**Dejong’s function**

clc

clf

x = -3:.1:3; y = -3:.1:3;

[X,Y] = meshgrid(x,y);

Z = X.^2+ Y.^2;

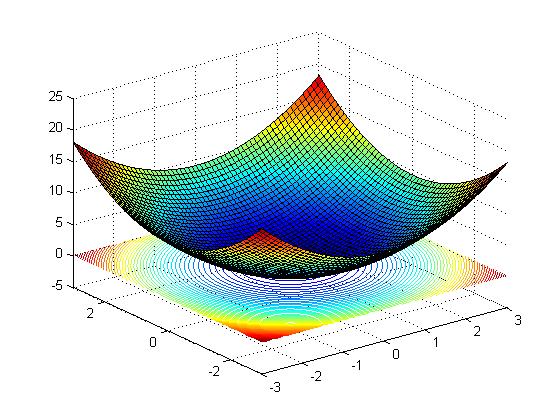
colormap(jet);

surf(X,Y,Z);

axis([-3 3 -3 3 -5 25]);

hold on

contour(X,Y, Z+1, 50)



**Rosnbrock function**

clc

clf

x = -3:.1:3; y = -3:.1:3;

[X,Y] = meshgrid(x,y);

Z = 100\*((Y-X.^2).^2) + ((1-X).^2);

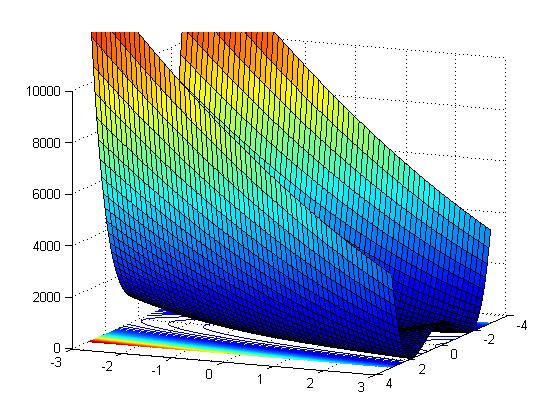
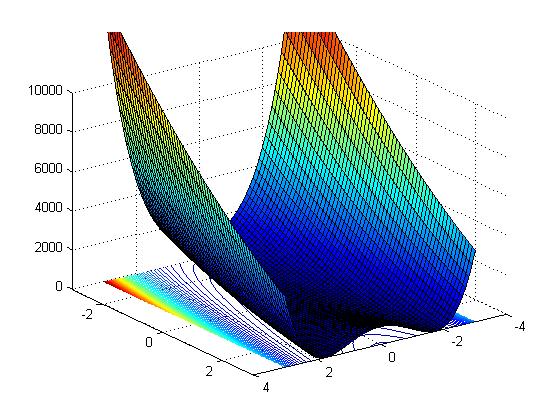
colormap(jet);

surf(X,Y,Z);

axis([-4 4 -3 3 -5 10000]);

hold on

contour(X,Y, Z+5, 50)



**Rastrigin function**

clc

clf

x = -3:.1:3; y = -3:.1:3;

[X,Y] = meshgrid(x,y);

Z = 10 + X.^2-10\*cos(2\*pi\*X) + Y.^2-10\*cos(2\*pi\*Y)

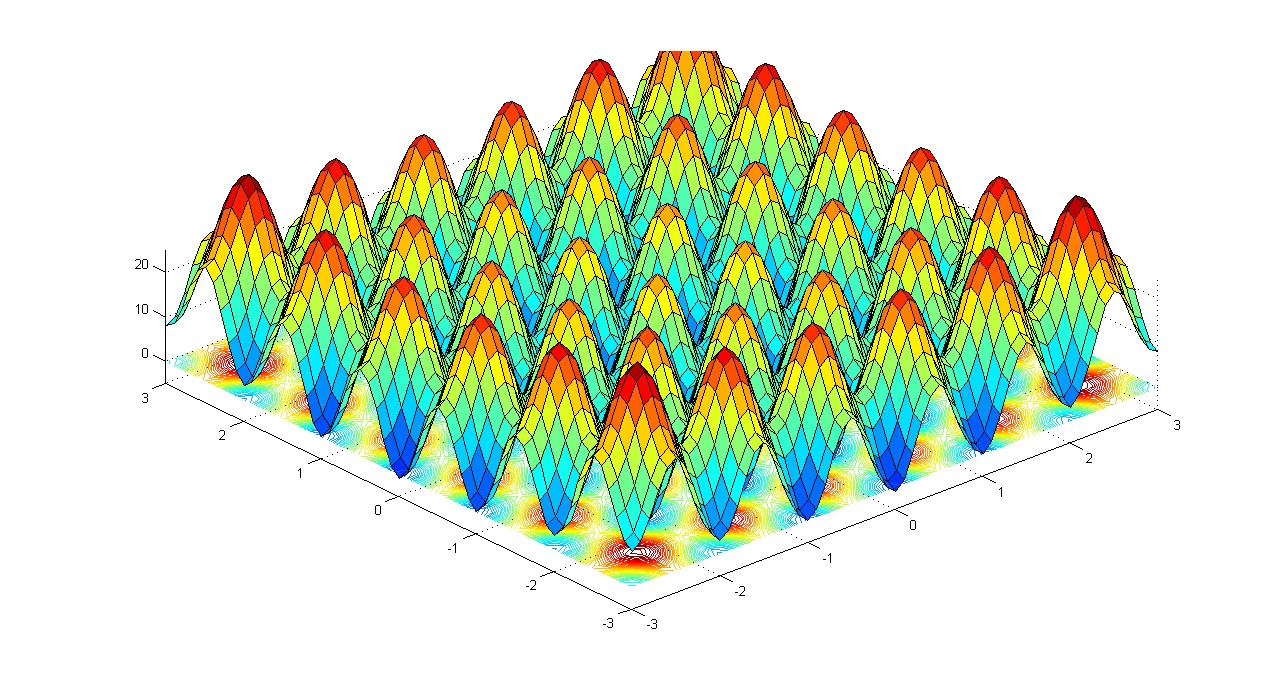
colormap(jet);

surf(X,Y,Z);

axis([-3 3 -3 3 -5 25]);

hold on

contour(X,Y, Z+1, 50)



**Newton's method**

!Newton's method

!ALGO: THE UPDATE SHOULD BE DONE IN THE FOLLOWING MANNER

! Xn+1= Xn + (Hn^-1)\*Gn where Hn^-1 is inverse of hessian matrix,and Gn is gradient

!FUNCTIONS/SUBROUTINE LIST:

! F; FX; FY; FXX; FYY; FXY; FYX; GRADIENT; HESSIAN,STEP

PROGRAM NEWTON\_METHOD

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,F0,F1,A(2,1),ALPHA(2,1),EPS

REAL (SELECTED\_REAL\_KIND(15,307)), PARAMETER :: h=10E-6

PRINT \*, 'ENTER THE INTIAL GUESS FOR COORDINATE X AND Y: '

READ\*, X,Y

A(1,1)=X

A(2,1)=Y

F0=F(X,Y)

EPS=EPSILON(1.)

PRINT \*, 'EPSILON NO IS=' , EPS

OPEN(10, FILE="DATA5\_11.DAT")

DO

PRINT \*, "X,Y ARE : ", A(1,1), A(2,1)

PRINT \*, "Value of F0 : ",F0

WRITE (10,\*) A, F0

CALL STEP(ALPHA, A(1,1), A(2,1))

A=A-ALPHA

F1=F(A(1,1),A(2,1))

IF (ABS(F0-F1)<EPS) EXIT

F0=F1

END DO

PRINT \*, 'THE FINAL SOLUTION IS : ', A

PRINT \*, "final Value of F : ",F1

WRITE(10,\*) A, F0

CLOSE(10)

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONTAINS

!THE ROSENBROCK-FUNCTION IS DEFINED HERE

FUNCTION F(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,F,PI

F=X\*\*2 + Y\*\*2

!F=100\*((Y-X\*\*2)\*\*2) + ((1-X)\*\*2)

!PI=4\*ATAN(1.)

!F=10+(X\*\*2-10\*COS(2\*PI\*X)) + (Y\*\*2 + 10\*COS(2\*PI\*Y))

END FUNCTION

!CALCULATING THE FIRST DERIVATIVE OF F WRT X

FUNCTION FX(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FX

FX= (F(X+h,Y)- F(X-h,Y)) / (2\*h)

END FUNCTION FX

!CALCULATING THE FIRST DERIVATIVE OF F WRT Y

FUNCTION FY(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FY

FY= (F(X,Y+h)- F(X,Y-h)) / (2\*h)

END FUNCTION FY

!CALCULATING THE SECOND DERIVATIVE OF FX WRT X

FUNCTION FXX(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FXX

FXX= (FX(X+h,Y)- FX(X-h,Y)) / (2\*h)

END FUNCTION FXX

!CALCULATING THE SECOND DERIVATIVE OF FY WRT Y

FUNCTION FYY(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FYY

FYY= (FY(X,Y+h)- FY(X,Y-h)) / (2\*h)

END FUNCTION FYY

!CALCULATING THE SECOND DERIVATIVE OF FY WRT X

FUNCTION FYX(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FYX

FYX= (FY(X+h,Y)- FY(X-h,Y)) / (2\*h)

END FUNCTION FYX

!CALCULATING THE SECOND DERIVATIVE OF FX WRT Y

FUNCTION FXY(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FXY

FXY= (FX(X,Y+h)- FX(X,Y-h)) / (2\*h)

END FUNCTION FXY

!FUNC CALCULATING STEP SIZE

SUBROUTINE STEP(ALPHA,X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,H(2,2),G(2,1),det,inv\_H(2,2),ALPHA(2,1)

CALL GRADIENT(G,X,Y)

CALL HESSIAN(H,X,Y)

!PRINT \*, 'HESSIAN=',H

det=H(1,1)\*H(2,2)-H(2,1)\*H(1,2)

inv\_H(1,1)= H(2,2)/det

inv\_H(1,2)=-H(1,2)/det

inv\_H(2,1)=-H(2,1)/det

inv\_H(2,2)=H(1,1)/det

ALPHA=MATMUL(inv\_H,G)

!PRINT \*,'ALPHA=',ALPHA

END SUBROUTINE STEP

!CALCULATE THE GRADIENT

SUBROUTINE GRADIENT(G,X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT(OUT) :: G(2,1)

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT (IN) :: X,Y

G(1,1)= FX(X,Y)

G(2,1)= FY(X,Y)

END SUBROUTINE GRADIENT

!CALCULATING THE HESSIAN METRIX

SUBROUTINE HESSIAN(H,X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT(OUT)::H(2,2)

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT(IN) :: X,Y

H(1,1)=FXX(X,Y)

H(1,2)=FXY(X,Y)

H(2,1)=FYX(X,Y)

H(2,2)=FYY(X,Y)

END SUBROUTINE HESSIAN

END PROGRAM NEWTON\_METHOD

**Steepest Descent**

!ALGO: THE UPDATE SHOULD BE DONE IN THE FOLLOWING MANNER

! Xn+1= Xn + ALPHA\*Pn where alpha is a constant, updated in each iteration

! Pn= -GRADIENT(FUNCTION(X))

! ALPHA= - Pn'\* Pn

! Pn'\*H\*Pn

!FUNCTIONS LIST:

! F; FX; FY; FXX; FYY; FXY; FYX; GRADIENT; HESSIAN,STEP

PROGRAM STEEPEST\_DESCENT

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,F0,F1,A(2,1),ALPHA,Pn(2,1),PI,EPS

REAL (SELECTED\_REAL\_KIND(15,307)), PARAMETER :: h=.00001

EPS=EPSILON(1.)

PI=4\*ATAN(1.)

PRINT \*, 'ENTER THE INTIAL GUESS FOR COORDINATE X AND Y: '

READ\*, X,Y

A(1,1)=X

A(2,1)=Y

F0=F(X,Y)

OPEN(10, FILE="DATA5\_21.DAT")

DO

PRINT\*, A,F0

WRITE (10,\*) A, F0

CALL GRADIENT(Pn,A(1,1),A(2,1))

ALPHA= MAGNITUDE(A(1,1),A(2,1))

A=A+ALPHA\*Pn

F1=F(A(1,1),A(2,1))

IF (ABS(F1-F0)<EPS) EXIT

F0=F1

END DO

PRINT \*, 'THE FINAL SOLUTION IS : ', A

WRITE (10,\*) A, F1

CONTAINS

!THE DE-JONG'S-FUNCTION IS DEFINED HERE

FUNCTION F(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,F

F=X\*\*2 + Y\*\*2

!F=100\*((y\*\*2-x\*\*2)\*\*2) + (1-x)\*\*2

!PI=4\*ATAN(1.)

!F=10+(X\*\*2-10\*COS(2\*PI\*X)) + (Y\*\*2 + 10\*COS(2\*PI\*Y)) ! CHECK METHOD FOR (34, 45)

END FUNCTION

!CALCULATING THE FIRST DERIVATIVE OF F WRT X

FUNCTION FX(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FX

FX= (F(X+h,Y)- F(X-h,Y)) / (2\*h)

END FUNCTION FX

!CALCULATING THE FIRST DERIVATIVE OF F WRT Y

FUNCTION FY(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FY

FY= (F(X,Y+h)- F(X,Y-h)) / (2\*h)

END FUNCTION FY

!CALCULATING THE SECOND DERIVATIVE OF FX WRT X

FUNCTION FXX(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FXX

FXX= (FX(X+h,Y)- FX(X-h,Y)) / (2\*h)

END FUNCTION FXX

!CALCULATING THE SECOND DERIVATIVE OF FY WRT Y

FUNCTION FYY(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FYY

FYY= (FY(X,Y+h)- FY(X,Y-h)) / (2\*h)

END FUNCTION FYY

!CALCULATING THE SECOND DERIVATIVE OF FY WRT X

FUNCTION FYX(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FYX

FYX= (FY(X+h,Y)- FY(X-h,Y)) / (2\*h)

END FUNCTION FYX

!CALCULATING THE SECOND DERIVATIVE OF FX WRT Y

FUNCTION FXY(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,FXY

FXY= (FX(X,Y+h)- FX(X,Y-h)) / (2\*h)

END FUNCTION FXY

!FUNC CALCULATING STEP SIZE

FUNCTION MAGNITUDE(X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)) :: X,Y,H(2,2),P(2,1),NR,DR,MAGNITUDE,Pt(1,2),temp(2,1)

CALL GRADIENT(P,X,Y)

Pt=TRANSPOSE(P)

CALL HESSIAN(H,X,Y)

Nr=MATMUL(Pt,P)

temp= MATMUL(H,P)

Dr=MATMUL(Pt,temp)

MAGNITUDE=-NR/DR

END FUNCTION MAGNITUDE

!CALCULATE THE GRADIENT

SUBROUTINE GRADIENT(G,X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT(OUT) :: G(2,1)

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT (IN) :: X,Y

G(1,1)= FX(X,Y)

G(2,1)= FY(X,Y)

END SUBROUTINE GRADIENT

!CALCULATING THE HESSIAN METRIX

SUBROUTINE HESSIAN(H,X,Y)

IMPLICIT NONE

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT(OUT)::H(2,2)

REAL (SELECTED\_REAL\_KIND(15,307)), INTENT(IN) :: X,Y

H(1,1)=FXX(X,Y)

H(1,2)=FXY(X,Y)

H(2,1)=FYX(X,Y)

H(2,2)=FYY(X,Y)

END SUBROUTINE HESSIAN

END PROGRAM STEEPEST\_DESCENT

**Downhill Simplex Method**

!FOLLOWING IS THE DOWNHILL SIMPLEX METHOD

!ALGO:

!1. START WITH THE SIMPLEX =DIM+1, AND FIND CENTROID OF ALL POINTS EXCLUDING WORST POINT

!2. TAKE THE REFLECTION(R) OF THE WORST POINT(S(3,:)) AND CHECK THE FUNCTION VALUE

! I. IF THE F(R) IS LESS THAN MIN

! THEN ASSIGN THE REFLECTED POIN AS NEW WORST POINT

! II. IF F(R) IS EVEN LESS THAN BEST POINT THEN EXPAND (E) IN THAT DIRECTION FURTHER

! A. IF E IS BETTER THAN R THEN ASSIGN THE E AS THE BEST POINT (S(1,:))

! B. IF R IS BETTER THAN E THEN ASSIGN THE R AS THE BEST POINT (S(1,:))

! III. UF THE F(R) IS GREATER THAN THE F(S(3,:)) THEN

! A. IF F(R)>F(S(2,:)) THEN CHECK FOR THE CONTRACTION(C).

! IF F(C) IS LESS THAN F(R) THEN ASSIGN C AS THE WORST POINT

! IF F(C) GREATER THAN F(R) THEN SHRINK THE SIMPLEX TOWARDS THE BEST POINT

!CONTAINED SUBROUTINES/FUNCTION ARE:

!SUB:DOWN\_HILL, SORT\_PTS,CENTROID; FUNC: F

PROGRAM SIMPLEX

REAL :: S(3,2),FTOL

PRINT \*, 'ENTER THE POINTS'

PRINT \*, 'ENTER THE POINT 1'

READ \*, S(1,1),S(1,2)

PRINT \*, 'ENTER THE POINT 2'

READ \*, S(2,1),S(2,2)

PRINT \*, 'ENTER THE POINTS 3'

READ \*, S(3,1),S(3,2)

!FTOL=EPSILON(1.)

FTOL=1.0E-5

CALL DOWN\_HILL(S,FTOL)

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONTAINS

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION F(P)

IMPLICIT NONE

REAL :: X,Y,F,P(2),PI

PI=4\*ATAN(1.)

X=P(1)

Y=P(2)

F= X\*\*2 + Y\*\*2

!F=100\*(Y-X\*\*2)\*\*2 + (1-X)\*\*2

!F=10+(X\*\*2-10\*COS(2\*PI\*X)) + (Y\*\*2 + 10\*COS(2\*PI\*Y))

END FUNCTION F

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE DOWN\_HILL(S,FTOL)

REAL, INTENT(INOUT) :: S(3,2), FTOL

REAL, DIMENSION(2) :: M,R, E, C

REAL :: FR, FE, FC, FUN(3), ALPHA=1., GAMMA=2., RHO=.5, SIGMA=.5, TINY=1.E-10

INTEGER :: I,J

OPEN(10, FILE="data31.DAT")

DO I=1,100

FUN(1)=F(S(1,:))

FUN(2)=F(S(2,:))

FUN(3)=F(S(3,:))

CALL SORT\_PTS(S,FUN) !STEP 1: SORT THE POINTS

WRITE (10,\*)((S(J,K),K=1,2),J=1,3), FUN(1)

RTOL= 2.0\*(ABS(FUN(3))-ABS(FUN(1)))/(ABS(FUN(1))+ABS(FUN(3)) + TINY )

IF (RTOL<FTOL) EXIT

PRINT \*, 'ITERATION NO=', I

PRINT\*, 'The SIMPLEX IS: '

DO J=1,3

PRINT \*, S(J,1),S(J,2)

END DO

CALL CENTROID(M,S) !2 CALCULATE THE CENTROID = M

PRINT \*, 'CENTROID',M

R= M + ALPHA\*(M-S(3,:)) !3 REFLECTION OF THE WORST POINT P3 OVER M ! RETURNS REFLECTED COORDINATE 'R'

FR=F(R)

IF (FR<FUN(1))THEN

E= R + GAMMA\*(R-M) ! RETURNS EXPANDED COORDINATE 'E'

FE=F(E)

IF(FE<F(R)) THEN

S(3,:)=E

FUN(3)=FE

ELSE

S(3,:)=R

FUN(3)=FR

END IF

ELSEIF(FR>FUN(1)) THEN

IF(FR<FUN(2))THEN !5A

S(3,:)=R

FUN(3)=FR

ELSEIF (FR>FUN(3)) THEN !5B

C= M + RHO\*(S(3,:)- M) !RETRURNS CONTRACTED POINT C

FC=F(C)

IF (FC<FUN(3))THEN

S(3,:)=C

ELSEIF(FC>FUN(3))THEN

S(2,:)=S(1,:)+SIGMA\*(S(2,:)-S(1,:)) ! SHRINK TOWARDS THE BEST SOLUTION CANDIDATE

S(3,:)=S(1,:)+SIGMA\*(S(3,:)-S(1,:))

END IF

ELSEIF (FR<FUN(3))THEN

S(3,:)=R

FUN(3)=FR

END IF

END IF

END DO

WRITE (10,\*)((S(J,K),K=1,2),J=1,3)

PRINT \*,'OBTAINED MIN POINT IS=', S(1,:)

CLOSE(10)

END SUBROUTINE DOWN\_HILL

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE SORT\_PTS(S,FUN)

REAL :: TEMP(2),FTEMP

REAL , INTENT(INOUT) :: S(3,2), FUN(3)

INTEGER :: I,J,N=3

DO I=1,N-1

DO J=1,N-1

IF (FUN(J)>FUN(J+1))THEN

TEMP=S(J,:)

S(J,:)=S(J+1,:)

S(J+1,:)=TEMP

FTEMP=FUN(J)

FUN(J)=FUN(J+1)

FUN(J+1)=FTEMP

END IF

END DO

END DO

END SUBROUTINE SORT\_PTS

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUBROUTINE CENTROID(C,S)

REAL, INTENT(OUT) :: C(2)

REAL, INTENT(IN) :: S(3,2)

C(1)=SUM(S(1:2,1))/2

C(2)=SUM(S(1:2,2))/2

END SUBROUTINE CENTROID

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

END PROGRAM SIMPLE

**RESULTS**

clc

clf

x = -100:.1:100; y = -100:.1:100;

[X,Y] = meshgrid(x,y);

Z = X.^2+ Y.^2;

hold on

contour(X,Y, Z+1, 50)

a=importdata('data.dat')

plot(A(:,1),A(:,2), '\*')

plot(A(:,1),A(:,2))

grid on

clc

clf

x = -20:.1:20; y = -20:.1:20;

[X,Y] = meshgrid(x,y);

Z = X.^2+ Y.^2;

hold on

contour(X,Y, Z+5, 500);

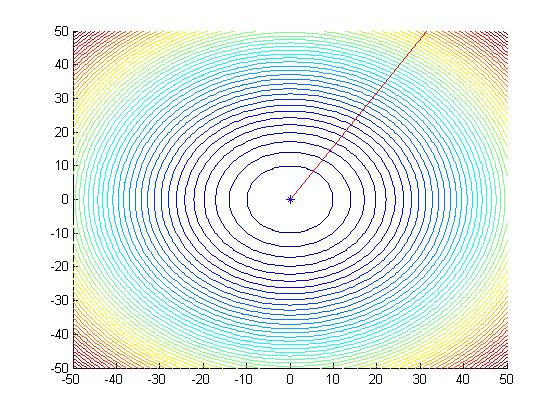
A=importdata('down-hill-simplex new-1.dat');

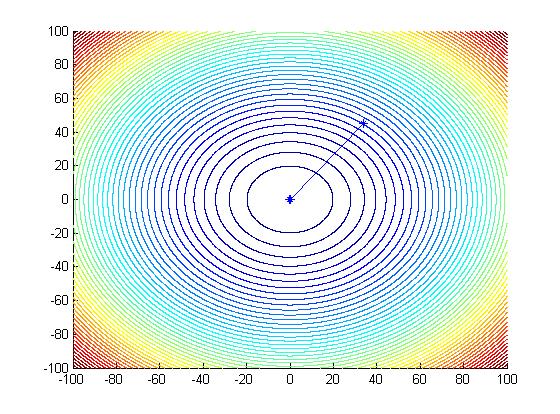
for i = 1:size(A,1)

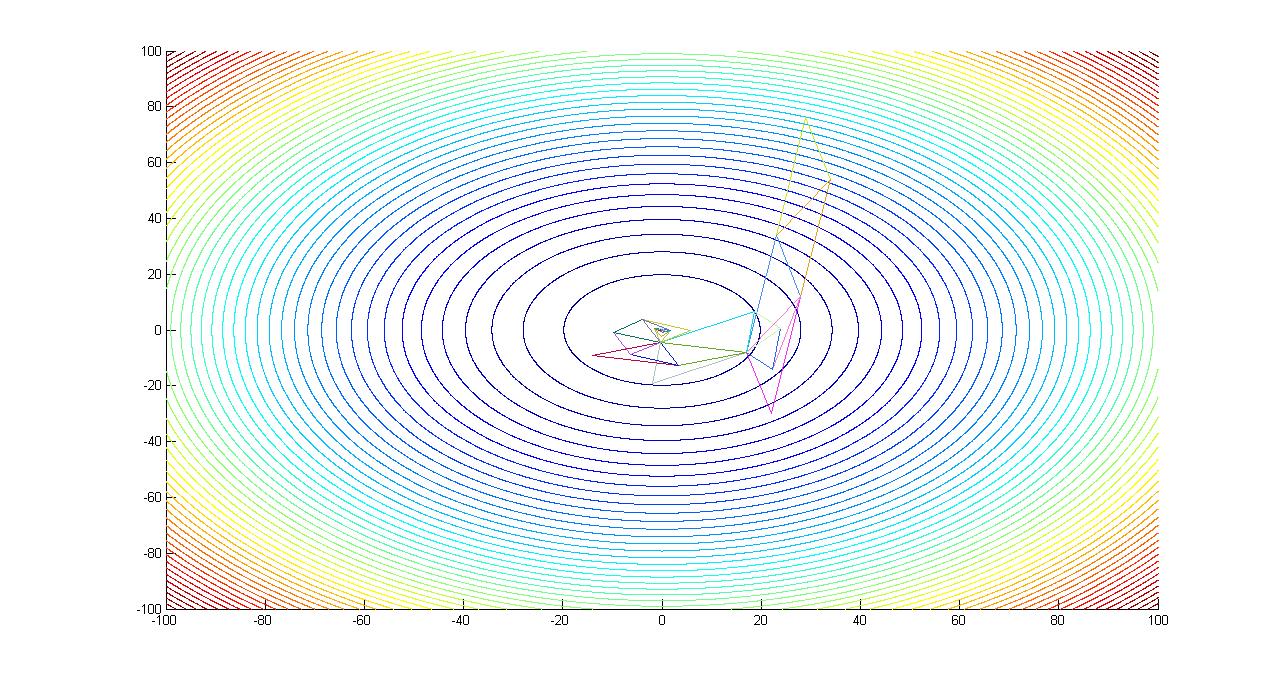
line(A(i,[1:2:5,1]), A(i,[2:2:6,2]), 'color',rand(1,3));

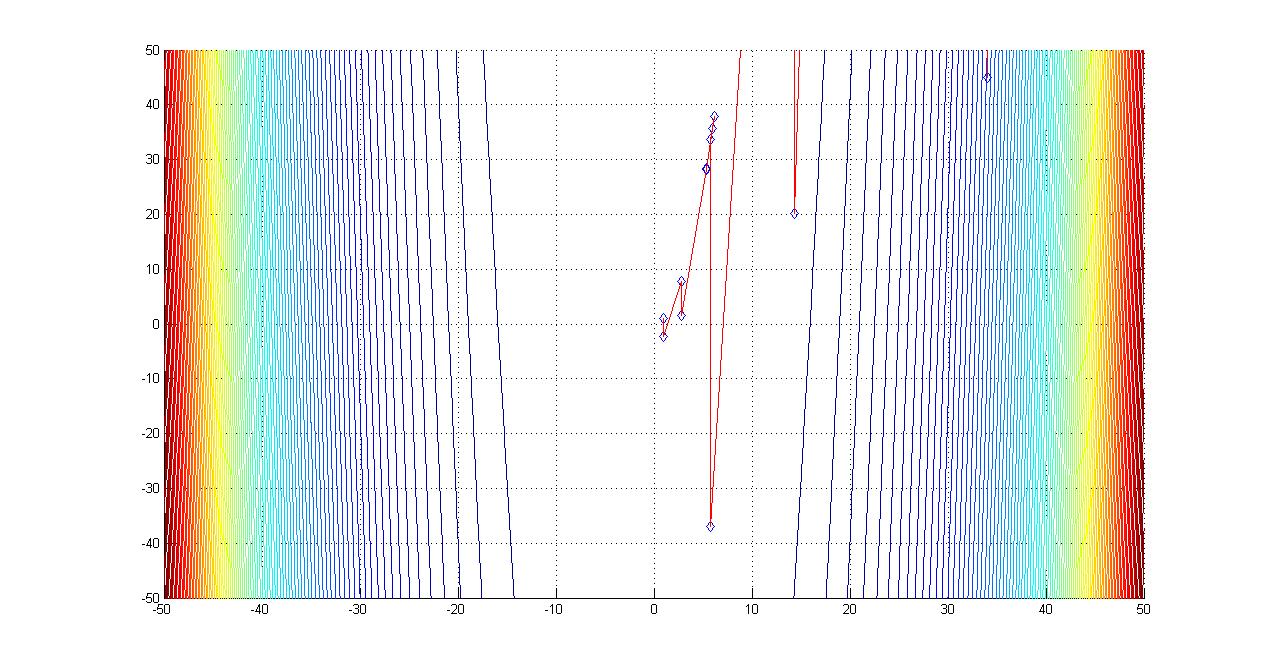
end

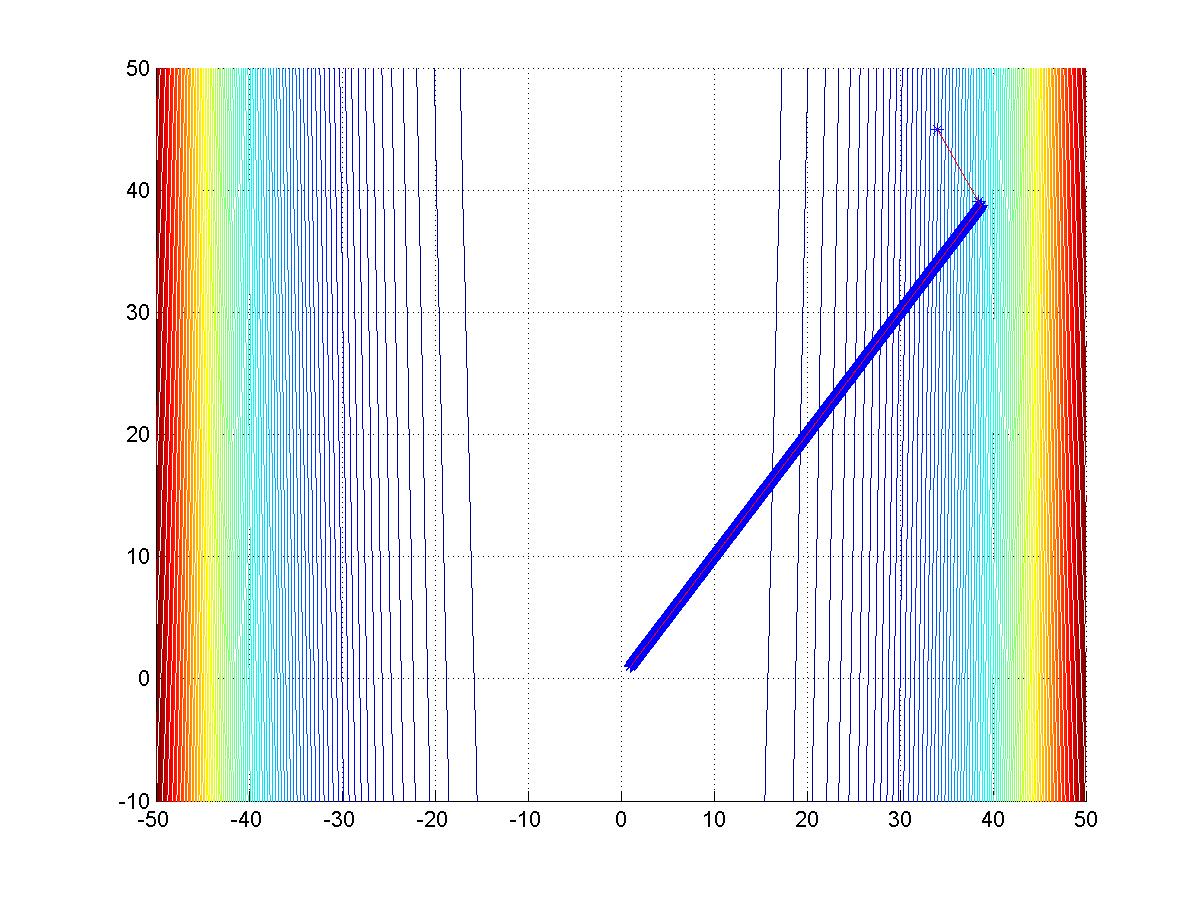
grid on

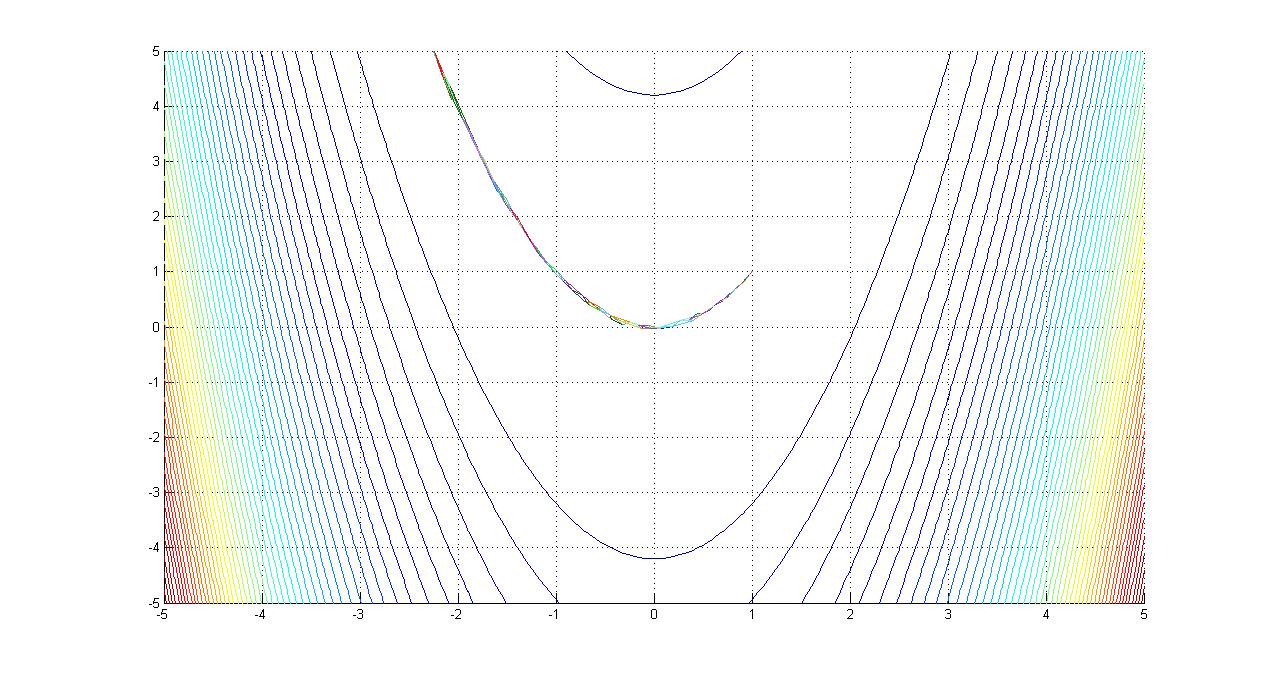


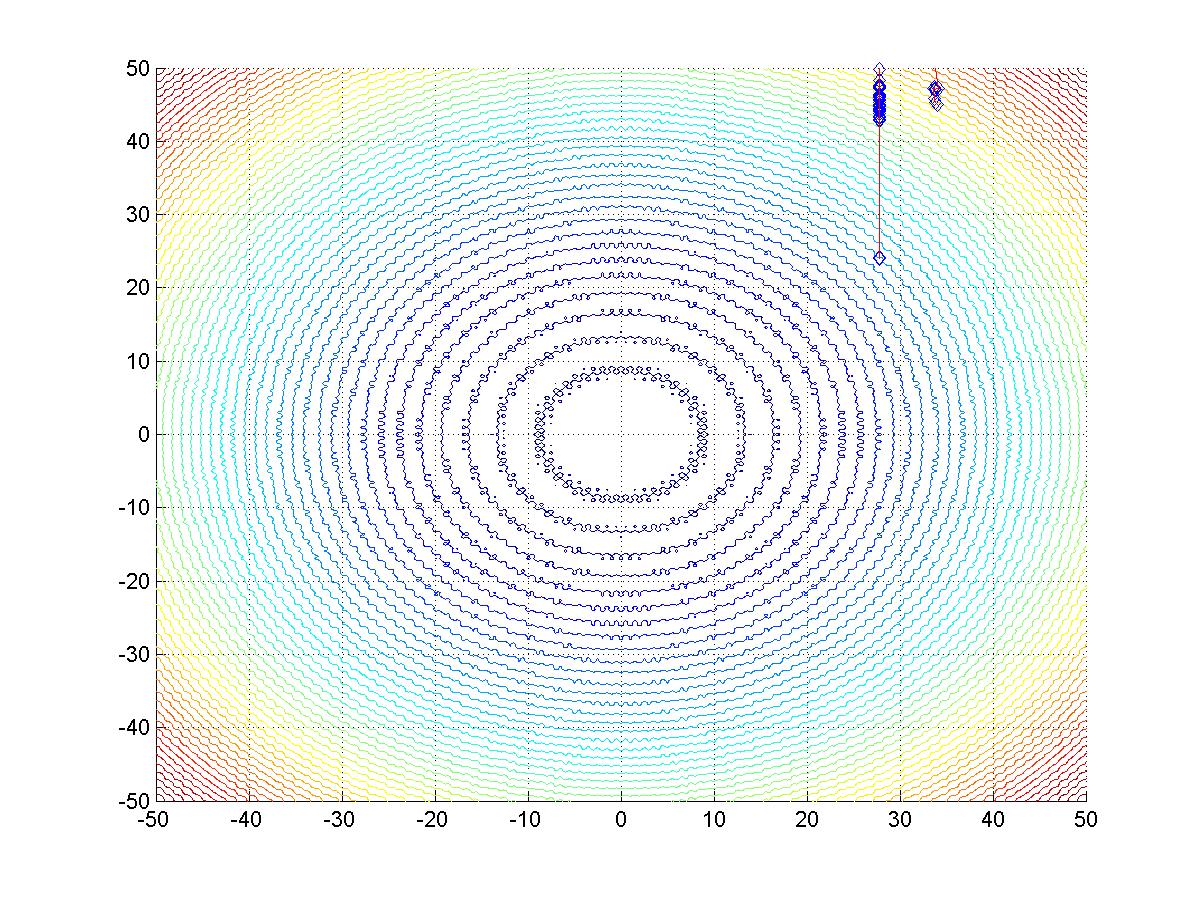


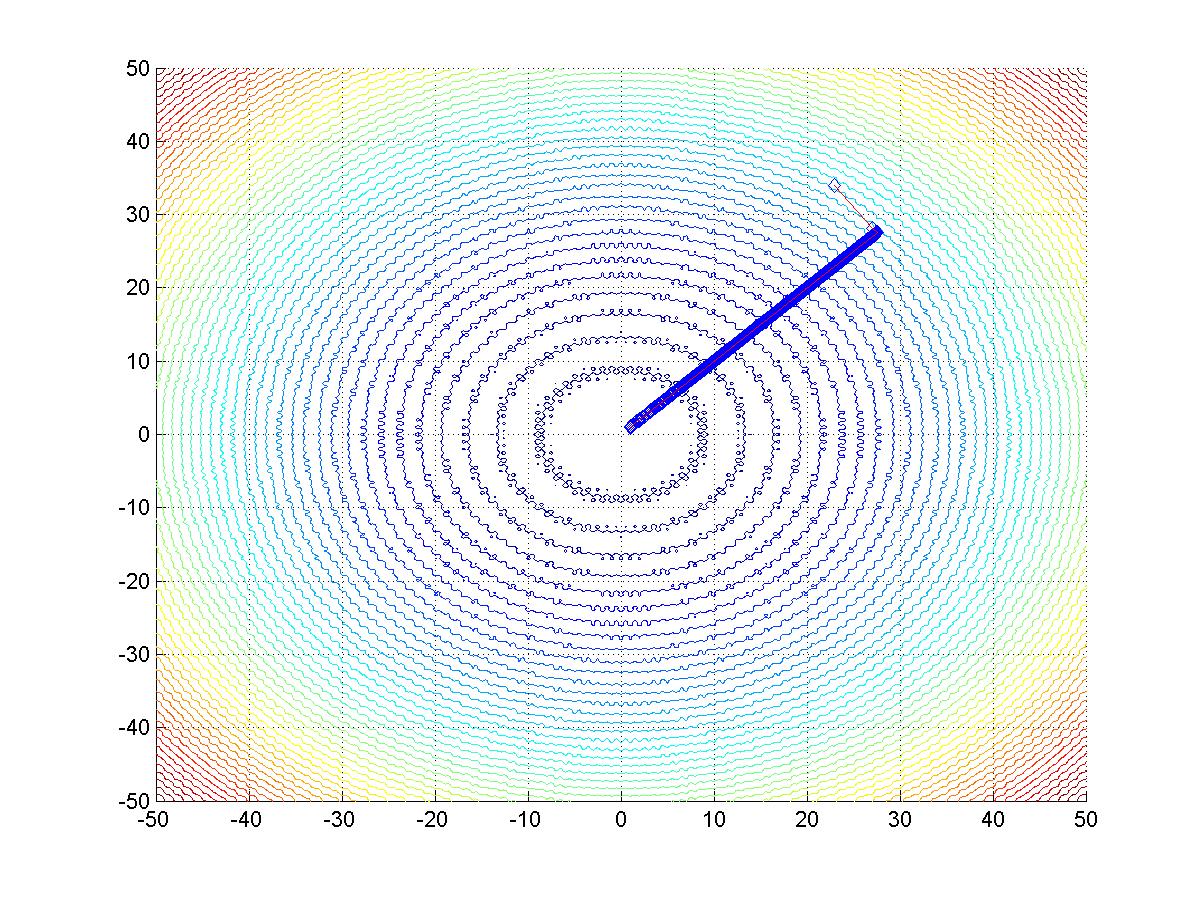


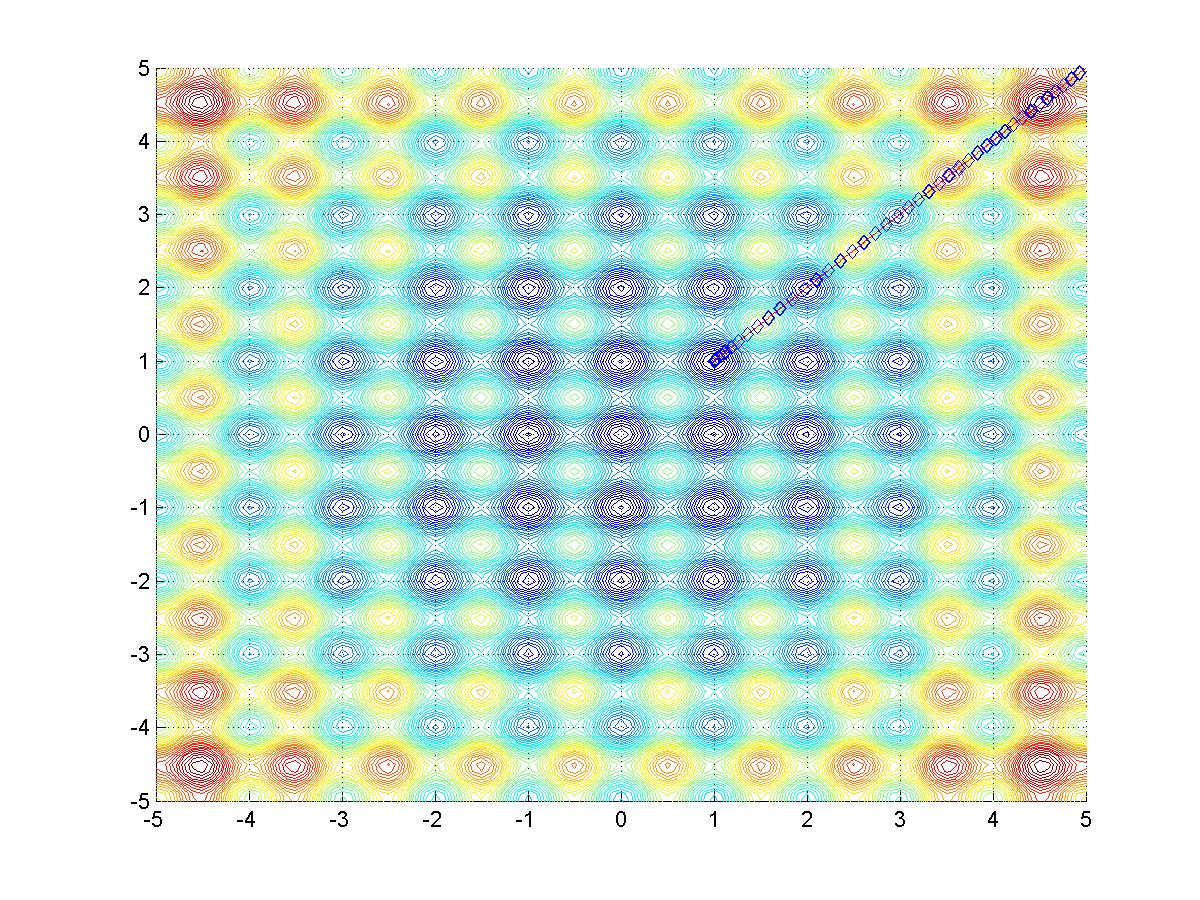


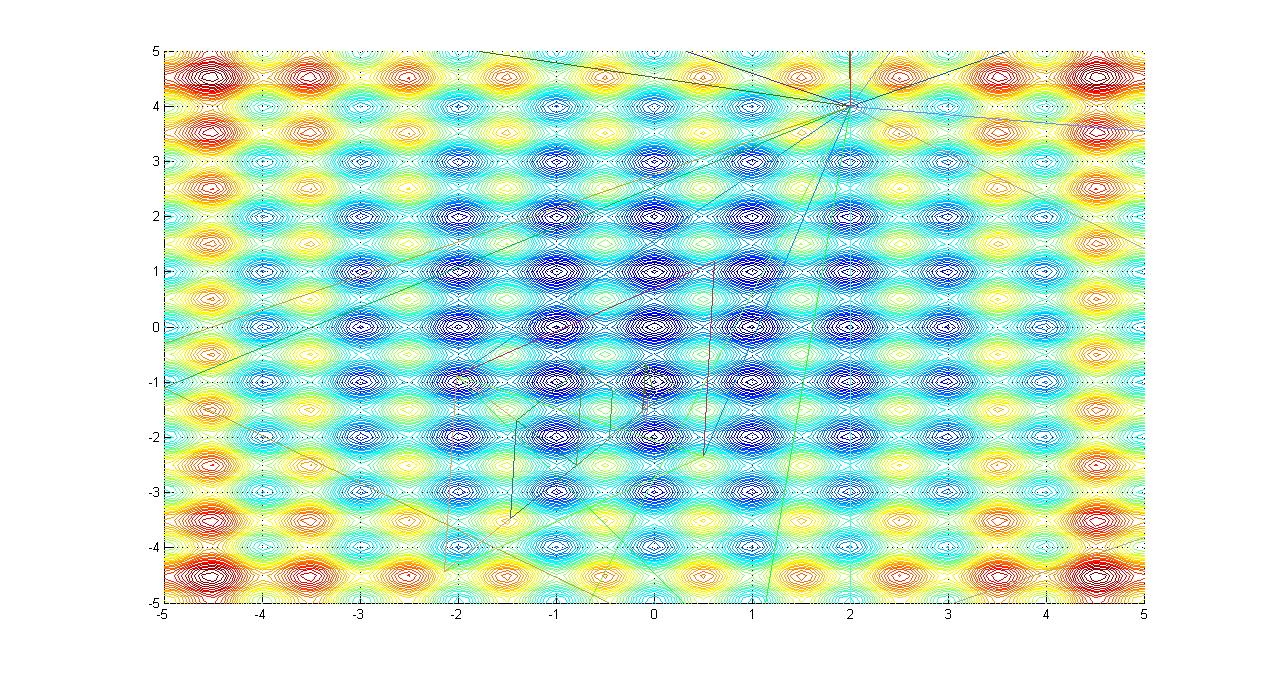












Iteration v/s function value

a=importdata('data5\_32.dat');

semilogy(a(:,7))

xlabel('No of iterations')

ylabel('function value')

title('Func val v/s Iteration')

a=importdata('data5\_33.dat');

plot(a(:,7))

xlabel('No of iterations')

ylabel('function value')

title('Func val v/s Iteration')

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

