```
% 2017 Fall Numerical Optimization Homework #1
% 2017. 09. 25
% 20161216 HAN SANGJUN
```

Problem 1 - find local minima (3*x^2 - 6*x + 7)

```
clc;
clear;
close all;

syms x;
f = symfun(3*x^2 - 6*x + 7, x);

% first derivative
fx = diff(f, x);

% ======================== Bisection method ========================
% automatically generate init where f(a)f(b) < 0
% but I need local minima, starting strictly with f(a) < 0, f(b) > 0,
 a < b
init_range = 10; % init range where to pick up between [-100 100]
while 1
    init = -init_range + (init_range + init_range) * rand(1, 2);
    if (fx(init(1)) < 0) && (fx(init(2)) > 0) && (init(1) < init(2))
        break;
    end;
end;
epsilon_0 = 0.005; % terminate condition when almost f(x) = 0
epsilon_ab = 0.001; % terminate condition when b - a < epsilon
[opt_x, opt_y, time] = bisection(fx, init, epsilon_0, epsilon_ab);
local_min = double(f(opt_x));
fprintf('1. Bisection - It takes %fsec to generate local minima %f at
 %f, when derivative of f(x) = %f\n', time, local_min, opt_x, opt_y);

% ======================== Newton's method ========================
% second derivative
fxx = diff(fx, x);

% automatically generate init
% but I need local minima, starting strictly with f' > 0
init_range = 10; % init range where to pick up between [-100 100]
while 1
    init = -init_range + (init_range + init_range) * rand(1);
    if (fx(init) > 0 && fxx(init) > 0)
        break;
    end;
end;
epsilon_0 = 0.005; % terminate condition when almost f(x) = 0
[opt_x, opt_y, time] = newton(fx, fxx, init, epsilon_0);
local_min = double(f(opt_x));
fprintf('1. Newton - It takes %fsec to generate local minima %f at %f,
 when derivative of f(x) = %f\n', time, local_min, opt_x, opt_y);
```

```
% ======================== Secant method ========================
% automatically generate init
% but I need local minima, starting strictly with a > 0, f(a) > 0, a <
 b
init_range = 10; % init range where to pick up between [-100 100]
while 1
    init = -init_range + (init_range + init_range) * rand(1, 2);

    if  (init(1) > 0) && (fx(init(1)) > 0) && (init(1) < init(2))
        break;
    end;
end;
epsilon_0 = 0.005; % terminate condition when almost f(x) = 0
[opt_x, opt_y, time] = secant(fx, init, epsilon_0);
local_min = double(f(opt_x));
fprintf('1. Secant - It takes %fsec to generate local minima %f at %f,
 when derivative of f(x) = %f\n', time, local_min, opt_x, opt_y);

% ======================== Regula falsi method
 ========================
% automatically generate init where f(a)f(b) < 0
% but I need local minima, starting strictly with f(a) < 0, f(b) > 0,
 a < b
init_range = 10; % init range where to pick up between [-100 100]
while 1
    init = -init_range + (init_range + init_range) * rand(1, 2);
    if (fx(init(1)) < 0) && (fx(init(2)) > 0) && (init(1) < init(2))
        break;
    end;
end;
epsilon_0 = 0.005; % terminate condition when almost f(x) = 0
[opt_x, opt_y, time] = regula(fx, init, epsilon_0);
local_min = double(f(opt_x));
fprintf('1. Regula - It takes %fsec to generate local minima %f at %f,
 when derivative of f(x) = %f\n', time, local_min, opt_x, opt_y);
```

```
1. Bisection - It takes 0.177240sec to generate local minima 4.000001
 at 0.999456, when derivative of f(x) = -0.003267
1. Newton - It takes 0.028498sec to generate local minima 4.000000 at
 1.000000, when derivative of f(x) = 0.000000
1. Secant - It takes 0.031304sec to generate local minima 4.000000 at
 1.000000, when derivative of f(x) = 0.000000
1. Regula - It takes 0.025210sec to generate local minima 4.000000 at
 1.000000, when derivative of f(x) = 0.000000
```

Published with MATLAB® R2015b