

MTRN4030 Exercise 2

Zachary Hamid

October 2019

1 Part 1 - Line Search and Gradient-based Optimization

1.1 Task 1

1. Assuming we start with some search range $[a_0, b_0]$ and a unimodal objective function, f , we choose two new points, a_1 and b_1 to evaluate the function f at in order to determine where the minimum (or maximum) is located. These points are chosen such that the reduction in range is symmetric, therefore:

$$a_1 - a_0 = b_1 - b_0 = \rho(b_0 - a_0), \rho < 0.5 \quad (1)$$

So for only one new evaluation of f , we choose ρ such that:

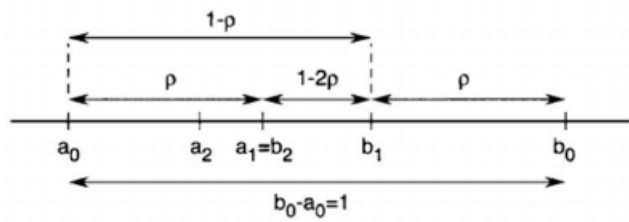


Figure 1: Finding value of ρ resulting in only one new evaluation of f [CITATION NEEDED]

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

Since $b_1 - a_0 = 1 - \rho$, and $b_1 - b_2 = 1 - 2\rho$:

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

We then get a quadratic equation from which we can solve for ρ :

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

Since it is required that $\rho < 0.5$ the solution that is taken from this quadratic is:

$$\rho = (3\sqrt{5})/2 \approx 0.382.$$

2. Given an objective function $f(\mathbf{x})$ with the assumption that the functions is continuous where derivatives exist, the Taylor series expansion about a point \mathbf{x}_n up to the quadratic term can be given as:

$$f(\mathbf{x}) \approx f(\mathbf{x}_n) + (\mathbf{x} - \mathbf{x}_n) \nabla f(\mathbf{x}_n) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_n)^T \nabla^2 f(\mathbf{x}_n) (\mathbf{x} - \mathbf{x}_n)$$

Where $\nabla f(\mathbf{x}_n)$ is the gradient of the function $f(\mathbf{x})$ at the point \mathbf{x}_n , and $\nabla^2 f(\mathbf{x}_n)$ is the Hessian matrix of the function $f(\mathbf{x})$ at the point \mathbf{x}_n .

3. From the Taylor series expansion given above setting $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_n$ we have:

$$f(\mathbf{x}) \approx f(\mathbf{x}_n) + \Delta \mathbf{x} \nabla f(\mathbf{x}_n) + \frac{1}{2} \Delta \mathbf{x}^T \nabla^2 f(\mathbf{x}_n) \Delta \mathbf{x} \quad (2)$$

From this, the derivative with respect to the infinitesimal change $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_n$ can be found to be:

$$\begin{aligned} \frac{df(\mathbf{x})}{d\Delta \mathbf{x}} &= \nabla f(\mathbf{x}_n) + 0.5 \frac{d(\Delta \mathbf{x}^T \nabla^2 f(\mathbf{x}_n) \Delta \mathbf{x})}{d\Delta \mathbf{x}} \\ &= \nabla f(\mathbf{x}_n) + 0.5 (2 \nabla^2 f(\mathbf{x}_n) \Delta \mathbf{x}) \\ &= \nabla f(\mathbf{x}_n) + \nabla^2 f(\mathbf{x}_n) \Delta \mathbf{x} \end{aligned}$$

4. Applying the first order necessary condition (FONC) to the derivative of the Taylor series expansion of objective function $f(\mathbf{x})$, given in 3. above:

$$\frac{df(\mathbf{x})}{d\Delta \mathbf{x}} = \mathbf{0} = \nabla f(\mathbf{x}_n) + \nabla^2 f(\mathbf{x}_n) \Delta \mathbf{x} \quad (3)$$

Rearranging for \mathbf{x} , we get an iterative formula that finds the optimal value for \mathbf{x} (ie. \mathbf{x}^*):

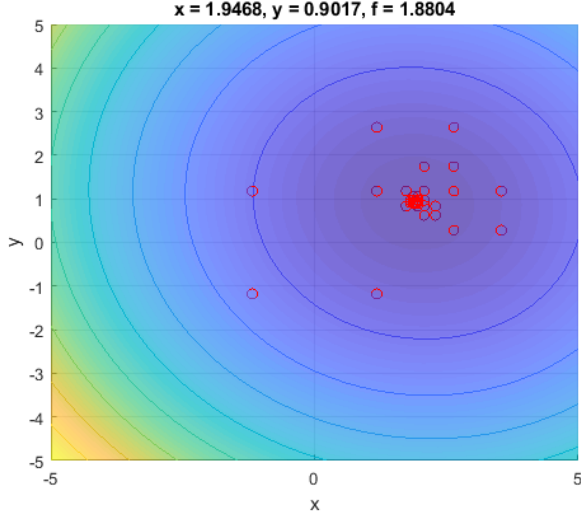
$$\mathbf{x}_{n+1} = \mathbf{x}_n - (\nabla^2 f(\mathbf{x}_n))^{-1} \nabla f(\mathbf{x}_n) \quad (4)$$

The iterative formula above is known as Newton's Method.

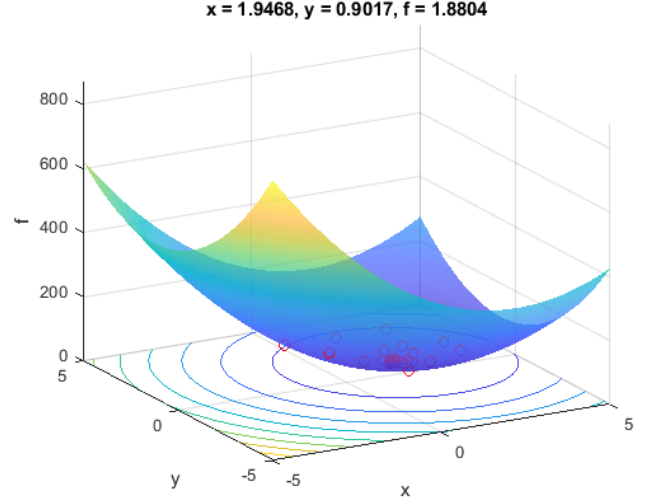
5.

1.2 Task 2

The solution derived by the 2-dimensional Golden Ratio search method using Matlab is shown in the figures below:



(a) x-y view of solution



(b) 3D view of function

Figure 2: Solution to task 2 using 2-dimensional Golden Ratio search algorithm

This solution can be verified by inspection of the characteristics of both the gradient vector and Hessian matrix of the function:

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

The gradient vector was found to be:

$$\nabla f = \begin{bmatrix} \delta f / \delta x \\ \delta f / \delta y \end{bmatrix} = \begin{bmatrix} 20x + y - 40 \\ x + 20y - 20 \end{bmatrix}$$

To locate the optimum design variables x^* , y^* , the first-order condition of $\nabla f = 0$ must be satisfied.

$$\frac{\delta f}{\delta x} = 20x + y - 40 = 0 \quad (5)$$

$$\frac{\delta f}{\delta y} = x + 20y - 20 = 0 \quad (6)$$

Rearranging Eq. 5 gives:

$$y = 40 - 20x \quad (7)$$

Substituting Eq. 7 into Eq. 6 and solving for x gives:

$$x = \frac{780}{399} \approx 1.95489$$

Substituting the above result into Eq. 7 gives:

$$y = \frac{120}{133} \approx 0.90226$$

Therefore an optimum (x^*, y^*) occurs at $(1.95489, 0.90226)$. To determine if this is a minimum (as required by the task description), the Hessian matrix characteristics must be inspected. The Hessian matrix can be defined as:

$$H = \nabla^2 f = \begin{bmatrix} \delta f / \delta x^2 & \delta f / \delta x \delta y \\ \delta f / \delta y \delta x & \delta f / \delta y^2 \end{bmatrix} = \begin{bmatrix} 20 & 1 \\ 1 & 20 \end{bmatrix}$$

The condition for the optimum point (x^*, y^*) to be a minimum is the Hessian matrix must be positive definite, that is, $\nabla^2 f > 0$. The method of principal minors can be used to determine this. This method looks at the determinants of the leading principal submatrices of the Hessian matrix. If these are all positive, the Hessian is positive definite.

$$A_0 = |20| = 20 > 0$$

$$A_1 = \begin{vmatrix} 20 & 1 \\ 1 & 20 \end{vmatrix} = 20 * 20 - 1 * 1 = 400 - 1 = 399 > 0$$

Since both A_0 and A_1 are positive, the Hessian matrix H is positive definite and therefore the optimum (x^*, y^*) found above is a minimum. Substituting this optimal value into the original function gives the minimum function value:

$$f(1.95489, 0.90226) \approx 1.8797$$

It can be seen that these values of x^*, y^* and f are all very similar to what was found using the Golden Section search method.

1.3 Task 3

Applying Newton's Method to the function given in question 6 of this task resulted in the minimum being found within a single iteration. To determine why this is the case, the objective function is first expanded:

$$f(x, y) = (x - 1)^2 + (y - 2)^2 = x^2 + y^2 - 2x - 4y + 5$$

This function can then be reformulated as:

$$f(x, y) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} - \mathbf{x}^T \mathbf{b} + c \quad (8)$$

Where:

$$\mathbf{Q} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, c = 5$$

In order for Newton's Method to converge from any initial point to the optimum in a single iteration, a few criteria must be met. Firstly, it is required that the gradient vector, $\mathbf{g} = \nabla f(x, y) = \mathbf{Q} \mathbf{x} - \mathbf{b}$:

$$\nabla f(x, y) = \begin{bmatrix} 2x - 2 \\ 2y - 4 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \mathbf{Q} \mathbf{x} - \mathbf{b}$$

Additionally, the Hessian matrix, \mathbf{H} must be equal to \mathbf{Q} :

$$\mathbf{H} = \nabla^2 f(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \mathbf{Q}$$

Finally, \mathbf{Q} needs to be symmetric and invertible:

$$\mathbf{Q}^T = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^T = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \mathbf{Q}$$

$$\mathbf{Q}^{-1} = \frac{1}{4} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

Now, from Newton's Method we have:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{H}(\mathbf{x}^k)^{-1} \mathbf{g}^k$$

Substituting the results of $\mathbf{H} = \mathbf{Q}$ and $\mathbf{g} = \mathbf{Q}\mathbf{x} - \mathbf{b}$ in and setting $k = 0$ since it is the first iteration:

$$\begin{aligned} \mathbf{x}^1 &= \mathbf{x}^0 - \mathbf{H}(\mathbf{x}^0)^{-1} \mathbf{g}^0 \\ &= \mathbf{x}^0 - \mathbf{Q}^{-1}[\mathbf{Q}\mathbf{x}^0 - \mathbf{b}] = \mathbf{Q}^{-1}\mathbf{b} \end{aligned}$$

The optimal point \mathbf{x}^* occurs when $\mathbf{g} = \nabla f(x, y) = \mathbf{0}$, therefore:

$$\begin{aligned} \mathbf{Q}\mathbf{x} - \mathbf{b} &= \mathbf{0} \\ \mathbf{Q}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &= \mathbf{Q}^{-1}\mathbf{b} = \mathbf{x}^* \end{aligned}$$

So it can be seen that, on iteration 1 (ie. when calculating \mathbf{x}^1) the result is the optimum \mathbf{x}^* , regardless of which initial point \mathbf{x}^0 is used. From this result, the exact point where the minimum occurs can be calculated for the objective function given in Eq. 8, and is given below:

$$\mathbf{x}^1 = \mathbf{x}^* = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Substituting this result into the original objective function yields the minimum function value:

$$f(1, 2) = (1 - 1)^2 + (2 - 2)^2 = 0$$

This result is consistent with the results shown in question 6 of the exercise 2 document.

2 Part 2 - Mechanical System Design

2.1 Task 4

The bracketing method employed for this task resulted in an initial search range of $\lambda \in [0.31, 0.91]$ for each method. From this, the number of iterations to find the optimal value of λ as well as the maximum function value corresponding to this λ is shown in the table below:

	Fibonacci	Bisection
Number of Iterations	3	5
Maximum Value $f(\lambda)$	0.3076	0.31

It can be seen that the Fibonacci method obtained the approximate optimal value in fewer iterations.

In terms of implementation complexity, each method has their own drawbacks, with the Fibonacci Method having computational overhead in terms of needing to compute the Fibonacci sequence and corresponding sequence of values for ρ . In addition to this, the algorithm itself is slightly more complex than the Bisection method, requiring more variable reassignments and recalculations. The Bisection method requires only recalculating a single value of x at each iteration. However, the drawback of the Bisection method is that it also requires the calculation of the function's derivative and can therefore be too computationally expensive for complex functions, and even impossible to apply in some instances.

2.2 Task 5

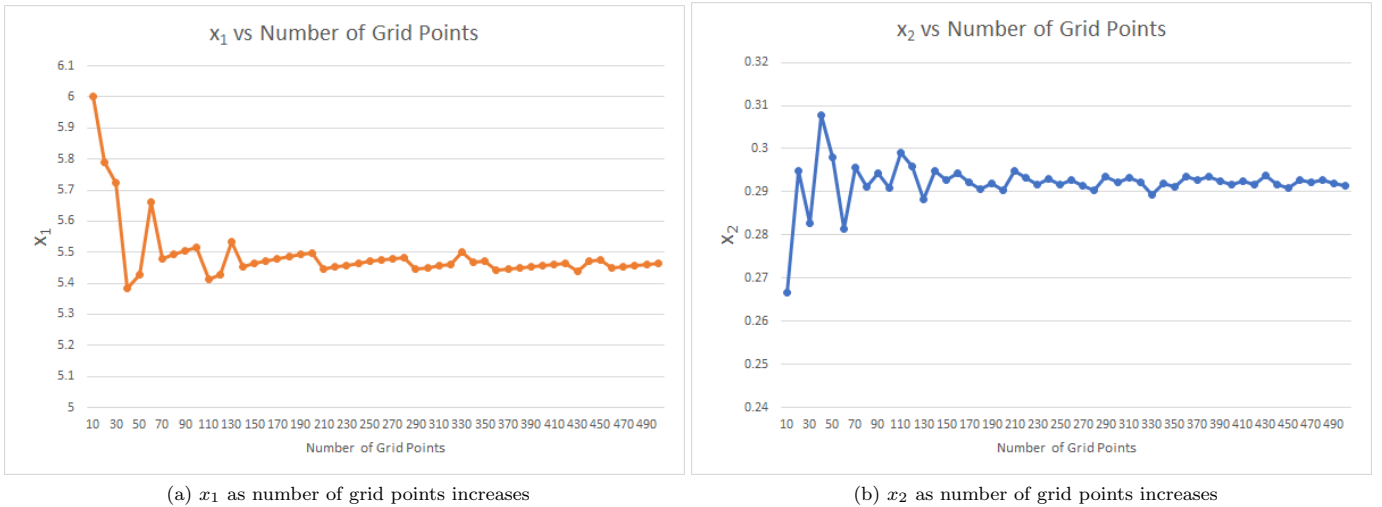


Figure 3: x_1 and x_2 of found minimum as number of grid points increases

It can be seen from Fig. 3 above that with an increasing number of grid points, there is a corresponding increase in accuracy as x_1 and x_2 oscillate closer and closer to their true values. This is intuitive as with an increase in number of grid points, there are more points to search from within the same space

of $2 \leq x_1 \leq 14$, and $0.2 \leq x_2 \leq 0.8$ as specified in the constraints of the design problem. This means that the chance that a grid point lay close to the true minimum solution increases with an increase in the number of grid points, which is an intuitive and expected result. The cost of this, is an increase in computation time to find the solution as there are more points to search through.