MTRN4030: Optimisation Lab 2 Written Report

By: Dylan Dam z5115759

Task 1

1. Let the reduction ratio in the golden search method be ρ. Find the specific choice of the golden section ratio in order to reduce the number of objective function evaluations. For a given search range Δ and the accepted tolerance τ, show how the number of iterations N can be obtained.

For the golden ratio, two boundaries need to be set for the search range, let them be a_0 and b_0.

$$b_0 - a_o = \Delta$$

Two intermediate points need to be chosen such that:

$$a_1 - a_0 = b_0 - b_1 = \rho(b_o - a_0) \; where \; \rho < 0.5$$

Each iteration the search range reduces by a factor of ρ

$$\rho(b_1 - a_0) = b_1 - b_2$$

$$b_1 - a_0 = (1 - \rho) * \Delta$$

$$\rho(1 - \rho) = 1 - 2\rho$$

$$\rho^2 - 3\rho + 1 = 0$$

Using the quadratic formula, we obtain two results:

$$\rho_1 = \frac{3 + \sqrt{5}}{2}$$

$$\rho_2 = \frac{3 - \sqrt{5}}{2}$$

We know that $\rho < 0.5$, therefore p_2 is chosen as the reduction ratio.

When given a tolerance, τ the number of iterations can be calculated via:

We know that the Golden ratio reduces the search range by a factor of 0.618 per iteration.

Therefore $(1 - \rho)^N * \Delta \approx 0.618^N * \Delta$

Iterations continues until the search range is less than the tolerance, it will need to satisfy the following inequality:

$$\tau > (0.618)^N * \Delta$$

$$\frac{\tau}{\Delta} > (0.618)^N$$

$$N > \frac{\ln\left(\frac{\tau}{\Delta}\right)}{\ln(0.618)}$$

2. Given the objective function $f(\boldsymbol{x})$ of an optimization problem, assume it is continuous where derivatives exist, give a Taylor series expansion about a point x_n up to the quadratic (2-nd derivative) term.

Using the Taylor series expansion and omitting all terms higher than second order we get:

$$f(x) \approx q(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(x_n)}{2!}(x - x_n)^2$$

3. Obtain the derivative of the series with respect to the infinitesimal change Δx = $\boldsymbol{x} - x_n$, and give the expression in terms of the Hessian matrix.

Using 2:

$$q'(x) = f'(x_n) + f''(x_n)(x - x_n)$$

$$g_n = \nabla f(x_n) = f'(x_n)$$

$$q'(x) = g_n + F(x_n) * \Delta x$$

4. Give the iterative equation to obtain the optimum solution x^* in the form of Newton's method.

Optimal solution occurs when $\nabla q(x) = 0$

$$0 = g_n + F(x_n) * \Delta x$$

$$-g_n = F(x_n) * \Delta x$$

$$\Delta x = -g_n * F(x_n)^{-1}$$

$$x^* - x_n = -g_n * F(x_n)^{-1}$$

$$x^* = x_n - g_n * F(x_n)^{-1}$$

5. Formulate the steepest descent optimization algorithm in an iterative expression including a control coefficient α on the step size.

Steepest descent relies in a theorem that the gradient would always point in the direction of the optimal solution. The form can be written as:

$$x_{n+1} = x_n \pm \alpha * \nabla f(x_n)$$

Where α is the step size and $\nabla f(x_n)$ is the directional vector. In steepest descent optimisation, the step size needs to be optimised so that the solution moves towards the solution quickly. The sign of $\alpha * \nabla f(x_n)$ determines if it is maximising or minimising. α is determined by finding the $argmin\, f(x^k - \alpha \nabla f(x^k))$. The iterative process is finding the gradient, calculating the optimal step size, find the next point and repeat.

Task 2

Verify your results by calculating the gradients, ∂f/∂x , ∂f/ then solve for the optimum design variables $x*$, $y*$, based on the first-order necessary condition, ∇f = 0. Verify that the solution is a minimum by the second-order necessary condition ∇^2> 0, i.e., the Hessian is positive definite. Put your answer in the written report.

Objective function

$$f(x, y) = 10(x-2)^2 + xy + 10(y-1)^2$$

$$\nabla f = 0$$

Gradient vector is:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 20(x-2) + y \\ x + 20(y-1) \end{bmatrix} = 0 \qquad \begin{matrix} (1) \\ (2) \end{matrix}$$

This forms two equation that we'll need to simultaneously solve in order to find the optimum solution.

$$From\ (1): y = -20(x-2) \quad (3)$$

$$Sub\ (3)\ into\ (2)$$

$$x + 20(-20(x-2) - 1) = 0$$

$$x^* = \frac{780}{399}$$

$$Sub\ x^*\ into\ (3)$$

$$y = -20\left(\frac{780}{399} - 2\right) = \frac{120}{133}$$

$$y^* = \frac{120}{133}$$

To verify that the Hessian matrix we need to determine that its eigenvector is > 0.

First, we need to construct the Hessian by further partial differentiating the gradient.

$$\frac{\partial^2 f}{\partial x^2} = 20$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = 1$$

$$\frac{\partial^2 f}{\partial y^2} = 20$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 20 & 1 \\ 1 & 20 \end{bmatrix}$$

$$\det \begin{bmatrix} 20 - \lambda & 1 \\ 1 & 20 - \quad \lambda \end{bmatrix}$$

$$(20 - \lambda)^2 - 1^2$$

$$\lambda^2 + 40\lambda - 399$$

$$(\lambda - 19)(\lambda - 21)$$

Since both eigenvectors (19 and 21) are positive, the Hessian is positive definite since eigenvalues are both positive.

Task 3

Objective Function

$$f(x, y) = (y - x^2)^2 + (1 - x)^2$$

Partial derivatives

$$\frac{\partial f}{\partial x} = 2x - 4x(-x^2 + y) - 2 = 0$$

$$\frac{\partial f}{\partial y} = -2x^2 + 2y = 0$$

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x - 4x(-x^2 + y) - 2 \\ -2x^2 + 2y \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 12x^2 - 4y + 2 & -4x \\ -4x & 2 \end{bmatrix}$$

Objective function

$$f(x, y) = (x - 1)^2 + (y - 2)^2$$

Partial derivatives

$$\frac{\partial f}{\partial x} = 2(x - 1) = 0$$

$$\frac{\partial f}{\partial y} = 2(y - 2) = 0$$

Gradient vector is:

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2(x - 1) \\ 2(y - 2) \end{bmatrix}$$

To calculate the Hessian, further partial differentiation

$$\frac{\partial^2 f}{\partial x^2} = 2$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = 0$$

$$\frac{\partial^2 f}{\partial y^2} = 2$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

As both eigenvalues are positive, therefore the function is positive definite and convex. When comparing the two Hessians of both objective functions, objective function 2 has a much simpler Hessian compared to objective function 1. Through the simpler Hessian function this reduces the iterations needed to find the optimal solution since objective function has x and y terms, this means the Hessian is dependent on the current point which would take longer compared to if the Hessian was independent from the point.

Task 4

Fibonacci and Bisection method were used to find the optimal solution for an objective function. The resulting estimations and the number of iterations for each method can be seen below.

```
Fibonacci's Iteration:
    10

Fibonacci's Lambda:
    0.4804    0.4806

Bisection's Iteration:
     5

Bisection's Lambda:
    0.4840
```

Figure 1: Results from Fibonacci and Bisection Optimisation

The Bisection method takes 5 iterations to find the solution whereas Fibonacci takes 10 iterations to find the solution. The property of Fibonacci is that the search range gets reduced to 0.618 of its original length up until the last iteration where the range is halved. This is inefficient compared to the Bisection method as the search range is continually halved after each iteration. Bisection method is more requires more computation as the objective function's derivative needs to be calculated.

Task 5

Number of Points in Mesh	Optimal Point (d)	Optimal Point (t)	Optimal Point (f)
100	6.000	0.267	27.712
10000	5.516	0.291	26.786
40000	5.498	0.291	26.675
90000	5.451	0.292	26.551
1000000	5.460	0.292	26.568

Table 1: Comparison Between Results and Resolution

From table 1 we can see that as the number of points increase, the quality of the solution diverges closer and closer to the optimal point. As the functions we are trying to optimise are continuous functions, the "true" solution may be computationally expensive to calculate. Therefore, we accept these approximations, but we have to ensure the solution contains a high enough resolution where the approximation can be obtained accurately. When comparing the 100-point mesh to the 1000000-point mesh we can see the change in d is 0.54 for d, a change in 0.25 for t and a 1.14 change for f. When comparing 10000 and 1000000 we can see that the values across d,t and f do not differ that much. Therefore, when picking a resolution, it is important to pick a resolution that adequately represents the function without diluting the sample space.