1 Exercise PP. 5

Use the steepest descent method to find the minimum of the unconstrained objective function $U(x, y)$ given by:

$$U(x, y) = 2x + \frac{20}{xy} + \frac{y}{3}$$

Solve the problem analytically and compare with the numerical results. Analyse the performance of the method.

1.1 Analytical Solution

Second Partial Derivative Test

Suppose that $f(x, y)$ is a differentiable real function of two variables and its second partial derivatives exist and are continuous. The Hessian matrix H of f is the 2x2 matrix of its the second partial derivatives.

$$H(x, y) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Define $D(x, y) = det(H(x, y)) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2$. Suppose that $(x, y) = (a, b)$ is a critical point of $f(x, y)$. Then,

- if $D(a, b) > 0$ and $\frac{\partial^2 f}{\partial x^2} < 0$ then (a, b) is a local maximum.
- if $D(a, b) > 0$ and $\frac{\partial^2 f}{\partial x^2} > 0$ then (a, b) is a local minimum.
- if $D(a, b) < 0$ then (a, b) is a saddle point.
- if $D(a, b) = 0$ then the test failed and (a, b) can be a local maximum, local minimum, saddle point or neither.

First we need to calculate the gradient of the function U.

$$\nabla U(x, y) = \left(2 - \frac{20}{x^2 y}, \frac{1}{3} - \frac{20}{xy^2}\right)$$

The second step is to determine the critical points of the function. Critical points are points where the function is defined and its gradient is a zero vector or undefined.

$$\nabla U(x, y) = 0 \Rightarrow \begin{bmatrix} 2 - \frac{20}{x^2 y} \\ \frac{1}{3} - \frac{20}{xy^2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} x = \sqrt[3]{\frac{5}{3}} \\ y = 2 \cdot 5^{\frac{1}{3}} \cdot 3^{\frac{2}{3}} \end{cases} \Rightarrow \begin{cases} x \approx 1.18563 \\ y \approx 7.11379 \end{cases}$$

Now, lets calculate $D(x, y)$:

$$\frac{\partial^2 U}{\partial x^2} = \frac{40}{x^3 y} \qquad \frac{\partial^2 U}{\partial y^2} = \frac{40}{xy^3} \qquad \frac{\partial^2 U}{\partial x \partial y} = \frac{20}{x^2 y^2}$$

$$D(x,y) = \frac{1200}{x^4y^4}$$

Evaluating $D(x,y)$ and $\frac{\partial^2 U}{\partial x^2}$ at the critical point obtained above, the results are :

$$D\left(\sqrt[3]{\frac{5}{3}}, 2\cdot 5^{\frac{1}{3}}\cdot 3^{\frac{2}{3}}\right) = \frac{1}{\sqrt[3]{75}} > 0$$

$$\frac{\partial^2 U}{\partial x^2}\left(\sqrt[3]{\frac{5}{3}}, 2\cdot 5^{\frac{1}{3}}\cdot 3^{\frac{2}{3}}\right) = \frac{1}{3\sqrt[3]{5}\cdot 3^{\frac{2}{3}}} > 0$$

So $(x,y) = \left(\sqrt[3]{\frac{5}{3}}, 2\cdot 5^{\frac{1}{3}}\cdot 3^{\frac{2}{3}}\right) \approx (1.18563, 7.11379)$ is a local minimum of the function U.

1.2 Numerical Solution

A numerical solution to find the minimum of the function U can be achieved by the steepest descent method and to perform it the following MATLAB scripts were used.

1.2.1 MATLAB Files

PP5_main.m

```
% Main Script
% Calls: PP5_data.m, PP5_search, PP5_plot, armijo

clear, clc
close(figure(1))     % Close open figure

% Initialize Data
PP5_data

% Open Contour Plot Figure
figure(1)
% Draw Contours at increments of 0.3
fcontour(f,[0 3 3 8])
xlabel('x','Interpreter','latex')
ylabel('y','Interpreter','latex')
title('f(x,y) = 2x + $\frac{20}{xy}$ + $\frac{y}{3}$','Interpreter','latex')
hold on

% Search Cycle
```

```
t=0;
while t<tmax && norm(grad_f(x(1),x(2))) > precision
    t=t+1;

    % Select the search direction
    PP5_search

    % Determine the step size
    alpha=armijo(f,d,x,grad_f);

    % New search point
    x_old=x;
    f_old=f_obj;

    x=x+alpha*d;
    f_obj=f(x(1),x(2));

    % Plot the current search path
     PP5_plot
end

% Display the results
fprintf('Number of Iterations: %d\n\n', t);
fprintf('Point of Minima: [%d , %d]\n\n', x(1),x(2));
fprintf('Objective Function Minimum Value after Optimization: %d\n\n',f_obj);
```

PP5_data.m

```
% Objective Function
f   = @(x,y) 2.*x+20./(x.*y)+y./3;
grad_f = @(x,y) [2-20./(x.^2.*y), -20./(x.*y.^2)+1/3];

% Maximum number of iterations
tmax = 2000;

% Initial Point
x = [1 4];
f_obj(1) = f(x(1),x(2));

% Precision (may not be achieved if the necessary number of iterations is greater than the maximum defined above)
precision = 1E-8;
```

PP5_search.m

```
% Computes the search direction as defined in the Steepest
   Descent Method.
d=-grad_f(x(1),x(2));
```

PP5_plot.m

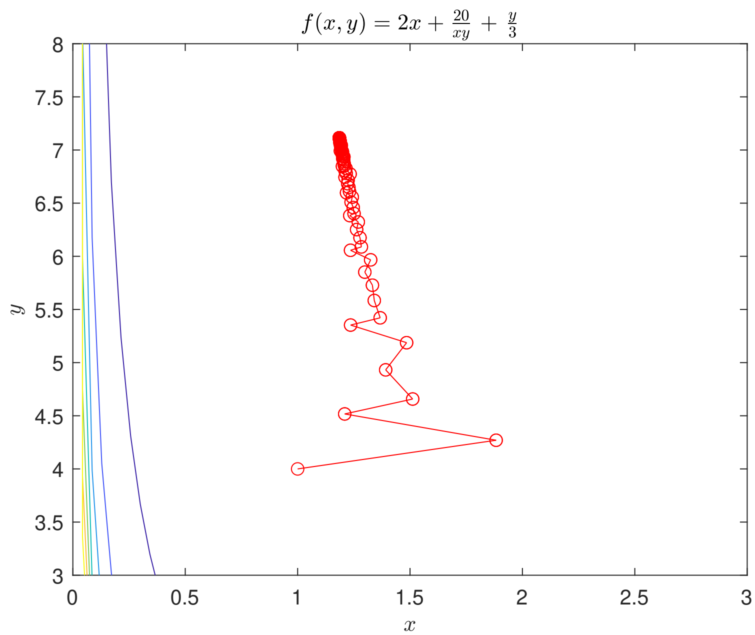
```
% Plots the iterated Points
x_coord = [x_old(1),x(1)];
y_coord = [x_old(2),x(2)];

plot(x_coord,y_coord,'o-r')
```

armijo.m

```
function [alpha]=armijo(f,d,x,grad_f)
% Armijo Linear Search
% f - objective function
% d - search direction
% x - search point
% grad_f - gradient of the objective function

delta=0.2967;
gamma=0.1;
c=0.9921;

% Initial guess for the step size alpha
a=c*abs(grad_f(x(1),x(2))*d.')/(norm(d))^2;

% Application of the Armijo Condition
while true
    if f(x(1)+a*d(1),x(2)+a*d(2))<=f(x(1),x(2))+gamma*a*grad_f(
       x(1),x(2))*d.'
        break
    else
        a=delta*a;
    end
end
alpha=a;
```

1.2.2 Results

```
Number of Iterations: 253
Point of Minima: [1.185631e+00 , 7.113786e+00]
Objective Function Minimum Value after Optimization: 7.113787e+00
```



1.3 Conclusions

The performance of the method is largely dependent in the initial search point, in the values δ, γ and c used in the *armijo.m* and in the precision defined in *PP5_data.m*.
In order to evaluate this differences, a modified version of *PP5_main.m* was used, where the plots and final output prints to the console were removed to improve speed. The initial search point was fixed (the same as the one in *PP5_data.m*), also the value γ was fixed since it was observed, initially, that, for this function, it doesn't have a big impact.

So the only values iterated through an exhaustive process were δ and c. For both values a linearly spaced vector of 500 elements was generated, respecting the values' domain, $(0, 1)$, and then a matrix was created with all the possible combinations of δ, γ and c. The last step was to introduce this values in the modified *PP5_main.m* and evaluate the the performance of the method for each combination of values, i.e., the number of iterations.

Using the process described above, the lowest number of iterations obtained was the one presented in the results above with $x_0 = (1, 4)$, $\delta = 0.2967$, $\gamma = 0.1$, $c = 0.9921$ and a $precision = 10^{-8}$. It was also noticed that the initial search point needs to be in the first quadrant, otherwise the algorithm won't work.

In conclusion, for the analysed function the analytical method is way better than the steepest descent method performed in the numerical solution, since the solution obtained by the latter is an approximation, which depends on various parameters as stated above, while the analytical one is an exact solution and, in terms of computation time and cost, is better. However, the analytical method isn't viable for all functions and a numerical method may be required.