Optimization Techniques Section 1

M. Fatih Amasyalı

What is optimization?

(mathematical) optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

optimal solution $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

Mehmet Fatih AMASYALI Optimization Techniques Lecture Notes YILDIZ TECHNICAL UNIVERSITY COMPUTER ENG DEPT.

Optimization example (Constraint Optimization)

- Find the largest possible rectangular area you can enclose, assuming you have L meters of fencing.
- f(x)=x*((L/2)-x)

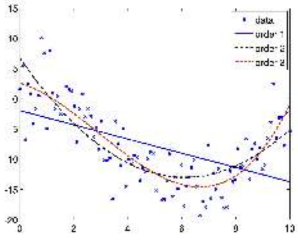
Optimization examples

portfolio optimization

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

data fitting

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error



Optimization Techniques

- Derivative Techniques
  - Gradient descent
  - Newton Raphson
  - LM
  - BFGS
- Non Derivative Techniques
  - Hill climbing
  - Genetic algorithms
  - Simulated annealing

Linear, nonlinear, convex functions

- f is a function $R^n \rightarrow R$

For every x1,x2 ∈ R^n (x1, x2 are n dimensional points)

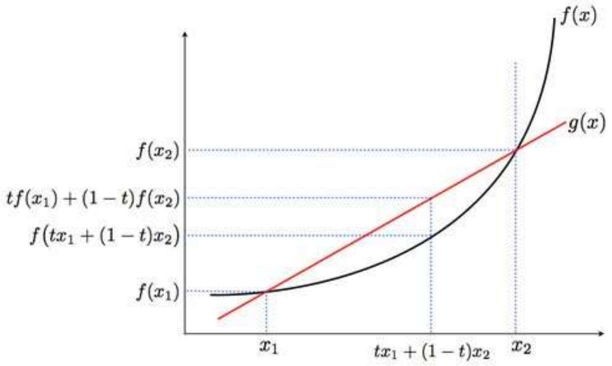
- f is linear if

  f(x1+x2)=f(x1)+f(x2) and f(r*x1)=r*f(x1)
- f(x)=2*x is linear and affine.
- f(x)=2*x+3 is affine but not linear.
- A linear function fixes the origin, whereas an affine function need not do so. An affine function is the composition of a linear function with a translation.
- f(x)=3 is constant, but not linear.

Linear, nonlinear, convex functions

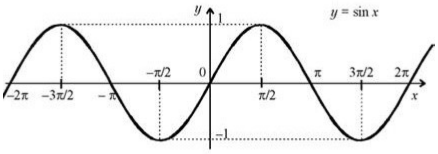
- f is convex if

  λ*f(x1) + β*f(x2) ≥ f(λ*x1 + β*x2). where λ+β=1, λ≥0, β≥0



Linear, nonlinear, convex functions

- f(x1)=3*x1+3
- f(x1,x2)=a*x1+b*x2+c
- f(x1,x2)=x2^2+x1^2+3*x1
- f(x)=sin(x) is convex when x ∈ [?,?]
- Concave?



Mehmet Fatih AMASYALI Optimization Techniques Lecture Notes YILDIZ TECHNICAL UNIVERSITY COMPUTER ENG DEPT.

How to find minimum point? Commonsense

- Use derivative
- Go to the negative direction of derivative
- But, step size?

General Algorithm

- initial guess
- While (improvement is significant) and (maximum number of iterations is not exceeded)
  - improve the result

Gradient Descent

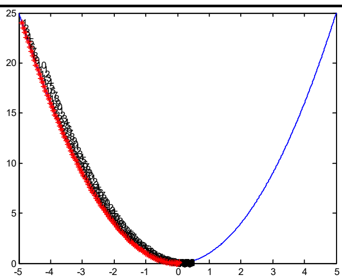
- Using only derivative sign
- Using also magnitude of the derivative
- Effect of the step size
- Effect of the starting point
- Local minimum problem

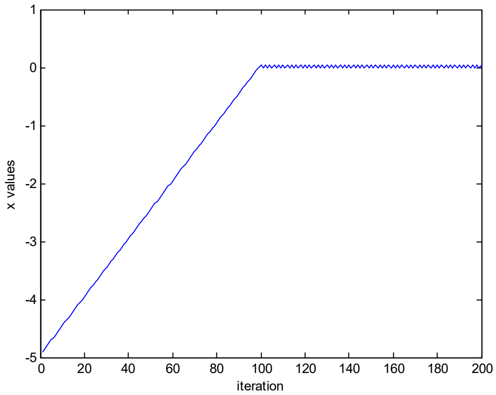

```
% The gradient descent algorithm with
constant stepsize

% f(x)=x^2, with derivative f'(x)=2x

clear all;
close all;

% plot objective function
x=-5:0.01:5;
y=zeros(1,length(x));
for i=1:length(x)
   y(i)=x(i)^2;
end
plot(x,y);

x_old = 0;
x_new = -4.9;
% The algorithm starts at x=-4.9
eps = 0.05; % step size
precision = 0.00001; % stopping condition1
max_iter=200; % stopping condition2
Xs=zeros(1,max_iter);
Ys=zeros(1,max_iter);
i=1;
while abs(x_new - x_old) > precision &&
max_iter>=i
   Xs(i)=x_new;
   Ys(i)=x_new^2;
   hold on;
   plot(Xs(i),Ys(i),'r*');
   text(Xs(i),Ys(i),int2str(i));

   x_old = x_new;
   df= 2 * x_old;
   x_new = x_old - eps * sign(df); % gradient sign
   i=i+1;
end
figure;
plot(Xs(1:i-1));
xlabel('iteration');
ylabel('x values');
```

gradient_desc_1.m

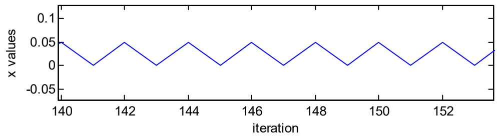
Result

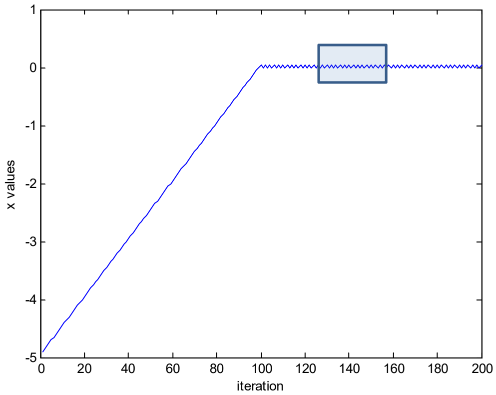
- f(x)=x^2
- f'(x)=2*x
- Max_iter=200
- Constant step size =0.05
- Starting point=-4.9



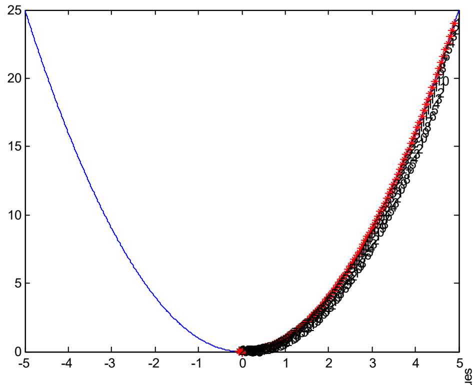


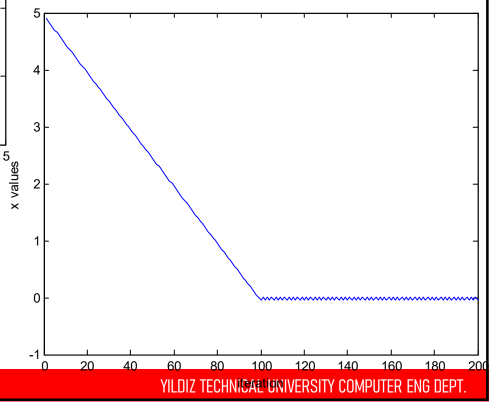
- It reaches the maximum iteration number
- It reached 0 at 100.iteration , then zig-zag.
- It selects 0 or 0.05 as the point f is minimum.





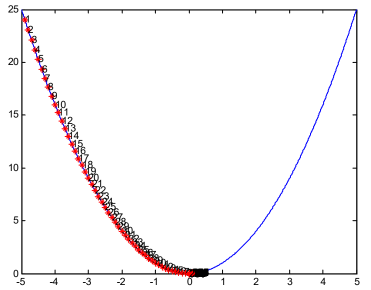
Starting point=4.9

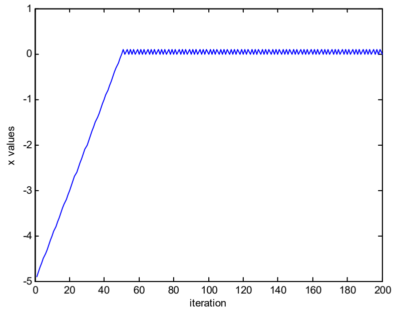




Step size=0.1

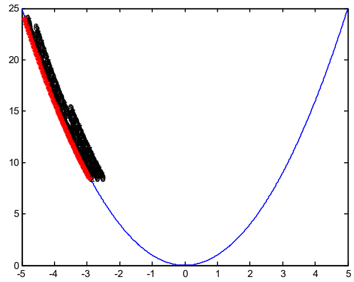
- It reaches the maximum iteration number
- It reached 0 at 50.iteration , then zig-zag.
- It selects 0 or 0.1 as the point f is minimum.

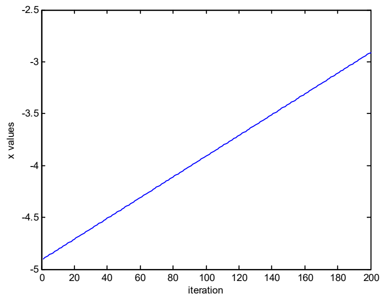




Step size=0.01

- It reaches the maximum iteration number
- It does not reached to 0.
- It selects -2.91 as the minimum point.
- It will zig-zag.



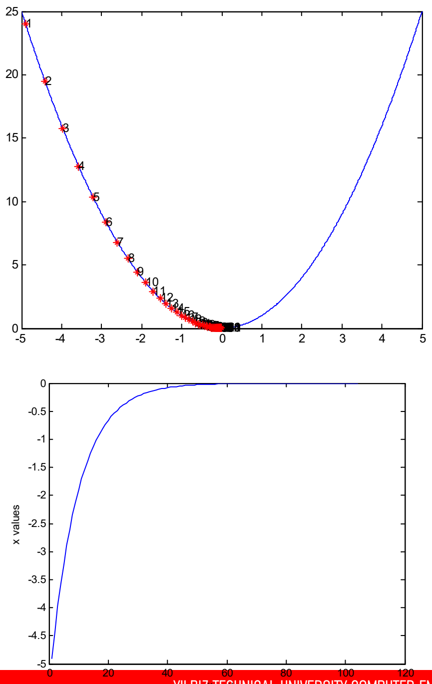


Gradient Descent

- Using also the magnitude of the derivative
- x_new = x_old - eps * df;
- instead of x_new = x_old - eps * sign(df);
- Can we reduce the required number of iterations?

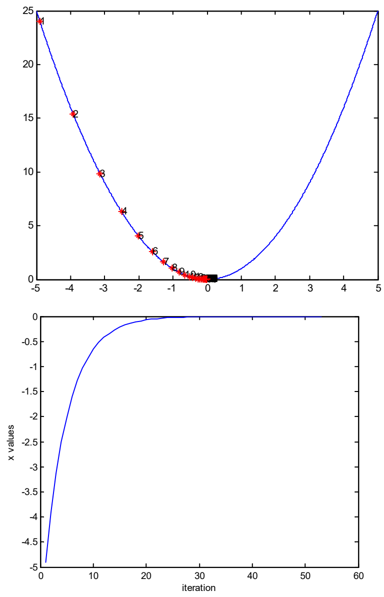
gradient_desc_2.m

- Step size=0.05
- Max_iter=200
- The number of iteration is 120.
- It does not zig-zag.
- It selects 0 ± 10^{-5} the point f is minimum.



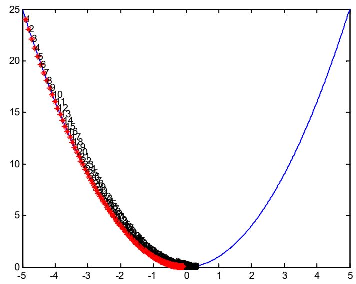
gradient_desc_2.m

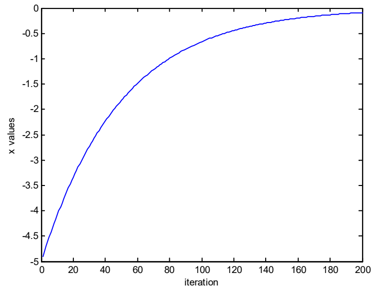
- Step size=0.1
- Max_iter=200
- The number of iteration is 60.
- It does not zig-zag.
- It selects 0 ± 10^{-5} the point f is minimum.



gradient_desc_2.m

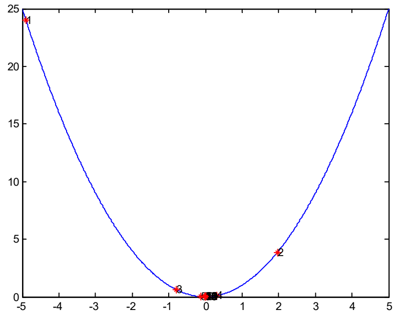
- Step size=0.01
- Max_iter=200
- It reaches the maximum iteration number.
- It does not zig-zag.
- It selects 0 ± 10^{-2} as the point f is minimum.
- It requires the more iteration.

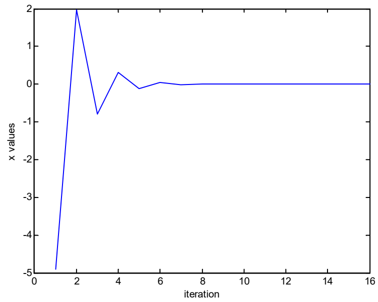




gradient_desc_2.m

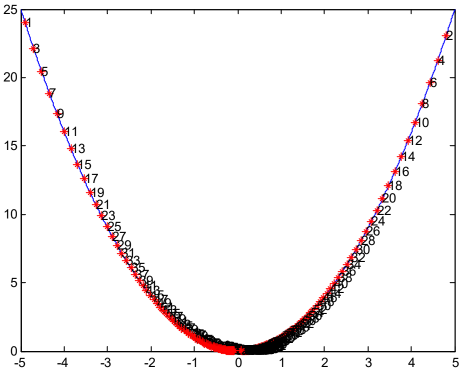
- Step size=0.7
- Max_iter=200
- The number of iteration is 16.
- It does not zig-zag.
- It selects 0 ± 10^{-5} the point f is minimum.

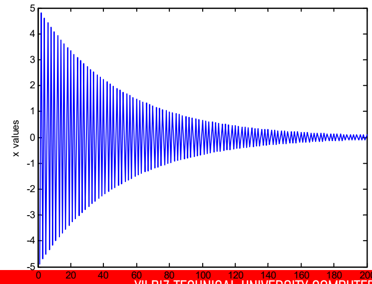




gradient_desc_2.m

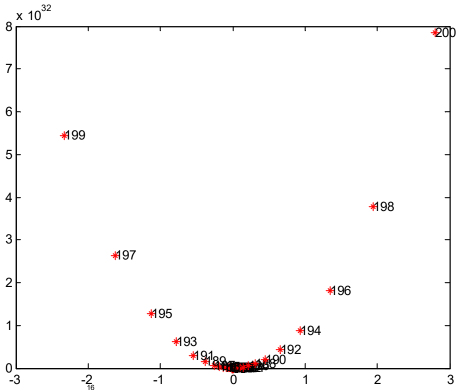
- Step size=0.99
- Max_iter=200
- It reached the maximum iteration number.
- It zig-zag, but zig-zag magnitude decreases.
- It selects 0 ± 10^{-1} the point f is minimum.
- It requires more iteration.

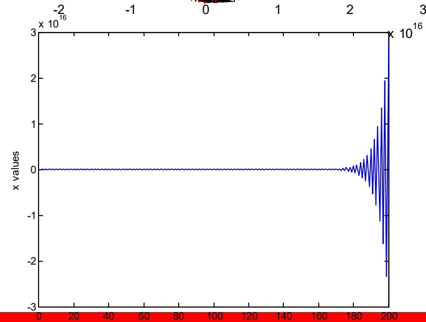




gradient_desc_2.m

- Step size=1
- Max_iter=200
- It reached the maximum iteration number.
- It does not converge.



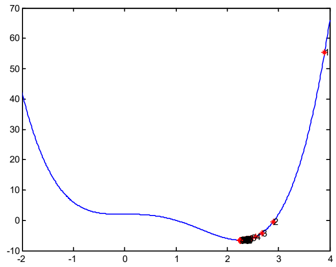


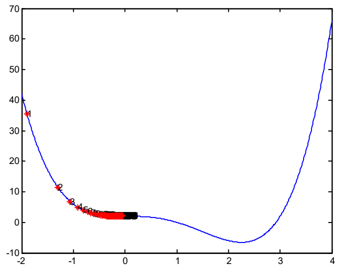
Starting point matters?

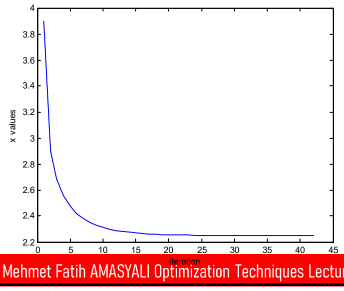
- f(x)=f(x)=x^4-3x^3+2 with derivative f'(x)=4x^3-9x^2
- It has two extreme points x=0, x=9/4
- gradient_desc_3.m

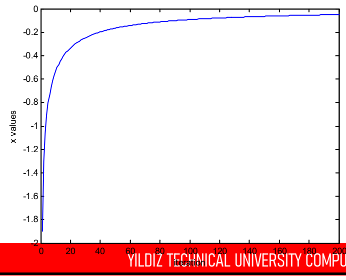
Step size=0.01

Starting point=3.9 Starting point=-1.9

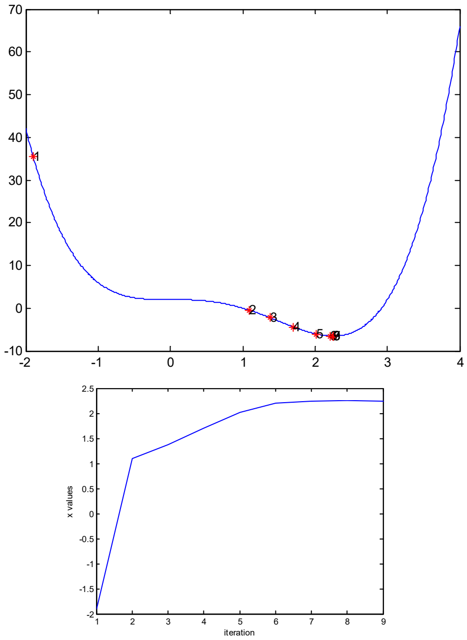








- Step size=0.05 can handle plateau, but not guaranteed.
- Step size=0.1 is very big for this function, it does not converge



Gradient descent

- Static zig-zag (hopeless) vs. slow converge (with hope)
- Very Small step size → slow converge, plateau problem
- Very big step size → do not converge
- Big and small step sizes depend on the objective function.

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
- http://2020insight.com/chasing-returns-risk-getting-burned/
- http://jitkomut.lecturer.eng.chula.ac.th/matlab/eng_probs.html