Assignment-4 Report

Name: Mohd Darish Khan
Roll: 2101MC29

Ques 1:

Code :

```
% Define the symbolic variables
syms x y lambda

% Define the objective function
f(x, y) = x + y;

% Define the constraint function
g(x, y) = x^2 + y^2 - 1;

% Define the Lagrange function
L = f - lambda * g;

% Compute the gradients of the Lagrange function with respect to x,
y, and lambda
grad_L_x = diff(L, x);
grad_L_y = diff(L, y);
grad_L_lambda = diff(L, lambda);

% Set up the system of equations
equations = [grad_L_x == 0, grad_L_y == 0, grad_L_lambda == 0];

% Solve the system of equations
solutions = solve(equations, [x, y, lambda]);

% Extract the solutions
x_sol = double(solutions.x);
y_sol = double(solutions.y);

% Evaluate the objective function at the solutions
f_values = f(x_sol, y_sol);

% Display the results
disp('Solutions (x, y):');
disp([x_sol, y_sol]);
disp('Objective function values at these points:');
disp(f_values);

% Find the maximum and minimum values
max_value = max(f_values);
min_value = min(f_values);
```

```
disp('Maximum value of f(x, y):');
disp(max_value);
disp('Minimum value of f(x, y):');
disp(min_value);
```

Output:

```
Solutions (x, y):
   -0.7071   -0.7071
    0.7071    0.7071

Objective function values at these points:
-2^(1/2)
 2^(1/2)

Maximum value of f(x, y):
2^(1/2)

Minimum value of f(x, y):
-2^(1/2)
```

Ques 2:
Code :

```
% Define the symbolic variables
syms x y z lambda

% Define the objective function
f(x, y, z) = 3*x^2 + y^2 - 2*z^2;

% Define the constraint function
g(x, y, z) = 3*x + 2*y - 8*z + 50;

% Define the Lagrange function
L = f - lambda * g;

% Compute the gradients of the Lagrange function with respect to x,
y, and lambda
grad_L_x = diff(L, x);
grad_L_y = diff(L, y);
grad_L_z = diff(L, z);
grad_L_lambda = diff(L, lambda);

% Set up the system of equations
equations = [grad_L_x == 0, grad_L_y == 0, grad_L_z == 0,
grad_L_lambda == 0];

% Solve the system of equations
solutions = solve(equations, [x, y, z, lambda]);

% Extract the solutions
x_sol = double(solutions.x);
y_sol = double(solutions.y);
z_sol = double(solutions.z);

% Evaluate the objective function at the solutions
f_values = f(x_sol, y_sol, z_sol);

% Display the results
disp('Solutions (x, y, z):');
disp([x_sol, y_sol, z_sol]);
disp('Objective function values at these points:');
disp(f_values);

% Find the maximum and minimum values
max_value = max(f_values);
min_value = min(f_values);

disp('Maximum value of f(x, y, z):');
disp(max_value);
disp('Minimum value of f(x, y, z):');
disp(min_value);
```

Output:

```
Solutions (x, y, z):
     2     4     8

Objective function values at these points:
-100

Maximum value of f(x, y, z):
-100

Minimum value of f(x, y, z):
-100
```

Ques 3:
Code :
```
% Define the symbolic variables
syms x y z lambda

% Define the objective function
f(x, y, z) = 4*x + 2*y + 6*z;

% Define the constraint function
g(x, y, z) = x^2 + y^2 + z^2 -14;

% Define the Lagrange function
L = f - lambda * g;

% Compute the gradients of the Lagrange function with respect to x,
y, and lambda
grad_L_x = diff(L, x);
grad_L_y = diff(L, y);
grad_L_z = diff(L, z);
grad_L_lambda = diff(L, lambda);

% Set up the system of equations
equations = [grad_L_x == 0, grad_L_y == 0, grad_L_z == 0,
grad_L_lambda == 0];

% Solve the system of equations
solutions = solve(equations, [x, y, z, lambda]);

% Extract the solutions
x_sol = double(solutions.x);
y_sol = double(solutions.y);
z_sol = double(solutions.z);

% Evaluate the objective function at the solutions
f_values = f(x_sol, y_sol, z_sol);

% Display the results
disp('Solutions (x, y, z):');
disp([x_sol, y_sol, z_sol]);
disp('Objective function values at these points:');
disp(f_values);

% Find the maximum and minimum values
max_value = max(f_values);
min_value = min(f_values);

disp('Maximum value of f(x, y, z):');
disp(max_value);
disp('Minimum value of f(x, y, z):');
disp(min_value);
```

Output:

```
Solutions (x, y, z):
    -2    -1    -3
     2     1     3

Objective function values at these points:
-28
 28

Maximum value of f(x, y, z):
28

Minimum value of f(x, y, z):
-28
```

Ques 4:
Code :

```
% Define the symbolic variables
syms x y z lambda

% Define the objective function
f(x, y) = 8*x^2 - 2*y;

% Define the constraint function
g(x, y) = x^2 + y^2 - 1;

% Define the Lagrange function
L = f - lambda * g;

% Compute the gradients of the Lagrange function with respect to x,
y, and lambda
grad_L_x = diff(L, x);
grad_L_y = diff(L, y);
grad_L_lambda = diff(L, lambda);

% Set up the system of equations
equations = [grad_L_x == 0, grad_L_y == 0, grad_L_lambda == 0];

% Solve the system of equations
solutions = solve(equations, [x, y, z, lambda]);

% Extract the solutions
x_sol = double(solutions.x);
y_sol = double(solutions.y);

% Evaluate the objective function at the solutions
f_values = f(x_sol, y_sol);

% Display the results
disp('Solutions (x, y):');
disp([x_sol, y_sol]);
disp('Objective function values at these points:');
disp(f_values);

% Find the maximum and minimum values
max_value = max(f_values);
min_value = min(f_values);

disp('Maximum value of f(x, y):');
disp(max_value);
disp('Minimum value of f(x, y):');
disp(min_value);
```

Output:

```
Solutions (x, y):
   -0.9922   -0.1250
    0.9922   -0.1250
         0    1.0000
         0   -1.0000

Objective function values at these points:
65/8
65/8
  -2
   2

Maximum value of f(x, y):
65/8

Minimum value of f(x, y):
-2
```

Ques 5:
Code :
```
% Define symbolic variables
syms x y lambda1 lambda2

% Objective function
f = x^2 + y^2;

% Constraints
g1 = y - x - 1;
g2 = x^2 - y - 2;

% Gradients of the objective function and constraints
grad_f = gradient(f, [x, y]);      % Gradient of the objective
function
grad_g1 = gradient(g1, [x, y]);    % Gradient of g1
grad_g2 = gradient(g2, [x, y]);    % Gradient of g2

% Stationarity condition
stationarity_x = grad_f(1) + lambda1 * grad_g1(1) + lambda2 *
grad_g2(1);
stationarity_y = grad_f(2) + lambda1 * grad_g1(2) + lambda2 *
grad_g2(2);

% Complementary slackness conditions
comp_slack_1 = lambda1 * g1;
comp_slack_2 = lambda2 * g2;

% Solve the system of equations
solution = solve([stationarity_x == 0, stationarity_y == 0,
comp_slack_1 == 0, comp_slack_2 == 0, g1 <= 0, g2 <= 0, lambda1 >=
0, lambda2 >= 0], [x, y, lambda1, lambda2]);

% Display the solutions
disp('Solutions:');
fprintf("x = %f\n", solution.x);
fprintf("y = %f\n", solution.y);
fprintf("lambda1 = %f\n", solution.lambda1);
fprintf("lambda2 = %f\n", solution.lambda2);
fprintf("Minimum function value = %f\n", solution.x^2 +
solution.y^2);
```

Output:

```
Solutions:
x = 0.000000
y = 0.000000
lambda1 = 0.000000
lambda2 = 0.000000
Minimum function value = 0.000000
```

Ques 6:

Code :

```
% Define symbolic variables
syms x y lambda1 lambda2

% Objective function
f = x^2 + y^2;

% Constraints
g1 = x - 7;
g2 = y^2 - x + 4;

% Gradients of the objective function and constraints
grad_f = gradient(f, [x, y]);      % Gradient of the objective
function
grad_g1 = gradient(g1, [x, y]);    % Gradient of g1
grad_g2 = gradient(g2, [x, y]);    % Gradient of g2

% Stationarity condition
stationarity_x = grad_f(1) + lambda1 * grad_g1(1) + lambda2 *
grad_g2(1);
stationarity_y = grad_f(2) + lambda1 * grad_g1(2) + lambda2 *
grad_g2(2);

% Complementary slackness conditions
comp_slack_1 = lambda1 * g1;
comp_slack_2 = lambda2 * g2;

% Solve the system of equations
solution = solve([stationarity_x == 0, stationarity_y == 0,
comp_slack_1 == 0, comp_slack_2 == 0, g1 <= 0, g2 <= 0, lambda1 >=
0, lambda2 >= 0], [x, y, lambda1, lambda2]);

% Display the solutions
disp('Solutions:');
fprintf("x = %f\n", solution.x);
fprintf("y = %f\n", solution.y);
fprintf("lambda1 = %f\n", solution.lambda1);
fprintf("lambda2 = %f\n", solution.lambda2);
fprintf("Minimum function value = %f\n", solution.x^2 +
solution.y^2);
```

Output:

```
Solutions:
x = 4.000000
y = 0.000000
lambda1 = 0.000000
lambda2 = 8.000000
Minimum function value = 16.000000
```