БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет НиДО

Специальность ИиТП

Контрольная работа № 1

по дисциплине «Методы оптимизации»

Выполнил студент: Дегтярев А.А.

группа 393551

Зачетная книжка № 902021-26

Минск 2016

**Часть 1.1 Симплекс-метод:**

Решить задачу f(x)=cx → max симплекс-методом

Вариант 2

C(x) = (1 1 1 1 1)

A = 1 1 0 2 0

0 -1 1 0 2

1 0 -1 1 -2

B = (3 1 -1)

В правой части присутствуют отрицательные значения, умножим 3-ю строку на (-1). Получим

А= 1 1 0 2 0

0 -1 1 0 2

-1 0 1 -1 2

B = 3 1 1

Так как базисный план в условиях задачи не оговорен, выполним поиск начального базисного плана используя искусственные переменные. Имея 3 ограничения – введем 3 новых переменных x6 x7 x8

А= 1 1 0 2 0 1 0 0

0 -1 1 0 2 0 1 0

-1 0 1 -1 2 0 0 1

Искусственные переменные лягут в основу базисного плана для задачи:



Решим эту задачу симплекс методом используя базисный план:

x = 0 0 0 0 0 3 1 1 [6,7,8]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 1 | 1 0 0  0 1 0  0 0 1 | 1 0 0  0 1 0  0 0 1 | [-1,-1,-1] | [0,0,-2,-1,-4] | 4 | [2,0,-1] |  | 1  js  6 |

x = 0 0 0 3/2 0 0 1 5/2 [4,7,8]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 2 | 2 0 0  0 1 0  -1 0 1 | ½ 0 0  0 1 0  ½ 0 1 | [-0.5,-1,-1] | 0.5,0.5,-2,0,-4 | 5 | [0,2,2] |  | 2  js  7 |

x = 0 0 0 3/2 1/2 0 0 3/2 [4,5,8]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 3 | 2 0 0  0 2 0  -1 2 1 | ½ 0 0  0 ½ 0  ½ -1 1 | [-0.5,1,-1] | 0.5,-1.5,0,0,0 | 2 | 1/2,-1/2,3/2 |  | 3  js  8 |

x = 0 1 0 1 1 [2,4,5]

Первый этап завершен, убираем столбцы с искусственными элементами и используя новый базисный план преступим к решению основной задачи:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 1 | 1 2 0  -1 0 2  0-1 2 |  | 5/6  -1/6  2/3 | -5/6  0  -1/2  0  0 | 1 | -1/3  2/3  -1/6 |  | 2  js  4 |

x = 3/2 3/2 0 0 5/4 [1,2,5]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 1 |  |  | 5/4  1/4  1/4 | 0  0  -1/2  5/4  0 | 3 | -1/3  2/3  -1/6 |  | 3  js  5 |

x = 3/2 3/2 5/2 0 0 [1,2,3]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 1 |  |  | 3/2  1/2  1/2 | 0  0  0  3/2  1 |  |  |  |  |

Оценки положительные – полученный базис является оптимальным

F(x) -> 5.5

|  |
| --- |
| function directSM  xj = [1,3]  Jb = [2,4]  A = [3,1,1,0,1,0;1,-2,0,1,0,1]  C = [0,0,0,0,-1,-1]  Ab = A(:,Jb)  B = Ab^(-1)  Cb = C(:,Jb)  u = Cb\*B  d = u\*A-C  [dj0,j0] = min(d)  j0=2  z = B\*A(:,[j0])  s = 2  t = xj(s)/z(s)  nxj = (xj'-(t\*z))  end |

**Часть1.2.. Двойственный симплекс-метод**

Решить задачи двойственным симплекс-методом

2. c=(1 4 1 -1)

a=(3 1 1 0

1 -2 0 1)

b=(1 1)

x(опт)=(0 1 0 3)

Начального базисного двойственного плана нет, для решения задачи будем использовать двухфазный двойственный симплекс-метод

Будем считать что задача приведена в канонической форме, тогда двойственная задача примет вид:

Для начала осуществим поиск базисного плана для прямой задачи для этого введем 2 новых переменных x5 и x6



Решим эту задачу симплекс методом используя базисный план:

x = 0 0 0 0 1 1 [5,6]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 1 | 1 0  0 1 | 1 0  0 1 | -1,-1 | -4 1 -1 -1 0 0 | 4 | [0,1] | 1 | 2  js  6 |

x = 0 0 0 1 1 0 [4,5]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 2 | 0 1  1 0 | 0 1  1 0 | -1,0 | -3,-1,-1,0,0,1 | 2 | [-2,1] | 1 | 2  js  5 |

x = 0 1 0 3 0 0 [2,4]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  | s |
| 2 | 1 0  -2 1 | 1 0  2 1 | 0,0 | 0 0 0 0 1 1 |  |  |  |  |

x = 0 1 0 3 0 0 [2,4]

Y=2/3 -1 jb=[2,4]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | A | B |  |  |  |  |  |  |
| 2 | 1 0  -2 1 | 1 0  2 1 | 1/3,1/6 |  |  |  |  |  |

Все компоненты >0 а это означает, что найденный базисный план

x = 0 1 0 3 0 0 [2,4] является оптимальным планом, а вектор y=(2/3,-1) – оптимальным двойственным планом.

|  |
| --- |
| function dualSM  y = [90/8,-26/8]  Jb = [3,4]  A = [1,-1,3,-2;1,-5,11,-6]  C = [1,1,-2,-3]  b = [1;9]  e = [0,0]  Ab = A(:,Jb)  B = Ab^(-1)  XB = B\*b  [djs,js] = min(XB)  e(js) = 1;  dy = e\*B  mj = dy\*A  [dj0,j0] = min(mj)  aj0 = (A(:,j0))'  sig = (C(:,j0)-(aj0 \* y'))/mj(j0)  ny = y + sig\*dy  end |

**Часть1.3.. Транспортная задача:**

**Вариант 1:**

B1 B2 B3 B4

A1 15 20 16 21 250

A2 25 13 5 11 130

A3 15 15 7 17 235

75 230 240 70

Условия баланса соблюдаются, Запасы равны потребностям. Модель задачи закрытая.

Используя метод наименьше стоимости построим первый опорный план задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | 16 | 21 | 250 |
| 25 | 13 | 5 | 11 | 130 |
| 15 | 15 | 7 | 17 | 235 |
| 75 | 230 | 240 | 70 |  |

X23 = 130

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | 16 | 21 | 250 |
| x | x | 5 | x | 130-130=0 |
| 15 | 15 | 7 | 17 | 235 |
| 75 | 230 | 240-130=110 | 70 |  |

X33 = 110

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | x | 21 | 250 |
| x | x | 5 | x | 0 |
| 15 | 15 | 7 | 17 | 235-110=125 |
| 75 | 230 | 110-110=0 | 70 |  |

X11=75

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | x | 21 | 250-75=175 |
| x | x | 5 | x | 0 |
| x | 15 | 7 | 17 | 125 |
| 75-75=0 | 230 | 0 | 70 |  |

X32 =125

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | x | 21 | 175 |
| x | x | 5 | x | 0 |
| x | 15 | 7 | x | 125-125=0 |
| 0 | 230-125 = 105 | 0 | 70 |  |

X12=105

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | x | 21 | 175-105=70 |
| x | x | 5 | x | 0 |
| x | 15 | 7 | x | 0 |
| 0 | 105-105=0 | 0 | 70 |  |

X14=70

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | x | 21 | 70-70=0 |
| x | x | 5 | x | 0 |
| x | 15 | 7 | x | 0 |
| 0 | 0 | 0 | 70-70=0 |  |

В результате получен первый опорный план, который является допустимым, все грузы вывезены потребность удовлетворена. Опорный план является невырожденным.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15[75] | 20[105] | 16 | 21[70] | 250 |
| 25 | 13 | 5[130] | 11 | 130 |
| 15 | 15[125] | 7[110] | 17 | 235 |
| 75 | 230 | 240 | 70 |  |

Значение целевой функции

F(x)=15\*75+20\*105+21\*70+5\*130+15\*125+7\*110=7990

**Улучшение опорного плана**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | V1=15 | V2=20 | V3=12 | V4=21 |
| U1=0 | 15[75] | 20[105] | 16 | 21[70] |
| U2=-7 | 25 | 13 | 5[130] | 11 |
| U3=-5 | 15 | 15[125] | 7[110] | 17 |

Опорный план не оптимален так как есть оценки клеток ui+vk>cij

(2;4): -7 + 21 >11 d = -7+21-11=3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15[75] | 20[105][+] | 16 | 21[70][-] | 250 |
| 25 | 13 | 5[130][-] | 11[+] | 130 |
| 15 | 15[125][-] | 7[110][+] | 17 | 235 |
| 75 | 230 | 240 | 70 |  |

Выбираем наименьший груз в минусовых клетках приьавляем 70 в плюсовых и вычитаем в минусовых

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15[75] | 20[175] | 16 | 21 | 250 |
| 25 | 13 | 5[50] | 11[70] | 130 |
| 15 | 15[55] | 7[180] | 17 | 235 |
| 75 | 230 | 240 | 70 |  |

Проверим оптимальность нового опорного плана:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | V1=15 | V2=20 | V3=12 | V4=18 |
| U1=0 | 15[75] | 20[175] | 16 | 21 |
| U2=-7 | 25 | 13 | 5[50] | 11[70] |
| U3=-5 | 15 | 15[55] | 7[180] | 17 |

Данный опорный план является оптимальным так как все оценки свободных клеток удовлетворяют условию ui+vk<=cij

**F(x)=15\*75+20\*175+5\*60+11\*70+15\*55+7\*180 = 7780**

Часть2.1.. Квадратичное программирование

*1.* c = (-8 -6 -4 -6), b = (2 3)

А =(1 0 2 1

0 1 -1 2)

D = (2 1 1 0

1 1 0 0

1 0 1 0

0 0 0 0)

X = (2 3 0 0), J = (1 2)

х(опт) = [17/10 12/5 0 3/10]

|  |  |
| --- | --- |
| X,J | (2,3,0,0) [1,2] |
|  | (-1 -1 -2 -6) |
|  | (1,1) |
|  | (0,0,-1,3) |
|  | 3 |
|  | 2 1  1 1 |
|  | 1 0  0 1 |
|  | 2;-1 |
|  | 1;0 |
| l\* = (-2,1) | y = (2,1) |
|  | 2 |
| .., | 1,inf,1/2: 1/2 j\* = 3 |
| X,J | (1,7/2,1/2,0) [1,2,3] |
|  | (-2 -1.5 -2.5 -6) |
|  | (2,3/2) |
|  | (0,0,0,-1) |
|  | 4 |
|  | 2 1 1  1 1 0  1 0 1 |
|  | 1 0 2  0 1-1 |
|  | 1;2 |
|  | 0;0;0 |
| l\* = (3,-4,-2) | y = (0,1) |
|  | 2 |
| .., | 1/3,7/8,1/4: 1/2 j\* = 3 |
| X,J | (7/4,5/2,0,1/4) [1,2,4] |

Поскольку *j\* ∈ J\* \ Jоп;*

Переходим к шагу 4

Очередное решение сформированной системы

|  |  |
| --- | --- |
|  | 2 1  1 1 |
|  | 1 0  0 1 |
|  | 1;2 |
|  | 0;0; |
| l\* = (-1,-2) | y = (4,3) |
|  | 10 |
| .., | 7/4,5/4: 1/20 j\* = 4 |
| X,J | (17/10,12/5,0,3/10) [1,2,4] |
|  | (-2.2 -1.9 -2.3 -6) |
|  | (2.2,1.9) |
|  | (0,0,0.2,0) |

Так как все компоненты вектора неотрицательны то текущий план является оптимальным, задача решена.

|  |  |
| --- | --- |
| X,J | (17/10,12/5,0,3/10) [1,2,4] |

|  |
| --- |
| function quadSM  x = [17/10,12/5,0,1/4]  Jb = [1,2,3]  Jop = [1,2]  c = [-8,-6,-4,-6]  A = [1,0,2,1;0,1,-1,2]  Ax = A(:,Jb)    D = [2,1,1,0;1,1,0,0;1,0,1,0;0,0,0,0]    cx = c'+D\*x'  cxt = cx'  cop = cxt(:,Jop)  Aop = A(:,Jop)  Bop = Aop^-1    ux = -cop \* Bop  d = A'\*ux' + cx  [dj0,j0] = min(d)  j0 = 4    DS = D(Jb,Jb)  AS = A(Jop,Jb)  Aj0 = A(:,j0)  Dj0 = D(Jb,j0)  % system 3.8  BS = AS^-1;  l = -Aj0'\*BS  y = -(Dj0+DS\*l')'\*BS'  % l = [3,-4,-2]  % y = [0,1]    dg = Dj0' \* l' + Aj0' \* y' + D(j0,j0)  t1 = -x(Jb(1))/l(1)  t2 = -x(Jb(2))/l(2)  % t3 = -x(Jb(3))/l(3)  tj0 = abs(d(j0)/dg)  t0 = min(abs([t1,t2,tj0]))  nx = x(Jb) + t0\*l  end |

Часть 2.2 Задание

Задание

1. Решить по указанию препо­давателя методами проекции градиента и комплексного поиска задачи нелинейного программирования, приведенные в табл. 3.23. Результаты оптимизации вывести на печатающее устройство.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вариант | Целевая функция | Ограничения | Начальный вектор x[0] | Значение f (x[0]) | Точка минимума x\* | Значение f(x\*) |
| II |  | ;;;;; | 0,0,2 | 8 | 1,0,1 | 3 |

1. По результатам оптимизации для каждого метода построить графики зависимости значений функции log(f(x[k]) — f(x\*)) от ко­личества вычислений N минимизируемой функции.
2. Вычислить точностные оценки алгоритмов по критерию опти­мальности

ξ=| f(x[k]) — f(x\*)|

и по координатам

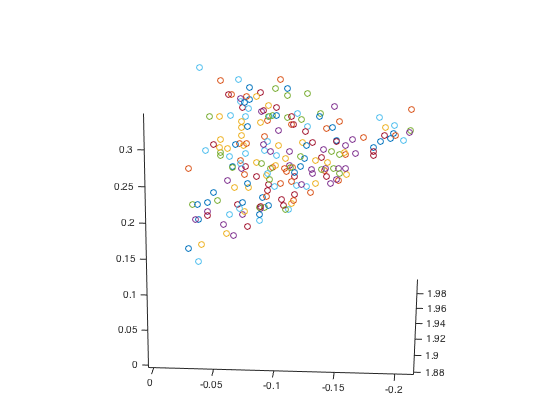
δ= (x[k] – x\*)T(x[k] – x\*).

Здесь f(x[k]) — значение целевой функции в точке x[k] после за­данного числа итераций k.

4. Провести сравнительный анализ алгоритмов по скорости сходимости, точности и времени поиска минимума.

**Отчет**

Ниже приведена область допустимых значений из 200 точек



Метод проекции градиента

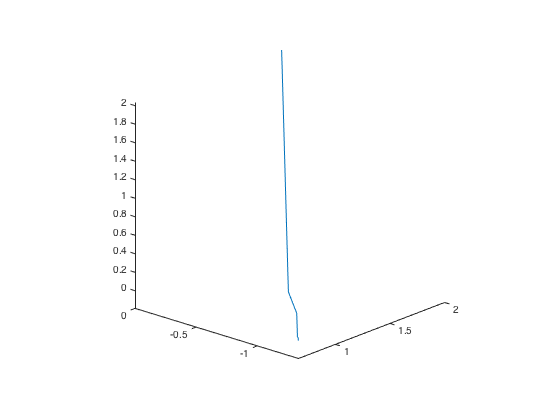
алгоритм метода проекции градиента состоит из следующих операций.

1. В точке х[k] определяется направление спуска р[k].

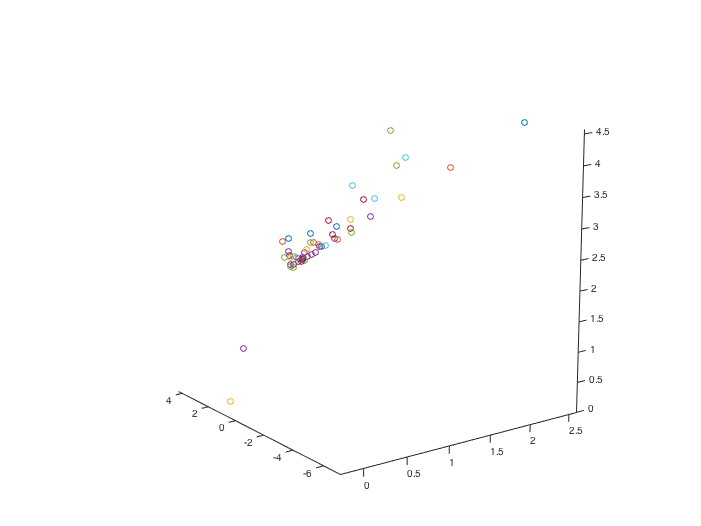
2. Находится величина шага аk.

3. Определяется новое приближение х[k+1].

Сходимость алгоритма выше чем метод Бокса, так для вычисления с точностью до 0.001 потребовалось 9 итераций



|  |
| --- |
| function res = gradProjMethod(f,x,xk,a,b,eps)  mag = eps\*2;  syms t0  resxk = [xk']  iter = 0;    while(mag > eps)  iter = iter + 1;  g = GradInPoint(f,x,xk);  %%searching for active limitations%%  act = a\*xk'-b  acti = [];  i = 0;  while i<length(act)  i = i + 1;  if(act(i)==0)  acti = [acti,i];  end  end    A = a(acti,[1,2,3])  if(rank(A)==0)  P = 1  else  P = eye(3)-A'\*((A\*A')^-1)\*A;  end    if(norm(P)<eps)  H = ((A\*A')^-1)\*A\*g' ;  [maxh,imaxh]=max(H)  acti(imaxh)=[];  A = a(acti,[1,2,3]);  P = eye(3)-A'\*((A\*A')^-1)\*A;  end  %%  S = P\*g';  af = subs(f,x,(xk - t0\*S'));  t = eps;  tval = double(subs(af,t0,eps));  for(nt = eps:eps:1)  ntval = double(subs(af,t0,nt));  if(ntval<tval)  t = nt;  tval = ntval;  end  end  prevxk = xk;  xk = xk - t\*S';  resxk = [resxk,xk']  mag = abs(norm(xk-prevxk));  end  res = struct('p',resxk,'i',iter)  end |

Сжатие комплекса методом Бокса к точке  


Код метода

|  |
| --- |
| function res = complexBoxMethod(f,x,x0,a,b,l,u,eps)  n = length(x)\*2;  poly = [x0'];  % creating random poly%  k = 1;  while (k<n)  p = l+rand(3,1).\*(u-l);  if(PointMeetsReq(a,b,p'))  poly = [poly,p];  k = k + 1;  else  % finding xc %  xp = sum(poly,2);  xc = xp/k;  %%%%%  attempt = 0;  while(~PointMeetsReq(a,b,p') && attempt < 100)  p = (p+xc)/2;  attempt = attempt + 1;  end  if(PointMeetsReq(a,b,p'))  poly = [poly,p];  k = k + 1;  end  end  end  scatter3(poly(1,:),poly(2,:),poly(3,:))  iter = 0  while true  iter = iter + 1  % finding xh %  ih = 0;  maxf = 0;  for i=1:1:k  fxp = double(subs(f,x,poly(:,i)'));  if(fxp>maxf)  maxf = fxp;  ih = i;  end  end  xh = poly(:,ih);  %%%%%    % finding xc %  xc = (sum(poly,2)-xh) / (k-1);  %%%%%  alpha = 1.3;  xr = xc-alpha\*(xc-xh);    while true  if(~PointMeetsReq(a,b,xr'))  for(i = 1:1:length(xr))  if(xr(i)<l)  xr(i)=l+eps;  end  if(xr(i)>u)  xr(i)=u-eps;  end  end    if(~PointMeetsReq(a,b,xr'))  xr = (xr + xc).\*0.5;  end  else  if(subs(f,x,xr')<subs(f,x,xh'))  poly(:,ih) = xr;  scatter3(poly(1,:),poly(2,:),poly(3,:))  fs = 0;    for i=1:1:length(poly)  fs = fs + double(subs(f,x,poly(:,i)'));  fs = fs - double(subs(f,x,xc'));  end    sig = (fs\*fs / (k-1))^ (1/2);  if(sig < eps)  res = poly;  return  else  break  end  else  xr = (xr + xc).\*0.5;  end  end  end  end  end |

Стоит отметить что метод является затратным в плане вычисления значений функции в точке, который необходимо делать для каждой точки комплекса, на каждой итерации метода. Так для точности 0.01 пришлось выполнить 130 итераций. Также стоит отметить, что в случае наличия точек в какой-то сильно малой допустимой области может привести к ошибкам вычислений.

Часть 2.3

##### Задание.

1. Построить линии уровня функций, приведенных в табл.
2. Минимизировать методами наискорейшего спуска и Флетчера – Ривса функции, приведенные в табл.
3. Результаты минимизации вывести в отчет.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вари-ант | Функция f(x) | Начальный вектор x[0] | Точка мини-мума | Зна-чение |
| VI |  | , | , |  |

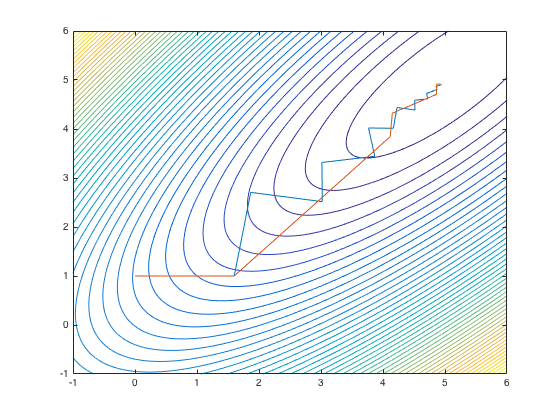
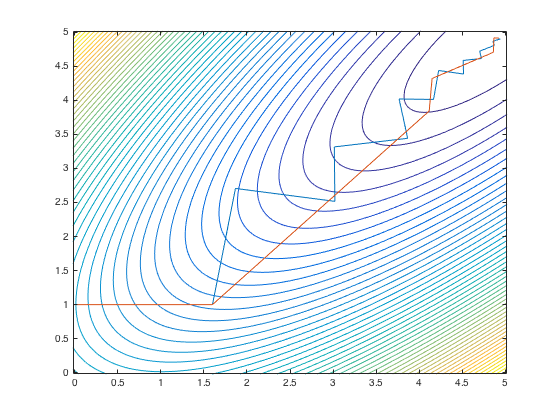
1. Построить траектории спуска по данным, полученным в результате выполнения п. 2.
2. Провести сравнительный анализ методов.

**Отчет**

Ниже приведен результат оптимизации с использованием метода наискорейшего спуска, и метода сопряженных градиентов Флетчера – Ривса:

Красный график – спуск по методу Ривса, (6 итераций e = 0.1, 11, e = 0.001)

Синий – метод наискорейшего спуска. (15 итераций e = 0.1, 39, e = 0.001)  
Как видно из графика и результатов – метод Флетчера – Ривса сходится быстрее.

****

|  |
| --- |
| function lab23  syms x1 x2  x = [x1,x2]  f = (x1-x2)^2+((x1+x2-10)/3)^2  x0 = [0,1]  eps = 0.001  %%Drawing levels of function  sx1 = -0.02:eps:5.02;  sx2 = -0.02:eps:5.02;  [X1,X2] = meshgrid(sx1,sx2);  Z = (X1-X2).^2+1/9\*(X1+X2-10).^2;  contour(X1,X2,Z,50)    hold on  res1 = fastestSlope(f,x,x0,eps)  plot(res1.p(1,:),res1.p(2,:))    res2 = fletcherReaves(f,x,x0,eps)  plot(res2.p(1,:),res2.p(2,:))  hold off  disp(res1.i)  disp(res2.i)  end      function grd = GradInPoint(f,x,x0)  grd = [];  for xn = x  dfxn = diff(f,xn);  grd = [grd,double(subs(dfxn,x,x0))]  end  return;  end      function res = fastestSlope(f,x,xk,eps)  mag = eps\*2;  syms t0  resxk = [xk']  iter = 0  while(mag > eps)  g = GradInPoint(f,x,xk)  af = subs(f,x,(xk - t0\*g));  t = eps;  tval = double(subs(af,t0,eps));  for(nt = eps:eps:1)  ntval = double(subs(af,t0,nt));  if(ntval<tval)  t = nt;  tval = ntval;  end  end  disp(t)  xk = xk - t\*g  resxk = [resxk,xk']  mag = norm(g);  iter = iter + 1;  end  res = struct('p',resxk,'i',iter)  end    function res = fletcherReaves(f,x,xk,eps)  mag = eps\*2;  syms t0  resxk = [xk']  gn = GradInPoint(f,x,xk)  iter = 0  while(mag > eps)  g = gn;  af = subs(f,x,(xk - t0\*g));  t = eps;  tval = double(subs(af,t0,eps));  for(nt = eps:eps:1)  ntval = double(subs(af,t0,nt));  if(ntval<tval)  t = nt;  tval = ntval;  end  end  xk = xk - t\*g  gn = GradInPoint(f,x,xk)  w = norm(gn)^2/norm(g)^2  gn = gn + w \* g  resxk = [resxk,xk']  mag = norm(gn);  iter = iter + 1;  end    res = struct('p',resxk,'i',iter)  end |