ECE-6500

ASSIGNMENT - 2

- UTKARSH PRATAP SINGH JADON
500711257

Q1] $f(x) = x_1^2 - x_1 x_2 + x_2^2 - 3x_2$

$$\nabla f = \begin{bmatrix} 2x_1 - x_2 \\ -x_1 + 2x_2 - 3 \end{bmatrix}$$

from the gradient, we get following two eq^n:

$$2x_1 - x_2 = 0 \quad \text{——} (1)$$

$$-x_1 + 2x_2 = 3 \quad \text{——} (2)$$

Solving these two eq^n, we get

$\boxed{x_1 = 1}$ $\boxed{x_2 = 2}$

$$\nabla^2 f = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Eigen values of Hessian are 3 and 1

$\therefore$ Hessian is Positive definite

Hence, $f(x)$ is strictly convex

$\therefore$ Local minima is also a global minima, i.e. $\underline{\underline{(1, 2)}}$

Q2) a) $A = \begin{bmatrix} 2 & 1 \\ 1 & 5 \end{bmatrix}$

Eigen values are given by

$$\lambda = m \pm \sqrt{m^2 - p}$$

where m - mean of trace

p - determinant of the matrix

Here, $\lambda_1 = 1.697$, $\lambda_2 = 5.302$

$\therefore$ A is Positive Definite

b) $B = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

$\lambda_1 = 3$, $\lambda_2 = 0$, $\lambda_3 = 3$

$\therefore$ B is Positive Semidefinite

c) $C = \begin{bmatrix} -3 & -3 & 0 \\ -3 & -10 & -7 \\ 0 & -7 & -8 \end{bmatrix}$

$\lambda_1 = -16.46$, $\lambda_2 = -4.23$, $\lambda_3 = -0.3$

$\therefore$ C is Negative Definite.

d) $D = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ $\lambda_1 = 3$, $\lambda_2 = -1$

$\therefore$ D is None

Q3|

$$f(x) = \left(\sum_{i=1}^{n} x_i^{p} \right)^{1/p} \qquad p < 1, \; p \neq 0$$

Show function is concave

$$\frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\sum_{i=1}^{n} x_i^{p} \right)^{1/p}$$

$$= \frac{1}{p} \left(\sum_{i=1}^{n} x_i^{p} \right)^{\frac{1}{p}-1} \cdot p \cdot (x_i^{p-1})$$

$$= \left(\sum_{i=1}^{n} x_i^{p} \right)^{\frac{1-p}{p}} \cdot (x_i^{p-1})$$

$$\frac{\partial f}{\partial x_i} = \left(\frac{f(x)}{x_i} \right)^{1-p}$$

Now, $$\frac{\partial^2 f}{\partial x_i^2} = \frac{1-p}{f(x)} \left(\frac{f(x)^2}{x_i^2} \right)^{1-p} - \frac{1-p}{x_i} \left(\frac{f(x)}{x_i} \right)^{1-p}$$

We need to show that,

$$y^T \nabla^2 f(x) y = \frac{1-p}{f(x)} \left(\left(\sum_{i=1}^{n} \frac{y_i f(x)^{1-p}}{x_i^{1-p}} \right)^2 - \sum_{i=1}^{n} \frac{y_i^2 f(x)^{2-p}}{x_i^{2-p}} \right) \leq 0$$

On comparing this with Cauchy-Schwarz inequality

$$a^T b \leq \|a\|_2 \|b\|_2$$

We get, $$a_i = \left(\frac{f(x)}{x_i} \right)^{-p/2}, \quad b_i = y_i \left(\frac{f(x)}{x_i} \right)^{1-p/2}$$

with $\sum_i a_i^2 = 1$

Q5] $U = \sum_{t=1}^{T} \beta^t u(t)$

where present value of utility derived from consumption is given by U and $0 < \beta < 1$ is discount factor.

maximize $U = \sum_{t=1}^{T} \beta^t u(ct)$

subject to $K_{t+1} = K_t + f(K_t) - ct$

and $t = 0, \dots T$

$K_t \geq 0$, $t = 1, \dots T+1$

with variables $c_1, \dots c_T$ and $K_1, \dots K_{T+1}$

The objective funct^n is concave, since it is positive weighted sum of concave functions.

Budget constraints are not convex, since the constraints are equality involving non-linear function f.

Modified problem:

maximize $U = \sum_{t=1}^{T} \beta^T u(ct)$

subject to $K_{t+1} \leq K_t + f(K_t) - ct$ $t = 0, \dots T$

$K_t \geq 0$, $t = 1, \dots T+1$

Budget inequalities can be written as

$K_{t+1} - K_t - f(K_t) + ct \leq 0$

where LHS is convex function of variables c and K.

By changing the equality constraints into inequalities, we are relaxing the constraints

Due to this, when we solve modified problem with inequality constraints, for any optimal solution we actually get equality for each of budget constraints

Let c^* and K^* be optimal for modified problem, for some period S,

$$K^*_{S+1} < K^*_S + f(K^*_S) - c^*_S$$

for any time period t, the above eqⁿ implies neither we are investing or consuming all of our funds.

New consumption stream $\tilde{c}$

$$\tilde{c}_t = \begin{cases} c^*_t & t \neq S \\ c^*_t + \epsilon & t = S \end{cases}$$ where ϵ is small positive number

$$\therefore \; K^*_{t+1} \leq K^*_S + f(K^*_S) - c^*_S$$

$V(\tilde{c}) \geq V(c^*)$ since 2 streams consume the same amount for every period except one, in which we consume more with $\tilde{c}$.

Let $\tilde{K}$ be asset stream that results from consumption stream $\tilde{c}$. All the constraints of original problem are satisfied for $\tilde{c}$ and $\tilde{K}$, yet c^* has lower objective than $\tilde{c}$. This contradicts optimality of c^*.

In conclusion, for c^*, we have

$$K^*_{t+1} = K^*_t + f(K^*_t) - c^*_t$$

Q6]

Bit rate $R_i = \alpha_i W_i \log(1 + \beta_i P_i / W_i)$

α_i, β_i are positive constants

$P_1 + \ldots + P_n = P_{total}$ $\quad P_{total} > 0$

$W_1 + \ldots + W_n = W_{total}$ $\quad W_{total} > 0$

Utility $\sum_{i=1}^{n} U_i(R_i)$

Substituting R_i in utility function U_i:

$$\text{maximize} \sum_{i=1}^{n} U(\alpha_i W_i \log(1 + \beta_i P_i / W_i))$$

It is given that utility funct^n U_i is non-decreasing and concave. If we show that R_i is a concave funct^n of (P_i, W_i), it will follow that $U(R_i)$ is also concave since U is non-decreasing concave function

$$\nabla = \begin{bmatrix} \dfrac{\alpha_i W_i}{1 + \frac{\beta_i P_i}{W_i}} \cdot \dfrac{\beta_i}{W_i} \\[2ex] \alpha_i \log\left(1 + \dfrac{\beta_i P_i}{W_i}\right) + \dfrac{1}{1 + \frac{\beta_i P_i}{W_i}} \left(\dfrac{-\beta_i P_i}{W_i^2}\right) \cdot \alpha_i W_i \end{bmatrix}$$

$$\nabla = \begin{bmatrix} \dfrac{\alpha_i \beta_i}{1 + \frac{\beta_i P_i}{W_i}} \\[2ex] \alpha_i \log\left(1 + \dfrac{\beta_i P_i}{W_i}\right) - \dfrac{\alpha_i \beta_i P_i}{W_i + \beta_i P_i} \end{bmatrix}$$

$\therefore$ Hessian is given by :

$$\nabla^2 R_i = \frac{-\alpha_i \beta_i^2}{W_i (1+ \beta_i P_i / W_i)^2} \begin{bmatrix} 1 \\ -P_i \end{bmatrix} \begin{bmatrix} 1 \\ -P_i \end{bmatrix}^T$$

We know that α_i, β_i, W_i and P_i are positive

$\therefore$ $\nabla^2 R_i$ is negative semi-definite

$\therefore$ R_i is convex

Hence, given problem is convex optimization problem

Q7] Projection : $P_C[x] = \underset{y \in C}{\arg\min} \; ||x-y||_2$

of $x \in \mathbb{R}^n$ onto convex set $C \in \mathbb{R}^n$

A vector y^* is the solution if and only if

$$(y^* - x)^T (y - y^*) \geq 0 \qquad \forall \; y \in C$$

$$\Rightarrow (x - y^*)^T (y - y^*) \leq 0$$

Let us assume y_1^* and y_2^* be the two minimizing vectors, then

$$(x - y_1^*)^T (y_2^* - y_1^*) \leq 0 \quad \text{——— (1)}$$

$$(x - y_2^*)^T (y_1^* - y_2^*) \leq 0 \quad \text{——— (2)}$$

Adding (1) + (2) : $(y_2^* - y_1^*) \left[(x - y_1^*)^T - (x - y_2^*)^T \right] \leq 0$

$$\Rightarrow \|y_2^* - y_1^*\|^2 \leq 0 \Rightarrow \boxed{y_1^* = y_2^*}$$

Here, it is proved that projection of a point $x \in \mathbb{R}^n$ onto a convex set $C \subset \mathbb{R}^n$ is unique.

Q8| a) $f(x) = \log(e^x + e^{-x})$ $\quad x^* = 0 \quad t = 1 \quad x^{(0)} = 1$ & $x^{(0)} = 1.1$

$$x^k = x^{k-1} - \frac{\nabla f(x^{k-1})}{\nabla^2 f(x^{k-1})}$$

Here, $\nabla f(x) = \frac{1}{e^x + e^{-x}}(e^x - e^{-x}) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

and $\nabla^2 f(x) = \frac{-1}{(e^x + e^{-x})^2}(e^x - e^{-x})^2 + \frac{\cancel{e^x + e^{-x}}}{\cancel{e^x + e^{-x}}}$

$$\nabla^2 f(x) = 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2$$

for $x^{(0)} = 1$

$\nabla f(x^0) = \frac{e^1 - e^{-1}}{e^1 + e^{-1}} = 0.7615$ $\qquad \nabla^2 f(x^0) = 1 - 0.7615^2 = \cancel{0.58002}$ 0.4199

$\therefore x^1 = 1 - \frac{0.7615}{\cancel{0.58002}\ 0.4199} \Rightarrow x^{(1)} = -0.8134$

for $k = 1$, $f(x^{(1)}) - p^* = 0.4311 - \log 2 = 0.13007$

Similarly, for $k=2$, $x^{(2)} = 0.4094$ $\quad f(x^{(2)}) - p^* = 0.0354$

$k=3$, $x^{(3)} = -0.0473$ $\quad f(x^{(3)}) - p^* = 4.8564 \times 10^{-4}$

$\therefore$ We can see that the method converges.

for $x^{(0)} = 1.1$

$k=1, \quad x^{(1)} = -1.129 \qquad f(x^{(1)}) - p^* = 0.2324$

$k=2, \quad x^{(2)} = 1.234 \qquad f(x^{(2)}) - p^* = 0.2702$

$k=3, \quad x^{(3)} = -1.695 \qquad f(x^{(3)}) - p^* = 0.4494$

$\therefore$ We can see that method diverges

b) $f(x) = -\log x + x \qquad x^* = 1 \qquad t = 1 \qquad x^{(0)} = 3$

$f'(x) = \frac{-1}{x} + 1 \qquad f''(x) = \frac{1}{x^2}$

$$x^{(1)} = x^{(0)} - \frac{f'(x^{(0)})}{f''(x^{(0)})}$$

$$= 3 - \frac{-\frac{1}{3}+1}{\frac{1}{3^2}} = 3 - \frac{2}{3} \times 9^3$$

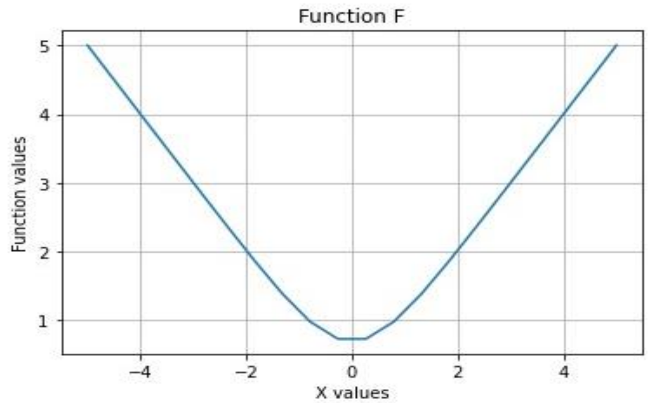
$$= 3-6$$

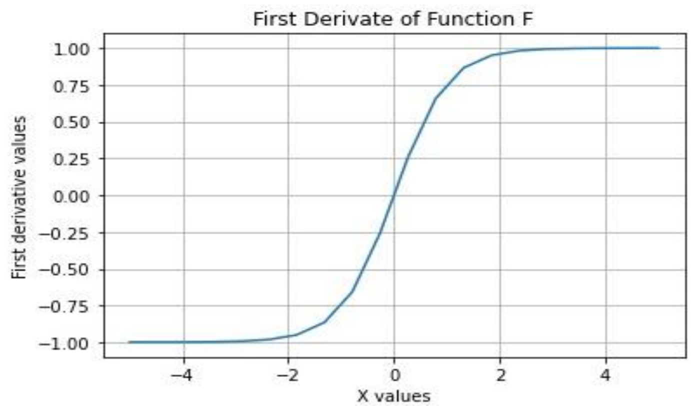
$$= -3$$

Here, $x^{(1)} = -3$. But domain of f is $\{x \mid x > 0\}$

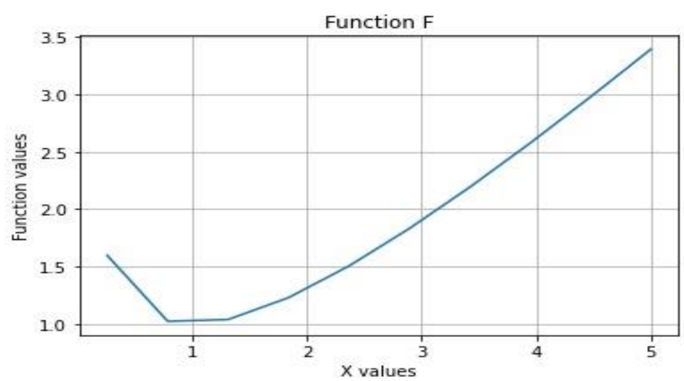
and $x^{(1)}$ lies outside dom. f

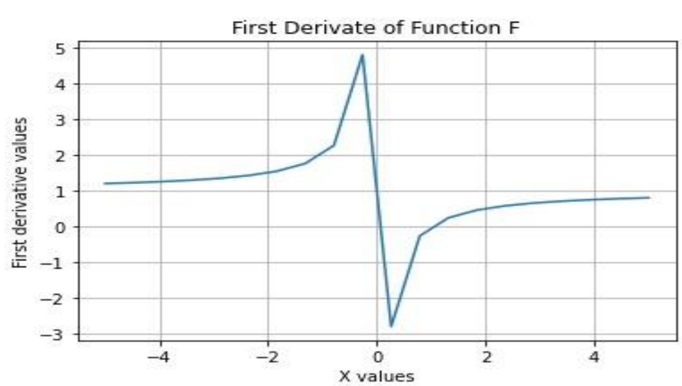
Q8a) F(x) = log(e^x + e^-x)





Q8b) F(x) = -log(x) + x





Q4]

$$\max \sum_{i=1}^{n} \log(\alpha_i + P_i)$$

such that $P_i \geq 0 \ \forall i$ and $\sum_{i=1}^{n} P_i \leq P_{max}$

Ignoring the constraint $P_i \geq 0$

and $1^T P = P_{max}$ for maximization

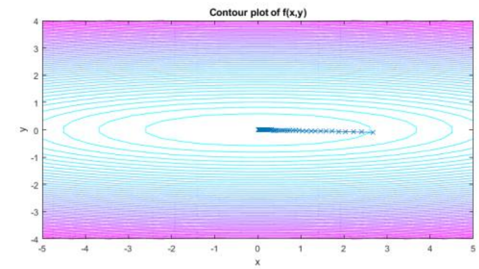
To maximize, we will have to make all the values equal.

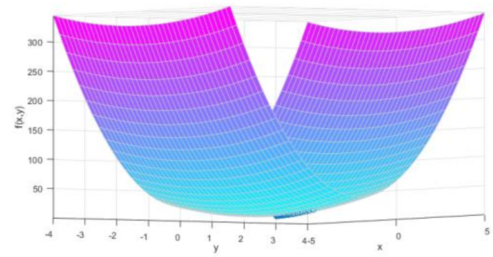
let $\bar{\alpha} = \frac{\sum_i^n \alpha_i}{n}$

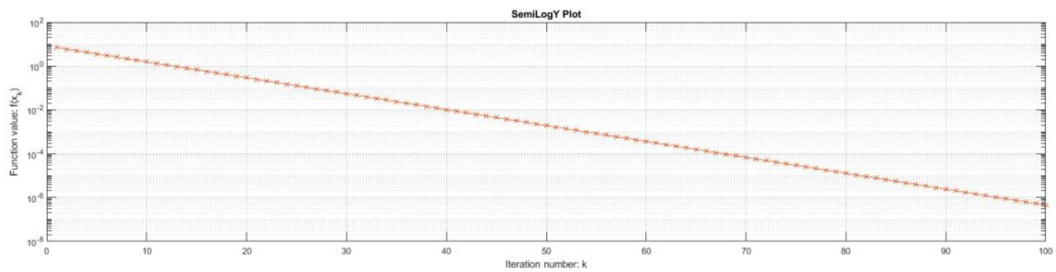
$$P^* = \begin{bmatrix} \frac{P_{max}}{n} - \alpha_1 + \bar{\alpha} \\ \frac{P_{max}}{n} - \alpha_2 + \bar{\alpha} \\ \vdots \\ \frac{P_{max}}{n} - \alpha_n + \bar{\alpha} \end{bmatrix}$$

Question 9 :

Heavy Ball method







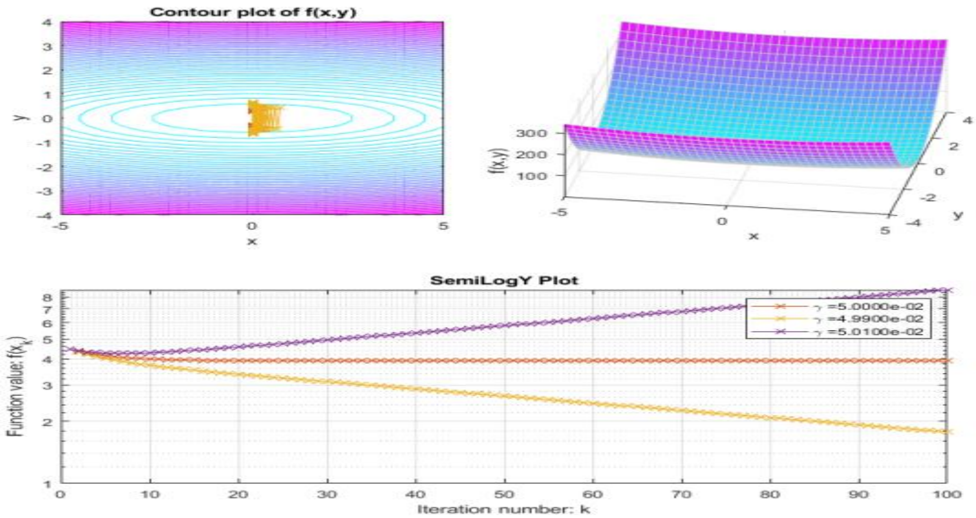
EXERCISE: Enter the step sizes (values of gamma) to be compared. Choose epsn < 1e-3 if required.

```
epsn = 1e-4;
beta = ((sqrt(L) - sqrt(mu))/ (sqrt(L) + sqrt(mu))^2);
beta_n = sqrt(beta);
gamma1 = 2/40;
gamma2 = 2*(1+ beta) / (mu + L);
```

This is the gradient update equation

```
            if k < 2
                xy_prev = xy;
                xy = xy - gamma1*grad_mat;
            else
                xy_buf = xy;
                xy = xy - gamma2*grad_mat + beta*(xy - xy_prev);
                xy_prev = xy_buf;
            end
        end
    end
```

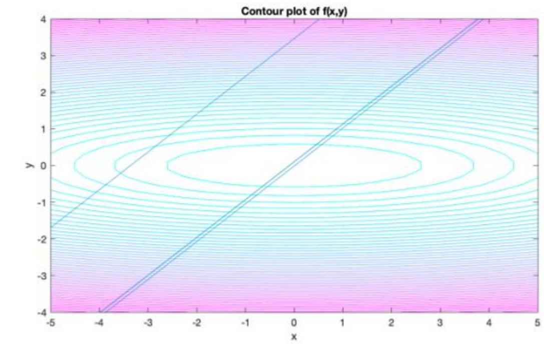
Gradient Descent Method:

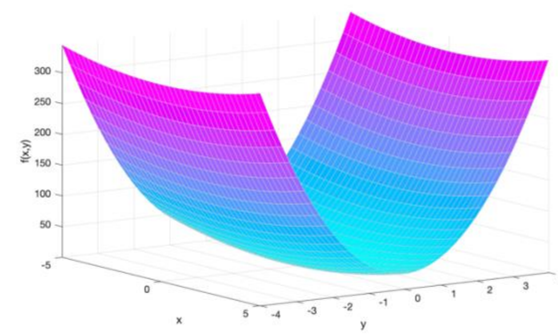


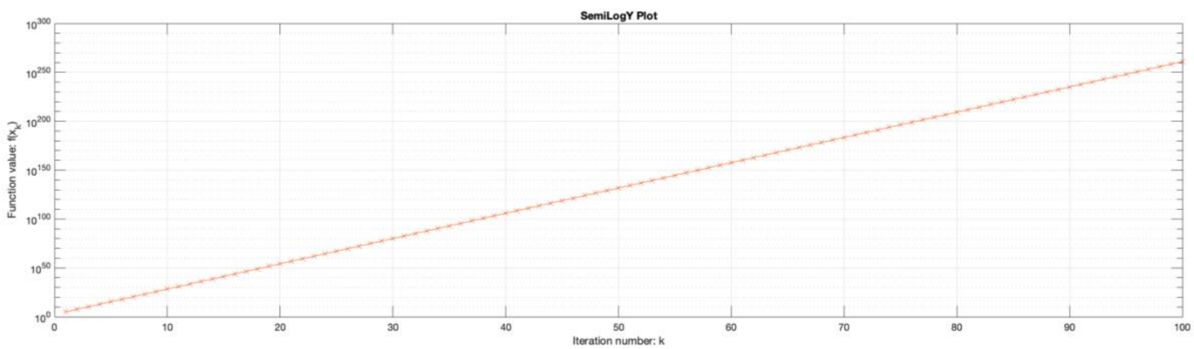
The red line belongs to the (B+Epsilon), purple line belongs to (B) and yellow line belongs to (B-epsilon)

From the above graphs, it can be observed that when the step size was increased marginally than lemma growth interval, the function values do not converge to the minimum.

Nesterov's Gradient Method







Numerical Optimization Methods

Choice of Optimization Method

```
main_method = 'gradient-constant';
```

Function to be minimized

```
f = inline('1*x^2+20*y^2','x','y');
```

EXERCISE: Find and enter mu and for f

```
mu = 40;
L = 2;
```

Create figure

```
fig_pos_x = 300;
fig_pos_y = 200;
fig_size_x = 800;
fig_size_y = (fig_size_x/1.61803398875)*1.5;
fig = figure;
set(gcf, 'Color', 'w');
set(gcf, 'Position', [fig_pos_x fig_pos_y fig_size_x fig_size_y]);

if strcmp(main_method,'gradient-constant')
    set(fig,'Name','Gradient method, constant stepsize','NumberTitle','off')
else
    error('Unknown main method!');
end
```

Parameters for calculations and optimization methods

```
num_iter = 100;
```

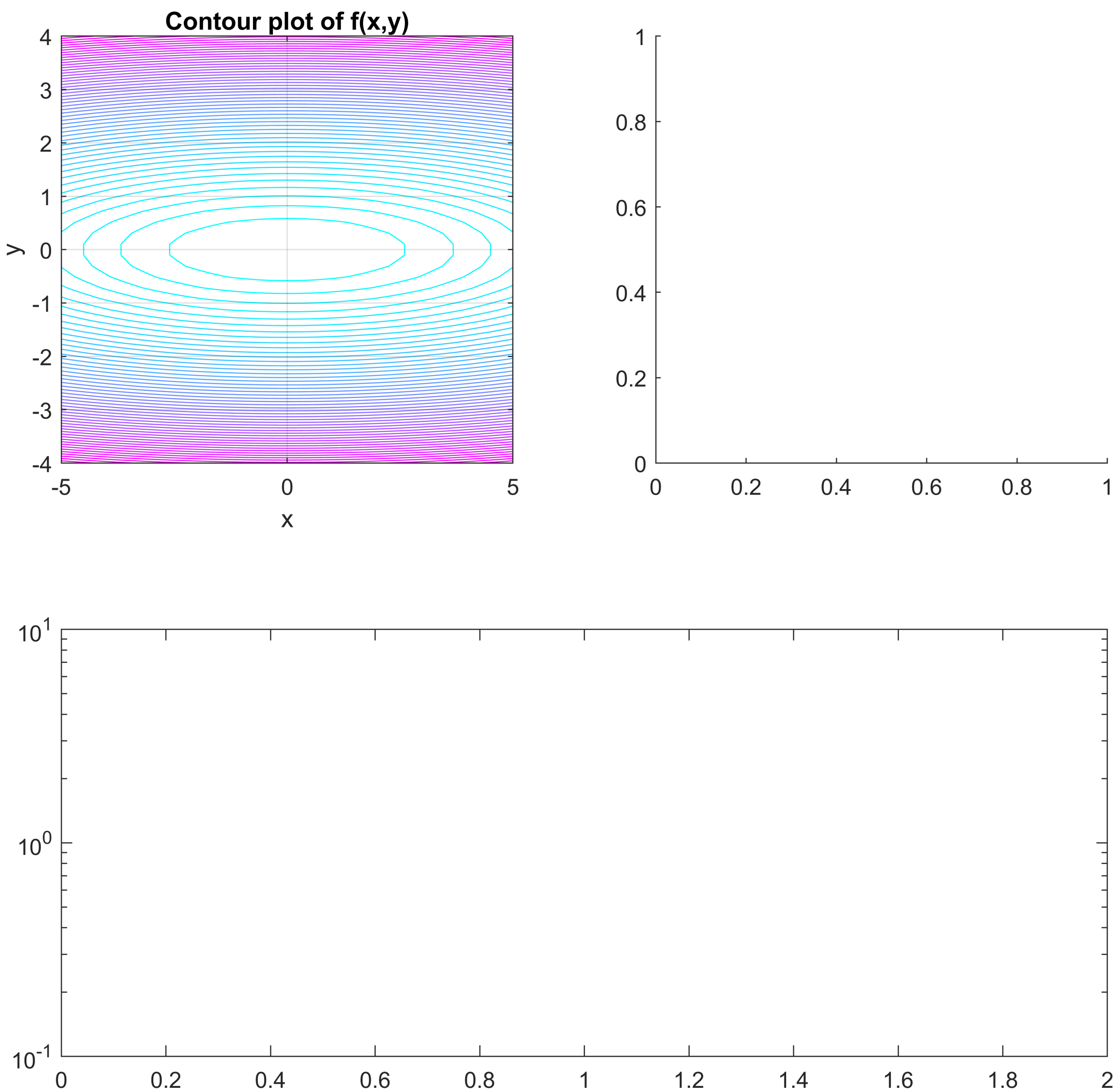
Create the two dimensional grid

```
x_min = -5;
x_max = 5;
x_num = 40;
y_min = -4;
y_max = 4;
y_num = 40;
x_vec = linspace(x_min, x_max, x_num);
y_vec = linspace(y_min, y_max, y_num);
[x_grid, y_grid] = meshgrid(x_vec, y_vec);

% Vectorize the function
```

```
f_vec = vectorize(f);
f_val_grid = f_vec(x_grid, y_grid);
```

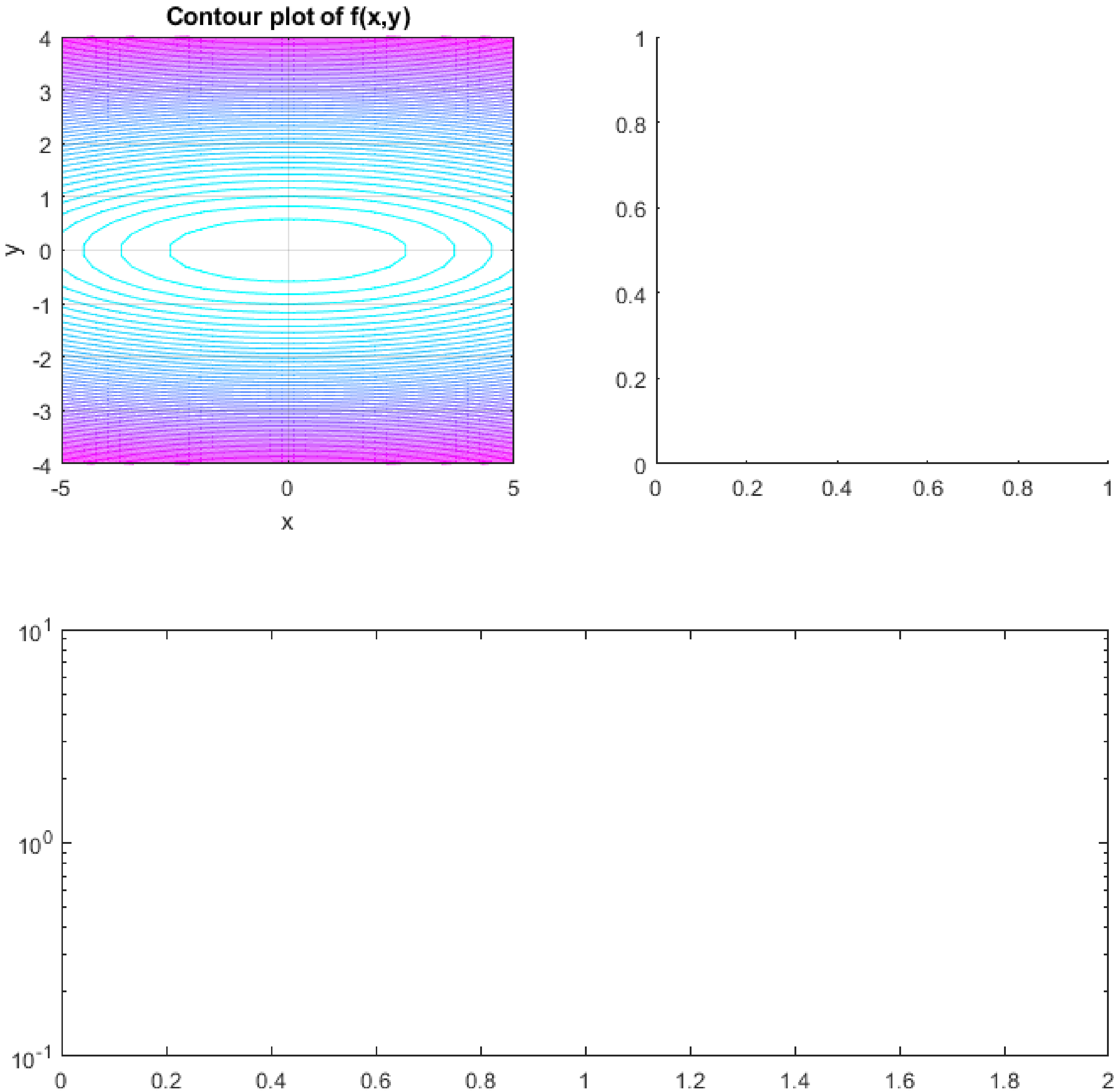
Prepare Plots

```
% Draw the contour plot
h_1 = subplot(2,2,1);
contour(x_vec,y_vec,f_val_grid,50);
title('Contour plot of f(x,y)')
xlabel('x')
ylabel('y')
colormap('cool');
grid
axis tight
axis manual
hold on;

% Surface Plot
h_2 = subplot(2,2,2);

% Draw the function value vs. iterations
h_3 = subplot(2,2,3:4);
semilogy(1,1)
hold on;
```



Get user input: location of point or exit command

```
subplot(h_1);
[x,y,button]=ginput(1);
```




```
if button == 3
    return;
end
xy_0 = [x; y];
```

EXERCISE: Enter the step sizes (values of gamma) to be compared. Choose epsn < 1e-3 if required.

```
epsn = 1e-4;
beta = ((sqrt(L) - sqrt(mu))/ (sqrt(L) + sqrt(mu))^2);
beta_n = sqrt(beta);
```

```
gamma1 = 2/40;
gamma2 = 2*(1+ beta) / (mu + L);
```

Main program loop

```
legendstr = [];
hline = [];

%for i = 1:ng

    % Storage for iterations
    fval_iter_vec = zeros(1,num_iter);
    x_iter_vec = zeros(2,num_iter);
    xy_prev = xy_0;
    xy = xy_0;
    prev_x_iter_vec = zeros(2,num_iter);

    for k = 1:num_iter

        % Save current iterate
        x = xy(1);
        y = xy(2);
        fval_iter_vec(k) = f(x,y);
        x_iter_vec(:,k) = xy;
```

EXERCISE: Complete the gradient vector for f(x,y) = x^2 + 20*y^2.

% % % x = xy(1); y = xy(2);

```
        %          grad_x = f_x(x,y)
        grad_x = 2*xy(1);
        %          grad_y = f_y(x,y)
        grad_y = 2*xy(2);
        grad_mat = [grad_x ; grad_y];
```

EXERCISE: Compute the Hessian matrix for f(x,y) = x^2 + 20*y^2.

% % % x = xy(1); y = xy(2);

```
        %          hxx = f_xx(x,y)
        hxx = 2;
        %          hyy = f_yy(x,y)
        hyy = 40;
        %          hxy = f_xy(x,y)
        hxy = 0;
        hessian_mat = [hxx hxy ; hxy hyy];
```

Run the optimization method

```
        if strcmp(main_method,'gradient-constant')
```

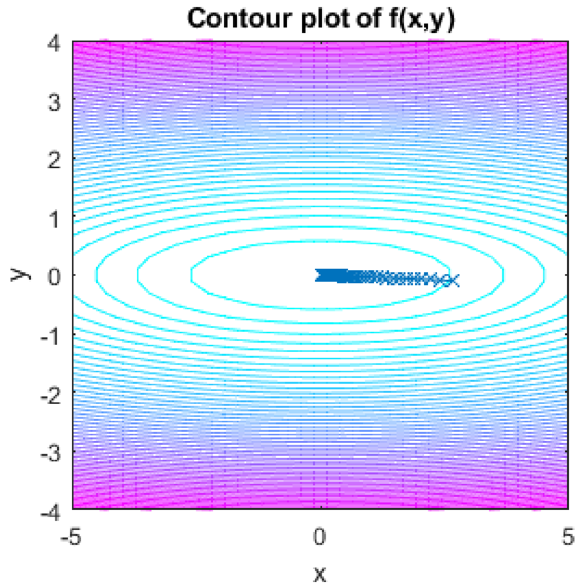
```
```

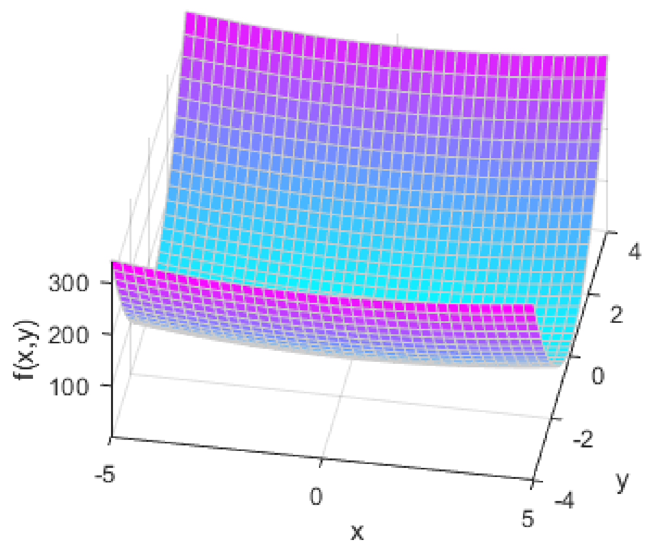
This is the gradient update equation

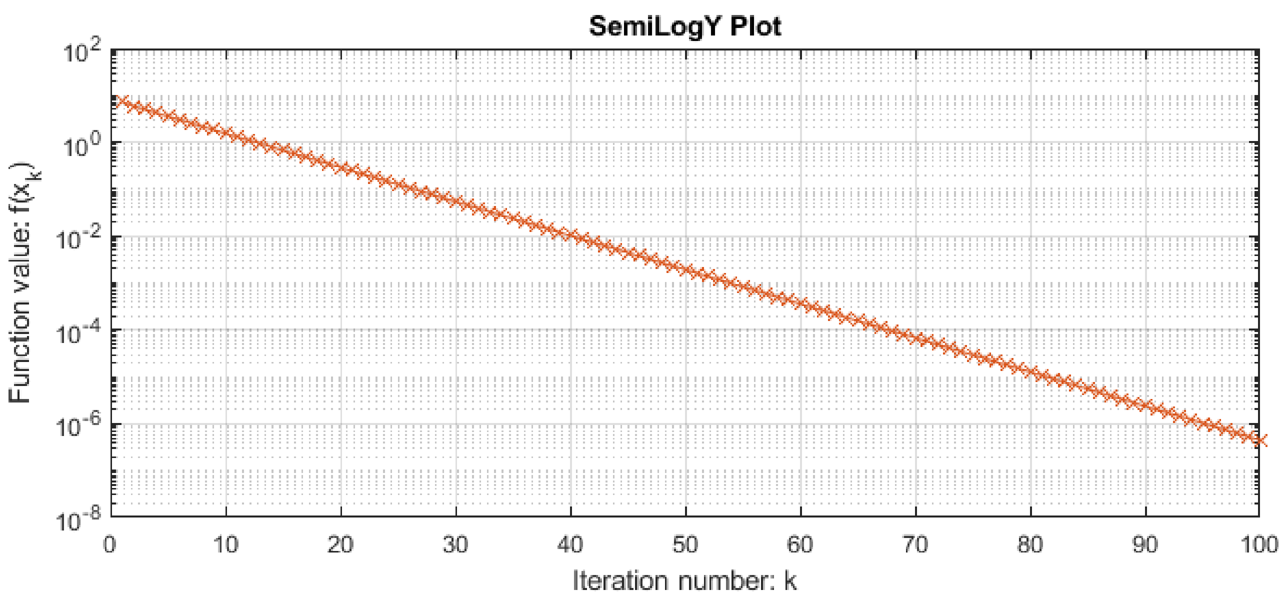
```
            if k < 2
                xy_prev = xy;
                xy = xy - gamma1*grad_mat;
            else
                xy_buf = xy;
                xy = xy - gamma2*grad_mat + beta*(xy - xy_prev);
                xy_prev = xy_buf;
            end
        end
    end
```

Plot the iterations and the result of the optimization

```
    subplot(h_1)
    contour(x_vec,y_vec,f_val_grid,50);
    title('Contour plot of f(x,y)')
    xlabel('x')
    ylabel('y')
    grid
    hold on
    plot(x_iter_vec(1,:),x_iter_vec(2,:),'x-');

    subplot(h_2)
    surface(x_vec,y_vec,f_val_grid,'EdgeColor',[.8 .8 .8]);
    hold on;
    view(10,55);
    colormap(cool);
    xlabel('x');
    ylabel('y');
    zlabel('f(x,y)');
    grid
    axis tight
    axis manual
    plot3(x_iter_vec(1,:),x_iter_vec(2,:),fval_iter_vec,'o');

    subplot(h_3)
    htemp = plot(1:num_iter,fval_iter_vec,'x-');
    hold on;
    title('SemiLogY Plot')
    xlabel('Iteration number: k');
    ylabel('Function value: f(x_k)')
    grid
```








```
%hline = [hline,htemp];
%tempstr = strcat('\gamma = ',num2str(gamma,'%10.4e\n'));
%legendstr = [legendstr; tempstr];
%legendin = cellstr(legendstr);
%legend(hline,legendin);
```