы является приобретение практических навыков по использованию программ численных методов для анализа и решения задач нелинейного программирования.

Порядок выполнения работы

1. Получить вариант задания у преподавателя.
2. Привести поставленную задачу к формальному математическому виду:

минимизировать $f(x_1, x_2, \ldots, x_n)$

при ограничениях

$h_i(x_1, x_2, \ldots, x_n) = 0,\ i = 1, 2, \ldots, m,$

$h_i(x_1, x_2, \ldots, x_n) \geq 0,\ i = m+1, m+2, \ldots, p,$

где n — число переменных;
m — число ограничений в виде равенств;
p — общее число ограничений (в виде равенств и неравенств);
$x_1, x_2, \ldots, x_n$ — переменные;
f — целевая функция;
$h_1, h_2, \ldots, h_m$ — функции, задающие ограничения в виде равенств;
$h_{m+1}, h_{m+2}, \ldots, h_p$ — функции, задающие ограничения в виде неравенств.

3. Исходя из содержательного смысла задачи оценить интервалы изменения переменных, в которых может (предположительно) находиться искомое оптимальное решение, и выбрать начальные приближения переменных внутри соответствующих интервалов, стараясь удовлетворить как можно большему числу ограничений, имеющихся в задаче.

4. Составить подпрограмму вычисления целевой функции и функций, определяющих ограничения (см. Указания по использованию подпрограмм). Составить основную программу решения поставленной задачи *методом скользящего допуска*, используя прототип этой программы (см. Указания по использованию подпрограмм).

5. Найти оптимальное решение поставленной задачи (оптимальную точку) $x^*_1, x^*_2, \ldots, x^*_n$, а также значение целевой функции и функций, задающих ограничения, в найденной оптимальной точке. Если оптимальное решение найти не удается, проверить математическую постановку задачи, ее программную реализацию (подпрограмму *PROBLM*), затем выбрать другие начальные приближения переменных и повторить решение задачи.

6. Выполнить параметрический анализ оптимального решения. Для этого ввести в задачу (т. е. в целевую функцию и (или) в ограничения) изменяемый параметр, указанный в варианте задания, и изменить надлежащим образом подпрограмму *PROBLM*. После этого найти оптимальные решения при различных значениях параметра (8-10 значений, заданных преподавателем), используя всякий раз в качестве начальных приближений переменных то оптимальное решение, которое было получено ранее (см. п. 5). Представить результаты параметрического анализа в виде табл. 3.1, после чего построить график зависимости оптимального значения целевой функции от исследуемого параметра (построение производить вручную) и сделать соответствующие содержательные выводы.

Таблица 3.1

Параметр	$f(x^*)$	x^*_1	x^*_2	. . .	x^*_n

Требования к отчету

Отчет должен содержать:

1) исходную постановку оптимизационной задачи;

2) формальную математическую постановку оптимизационной задачи с необходимыми выкладками;

3) обоснованные начальные приближения независимых переменных и найденное оптимальное решение (включая значения целевой функции и ограничений в оптимальной точке);

4) результаты параметрического анализа оптимального решения и соответствующие выводы.

Указания по использованию подпрограмм

Для решения задач нелинейного программирования с ограничениями общего вида может использоваться *метод скользящего (нежесткого) допуска*.

Указанный метод позволяет улучшить значение целевой функции как за счет информации, получаемой в допустимых точках (т. е. в точках, удовлетворяющих всем ограничениям задачи), так и за счет информации, которую удается получить при прохождении через некоторые точки, являющиеся близкими к допустимым («почти» допустимые точки). Интервалы, в пределах которых точки можно считать «почти» допустимыми, в ходе оптимизационного поиска постепенно сокращаются, так что в пределе (по мере приближения к искомому решению задачи нелинейного программирования) учитываются только допустимые точки. При этом критерий, который определяет меру «почти» допустимости, называется *критерием скользящего допуска*. Этот критерий монотонно уменьшается на каждом шаге оптимизационного поиска и стремится к нулю по мере приближения к искомому оптимальному решению.

Подпрограмма, реализующая метод скользящего допуска, называется *FLEX*. Для использования этой подпрограммы необходимо составить следующую основную программу:

```
PROGRAM MAIN
DIMENSION X(50), X1(50,50), X2(50,50), R(100), SUM(50), F(50), SR(50),
        ROLD(100)
COMMON /FLEXC1/ NX, NC, NIC, STEP, ALFA, BETA, GAMA, IN, INF, FDIFER,
        SEQL, K1, K2, K3, K4, K5, K6, K7, K8, K9, X, X1, X2, R, SUM, F, SR,
        ROLD, SCALE, FOLD
EXTERNAL PROBLM
INTEGER NPR, INFO
REAL SIZE, CONVER
NX = (см. ниже)
NC = (см. ниже)
NIC = (см. ниже)
NPR = (см. ниже)
SIZE = (см. ниже)
CONVER = (см. ниже)
```

```
X(1) = (см. ниже)
X(2) = (см. ниже)
. . . . . . . . . . . . . .
X(NX) = (см. ниже)
CALL FLEX(PROBLM, NPR, SIZE, CONVER, INFO)
PRINT *, 'INFO =', INFO
STOP
END
```

Перед вызовом подпрограммы *FLEX* необходимо задать следующие исходные данные (см. соответствующие операторы присваивания):

NX — число переменных в задаче;

NC — число ограничений в виде равенств (допускается задание *NC* = 0, если в задаче нет таких ограничений);

NIC — число ограничений в виде неравенств (допускается задание *NIC* = 0, если в задаче нет таких ограничений);

NPR — номер файла для вывода (печати) информации о ходе решения задачи. Рекомендуется задать *NPR* = 6;

SIZE — начальный размер шага поиска. Рекомендуется задать следующее значение:

$$SIZE \approx \min\left(\frac{0{,}2}{NX}\sum_{i=1}^{NX} d_i, d_1, d_2, \ldots, d_{NX}\right),$$

где d_i — размер оцененного интервала по переменной x_i;

CONVER — малое положительное число, используемое в качестве индикатора сходимости поиска (рекомендуется задавать *CONVER* = $10^{-4} \div 10^{-5}$);

X(1), X(2), ..., X(NX) — начальные приближения переменных, определяющие точку, с которой начинается поиск. Желательно (но не обязательно), чтобы эта начальная точка удовлетворяла, по возможности, всем имеющимся ограничениям в виде равенств и неравенств.

После выполнения подпрограммы *FLEX* возвращается признак завершения *INFO*. Этот признак может быть следующим:

INFO = 0 — получено окончательное (оптимальное) решение, удовлетворяющее критерию скользящего допуска;

INFO = 1 — невозможно найти очередное «почти» допустимое решение. В ряде случаев это может быть связано с тем, что либо ограничения задачи находятся в противоречии друг с другом, либо неудачно выбраны начальные приближения переменных;

INFO = 2 — число переменных превышает 50;

INFO = 3 — общее число ограничений превышает 100;

INFO = 4 — *SIZE* $\le$ 0 и (или) *CONVER* $\le$ 0.

Решаемая задача нелинейного программирования должна быть запрограммирована в виде подпрограммы *PROBLM*, которая вычисляет целевую функцию и функции, задающие ограничения, по заданным значениям переменных. Эта подпрограмма должна иметь следующий вид:

```
      SUBROUTINE PROBLM(INQ)
      DIMENSION (см. строки 2-3 в основной программе)
      COMMON (см. строки 4-6 в основной программе)
      GO TO (10, 20, 30), INQ
C     Вычисление ограничений-равенств
   10 R(1) = h1(x1, x2, ..., xn)
```

```
      R(2) = h2(x1, x2, ..., xn)
      . . . . . . . . . . . . . . . . .
      R(NC) = hm(x1, x2, ..., xn)
      RETURN
C     Вычисление ограничений-неравенств
   20 R(NC+1) = hm+1(x1, x2, ..., xn)
      R(NC+2) = hm+2(x1, x2, ..., xn)
      . . . . . . . . . . . . . . . . .
      R(NC+NIC) = hp(x1, x2, ..., xn)
      RETURN
C     Вычисление целевой функции
   30 R(NC+NIC+1) = f(x1, x2, ..., xn)
      RETURN
      END
```

При вызове подпрограммы *PROBLM* (из подпрограммы *FLEX*) текущие значения переменных x_1, x_2, ..., x_n передаются в элементах массива *X*(1), *X*(2), ..., *X*(*NX*). Не разрешается изменять никакие переменные и массивы (кроме массива *R*) из общего блока *FLEXC*1.

Перед выполнением работы файл с исходным текстом подпрограммы *FLEX* необходимо переписать с библиотечного диска на рабочий.

Варианты задач к работе №3

1. В механическом цехе установлены один сверлильный и пять фрезерных станков, предназначенных для производства узлов, состоящих из деталей *A*, *B* и *C*. Производительность станков выражается следующим образом:

Деталь	Время обработки, *мин*	
	сверлильный	фрезерный
A	3	20
B	5	15
C	2	10

Загрузка станков должна быть равномерной, так чтобы разница во времени работы любых двух станков не превышала 30 *мин* в день (предполагается, что загрузка фрезерных станков равномерная). Каким должно быть расписание обработки деталей на станках, при котором дневное производство готовых узлов максимально, если рабочий день длится 8 *ч* ?

Исследуемый параметр: производительность сверлильного станка про обработке детали *B*.

2. В отделе технического контроля (ОТК) машиностроительного предприятия работают контролеры I, II и III разрядов. Норма выработки ОТК за 8-часовой день составляет не менее 1800 изделий. Контролер I разряда проверяет 25 изделий в час, причем не ошибается в 98% случаев. Контролер II разряда проверяет 15 изделий в час, а его точность составляет 95%. Контролер III разряда проверяет 12 изделий в час с точностью 95%. Заработная плата контролера I разряда равна 4 тыс. руб. за час, контролера II разряда — 3 тыс. руб. за час, контролера III разряда — 2,5 тыс. руб. за час. При каждой ошибке контролера предприятие несет убыток в размере 2 тыс. руб. Предприятие может использовать 8 контролеров I разряда, 10 контролеров II разряда и 15 контролеров III разряда. При каком составе ОТК общие затраты на контроль будут минимальными?

Исследуемый параметр: заработная плата контролера I разряда.

3. Текстильное предприятие выпускает продукты *A*, *B* и *C*. На производство единицы продукта *A* требуется затратить 1 *ч* труда ИТР, 10 *ч* физического труда и 3 *кг* сырья. Для единицы продукта *B* соответствующие показатели равны 2 *ч*, 4 *ч* и 2 *кг*, а для единицы продукта *C* — 1 *ч*, 5 *ч* и 1 *кг*. Ресурсы предприятия составляют 100 *ч* труда ИТР, 700 *ч* физического труда и 400 *кг* сырья. При оптовых закупках покупателям предоставляются скидки, так что прибыль от продажи продукции изменяются следующим образом:

Продукт *A*		Продукт *B*		Продукт *C*	
Продажа, ед.	Прибыль, тыс. руб./ед.	Продажа, ед.	Прибыль, тыс. руб./ед.	Продажа, ед.	Прибыль, тыс. руб./ед.
0-40 40-100 100-150 ≥ 150	10 9 8 7	0-50 50-100 ≥ 100	6 4 3	0-100 ≥ 100	5 4

Каким должен быть наиболее доходный производственный план?

Исследуемый параметр: затраты сырья для производства единицы продукта *B*.

4. Требуется спроектировать прямоугольную конструкцию в форме параллелепипеда с открытой передней стенкой. Конструкция должна иметь объем 16 000 $м^3$, но периметр ее основания не должен превышать 220 *м*, глубина должна быть не больше 60 *м*, а ширина — не больше 80 *м*. Кроме того, ширина не должна превышать утроенной глубины, а высота не должна превышать двух третей ширины. Стоимость гофрированного материала, из которого изготавливаются крыша и три стенки конструкции, составляет 1200 *руб* за квадратный метр. Какими должны быть размеры конструкции, чтобы затраты на материалы были как можно меньше?

Исследуемый параметр: ограничение на ширину конструкции.

5. Для осуществления капиталовложений в начале 1-го года имеется сумма в 100 млн. руб., а в начале 2-го года — 200 млн. руб. Инвестиции можно производить в начале 1-го, 2-го и 3-го года. Кроме того, имеющиеся деньги можно помещать в банк на срочный счет, получая при этом 5% годовых. В каждом периоде вклад каждого вида должен быть не меньше 10 млн. руб. и не больше 50 млн. руб. Прибыль на инвестированный капитал составляет 18%, 14% и 10% годовых в конце 1-го, 2-го и 3-го года соответственно. Как следует осуществлять капиталовложения, чтобы максимизировать общий доход?

Исследуемый параметр: прибыль (в %%) на инвестированный капитал в конце 1-го года.

6. Для производства трех продуктов *A*, *B* и *C* необходимы два химических процесса. На производство единицы продукта *A* требуется процесс 1 в течение 2 *ч* и процесс 2 в течение 4 *ч*. На производство единицы продукта *B* требуется процесс 1 в течение 1,5 *ч* и процесс 2 в течение 6 *ч*. На производство единицы продукции *C* требуется процесс 1 в течение 2 *ч* и процесс 2 в течение 5 *ч*. Процесс 1 можно использовать в течение 16 *ч* в сутки, процесс 2 — круглосуточно (24 *ч*). При производстве продукта *C* в качестве побочного также получается продукт *H*. Некоторую часть этого побочного продукта можно продать, а остаток уничтожается. Прибыль по продукту *A* равна 4 тыс. руб. за единицу, по продукту *B* — 6 тыс. руб. за единицу, по продукту *C* — 10 тыс. руб. за единицу. Прибыль по продукту *H* составляет 3 тыс. руб. за единицу, а затраты на его ликвидацию — 2 тыс. руб. за единицу. По прогнозу возможность сбыта продукта *H* составляет 5 единиц в сутки. При выпуске единицы продукта *C* выход продукта *H* равен двум единицам. При каком суточном производстве про-

дуктов *A*, *B* и *C* (с учетом продукта *H*) можно получить максимальную прибыль от их реализации?

Исследуемый параметр: затраты на ликвидацию продукта *H*.

7. Торговое предприятие приобрело 50 телевизоров, заплатив по 200 тыс. руб. за каждый. Из-за непостоянства спроса на эти телевизоры их продажная цена изменяется каждый месяц. Кроме того, каждый месяц изменяются затраты на хранение телевизоров. Соответствующие исходные данные следующие:

Число телевизоров	0-10	11-25	26-45	46 и более
Затраты на хранение одного телевизора, тыс. руб.				
Месяц 1	18	16	14	12
Месяц 2	26	24	22	20
Месяц 3	24	22	20	18
Продажная цена одного телевизора, тыс. руб.				
Месяц 1	450	400	350	300
Месяц 2	400	350	300	250
Месяц 3	550	500	450	400
Месяц 4	500	450	400	350

Сколько телевизоров надо продавать в 1-м, 2-м, 3-м и 4-м месяце, чтобы получить максимальную прибыль?

Исследуемый параметр: продажная цена одного телевизора в 4-м месяце в случае продажи более 45 телевизоров.

8. Химическое предприятие производит растворитель особого состава в трех вариантах *A*, *B* и *C*, отличающихся по чистоте. Прибыль по *A* составляет 4000 *руб/л*, по *B* — 3000 *руб/л*, по *C* — 2500 *руб/л*. Время производства продукта *A* в два раза превышает время производства продукта *B* и в три раза время производства продукта *C*. При условии выпуска только продуктов *B* и *C* предприятие может производить их в общем количестве до 1000 *л* в день. По техническим причинам при выпуске всех трех продуктов общее производство не превышает 800 *л* в день. Контракт предусматривает, что каждый день должно производиться не менее 150 *л* продукта *B*. Каковы оптимальные объемы выпуска продуктов *A*, *B* и *C* при условии, что всю производимую продукцию можно реализовать?

Исследуемый параметр: ограничение на ежедневное производство продукта *B*.

9. Финансовый директор разрабатывает план осуществления денежных вкладов. Общая сумма вкладываемых денег составляет 280 млн. руб. Имеется шесть типов ценных бумаг, доход по которым составляет 3,5%, 2,5%, 3,0%, 4,5%, 5,0% и 4,0% соответственно. По существующему соглашению не менее 35% от общей суммы должны составлять вложения в ценные бумаги 1-го и 2-го типов, и не более 40% — в ценные бумаги 3-го и 4-го типов или 5-го и 6-го типов. Каким должен быть оптимальный план капиталовложений?

Исследуемый параметр: ограничение (в %%) на капиталовложения в ценные бумаги 1-го и 2-го типов.

10. Со склада необходимо вывезти 20 000 $м^3$ товара, причем 1 $м^3$ товара весит 820 *кг*. Хранение на складе 1 *т* товара в течение одних суток стоит 30 *руб*. Раз в сутки к складу подается поезд, которым можно вывозить товар. Для перевозки товара необходимо изготовить контейнер в форме параллелепипеда, ширина которого не превышает 6 *м*, а высота — 8 *м*. Материал, из которого изготавливается контейнер, представляет собой отходы производства. Всего имеется 8000 $м^2$ таких отходов, причем 1 $м^2$ отходов весит 20 *кг*. Стоимость изготовления контейнера объема *V*

равна $150\sqrt{V}$ *руб*. Стоимость перевозки по железной дороге 1 *т* груза (брутто) равна 1300 *руб*. Какие размеры должен иметь контейнер, чтобы суммарная стоимость перевозки товара (с учетом затрат на его хранение) была как можно меньше?

Исследуемый параметр: ограничение на запас отходов, из которых изготавливается контейнер.

11. Четыре шахты поставляют каменный уголь пяти газоперерабатывающим заводам. Расходы на транспортировку угля (*руб*/*т*), а также данные по предложению и спросу следующие:

	Завод 1	2	3	4	5	Предложение, *т*
Шахта 1	100	90	80	90	70	500
Шахта 2	40	80	120	70	90	900
Шахта 3	60	30	40	20	80	700
Шахта 4	70	60	50	40	30	600
Спрос, *т*	400	700	400	500	700	

При какой схеме транспортировки угля общие затраты на транспортировку будут минимальными?

Исследуемый параметр: производственная мощность 3-й шахты.

12. Предприятию необходимо спланировать производство своей продукции в течение четырех недель. Стоимость производства единицы продукции равна 10 тыс. руб. в течение первых двух недель и 15 тыс. руб. в течение последующих двух недель. Необходимо удовлетворить недельные потребности заказчика в продуктах, которые равны 300, 700, 900 и 800 ед. соответственно. Максимальное производство данной продукции в одну неделю составляет 700 ед. Кроме того, в течение второй и третьей недель предприятие имеет возможность применять сверхурочные работы. При использовании сверхурочных работ недельный выпуск продукции может возрасти на 200 ед., но стоимость производства единицы дополнительной продукции выше на 5 тыс. руб. Стоимость хранения единицы продукции составляет 3 тыс. руб. в неделю. Как спланировать производство продукции, чтобы минимизировать общие производственные затраты?

Исследуемый параметр: стоимость хранения единицы продукции.

13. Предприятие должно удовлетворить спрос на продукцию в течение трех месяцев. В каждом месяце продукция может выпускаться в рабочее и сверхурочное время. Имеются следующие данные:

Месяц	Производительность, ед.		Стоимость производства, тыс. руб. / ед.		Спрос, ед.
	Рабочее время	Сверхурочное время	Рабочее время	Сверхурочное время	
1	100	20	14	18	60
2	100	10	17	22	80
3	60	20	17	22	140

Стоимость хранения единицы продукции на складе в течение одного месяца равна 500 руб. Уровень запасов в начале первого месяца составляет 15 единиц. Каким должен быть оптимальный производственный план?

Исследуемый параметр: стоимость хранения единицы продукции.

14. Нефтеперерабатывающее предприятие производит смеси *A* и *B* следующим образом. Сначала смешиваются два потока сырья: V_1 с содержанием серы 3% и V_2 с содержанием серы 1%. Затем полученная смесь смешивается в двух различных пропорциях с третьим потоком сырья V_3, имеющим 2% серы; при этом получаются смеси *A* и *B*. Смесь *A* должна производиться в количестве не более 100 *л* в час, причем содержание серы не должно превосходить 2,5%; для смеси *B* аналогичные показатели равны 200 *л* в час и 1,5%. Стоимость сырья для потоков V_1, V_2 и V_3 равна соответственно 600, 1600 и 1000 руб. за 1 литр. Продажная цена смесей *A* и *B* составляет соответственно 900 и 1000 руб. за 1 литр. При каких объемах потоков сырья можно получить максимальный доход?

Исследуемый параметр: продажная цена смеси *B*.

15. Кирпичный завод получил заказ на кирпич. который должен быть выполнен в ближайшие 24 часа. Транспортный отдел располагает семью грузовиками, имеющими грузоподъемность 15, 12, 13, 16, 7, 8 и 6 тонн. Эксплуатационные расходы для этих грузовиков составляют соответственно 25, 31, 22, 30, 44, 48 и 41 тыс. руб. на каждый рейс. Какой должна быть оптимальная схема перевозки, если время на выполнение одного рейса (до заказчика и обратно) составляет 8 часов, а всего было заказано 200 тонн кирпича?

Исследуемый параметр: размер заказа.

Исходный текст подпрограммы *FLEX*

```
      SUBROUTINE FLEX(PROBLM,NPR,SIZE,CONVER,INFO)
      EXTERNAL PROBLM
      INTEGER NPR,INFO
      REAL SIZE,CONVER
C
C     * * * * * * P R O G R A M   F L E X I P L E X * * * * * *
C
C     NX      TOTAL NUMBER OF INDEPENDENT VARIABLES
C     NC      TOTAL NUMBER OF EQUALITY CONSTRAINTS
C     NIC     TOTAL NUMBER OF INEQUALITY CONSTRAINTS
C     SIZE    EDGE LENGTH OF THE INITIAL POLYHEDRON
C     CONVER  CONVERGENCE CRITERION FOR TERMINATION OF THE SEARCH
C     ALFA    THE REFLECTION COEFFICIENT
C     BETA    THE CONTRACTION COEFFICIENT
C     GAMA    THE EXPANSION COEFFICIENT
C     X(I)    THE ASSUMED VECTOR TO INITIATE THE SEARCH
C     FDIFER  THE TOLERANCE CRITERION FOR CONSTRAINT VIOLATION
C     ICONT   A COUNTER TO RECORD STAGE COMPUTATIONS
C     NCONT   A COUNTER TO PRINT INFORMATION EVERY (NX+1) STAGE
C     LOW     AN INDEX TO IDENTIFY INFORMATION RELATED TO THE LOWEST
C             VALUE OF OBJECTIVE FUNCTION IN MOST RECENT POLYHEDRON
C     LHIGH   AN INDNX TO IDENTIFY INFORMATION RELATED TO LARGEST VALUE
C             OF OBJECTIVE FUNCTION IN MOST RECENT POLYHEDRON
C     LSEC    AN INDEX TO IDENTIFY INFORMATION RELATED TO THE SECOND
C             LARGEST VALUE OF OBJECTIVE FUNCTION IN MOST RECENT
C             POLYHEDRON
C
      DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
     1 ROLD(100),H(50)
      COMMON /FLEXC1/NX,NC,NIC,STEP,ALFA,BETA,GAMA,IN,INF,FDIFER,SEQL,
     1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
      COMMON /FLEXC2/LFEAS,L5,L6,L7,L8,L9,R1A,R2A,R3A
      ALFA=1.0
      BETA=0.5
      GAMA=2.0
      STEP=SIZE
      K1=NX+1
      K2=NX+2
      K3=NX+3
      K4=NX+4
      K5=NX+5
      K6=NC+NIC
      K7=NC+1
      K8=NC+NIC
      K9=K8+1
      N=NX-NC
      N1=N+1
      IF (N1.GE.3) GO TO 50
      N1=3
      N=2
   50 N2=N+2
      N3=N+3
      N4=N+4
      N5=N+5
      N6=N+6
      N7=N+7
      N8=N+8
      XN=N
      XNX=NX
      XN1=N1
      R1A=0.5*(SQRT(5.0)-1.0)
      R2A=R1A*R1A
```

```
      R3A=R2A*R1A
      L5=NX+5
      L6=NX+6
      L7=NX+7
      L8=NX+8
      L9=NX+9
      ICONT=1
      NCONT=1
      FDIFER=2.0*(NC+1)*STEP
      FOLD=FDIFER
      IN=N1
      CALL SUMR(PROBLM)
      SR(N1)=SQRT(SEQL)
      IF (NPR.GT.0) WRITE (NPR,763) FDIFER,SR(N1)
      IF (SR(N1).LT.FDIFER) GO TO 341
      CALL WRITEX(PROBLM,NPR)
      IF (NPR.GT.0) WRITE (NPR,757)
      INF=N1
      STEP=0.05*FDIFER
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
      IF (NPR.GT.0) WRITE (NPR,764)
      IF (NPR.GT.0) WRITE (NPR,116) (J,X2(INF,J),J=1,NX)
      IF (NPR.GT.0) WRITE (NPR,765) SR(INF)
  341 IF (NPR.GT.0) WRITE (NPR,35)
      IF (NPR.GT.0) WRITE (NPR,758) ICONT,FDIFER
      CALL WRITEX(PROBLM,NPR)
      FTER=R(K9)
C
C     COMPUTE CENTROID OF ALL VERTICES OF INITIAL POLYHEDRON
C
  237 STEP1=STEP*(SQRT(XNX+1.0)+XNX-1.0)/(XNX*SQRT(2.0))
      STEP2=STEP*(SQRT(XNX+1.0)-1.0)/(XNX*SQRT(2.0))
      ETA=(STEP1+(XNX-1.0)*STEP2)/(XNX+1.0)
      DO 4 J=1,NX
      X(J)=X(J)-ETA
    4 CONTINUE
      CALL START
      DO 9 I=1,N1
      DO 9 J=1,NX
      X2(I,J)=X1(I,J)
    9 CONTINUE
      DO 5 I=1,N1
      IN=I
      DO 6 J=1,NX
    6 X(J)=X2(I,J)
      CALL SUMR(PROBLM)
      SR(I)=SQRT(SEQL)
      IF (SR(I).LT.FDIFER) GO TO 8
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
    8 CALL PROBLM(3)
      F(I)=R(K9)
    5 CONTINUE
 1000 STEP=0.05*FDIFER
      ICONT=ICONT+1
C
C     SELECT LARGEST VALUE OF OBJECTIVE FUNCTION FROM POLYHEDRON
C     VERTICES
C
      FH=F(1)
      LHIGH=1
      DO 16 I=2,N1
      IF (F(I).LT.FH) GO TO 16
      FH=F(I)
      LHIGH=I
   16 CONTINUE
```

```
C
C     SELECT MINIMUM VALUE OF OBJECTIVE FUNCTION FROM POLYHEDRON
C     VERTICES
C
   41 FL=F(1)
      LOW=1
      DO 17 I=2,N1
      IF (FL.LT.F(I)) GO TO 17
      FL=F(I)
      LOW=I
   17 CONTINUE
      DO 86 J=1,NX
   86 X(J)=X2(LOW,J)
      IN=LOW
      CALL SUMR(PROBLM)
      SR(LOW)=SQRT(SEQL)
      IF (SR(LOW).LT.FDIFER) GO TO 87
      INF=LOW
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
      CALL PROBLM(3)
      F(LOW)=R(K9)
      GO TO 41
   87 CONTINUE
C
C     FIND CENTROID OF POINTS WITH I DIFFERENT THAN LHIGH
C
      DO 19 J=1,NX
      SUM2=0.0
      DO 20 I=1,N1
   20 SUM2=SUM2+X2(I,J)
   19 X2(N2,J)=1.0/XN*(SUM2-X2(LHIGH,J))
      SUM2=0.0
      DO 36 I=1,N1
      DO 36 J=1,NX
      SUM2=SUM2+(X2(I,J)-X2(N2,J))**2
   36 CONTINUE
      FDIFER=(NC+1)/XN1*SQRT(SUM2)
      IF (FDIFER.LT.FOLD) GO TO 98
      FDIFER=FOLD
      GO TO 198
   98 FOLD=FDIFER
  198 CONTINUE
      FTER=F(LOW)
  137 NCONT=NCONT+1
      IF (NCONT.LT.4*N1) GO TO 37
      IF (ICONT.LT.1500) GO TO 337
      FOLD=0.5*FOLD
  337 NCONT=0
      IF (NPR.GT.0) WRITE (NPR,35)
      IF (NPR.GT.0) WRITE (NPR,758) ICONT,FDIFER
      CALL WRITEX(PROBLM,NPR)
   37 IF (FDIFER.LT.CONVER) GO TO 81
C
C     SELECT SECOND LARGEST VALUE OF OBJECTIVE FUNCTION
C
      IF (LHIGH.EQ.1) GO TO 43
      FS=F(1)
      LSEC=1
      GO TO 44
   43 FS=F(2)
      LSEC=2
   44 DO 18 I=1,N1
      IF (LHIGH.EQ.I) GO TO 18
      IF (F(I).LT.FS) GO TO 18
      FS=F(I)
      LSEC=I
```

```
   18 CONTINUE
C
C     REFLECT HIGH POINT THROUGH CENTROID
C
      DO 61 J=1,NX
      X2(N3,J)=X2(N2,J)+ALFA*(X2(N2,J)-X2(LHIGH,J))
   61 X(J)=X2(N3,J)
      IN=N3
      CALL SUMR(PROBLM)
      SR(N3)=SQRT(SEQL)
   89 IF (SR(N3).LT.FDIFER) GO TO 82
      INF=N3
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
   82 CALL PROBLM(3)
      F(N3)=R(K9)
      IF (F(N3).LT.F(LOW)) GO TO 84
      IF (F(N3).LT.F(LSEC)) GO TO 92
      GO TO 60
   92 DO 93 J=1,NX
   93 X2(LHIGH,J)=X2(N3,J)
      SR(LHIGH)=SR(N3)
      F(LHIGH)=F(N3)
      GO TO 1000
C
C     EXPAND VECTOR OF SEARCH ALONG DIRECTION THROUGH CENTROID AND
C     REFLECTION VECTOR
C
   84 DO 23 J=1,NX
      X2(N4,J)=X2(N3,J)+GAMA*(X2(N3,J)-X2(N2,J))
   23 X(J)=X2(N4,J)
      IN=N4
      CALL SUMR(PROBLM)
      SR(N4)=SQRT(SEQL)
      IF (SR(N4).LT.FDIFER) GO TO 25
      INF=N4
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
   25 CALL PROBLM(3)
      F(N4)=R(K9)
      IF (F(LOW).LT.F(N4)) GO TO 92
      DO 26 J=1,NX
   26 X2(LHIGH,J)=X2(N4,J)
      F(LHIGH)=F(N4)
      SR(LHIGH)=SR(N4)
      GO TO 1000
   60 IF (F(N3).GT.F(LHIGH)) GO TO 64
      DO 65 J=1,NX
   65 X2(LHIGH,J)=X2(N3,J)
   64 DO 66 J=1,NX
      X2(N4,J)=BETA*X2(LHIGH,J)+(1.0-BETA)*X2(N2,J)
   66 X(J)=X2(N4,J)
      IN=N4
      CALL SUMR(PROBLM)
      SR(N4)=SQRT(SEQL)
      IF (SR(N4).LT.FDIFER) GO TO 67
      INF=N4
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
   67 CALL PROBLM(3)
      F(N4)=R(K9)
      IF (F(LHIGH).GT.F(N4)) GO TO 68
      DO 69 J=1,NX
      DO 69 I=1,N1
   69 X2(I,J)=0.5*(X2(I,J)+X2(LOW,J))
      DO 70 I=1,N1
      DO 71 J=1,NX
```

```
   71 X(J)=X2(I,J)
      IN=I
      CALL SUMR(PROBLM)
      SR(I)=SQRT(SEQL)
      IF (SR(I).LT.FDIFER) GO TO 72
      INF=I
      CALL FEASBL(PROBLM,SIZE,NPR)
      IF (FOLD.LT.1.0E-9) GO TO 80
   72 CALL PROBLM(3)
   70 F(I)=R(K9)
      GO TO 1000
   68 DO 73 J=1,NX
   73 X2(LHIGH,J)=X2(N4,J)
      SR(LHIGH)=SR(N4)
      F(LHIGH)=F(N4)
      GO TO 1000
   81 IF (NPR.GT.0) WRITE (NPR,760) ICONT,FDIFER
      CALL WRITEX(PROBLM,NPR)
      INFO=0
      RETURN
   80 IF (NPR.GT.0) WRITE (NPR,760) ICONT,FDIFER
      CALL WRITEX(PROBLM,NPR)
      INFO=1
      RETURN
  763 FORMAT (//' Начальный скользящий допуск = ',1PE12.5///' Сумма нару
     1шенных ограничений = ',1PE12.5)
  757 FORMAT (//' Начальный вектор независимых переменных не удовлетворя
     1ет начальному'/' скользящему допуску')
  764 FORMAT (//' Программа нашла вектор независимых переменных, который
     1 удовлетворяет'/' начальному скользящему допуску:')
  116 FORMAT (/(4(' X(',I2,')=',1PE12.5,1X)))
  765 FORMAT (//' Сумма нарушенных ограничений = ',1PE12.5)
   35 FORMAT (//' * * * * * * * * * * * * * * * * * * * * * * * * *')
  758 FORMAT (//' Номер шага вычислений = ',I6///' Скользящий допуск = '
     1 ,1PE12.5)
  760 FORMAT (//' Общее число шагов вычислений = ',I6///' Достигнутая то
     1чность решения = ',1PE12.5)
      END
      SUBROUTINE START
      DIMENSION A(50,50)
      DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
     1 ROLD(100)
      COMMON /FLEXC1/NX,NC,NIC,STEP,ALFA,BETA,GAMA,IN,INF,FDIFER,SEQL,
     1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
      COMMON /FLEXC2/LFEAS,L5,L6,L7,L8,L9,R1A,R2A,R3A
      VN=NX
      STEP1=STEP/(VN*SQRT(2.0))*(SQRT(VN+1.0)+VN-1.0)
      STEP2=STEP/(VN*SQRT(2.0))*(SQRT(VN+1.0)-1.0)
      DO 1 J=1,NX
    1 A(1,J)=0.0
      DO 2 I=2,K1
      DO 4 J=1,NX
    4 A(I,J)=STEP2
      L=I-1
      A(I,L)=STEP1
    2 CONTINUE
      DO 3 I=1,K1
      DO 3 J=1,NX
    3 X1(I,J)=X(J)+A(I,J)
      RETURN
      END
      SUBROUTINE FEASBL(PROBLM,SIZE,NPR)
      EXTERNAL PROBLM
      REAL SIZE
      INTEGER NPR
C
C     SUBROUTINE FEASBL MINIMIZES THE SUM OF THE SQUARE VALUES OF THE
```

```
C      VIOLATED CONSTRAINTS; IT IS CALLED EVERY TIME THE COMBINED VALUE
C      OF THE VIOLATED CONSTRAINTS EXCEEDS THE VALUE OF THE TOLERANCE
C      CRITERION FOR THE CURRENT STAGE
C
       DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
      1 ROLD(100),R1(100),R2(100),R3(100),FLG(10),H(50)
       COMMON /FLEXC1/NX,NC,NIC,STEP,DUM1,DUM2,DUM3,IN,INF,FDIFER,SEQL,
      1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
       COMMON /FLEXC2/LFEAS,L5,L6,L7,L8,L9,R1A,R2A,R3A
       ALFA=1.0
       BETA=0.5
       GAMA=2.0
       XNX=NX
       ICONT=0
       LCHEK=0
       ICHEK=0
       VOLD=-1960.0
       KOLD=0
    25 CALL START
       DO 3 I=1,K1
       DO 4 J=1,NX
     4 X(J)=X1(I,J)
       IN=I
       CALL SUMR(PROBLM)
     3 CONTINUE
C
C      SELECT LARGEST VALUE OF SUM(I) IN SIMPLEX
C
    28 SUMH=SUM(1)
       INDEX=1
       DO 7 I=2,K1
       IF (SUM(I).LE.SUMH) GO TO 7
       SUMH=SUM(I)
       INDEX=I
     7 CONTINUE
C
C      SELECT MINIMUM VALUE OF SUM(I) IN SIMPLEX
C
       SUML=SUM(1)
       KOUNT=1
       DO 8 I=2,K1
       IF (SUML.LE.SUM(I)) GO TO 8
       SUML=SUM(I)
       KOUNT=I
     8 CONTINUE
C
C      FIND CENTROID OF POINTS WITH I DIFFERENT THAN INDEX
C
       DO 9 J=1,NX
       SUM2=0.0
       DO 10 I=1,K1
    10 SUM2=SUM2+X1(I,J)
       X1(K2,J)=1.0/XNX*(SUM2-X1(INDEX,J))
C
C      FIND REFLECTION OF HIGH POINT THROUGH CENTROID
C
       X1(K3,J)=2.0*X1(K2,J)-X1(INDEX,J)
     9 X(J)=X1(K3,J)
       IN=K3
       CALL SUMR(PROBLM)
       IF (SUM(K3).LT.SUML) GO TO 11
C
C      SELECT SECOND LARGEST VALUE IN SIMPLEX
C
       IF (INDEX.EQ.1) GO TO 38
       SUMS=SUM(1)
       GO TO 39
```

```
   38 SUMS=SUM(2)
   39 DO 12 I=1,K1
      IF ((INDEX-I).EQ.0) GO TO 12
      IF (SUM(I).LE.SUMS) GO TO 12
      SUMS=SUM(I)
   12 CONTINUE
      IF (SUM(K3).GT.SUMS) GO TO 13
      GO TO 14
C
C     FORM EXPANSION OF NEW MINIMUM IF REFLECTION HAS PRODUCED ONE
C     MINIMUM
C
   11 DO 15 J=1,NX
      X1(K4,J)=X1(K2,J)+2.0*(X1(K3,J)-X1(K2,J))
   15 X(J)=X1(K4,J)
      IN=K4
      CALL SUMR(PROBLM)
      IF (SUM(K4).LT.SUML) GO TO 16
      GO TO 14
   13 IF (SUM(K3).GT.SUMH) GO TO 17
      DO 18 J=1,NX
   18 X1(INDEX,J)=X1(K3,J)
   17 DO 19 J=1,NX
      X1(K4,J)=0.5*X1(INDEX,J)+0.5*X1(K2,J)
   19 X(J)=X1(K4,J)
      IN=K4
      CALL SUMR(PROBLM)
      IF (SUMH.GT.SUM(K4)) GO TO 6
C
C     REDUCE SIMPLEX BY HALF IF REFLECTION TO PRODUCE A LARGER VALUE
C     THAN THE MAXIMUM
C
      DO 20 J=1,NX
      DO 20 I=1,K1
   20 X1(I,J)=0.5*(X1(I,J)+X1(KOUNT,J))
      DO 29 I=1,K1
      DO 30 J=1,NX
   30 X(J)=X1(I,J)
      IN=I
      CALL SUMR(PROBLM)
   29 CONTINUE
    5 SUML=SUM(1)
      KOUNT=1
      DO 23 I=2,K1
      IF (SUML.LT.SUM(I)) GO TO 23
      SUML=SUM(I)
      KOUNT=I
   23 CONTINUE
      SR(INF)=SQRT(SUM(KOUNT))
      DO 27 J=1,NX
   27 X(J)=X1(KOUNT,J)
      GO TO 26
    6 DO 31 J=1,NX
   31 X1(INDEX,J)=X1(K4,J)
      SUM(INDEX)=SUM(K4)
      GO TO 5
   16 DO 21 J=1,NX
      X1(INDEX,J)=X1(K4,J)
   21 X(J)=X1(INDEX,J)
      SUM(INDEX)=SUM(K4)
      SR(INF)=SQRT(SUM(K4))
      GO TO 26
   14 DO 22 J=1,NX
      X1(INDEX,J)=X1(K3,J)
   22 X(J)=X1(INDEX,J)
      SUM(INDEX)=SUM(K3)
      SR(INF)=SQRT(SUM(K3))
```

```
   26 ICONT=ICONT+1
      DO 36 J=1,NX
   36 X2(INF,J)=X(J)
      IF (ICONT.LT.2*K1) GO TO 50
      ICONT=0
      DO 24 J=1,NX
   24 X(J)=X1(K2,J)
      IN=K2
      CALL SUMR(PROBLM)
      DIFER=0.0
      DO 57 I=1,K1
   57 DIFER=DIFER+(SUM(I)-SUM(K2))**2
      DIFER=1.0/(K7*XNX)*SQRT(DIFER)
      IF (DIFER.GT.1.0E-5) GO TO 50
C
C     IF FLEXIBLE SIMPLEX METHOD FAILED TO SATISFY THE CONSTRAINTS
C     WITHIN THE TOLERANCE CRITERION FOR THE CURRENT STAGE, THE SEARCH
C     IS PERTURBED FROM THE POSITION WHERE THE X VECTOR IS STUCK AND
C     THEN FEASBL IS REPEATED ONCE MORE FROM THE BEGINNING
C
   51 IN=K1
      STEP=20.0*FDIFER
      CALL SUMR(PROBLM)
      SR(INF)=SQRT(SEQL)
      DO 52 J=1,NX
   52 X1(K1,J)=X(J)
      DO 53 J=1,NX
      FACTOR=1.0
      X(J)=X1(K1,J)+FACTOR*STEP
      X1(L9,J)=X(J)
      IN=L9
      CALL SUMR(PROBLM)
      X(J)=X1(K1,J)-FACTOR*STEP
      X1(L5,J)=X(J)
      IN=L5
      CALL SUMR(PROBLM)
   56 IF (SUM(L9).LT.SUM(K1)) GO TO 54
      IF (SUM(L5).LT.SUM(K1)) GO TO 55
      GO TO 97
   54 X1(L5,J)=X1(K1,J)
      SUM(L5)=SUM(K1)
      X1(K1,J)=X1(L9,J)
      SUM(K1)=SUM(L9)
      FACTOR=FACTOR+1.0
      X(J)=X1(K1,J)+FACTOR*STEP
      IN=L9
      CALL SUMR(PROBLM)
      GO TO 56
   55 X1(L9,J)=X1(K1,J)
      SUM(L9)=SUM(K1)
      X1(K1,J)=X1(L5,J)
      SUM(K1)=SUM(L5)
      FACTOR=FACTOR+1.0
      X(J)=X1(K1,J)-FACTOR*STEP
      IN=L5
      CALL SUMR(PROBLM)
      GO TO 56
C
C     ONE DIMENSIONAL SEARCH BY GOLDEN SECTION ALONG EACH COORDINATE
C
   97 H(J)=X1(L9,J)-X1(L5,J)
      X1(L6,J)=X1(L5,J)+H(J)*R1A
      X(J)=X1(L6,J)
      IN=L6
      CALL SUMR(PROBLM)
      X1(L7,J)=X1(L5,J)+H(J)*R2A
      X(J)=X1(L7,J)
```

```
      IN=L7
      CALL SUMR(PROBLM)
      IF (SUM(L6).GT.SUM(L7)) GO TO 68
      X1(L8,J)=X1(L5,J)+(1.0-R3A)*H(J)
      X1(L5,J)=X1(L7,J)
      X(J)=X1(L8,J)
      IN=L8
      CALL SUMR(PROBLM)
      IF (SUM(L8).GT.SUM(L6)) GO TO 76
      X1(L5,J)=X1(L6,J)
      SUM(L5)=SUM(L6)
      GO TO 75
   76 X1(L9,J)=X1(L8,J)
      SUM(L9)=SUM(L8)
      GO TO 75
   68 X1(L9,J)=X1(L6,J)
      X1(L8,J)=X1(L5,J)+R3A*H(J)
      X(J)=X1(L8,J)
      IN=L8
      CALL SUMR(PROBLM)
      STEP=SIZE
      SUM(L9)=SUM(L6)
      IF (SUM(L7).GT.SUM(L8)) GO TO 71
      X1(L5,J)=X1(L8,J)
      SUM(L5)=SUM(L8)
      GO TO 75
   71 X1(L9,J)=X1(L7,J)
      SUM(L9)=SUM(L7)
   75 IF (ABS(X1(L9,J)-X1(L5,J)).GT.0.01*FDIFER) GO TO 97
      X1(K1,J)=X1(L7,J)
      X(J)=X1(L7,J)
      SUM(K1)=SUM(L5)
      SR(INF)=SQRT(SUM(K1))
      IF (SR(INF).LT.FDIFER) GO TO 760
   53 CONTINUE
      ICHEK=ICHEK+1
      STEP=FDIFER
      IF (ICHEK.LE.2) GO TO 25
 1988 FOLD=1.0E-10
      IF (NPR.GT.0) WRITE (NPR,853)
      IF (NPR.GT.0) WRITE (NPR,850)
      IF (NPR.GT.0) WRITE (NPR,851) (J,X(J),J=1,NX)
      IF (NPR.GT.0) WRITE (NPR,852) FDIFER,SR(INF)
      GO TO 46
  760 DO 761 J=1,NX
      X2(INF,J)=X1(K1,J)
  761 X(J)=X1(K1,J)
   50 IF (VOLD.NE.SR(INF)) GO TO 10050
      IF (KOLD.GE.10) GO TO 1988
      KOLD=KOLD+1
      GO TO 20050
10050 VOLD=SR(INF)
      KOLD=0
20050 IF (SR(INF).GT.FDIFER) GO TO 28
C
C     MODIFIED LAGRANGE INTERPOLATION FOR TIGHT INEQUALITIES
C
      IF (SR(INF).GT.0.0) GO TO 35
      CALL PROBLM(3)
      FINT=R(K9)
      DO 139 J=1,NX
  139 X(J)=X2(INF,J)
      CALL PROBLM(2)
      DO 40 J=K7,K8
   40 R1(J)=R(J)
      DO 41 J=1,NX
   41 X(J)=X1(KOUNT,J)
```

```
      CALL PROBLM(2)
      DO 42 J=K7,K8
   42 R3(J)=R(J)
      DO 43 J=1,NX
      H(J)=X1(KOUNT,J)-X2(INF,J)
   43 X(J)=X2(INF,J)+0.5*H(J)
      CALL PROBLM(2)
      FLG(1)=0.0
      FLG(2)=0.0
      FLG(3)=0.0
      DO 44 J=K7,K8
      IF (R3(J).GE.0.0) GO TO 44
      FLG(1)=FLG(1)+R1(J)*R1(J)
      FLG(2)=FLG(2)+R(J)*R(J)
      FLG(3)=FLG(3)+R3(J)*R3(J)
   44 CONTINUE
      SR(INF)=SQRT(FLG(1))
      IF (SR(INF).LT.FDIFER) GO TO 35
      ALFA1=FLG(1)-2.0*FLG(2)+FLG(3)
      BETA1=3.0*FLG(1)-4.0*FLG(2)+FLG(3)
      RATIO=BETA1/(4.0*ALFA1)
      DO 45 J=1,NX
   45 X(J)=X2(INF,J)+H(J)*RATIO
      IN=INF
      CALL SUMR(PROBLM)
      SR(INF)=SQRT(SEQL)
      IF (SR(INF).LT.FDIFER) GO TO 465
      DO 49 I=1,20
      DO 48 J=1,NX
   48 X(J)=X(J)-0.05*H(J)
      CALL SUMR(PROBLM)
      SR(INF)=SQRT(SEQL)
      IF (SR(INF).LT.FDIFER) GO TO 465
   49 CONTINUE
  465 CALL PROBLM(3)
      IF (FINT.GT.R(K9)) GO TO 46
      SR(INF)=0.0
      GO TO 35
   46 DO 47 J=1,NX
   47 X2(INF,J)=X(J)
   35 CONTINUE
      DO 335 J=1,NX
  335 X(J)=X2(INF,J)
      RETURN
  853 FORMAT (//' * * * * * * Неудача при поиске квазидопустимой точки *
     1 * * * * *')
  850 FORMAT (//' Невозможно удовлетворить ограничениям задачи; поиск пр
     1екращен'///' Пожалуйста, задайте другое начальное приближение и по
     2вторите решение задачи')
  851 FORMAT (//' Вектор независимых переменных, для которого нельзя удо
     1влетворить ограничениям:'//(4(' X(',I2,')=',1PE12.5,1X)))
  852 FORMAT (//' Скользящий допуск = ',1PE12.5///' Корень из суммы квад
     1ратов нарушенных ограничений = ',1PE12.5)
      END
      SUBROUTINE SUMR(PROBLM)
      EXTERNAL PROBLM
C
C     THIS SUBROUTINE COMPUTES THE SUM OF THE SQUARE VALUES OF THE
C     VIOLATED CONSTRAINTS IN ORDER TO BE COMPARED WITH THE TOLERANCE
C     CRITERION
C
      DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
     1 ROLD(100)
      COMMON /FLEXC1/NX,NC,NIC,STEP,ALFA,BETA,GAMA,IN,INF,FDIFER,SEQL,
     1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
      COMMON /FLEXC2/LFEAS,L5,L6,L7,L8,L9,R1A,R2A,R3A
      SUM(IN)=0.0
```

```
      CALL PROBLM(2)
      SEQL=0.0
      IF (NIC.EQ.0) GO TO 4
      DO 1 J=K7,K8
      IF (R(J).GE.0.0) GO TO 1
      SEQL=SEQL+R(J)*R(J)
    1 CONTINUE
    4 IF (NC.EQ.0) GO TO 3
      CALL PROBLM(1)
      DO 2 J=1,NC
    2 SEQL=SEQL+R(J)*R(J)
    3 SUM(IN)=SEQL
    5 RETURN
      END
      SUBROUTINE WRITEX(PROBLM,NPR)
      EXTERNAL PROBLM
      INTEGER NPR
      DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
     1 ROLD(100)
      COMMON /FLEXC1/NX,NC,NIC,STEP,ALFA,BETA,GAMA,IN,INF,FDIFER,SEQL,
     1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
      COMMON /FLEXC2/LFEAS,L5,L6,L7,L8,L9,R1A,R2A,R3A
      IF (NPR.GT.0) WRITE (NPR,1) (J,X(J),J=1,NX)
    1 FORMAT (//' Вектор независимых переменных:'//(4(' X(',I2,')=',
     1 1PE12.5,1X)))
      IF (NC.LE.0) GO TO 3
      CALL PROBLM(1)
      IF (NPR.GT.0) WRITE (NPR,2) (J,R(J),J=1,NC)
    2 FORMAT (//' Значения ограничений-равенств:'//(4(' E(',I2,')=',
     1 1PE12.5,1X)))
    3 IF (NIC.LE.0) GO TO 5
      CALL PROBLM(2)
      IF (NPR.GT.0) WRITE (NPR,4) (J,R(NC+J),J=1,NIC)
    4 FORMAT (//' Значения ограничений-неравенств:'//(4(' Z(',I2,')=',
     1 1PE12.5,1X)))
    5 CALL PROBLM(3)
      IF (NPR.GT.0) WRITE (NPR,6) R(NC+NIC+1)
    6 FORMAT (//' Значение целевой функции = ',1PE12.5)
      RETURN
      END
```

Пример использования подпрограммы *FLEX*

```
      PROGRAM DEMO
*
      DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
     1 ROLD(100)
      COMMON /FLEXC1/NX,NC,NIC,STEP,ALFA,BETA,GAMA,IN,INF,FDIFER,SEQL,
     1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
*
      EXTERNAL PROBLM
      INTEGER NPR,INFO
      REAL SIZE,CONVER
*
      NX=2
      NC=1
      NIC=1
      NPR=6
      SIZE=1.0
      CONVER=1E-6
      X(1)=1.0
      X(2)=1.0
*
      CALL FLEX(PROBLM,NPR,SIZE,CONVER,INFO)
*
      WRITE (NPR,10) INFO
```

```
      IF (INFO.EQ.0) WRITE (NPR,20)
      IF (INFO.EQ.1) WRITE (NPR,30)
      IF (INFO.EQ.2) WRITE (NPR,40)
      IF (INFO.EQ.3) WRITE (NPR,50)
      IF (INFO.EQ.4) WRITE (NPR,60)
*
      STOP
*
   10 FORMAT (' Решение задачи закончено с кодом INFO = ',I1,', который
     1означает, что')
   20 FORMAT (' получено окончательное решение, удовлетворяющее критерию
     1 скользящего допуска')
   30 FORMAT (' невозможно найти очередное квазидопустимое решение')
   40 FORMAT (' число независимых переменных превышает 50')
   50 FORMAT (' общее число ограничений превышает 100')
   60 FORMAT (' входные параметры SIZE или CONVER заданы некорректно')
*
      END
      SUBROUTINE PROBLM(INQ)
*
      INTEGER INQ
*
      DIMENSION X(50),X1(50,50),X2(50,50),R(100),SUM(50),F(50),SR(50),
     1 ROLD(100)
      COMMON /FLEXC1/NX,NC,NIC,STEP,ALFA,BETA,GAMA,IN,INF,FDIFER,SEQL,
     1 K1,K2,K3,K4,K5,K6,K7,K8,K9,X,X1,X2,R,SUM,F,SR,ROLD,SCALE,FOLD
*
      GO TO (10,20,30),INQ
*
*     ОГРАНИЧЕНИЯ-РАВЕНСТВА R(1),...,R(NC)
*
   10 R(1)=(2.0*X(1)-2.0*X(2))-1.0
      RETURN
*
*     ОГРАНИЧЕНИЯ-НЕРАВЕНСТВА R(NC+1),...,R(NC+NIC)
*
   20 R(NC+1)=5.0-(X(1)**2+X(2)**2)
      RETURN
*
*     ЦЕЛЕВАЯ ФУНКЦИЯ R(NC+NIC+1)
*
   30 R(NC+NIC+1)=X(1)**2+2.0*X(2)**2
      RETURN
*
      END
```

```
Начальный скользящий допуск =  4.00000E+00

Сумма нарушенных ограничений =  1.00000E+00

* * * * * * * * * * * * * * * * * * * * * * * *

Номер шага вычислений =      1

Скользящий допуск =  4.00000E+00

Вектор независимых переменных:

X( 1)= 1.00000E+00  X( 2)= 1.00000E+00  X(

Значения ограничений-равенств:

E( 1)=-1.00000E+00  E(

Значения ограничений-неравенств:

Z( 1)= 3.00000E+00  Z(
```

```
Значение целевой функции =  3.00000E+00
* * * * * * * * * * * * * * * * * * * * * * * *
Номер шага вычислений =     12
Скользящий допуск =  8.46788E-03
Вектор независимых переменных:
X( 1)= 3.13801E-01  X( 2)=-1.78877E-01  X(
Значения ограничений-равенств:
E( 1)=-1.46434E-02  E(
Значения ограничений-неравенств:
Z( 1)= 4.86953E+00  Z(
Значение целевой функции =  1.62465E-01
* * * * * * * * * * * * * * * * * * * * * * * *
Номер шага вычислений =     24
Скользящий допуск =  3.82381E-05
Вектор независимых переменных:
X( 1)= 3.18994E-01  X( 2)=-1.81001E-01  X(
Значения ограничений-равенств:
E( 1)=-9.29832E-06  E(
Значения ограничений-неравенств:
Z( 1)= 4.86548E+00  Z(
Значение целевой функции =  1.67280E-01
* * * * * * * * * * * * * * * * * * * * * * * *
```

(шаги с 25 по 83 не показаны)

```
* * * * * * * * * * * * * * * * * * * * * * * *
Номер шага вычислений =     84
Скользящий допуск =  1.38377E-06
Вектор независимых переменных:
X( 1)= 3.33389E-01  X( 2)=-1.66610E-01  X(
Значения ограничений-равенств:
E( 1)=-2.59280E-06  E(
Значения ограничений-неравенств:
Z( 1)= 4.86109E+00  Z(
Значение целевой функции =  1.66666E-01
Общее число шагов вычислений =     85
```

```
Достигнутая точность решения =  5.31637E-07

Вектор независимых переменных:

X( 1)= 3.33389E-01  X( 2)=-1.66611E-01  X(

Значения ограничений-равенств:

E( 1)=-1.13249E-06  E(

Значения ограничений-неравенств:

Z( 1)= 4.86109E+00  Z(

Значение целевой функции =  1.66666E-01

Решение задачи закончено с кодом INFO = 0, который означает, что
получено окончательное решение, удовлетворяющее критерию скользящего допуска
Stop - Program terminated.
```