Numerical Optimization

Dimitrios Nentidis

November 2023

1 Introduction

In this first assignment the task was to implement three different methods to minimize the following function.

$$f(x) = \sum_{i=1}^{n-1} (x_i + x_{i+1} - 3)^2 + (x_i - x_{i+1} + 1)^4$$

For initial conditions of $X_0 = [2, 2, ..., 2]$.

2 For $n = 2$

For the specific case when n=2 the function takes the form of $f(x) = (x_1 + x_2 - 3)^2 + (x_1 - x_2 + 1)^4$. The graphical representation of this function is illustrated in fig. 1, where the point marked corresponds to the initial point $X_0 = [2, 2, 2]$. Notably, the function is convex.

The absolute minimum value of f(x) is achieved when $x_1 = 1$ and $x_2 = 2$, resulting in $f(x) = 0$. This minimum point is highlighted on the plot. The convex nature of the function indicates that it has a single, global minimum, making optimization relatively straightforward.

For the Gradient only methods, Steepest Descent and Conjugate Gradient, convergence was not an issue. Newton's method did not converge without some modifications to add

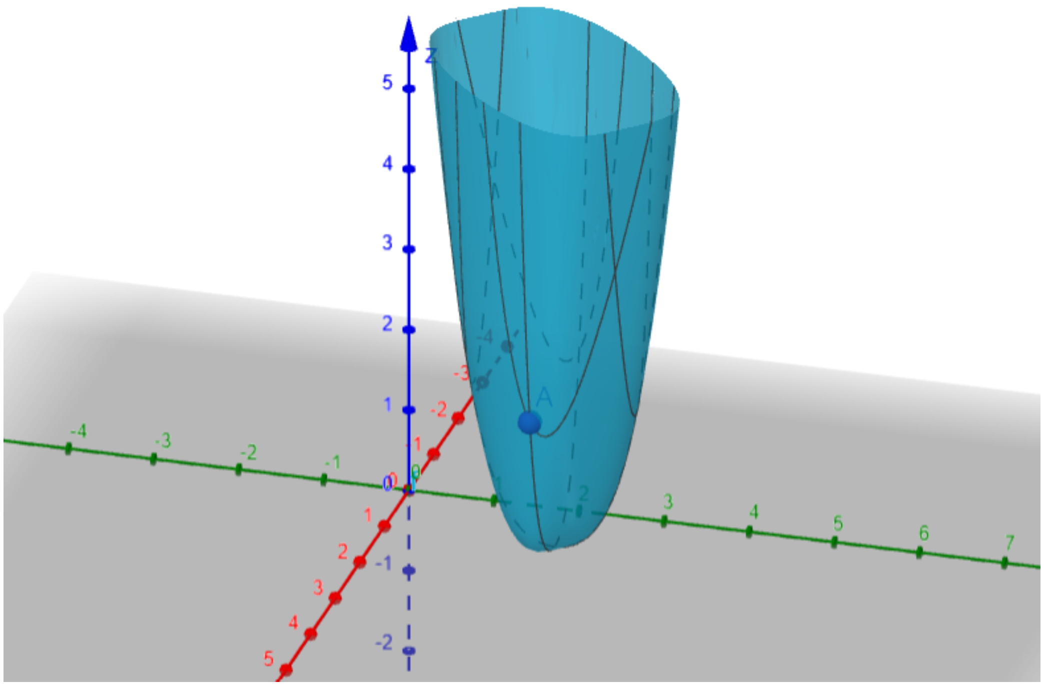

Figure 1: Generalized Tridiagonal 1 function, for n=2.

2.1 Gradient only based algorithms

The Steepest Descent Method exhibited convergence after a substantial 905 iterations, revealing the optimal solution of $x_1 = 1.013$ and $x_2 = 1.987$. The resulting optimal objective function value was at 4.5881×10^{-7}.

As depicted in fig. 6, the method takes numerous small steps before finally converging to the minimum. Notably, altering the initial conditions, whether closer or further from the global minimum, did not significantly impact the algorithm's performance. The method consistently converged within the set tolerance of $tol = 1 \times 10^{-4}$, and even with different initial conditions, it reliably reached convergence in under 1000 iterations.

To maintain consistency, a gamma value of 0.1 was selected, as it had proven effective for different number of parameters, n.

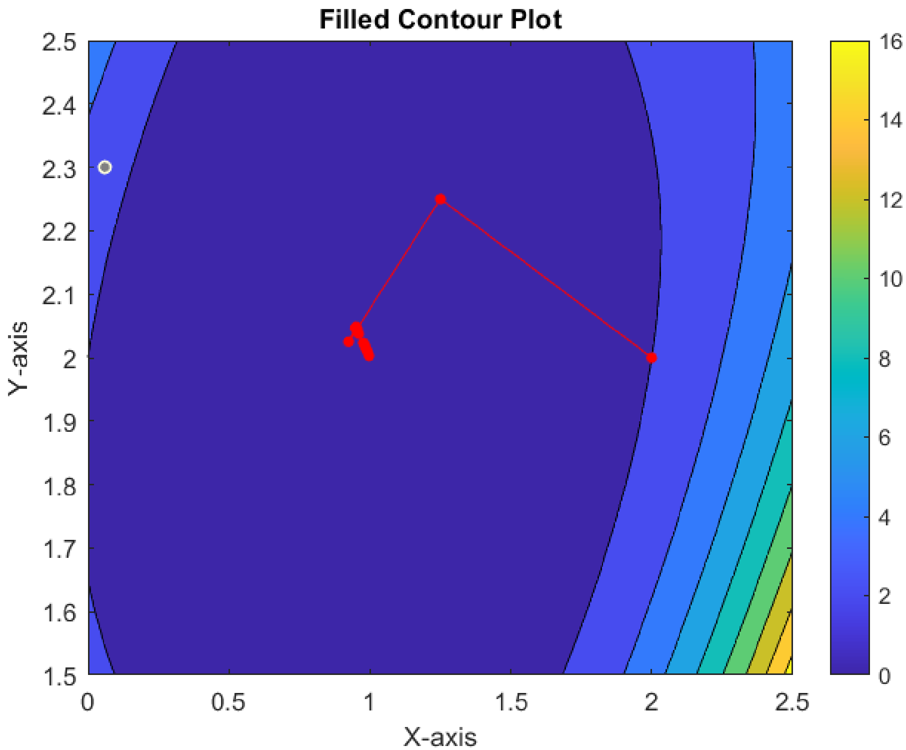

Figure 2: Steepest Descent converging path.

Another method that can be used is the Conjugate Gradient. After 173 iterations, the method converged to the optimal solution: $x_1 = 0.997 and x2 = 2.003$. The achieved optimal objective function value is impressively close to zero, specifically 1.3037×10^{-9}

Comparing the two methods, the Steepest Descent Method demonstrates a slower convergence rate, requiring significantly more iterations (905) compared to the Conjugate Gradient Method (173). The larger number of iterations suggests that the Steepest Descent Method may be computationally more expensive for this specific problem. Additionally, the achieved optimal objective function value, though small, is relatively higher compared to the Conjugate Gradient Method. Overall, the Conjugate Gradient Method appears to outperform the Steepest Descent Method in terms of efficiency and speed of convergence.

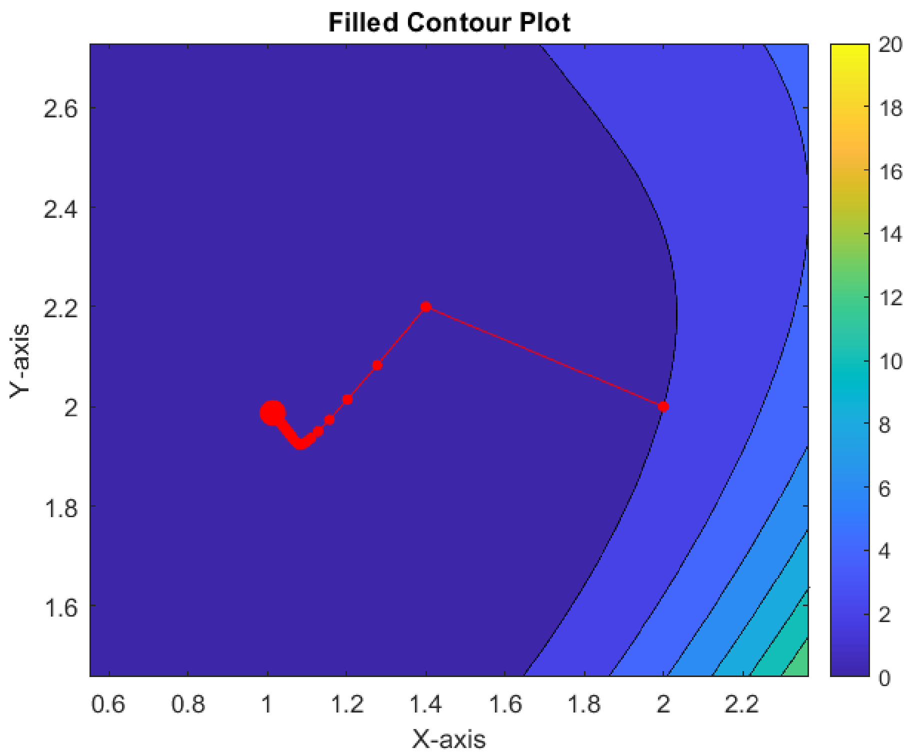

Figure 3: Path taken by the Conjugate Gradient method.

In the Steepest Descent method, each step taken along its convergence path is orthogonal to the previous one. This orthogonality ensures a straightforward and direct progression towards the minimum.

On the other hand, the Conjugate Gradient method shares the same initial step as Steepest Descent but subsequently takes steps that are conjugate to the previous. The two paths can be found side by side in fig. 4.

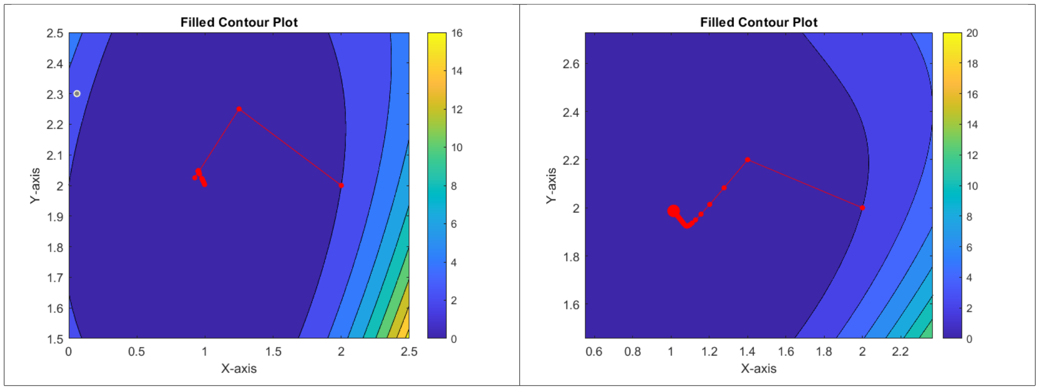

Figure 4: Steepest Descent and Conjugate Gradient converging paths.

The implementation of both of these algorithms entailed a constant gamma value equal to 0.1. This along with the nature of the function, whose gradient values are quite small approaching the global minimum, led to very shallow steps taken in the same direction. As can be seen in fig. 5 this greatly reduces the optimizing efficiency.

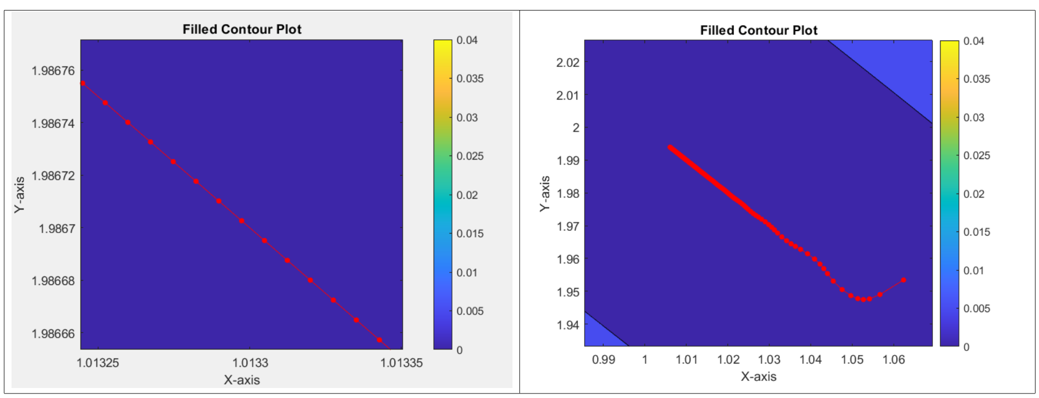

Figure 5: Shallow steps in the same direction.

To solve this a backtracking line search function that adapts the gamma value in each iteration can be implemented to both of these, resulting in greatly improved efficiency. Here gamma is reduced until the Armijo condition is met. In this case, the Gradient Descent algorithm converged in only 25 iterations, whereas the Steepest Descent interestingly, only needed 3 iterations, converging to the exact value.

2.2 Newton's Method

Before the assessing the performance of this method it is deemed appropriate to explain the inner workings of it's algorithm. Although this algorithm bares great resemblance to Steepest Descent, it is different in one crucial area. The step that is taken each time is dependant both on the Gradient of the function on this point and the Hessian which essentially is the Jacobian of the Gradient. For this algorithm to converge it is necessary for the Hessian matrix to be positive semi definite in each step.

This approach to finding the minimum of the function did not converge. This is because the Hessian after six iterations stopped being positive semi definite. To prevent this from happening a modified version of this algorithm called the Levenbrg - Marquardt method can be used. This method essentially forces the Hessian to be positive semi definite by adding to it a diagonal matrix as many times as necessary.

The Levenberg-Marquardt method demonstrated efficiency in optimizing the objective function, converging to an optimal solution of $x_1 = 1.0006$ and $x_2 = 1.9994$. The achieved optimal objective function value of 2.0947×10^{-12} indicates a highly accurate solution. Interestingly, this method required 58 iterations to converge to this point regardless of the initial point.

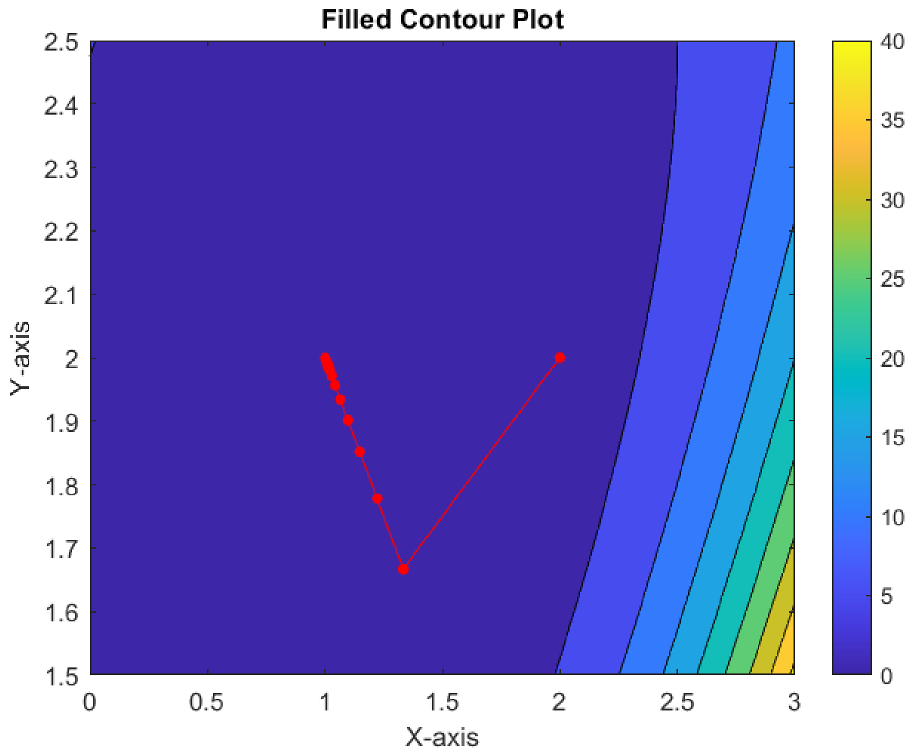

Figure 6: Levenberg - Marquardt converging path.

3 For $n = 5$

Adding more dimensions to the problem reduces the intuitive understanding and makes it impossible to get any sort of visualized path through a contour plot.

3.1 Gradient only based algorithms

For the initial point $X_0 = [2, 2, 2, 2, 2]$ and a tolerance of $tol = 1 \times 10^{-4}$ the Steepest Descent method converged in 990 iterations to $A = [0.66378, 0.73528, 0.84275, 1.0125, 1.987$ with an objective value of 7.7487. The algorithm did not fail to converge for a number of other initial points. Noteworthy is that to achieve a $tol = 1 \times 10^{-5}$ accuracy the iterations increased to 4654. Here the gamma was constantly set at 0.1 whereas the algorithm failed to converge using a variable gamma. It essentially terminated after only 2 iterations as the Armijo condition was met, this might be attributed to a bug in the function.

The Conjugate Gradient algorithm did not converge for a constant gamma value. When changing to a variable one, using the same backtracking line search function, with the same initial point as above, the objective function was minimized at a value of 3.3780 after 83 iterations. Noteworthy here is that this method is highly dependant initial conditions. For instance, changing the initial point to $X_0 = [100, 100, 100, 100, 100]$, the objective function value becomes 6.3075.

3.2 Newton's method

For this section two different approaches were tried. Calculating the Hessian and the Gradient numerically, using the same functions as before and symbolically, using the corresponding matlab toolbox. For the numerical approach, the Newton's method unsurprisingly did not converge in this case either. This time however the Levenberg - Marquardt alteration did not converge as well, regardless of the initial conditions as the Hessian proved to not be positive semi definite from the first step.

Symbolically only the Levenberg - Marquardt method was tried as it was deemed the most probable to be successful. Using the already built in functions the Hessian became positive semi-definite and the method converged to an objective value of 2.2787 at $B = [1.0368, 1.3699, 1.5, 1.6301, 1.9632]$. This value is the lowest from all the methods used. Moreover, it only required 5 iterations to approach this solution when starting from $X_0 = [2, 2, 2, 2, 2]$ and 33 starting from a completely random point $X_1 = [10, 9, 91848, 93, 1]$.

4 Conclusions

It is important to describe in detail the methods used in approaching this assignment. For $n = 2$ the functions for Steepest Descent and Conjugate Gradient were used both with and without a variable gamma value for the step. Here the Newton's method failed to converge so an alteration, the Levenberg - Marquardt method was implemented. All three converged to the same point approximately in a number of iterations. Here the Hessian was correctly defined numerically so there was no reason to attempt a symbolic

Method/Parameter	St. D.	CG	Newton's	L-M
Converged	$4.5881e-8$	$1.3037e-10$	No	$2.0947e-13$
Iterations	905	173	-	58
Iterations (Armijo)	3	25	-	-
Stable	Yes	Yes	-	Yes

Table 1: Performance Evaluation Table for $n = 2$.

approach. However, for $n = 5$ the process was not as smooth. Even though the Steepest Descent and the Conjugate Gradient converged, due to ill definition of the function used to calculate the Hessian, there was no way for the Hessian to be positive semi definite, even from the first iteration (the ill definition was not present for $n = 2$). This meant that another way to calculate the Hessian had to be implemented, in this case a symbolic approach. Using multiple different methods to approach this assignment, led to a better understanding of the inner workings of each method. On the other hand this increased the difficulty to draw conclusions between the various methods since the calculations were not made using a consistent method.

The findings of this assignment can be summed up in the next two tables. The algorithms are judged based on three parameters. If they converged, how many iterations did it take them to converge and how stable they were i.e. if their optimum value was dependent on initial conditions.

For $n = 2$ both the gradient based algorithms and the Levenberg - Marquardt converged to approximately the same point which is indeed the global minimum. The fastest to converge was the Steepest Descent using the Backtracking line search function to select the gamma value in each iteration.

$For n = 5$ the Gradient based algorithms did not fail to converge, however they did not converge to the same point. The Steepest Descent method converged to a point where the value of the objective function was 7.7487 whereas under some initial conditions the Conjugate Gradient reduced this value to 3.3780. Note that the Conjugate Gradient utilized a variable gamma value and was thus more effective than the Steepest Descent, converging to the solution in only 83 iterations. This difference became more prominent when the tolerance level was increased. The Steepest Descent needed 4654 to the 268 of the Conjugate Gradient. Interestingly under every initial conditions tried, the Conjugate Gradient managed to find a point where the

Method/Parameter	Steepest D.	CG	Newton's	L- M
Converged	7.7487	3.3780	No	2.2787 (sym)
Iterations	990	83	-	-
Stable	Yes	No	-	-
Iterations (sym)	-	-	-	5
Stable (sym)	-	-	-	Yes

Table 2: Performance Evaluation Table for $n = 5$.

value of the objective function was lower than the Steepest Descent's which always converged to 7.7487. As mentioned above the methods requiring the inversion of the Hessian did not converge using a numerical approach. As mentioned above when a symbolic approach was implemented, the Levenberg - Marquardt method converged to a value of 2.2787 in very few iterations, regardless of initial conditions. It becomes apparent that since every method converged to a different point, there are more than one local minimums.

5 Appendix

Here are the two Hessians for $n = 5$. The first was calculated numerically by a custom function, while the second one using a built in symbolic calculation.

Numerically calculated Hessian:

$$\begin{bmatrix} 14 & -10 & 0 & 0 & 0 \\ -10 & 14 & -10 & 0 & 0 \\ 0 & -10 & 14 & -10 & 0 \\ 0 & 0 & -10 & 14 & -10 \\ 0 & 0 & 0 & -10 & 14 \end{bmatrix}$$

Symbolically calculated Hessian:

$$\begin{bmatrix} 14 & -10 & 0 & 0 & 0 \\ -10 & 28 & -10 & 0 & 0 \\ 0 & -10 & 28 & -10 & 0 \\ 0 & 0 & -10 & 28 & -10 \\ 0 & 0 & 0 & -10 & 14 \end{bmatrix}$$