Optimization Techniques
Section 3

M. Fatih Amasyalı

Approximating Derivatives

- In many instances, the finding f'(x) is difficult or impossible to encode. The Finite difference Newton method approximates the derivative:
- Forward difference

f'(x)≈(f(x+delta)-f(x))/ delta

- Backward difference

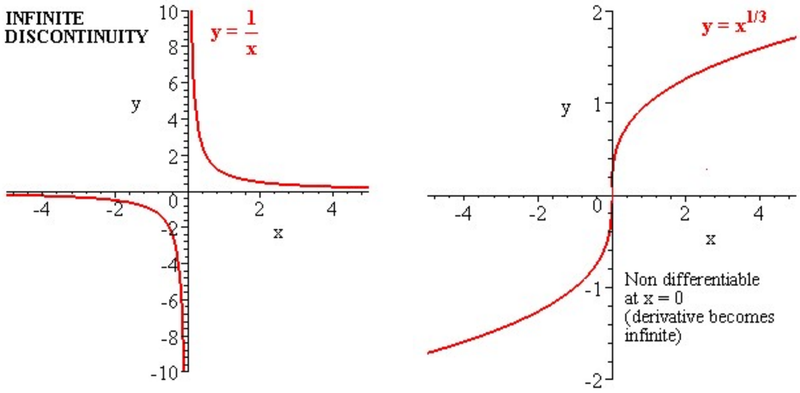
f'(x)≈(f(x)-f(x-delta))/ delta

- Central difference

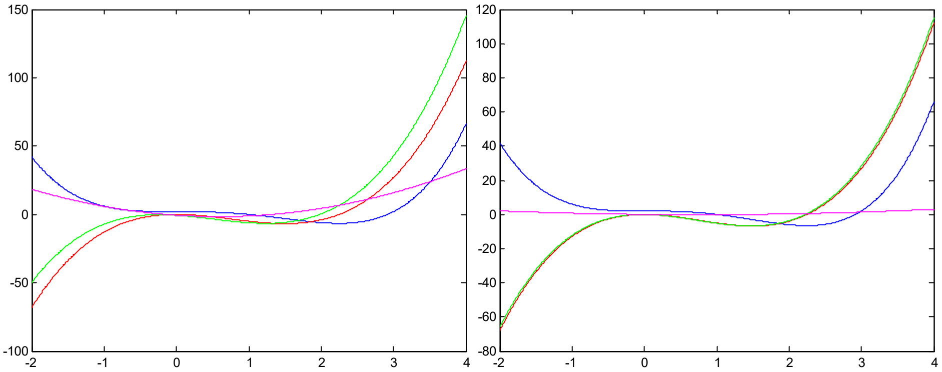
f'(x)≈(f(x+delta/2)-f(x-delta/2))/ delta

The choice of delta matters.

Approximating Derivatives



Forward difference Newton method



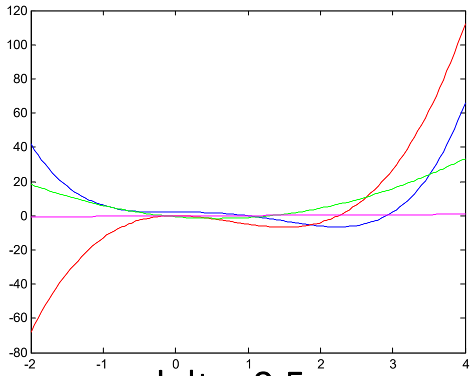
delta=0.5

delta=0.05

Blue: f(x)
Red: f'(x)
Green: approximated f'(x)
Magenta: (error) green-red

finite_difference_Newton.m

Forward difference vs. Central difference



delta=0.5

Blue: f(x)
Red: f'(x)
Green: (error) forward
Magenta: (error) central

finite_difference_Newton_2.m

Approximating higher order Derivatives

- According to the Central difference
- h=delta
- f'(x)=(f(x+h/2)-f(x-h/2))/h
- f''(x)=(f'(x+h/2)-f'(x-h/2))/h
- f'(x+h/2)=(f(x+h/2+h/2)-f(x+h/2-h/2))/h
- f'(x+h/2)=(f(x+h)-f(x))/h
- f'(x-h/2)=(f(x-h/2+h/2)-f(x-h/2-h/2))/h
- f'(x-h/2)=(f(x)-f(x-h))/h

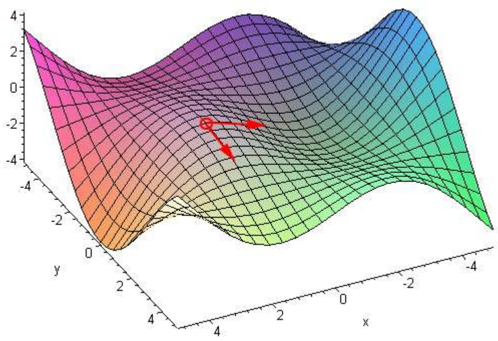
Approximating higher order Derivatives

- f''(x)=(f'(x+h/2)-f'(x-h/2))/h
- f'(x+h/2)=(f(x+h)-f(x))/h
- f'(x-h/2)=(f(x)-f(x-h))/h

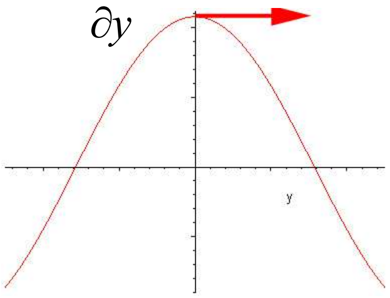
- f''(x)= (((f(x+h)-f(x))/h) - ((f(x)-f(x-h))/h)) /h
- f''(x)= (f(x+h)-2*f(x)+f(x-h)) / h^2

- See the approximating to the partial derivatives: http://en.wikipedia.org/wiki/Finite_difference

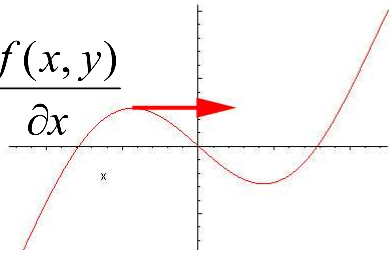
Two or more dimensions



$$\frac{\partial f(x,y)}{\partial y}$$



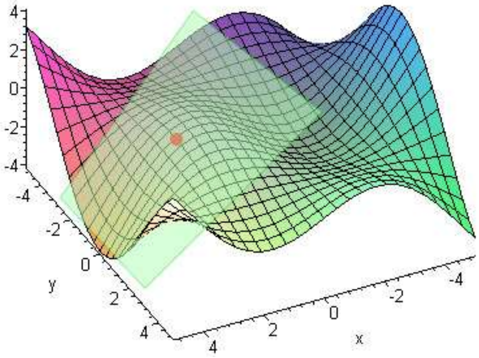
$$\frac{\partial f(x,y)}{\partial x}$$



- Definition: The gradient of f: $R^n \rightarrow R$ is a function ∇f: $R^n \rightarrow R^n$ given by

$$\nabla f(x_1,\ldots,x_n) := \left(\frac{\partial f}{\partial x_1},\ldots,\quad \frac{\partial f}{\partial x_n}\right)^T$$

- The gradient defines (hyper) plane approximating the function

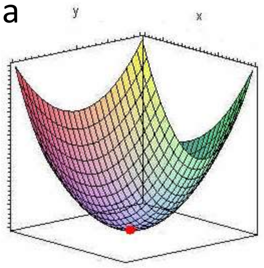


$$\Delta z = \frac{\partial f}{\partial x}\cdot \Delta x + \frac{\partial f}{\partial y}\cdot \Delta y$$

- Given the quadratic function

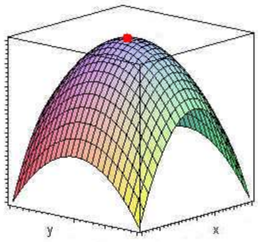
$$f(x)=(1/2)\, x^T q\, x + b^T x + c$$

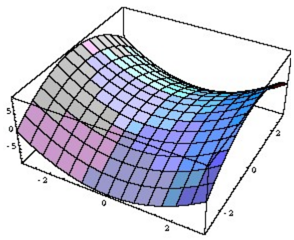
If *q* is positive definite, then *f* is a parabolic "bowl."



http://en.wikipedia.org/wiki/Positive-definite_matrix

- Two other shapes can result from the quadratic form.
  - If q is negative definite, then *f* is a parabolic "bowl" up side down.
  - If q is indefinite then *f* is a saddle.





quadratic_functions.m

```
% quadratic functions in n dimensions

% f(x)=(1/2) * xT * q * x + bT * x + c
%f: Rn--> R
%q--> n*n
%b--> n*1
%c--> 1*1

clear all;
close all;
% n=2
q=[1 0.5; 0.5 -2];
b=[1 ;1];
c=0.5;
```

```
x1=-5:0.5:5;
x2=x1;
z=zeros(length(x1),length(x1));
for i=1:length(x1)
  for j=1:length(x2)
    x=[x1(i); x2(j)];
    z(i,j)=(1/2)*x'*q*x+b'*x+c;
  end
end

surfc(x1,x2,z)
figure;
contour(x1,x2,z)
```


quadratic_functions.m

$f(x)=(1/2)\, x^T\, q\, x + b^T\, x + c$

- $x = [x_1\ x_2\ x_3\ \dots\ x_n]^T$
- $q = [x_1^2\ x_1x_2\ x_1x_3\ \dots\ x_1x_n$
  $x_2x_1\ x_2^2\ x_2x_3\ \dots\ x_2x_n$
  $\dots$
  $x_nx_1\ x_nx_2\ x_nx_3\ \dots\ x_n^2]$
  coefficients
- $b = [x_1\ x_2\ \dots\ x_n]^T$
  coefficients
- c = constant
- $f''(x) = q$

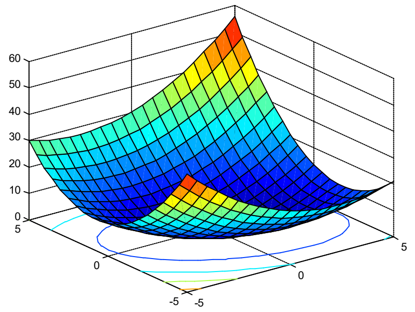
$q = [\,1\ 2\ ;2\ 1]$
$b = [1\ ;\ 3]$
$c = 2$

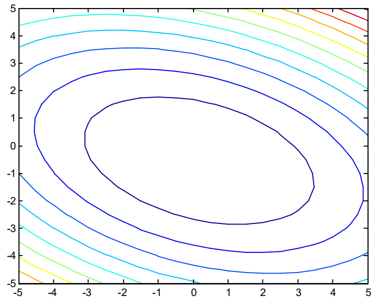
f(x)=?

$f(x)=(x_1^2+2x_1x_2+2x_2x_1+x_2^2)/2+x_1+3x_2+2$

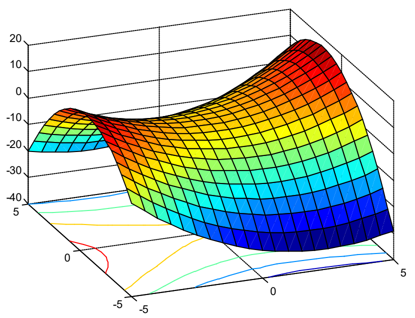
$f(x)=(x_1^2+4x_1x_2+x_2^2)/2+x_1+3x_2+2$

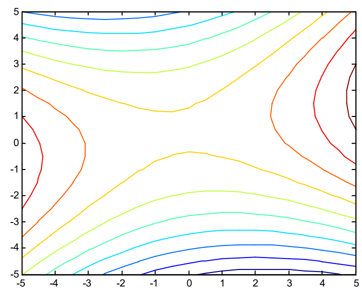
q=[1 0.5; 0.5 2]; b=[0.1 ;1]; c=0.5;





q=[1 0.5; 0.5 -2]; b=[0.1 ;1]; c=0.5;





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

- q,b,c ?
- (½)*q = [1 4; 0 3] or [1 3; 1 3]
  or **[1 2; 2 3] (symmetric)**
  q= [2 4; 4 6]
- b = [0; 3]
- c = 2

- *Hessian* of *f* : the second derivative of f

$$\boldsymbol{F} = D^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(\boldsymbol{x}) \\ \frac{\partial^2 f}{\partial x_1 \partial x_2}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_2^2}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_2}(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\boldsymbol{x}) \end{bmatrix}$$

f''(x) = q

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

```
syms x1;
syms x2;
syms expr;
% diff(expr,n,v) differentiate expr n times with respect to v .
expr=x1^2+3*x2^2+4*x1*x2+3*x2+2;
ddx=diff(expr,2,x1);
dx=diff(expr,1,x1);
dy=diff(expr,1,x2);
dxdy=diff(dx,1,x2);
ddy=diff(expr,2,x2);

q=[ddx dxdy; dxdy ddy]
```

```
q =

[ 2, 4]
[ 4, 6]
```


Quadratic functions in 2 dims.

$f(x)=(1/2)\, x^T q\, x + b^T x + c \qquad x = [x_1\ x_2]^T$

q=[1 0.5; 0.5 2];

b=[-0.5 ;-0.5];

c=0.5;

f(x)=(1/2) [x1 x2][1 0.5; 0.5 2] [x1;x2] + [-0.5 -0.5] [x1;x2]+0.5

f(x)=(1/2) [x1+0.5*x2 0.5*x1+2*x2] [x1;x2] – (0.5*x1+0.5*x2)+0.5

f(x)=(1/2)(x1^2+0.5*x1*x2+0.5*x1*x2+2*x2^2) -0.5*x1-0.5*x2+0.5

f(x)=(1/2) x1^2+x1*x2+2*x2^2) -0.5*x1-0.5*x2+0.5

f(x)=(x1^2)/2+(x1*x2)/2+x2^2- 0.5*x1-0.5*x2+0.5

Quadratic functions in 2 dims.

f(x)=(1/2) x^T q x + b^T x + c   x = $[x_1\ x_2]^T$

q=[1 0.5; 0.5 2];

b=[-0.5 ;-0.5];

c=0.5;

f(x)=(x1^2)/2+(x1*x2)/2+x2^2- 0.5*x1-0.5*x2+0.5

df/dx1 = x1+x2/2 -0.5

df/dx2 = x1/2+2*x2 -0.5

df=[df/dx1; df/dx2]

df/dx1x2=df/dx2x1=1/2

df/dx1x1=1

df/dx2x2=2

ddf=[df/dx1x1 df/dx1x2 ; df/dx2x1 df/dx2x2] = q

Opt. in 2 dims.

x = $[x_1\ x_2]^T$

```
% gradient decent
x_new = x_old - eps * df;
% [2,1] = [2,1] - [1,1]*[2,1]
% newton raphson
x_new = x_old - df/ddf;
x_new = x_old - inv(ddf)*df;
% [2,1] = [2,1] - [2,2]*[2,1]
```

Opt. in N dims.

% gradient decent $x = [x_1\ x_2\ x_3\ \dots\ x_n]^T$

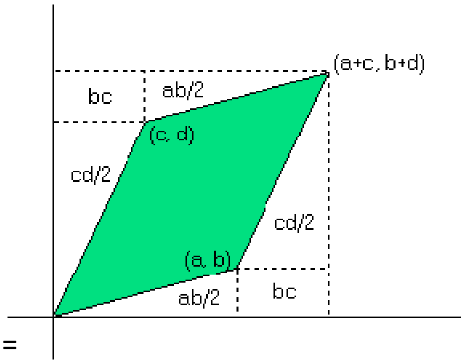
```
x_new = x_old - eps * df;
% [n,1] = [n,1] - [1,1]*[n,1]
% newton raphson
x_new = x_old - df/ddf;
x_new = x_old - inv(ddf)*df;
% [n,1] = [n,1] - [n,n]*[n,1]
```


Matrix inversion

- A is a square matrix (n*n)
- I is the identitiy matrix (n*n)
- $A*A^{-1}=I$
- A^{-1} is the inversion of A
- A matrix has an inverse if the determinant $|A|\neq0$

Geometric meaning of the determinant

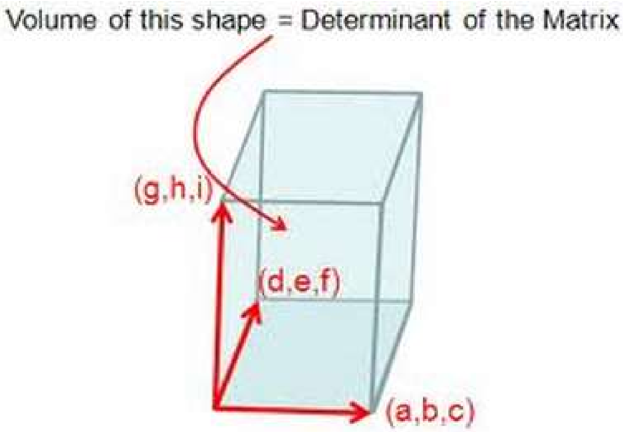
- A=[a b ; c d]
- det(A) is the area of the green parallelogram with vertices at (0,0), (*a*, *b*), (*a*+*c*, *b*+*d*), (*c*,*d*).



The area of the big rectengular=
(a+c)*(b+d)=a*b+a*d+c*d+c*b
The area of the green parallelogram =
=a*b+a*d+c*d+c*b – 2*c*b -2*(a*b)/2 -2*(d*c)/2
=a*d-c*b

Geometric meaning of the determinant

- In 3 dimensions:



Matrix inversion

- For a 2*2 matrix (A)= [a b ; c d]
- A^{-1} =[e f ; g h]
- det(A)=a*d-c*b
- [a b ; c d]*[e f ; g h]= [1 0 ; 0 1]
- a*e+b*g=1
- a*f+b*h=0
- c*e+d*g=0
- c*f+d*h=1

a*f=-b*h
f=-(b*h)/a
-(c*b*h)/a+d*h=1
h*(d-(c*b)/a)=1
h=a/(a*d-c*b)
h=a/det(A)

Matrix inversion

For a 2×2 matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

the matrix inverse is

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

h=a/det(A)

Matrix inversion

- For a 3×3 matrix the inverse may be written as:

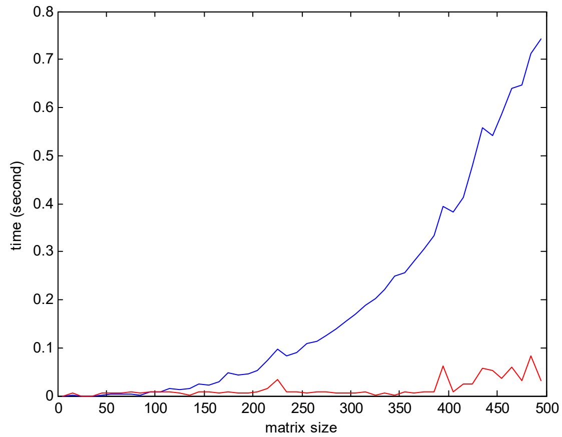
$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$\mathbf{A}^{-1} = \frac{\begin{bmatrix} ei-fh & hc-ib & bf-ce \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix}}{|\mathbf{A}|} = \frac{\begin{bmatrix} ei-fh & hc-ib & bf-ce \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix}}{aei+bfg+cdh-gec-hfa-idb}$$

A general n*n matrix can be inverted using methods such as the Gauss-Jordan elimination, Gauss elimination or LU decomposition.

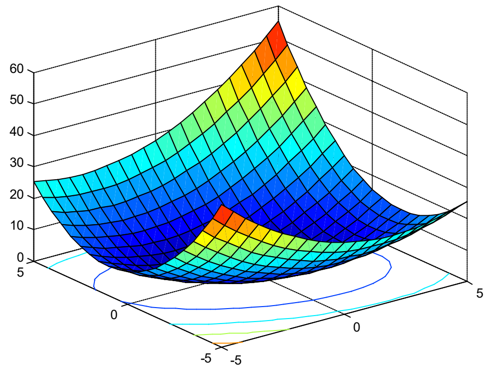
The cost of Matrix inversion

- inversion_time.m



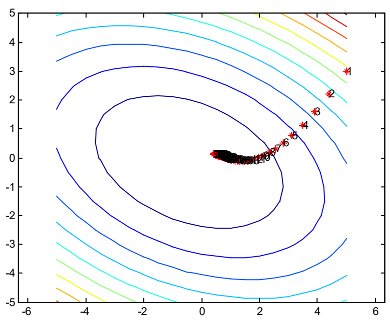
Blue mean time
Red std time

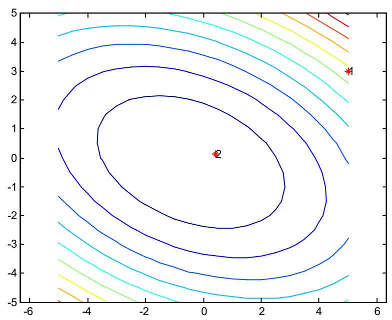
q=[1 0.5; 0.5 2];
b=[-0.5 ;-0.5];
c=0.5;



Gradient Descent
stepsize=0.1

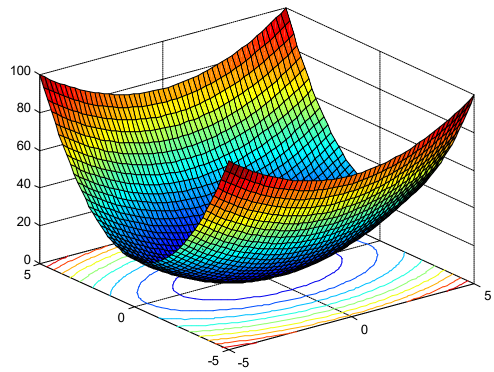
Newton Raphson



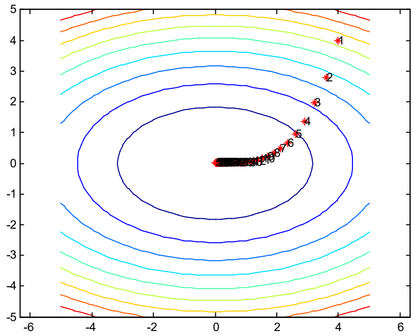


opt_Ndim.m

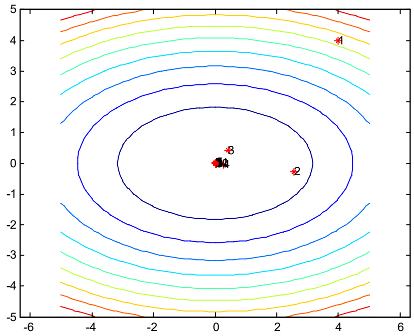
Find the minimum of f(x1,x2)=(x1*x1)+(3*x2*x2)



Gradient Descent
stepsize=0.05
do not converged at
50 iteration



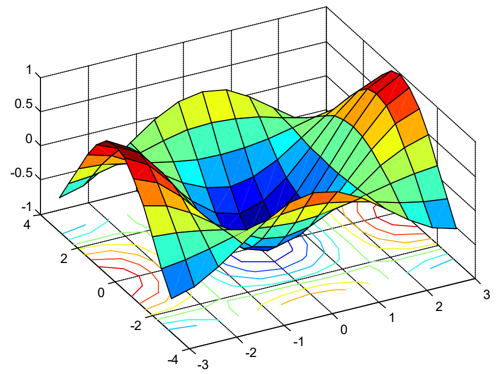
Steepest Descent
converged at the 12^{th}
iteration
Attention to
orthogonal updates



steepest_desc_2dim.m

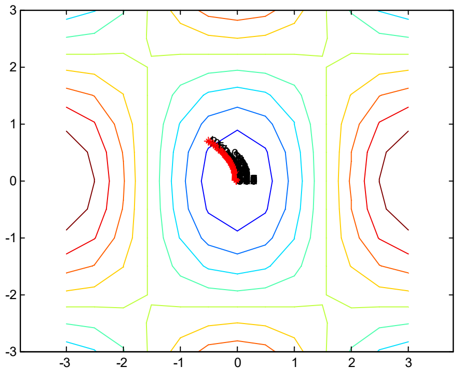
Griewank function

- f=((x1^2/4000)+(x2^2/4000))-(cos(x1)*cos(x2/(sqrt(2))))

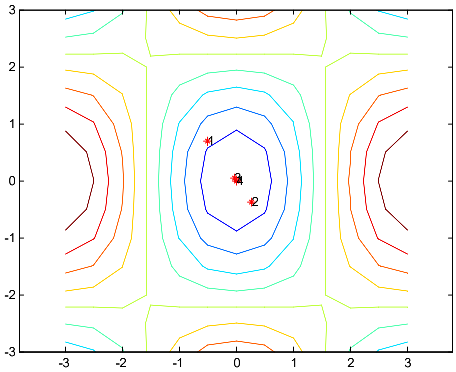


opt_Ndim_general.m

Gradient Descent
stepsize=0.1
converged at 118 th
iteration



Newton Raphson
converged at the 4th
iteration

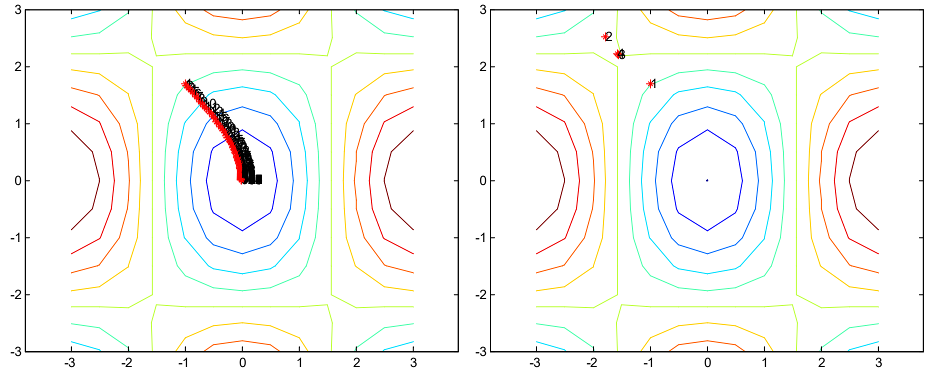


x0=[-0.5 ; 0.7]

opt_Ndim_general.m

Gradient Descent
stepsize=0.1
converged at 142 th iteration

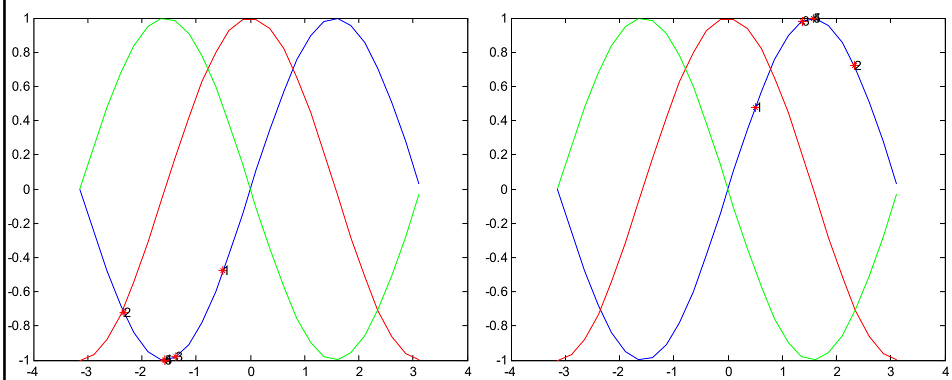
Newton Raphson
converged at the 5th iteration, but where?



x0=[-1 ; 1.7]

opt_Ndim_general.m

What happened to the Newton Raphson?



f(x)=sin(x)
Blue f
Red f'
Green f''

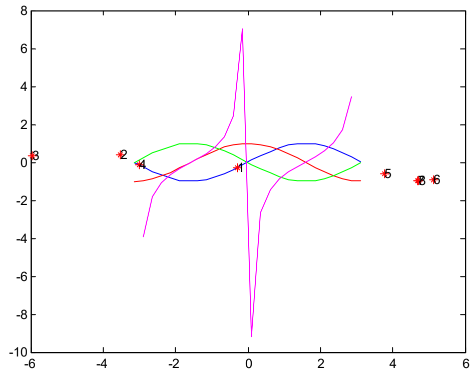
x0=-0.5

x0=0.5

Attention to the signs of f' and f''

newton_raphson_2.m

What happened to the Newton Raphson?

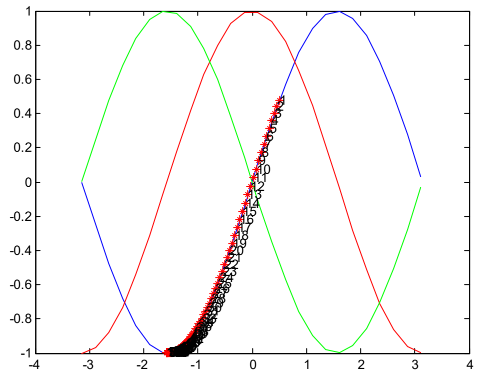


f(x)=sin(x)
Blue f
Red f’
Green f’’
Magenta f’/f’’

x0=-0.3
f’ / f’’ is not continuous

newton_raphson_2.m

What happens if we use Gradient descent?

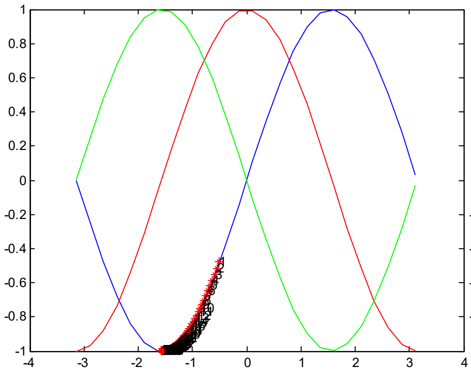


f(x)=sin(x)
Blue f
Red f’
Green f’’

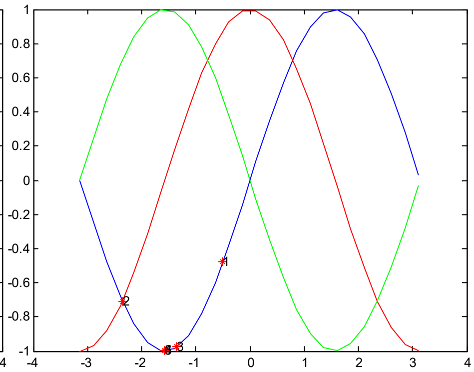
Step size=0.05
f’ is positive, f’’ is not used

newton_raphson_2.m

Optimization using approximated derivatives



Gradient descent converged at 174th iteration



Newton Raphson converged at 7th iteration

f(x)=sin(x)
Blue f
Red real f'
Green real f''

newton_raphson_3.m

Some more comparisons

- opt_Ndim_general.m
- Nigthmares of a convex optimization, because of local minimums
- acckley f=(-20*exp(-0.2*sqrt((1/2)*(x1.^2+x2.^2))) -
-     exp((1/2)*(cos(2*pi*x1) + cos(2*pi*x2))) +
-     20 + exp(1) + 5.7);
- griewank f=((x1^2/4000)+(x2^2/4000))-(cos(x1)*cos(x2/(sqrt(2))));
- rastrigin f=10*2 + x1.^2 + x2.^2 - 10*cos(2*pi*x1) - 10*cos(2*pi*x2);
- rosen f=100*(x1^2-x2)^2+(x1-1)^2;
- schwell f=(abs(x1)+abs(x2))+(abs(x1)*abs(x2));

References

- http://math.tutorvista.com/calculus/newton-raphson-method.html
- http://math.tutorvista.com/calculus/linear-approximation.html
- http://en.wikipedia.org/wiki/Newton's_method
- http://en.wikipedia.org/wiki/Steepest_descent
- http://www.pitt.edu/~nak54/Unconstrained_Optimization_KN.pdf
- http://mathworld.wolfram.com/MatrixInverse.html
- http://lpsa.swarthmore.edu/BackGround/RevMat/MatrixReview.html
- http://www.cut-the-knot.org/arithmetic/algebra/Determinant.shtml
- Matematik Dünyası, MD 2014-II, Determinantlar
- http://www.sharetechnote.com/html/EngMath_Matrix_Main.html
- Advanced Engineering Mathematics , Erwin Kreyszig, 10th Edition, John Wiley & Sons, 2011
- http://en.wikipedia.org/wiki/Finite_difference
- http://ocw.usu.edu/Civil_and_Environmental_Engineering/Numerical_Methods_in_Civil_Engineering/NonLinearEquationsMatlab.pdf
- http://www-math.mit.edu/~djk/calculus_beginners/chapter09/section02.html
- http://stanford.edu/class/ee364a/lectures/intro.pdf