Global Positioning System (GPS) Algorithm

Nonlinear optimization

Erik Sandström

ECE174, Department of Electrical and Computer Engineering
University of California, San Diego

December 7, 2017

Contents

1 Problem description and goals

The objective of this project is to utilize nonlinear optimization algorithms to show the mathematical steps needed to calculate the position of a receiver on earth to high accuracy, when the receiver clock is not accurate. This can be done by using satellite data from four satellites, while only three are needed when the accuracy of the receiver clock is high. The problem is relevant in order to utilize the non accurate clock in modern day cell phones in order to determine the position of the phone. Two methods of optimization will be used, the gradient descent method and the Gauss-Newton method, and their performance will be compared. `MatLab` is used as a tool to implement the concept.

2 Theory

A cartesian right handed coordinate system with the origin positioned at the center of the earth is used and the unit of measurement utilized is in units of earth Radii. The true range between a receiver and a satellite denoted ℓ is defined as

$$R_\ell(S) = ||(S - S_\ell)|| = \sqrt{(S - S_\ell)^T (S - S_\ell)} \quad \ell = 1, ..., 4, \tag{1}$$

where $S = (x, y, z)^T$ is defined as the position of the receiver and $S = (x_\ell, y_\ell, z_\ell)$ is defined as the position of the ℓ:th satellite. The true range can, due to the inaccuracy of the receiver clock not be calculated explicitly so instead the receiver measures a pseudorange denoted by y_ℓ and defined as

$$y_\ell = R_\ell(S) + b + v_\ell, \quad \ell = 1, ..., 4, \tag{2}$$

where b is called the clock bias error describing the inaccuracy of the receiver clock and v_ℓ is Gaussian noise of the measurement (where each v_ℓ is i.d.d) caused by uncertainty in the effective velocity of light through the medium between the satellite and the receiver. One can suggest that multiple measurements of y_ℓ can be made within a short amount of time so that the satellites and the receiver are considered motionless to increase the accuracy of the measured pseudorange, but in the following discussion for simplicity $v_\ell = 0$ and only one measurement of $y = (y_1, y_2, y_3, y_4)^T$ will be considered. Equation 2 can be described in vector form as

$$\begin{aligned} y &= R(S) + eb + v \qquad (3) \\ &= h(X) + v \qquad (4) \end{aligned}$$

where

$$h(X) = (R_1(S) \;\; R_2(S) \;\; R_3(S) \;\; R_4(S))^T + (1 \;\; 1 \;\; 1 \;\; 1)^T b. \tag{5}$$

To linearize y, one only needs to take the true range R(S) into consideration since the clock bias b is already a linear parameter. Doing a multivariate Taylor series expansion of $R_\ell(S)$ around an estimate $\hat{S}$ yields

$$R_\ell(S) \approx R(\hat{S}) + \frac{\partial R(\hat{S})}{\partial S}\Delta S \tag{6}$$

The derivative of $R_\ell(S)$ with respect to S can be further simplified

$$\begin{aligned}
\frac{\partial R(\hat{S})}{\partial S} &= \frac{\partial\sqrt{(\hat{S}-S_\ell)^T(\hat{S}-S_\ell)}}{\partial S} &(7)\\
&= \frac{\partial\sqrt{(\hat{S}-S_\ell)^T(\hat{S}-S_\ell)}}{\partial(\hat{S}-S_\ell)}\frac{\partial(\hat{S}-S_\ell)}{\partial S} &(8)\\
&= \frac{1}{2}\frac{1}{\sqrt{(\hat{S}-S_\ell)^T(\hat{S}-S_\ell)}}\frac{\partial(\hat{S}-S_\ell)^T(\hat{S}-S_\ell)}{\partial(\hat{S}-S_\ell)}I &(9)\\
&= \frac{1}{2R_\ell(\hat{S})}2(\hat{S}-S_\ell)^T &(10)\\
&= \frac{(\hat{S}-S_\ell)^T}{R_\ell(\hat{S})} = r_\ell^T(\hat{S}) &(11)
\end{aligned}$$

Using equation 11 and inputing the linearized version of $R_\ell(S)$ in equation 2 yields,

$$y = R_\ell(\hat{S}) + \frac{(\hat{S}-S_\ell)^T}{R_\ell(\hat{S})}\Delta S + b + v. \tag{12}$$

This shows that a larger value of the true range $R_\ell(\hat{S})$ yields a more accurate model since the difference term will affect the model less.

To derive the algorithms used in this paper, the loss function needs to be defined. Using equation 4, one finds that, due to the Gaussian noise present in the measurement, likelihood function can be written as a constant times the exponential function to the power of the negative loss function. The optimization problem at hand is to minimize the loss function defined by

$$\ell(X) = \frac{1}{2}||y - h(X)||^2, \tag{13}$$

in order to maximize the likelihood function of the estimates S and b (found in X) for a given y. To find the value of X which minimizes equation 13, the family of gradient descent methods can be used. This method is defined by, for every iterative step ΔX_k, moving in the direction opposite to the gradient of $\ell(X)$. It does not necessarily mean moving directly in the opposite direction of the gradient, but a component parallell to the negative gradient is needed for a method to belong to this family. Assume that the path along which will be walked in order to reach the minima is parametrised as X(t), where t is a parameter (e.g. time). Then the change in amplitud with respect to t is

$$\frac{\partial\ell}{\partial t} = \frac{\partial\ell}{\partial X}\frac{\partial X}{\partial t} = (\nabla_X\ell)^T\frac{\partial X}{\partial t}, \tag{14}$$

where the $(\nabla_X \ell)^T$ is the gradient of ℓ. To force the derivative of the loss function with respect to t to be negative, the direction in which X is chosen corresponds to the derivative of X being

$$\frac{\partial X}{\partial t} = -Q(X)\nabla_X \ell, \tag{15}$$

where Q(X) is an arbitrary positive definite matrix. Putting equation 15 into equation 14 yields

$$\frac{\partial \ell}{\partial t} = -(\nabla_X \ell)^T Q(X) \nabla_X \ell = -||\nabla_X \ell||^2_{Q(X)} \leq 0, \tag{16}$$

which means that the chosen path always decreases the value of the loss function or keeps it constant, if a minima has been found. To make the algorithm implementable, it needs to be discretized. This is done by approximating the derivative of X with respect to t with a forward difference equation yielding

$$\frac{\partial X}{\partial t} \approx \frac{X(t_{k+1}) - X(t_k)}{\Delta t} = \frac{X_{k+1} - X_k}{\Delta t}. \tag{17}$$

Putting equation 17 into equation 15 yields

$$\hat{X}_{k+1} = \hat{X}_k - \Delta t Q(\hat{X}_k)\nabla_X \ell(\hat{X}_k) = \hat{X}_k - \alpha_k Q(\hat{X}_k)\nabla_X \ell(\hat{X}_k), \tag{18}$$

where Δt has been replaced by α_k. The real positive number α_k is called the step size parameter the choice of α_k is critical for convergence of the algorithm. Note that the estimates of X are denoted by $\hat{X}$. The derived formula is called the generalized gradient descent algorithm since the matrix Q can be chosen in different ways, yielding algorithms with different properties. Two common choices of Q will be further discussed. The gradient of the loss function can, however, be further simplified. Denoting $e(X) = y - h(X)$ yields

$$\begin{aligned}\nabla_X \ell(X) &= \Big(\frac{1}{2}\frac{\partial \ell}{\partial e}\frac{\partial e}{\partial x}\Big)^T = (e^T(X)\frac{e(X)}{\partial X})^T = -\Big(\frac{h(X)}{\partial X}\Big)^T (y - h(X)) \qquad (19)\\ &= -H^T(X)(y - h(X)), \qquad (20)\end{aligned}$$

where the Jacobian of h(X) is denoted H(X). The Jacobian matrix H(X) is square (4×4) and invertible since it is assumed that h(X) is locally one-to-one. Putting the derived expression in equation 20 into equation 18 yields

$$\hat{X}_{k+1} = \hat{X}_k + \alpha_k Q(\hat{X}_k) H^T(\hat{X}_k)(y - h(\hat{X}_k)) \tag{21}$$

2.1 Gradient Descent Method

Choosing Q = I, the iterative formula becomes

$$\begin{aligned}\hat{X}_{k+1} &= \hat{X}_k + \alpha_k H^T(\hat{X}_k)(y - h(\hat{X}_k)) \qquad (22)\\ &= \hat{X}_k + \Delta X_k, \qquad (23)\end{aligned}$$

This means that for each step ΔX_k, the path that is walked is in the direction directly opposite to the gradient. This ensures that every step generates a locally optimum descent. The algorithm is very easily implemented and the computations needed to perform each step is low. The efficiency of the method is the biggest drawback and will be discussed later. The terms in equation 23 can be concretized by putting in the data. In a real life situation, the receiver will measure the pseudorange to every satellite. Since this is a simulation project, this data need to be artificially generated. This is done by assuming locations for four satellites, each denoted S_ℓ, assuming a receiver location S and a block bias error b. Putting these assumptions into equation 2 yields the pseudorange. The function h(X) can be expanded to the following using equation 5

$$h(X) = \begin{bmatrix} R_1(S) + b \\ R_2(S) + b \\ R_3(S) + b \\ R_4(S) + b \end{bmatrix} = \begin{bmatrix} ((x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2)^{1/2} + b \\ ((x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2)^{1/2} + b \\ ((x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2)^{1/2} + b \\ ((x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2)^{1/2} + b \end{bmatrix}, \tag{24}$$

The Jacobian matrix can be written in the following form

$$H(X) = \frac{\partial h(X)}{\partial X} = \begin{bmatrix} \frac{\partial R(S)}{\partial S} & \frac{\partial eb}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{(S-S_1)^T}{R_1(S)} & 1 \\ \frac{(S-S_2)^T}{R_2(S)} & 1 \\ \frac{(S-S_3)^T}{R_3(S)} & 1 \\ \frac{(S-S_4)^T}{R_4(S)} & 1 \end{bmatrix} = \begin{bmatrix} \frac{x-x_1}{R_1(S)} & \frac{y-y_1}{R_1(S)} & \frac{z-z_1}{R_1(S)} & 1 \\ \frac{x-x_2}{R_2(S)} & \frac{y-y_2}{R_2(S)} & \frac{z-z_2}{R_2(S)} & 1 \\ \frac{x-x_3}{R_3(S)} & \frac{y-y_3}{R_3(S)} & \frac{z-z_3}{R_3(S)} & 1 \\ \frac{x-x_4}{R_4(S)} & \frac{y-y_4}{R_4(S)} & \frac{z-z_4}{R_4(S)} & 1 \end{bmatrix} \tag{25}$$

The initial condition $\hat{X}_0$ will be chosen as a crude estimate of the receiver location.

2.2 Gauss-Newton Method

To find the value of X which minimizes equation 13, the Gauss-Newton method can be used. This method is defined by a minimization of the linearization of

$$e(X) = y - h(X), \tag{26}$$

where for every iterative step ΔX_k, the method moves towards the minima but not necessarily in the direction opposite to the gradient. The Gauss-Newton method rather takes more than local points into account when determining direction of descent and therefore a more optimum path can be chosen, with hopefully less iterative steps needed in order to reach the minimum value of the loss function. The iterative formula is found by chosing

$$Q(X) = (H^T(X)H(X))^{-1}, \tag{27}$$

and putting this into equation 21 and using the fact that the Jacobian matrix is invertible yields the Gauss-Newton iterative algorithm

$$\begin{aligned} \hat{X}_{k+1} &= \hat{X}_k + \alpha_k H^{-1}(\hat{X}_k)(y - h(\hat{X}_k)) && (28) \\ &= \hat{X}_k + \Delta X_k. && (29) \end{aligned}$$

3 Method

The initial condition $\hat{X}_0$ is chosen as a crude estimate of the receiver location.

$$\hat{S}_0 = (0.933100\ \ 0.250000\ \ 0.258819)^T\ \ Earth\ \ Radii\ \ (ER) \tag{30}$$

The initial clock bias error $\hat{b}_0 = 0$. To simulate y as described in the previous section, assumed positions of the receiver, satellites and clock bias error are needed. The values are taken to be

$$S = (1.0000\ \ 0.0000\ \ 0.0000)^T\ \ ER \tag{31}$$

$$S_1 = (3.5852\ \ 2.0700\ \ 0.0000)^T\ \ ER \tag{32}$$

$$S_2 = (2.9274\ \ 2.9274\ \ 0.0000)^T\ \ ER \tag{33}$$

$$S_3 = (1.4159\ \ 0.0000\ \ 3.8904)^T\ \ ER \tag{34}$$

$$S_4 = (1.4159\ \ 0.0000\ \ 3.8904)^T\ \ ER \tag{35}$$

and

$$b = 2.354788068 \times 10^{-3}\ \ ER \tag{36}$$

For both the gradient descent method and Gauss-Newton method a loop is used to update the iterations. Some form of termination criteria are needed in order to know when an optimum solution has been found. When the iteration steps $||\Delta X_k||$ are smaller than some predefined value, the loop will terminate. If the steps do not meet this predefined value, a maximum allowed iteration steps will be enforced on the loop as well. In this project the termination criteria are $min||\Delta X_k|| = 10^{-8}$ and the maximum amount of iterations allowed are $maxItr = 10^6$. The value of y is normally measured by the receiver, but since no measurements can be done for the purpose of this paper, a synthetic variant of y will be created since we assum a certain position of the receiver and value of the clock bias error. With these values, y can be generated according to

$$y = R(\hat{S}_0) + e\hat{b}_0 + v \tag{37}$$

Furthermore, the term v describing the noise will be assumed to be nonexistent troughout the rest of the paper.

4 Results

4.1 Gradient Descent Method

All plots have been made in a logarithmic scale of the vertical axis to present the data in a clearer way.

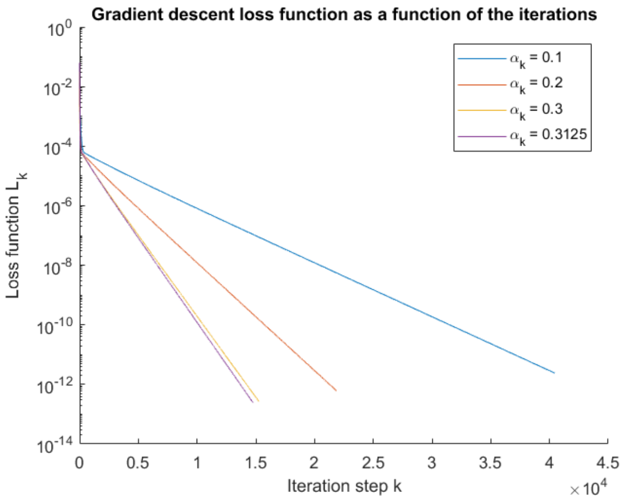

Figure 1: The loss function L_k decreases exponentially and depends on α_k.

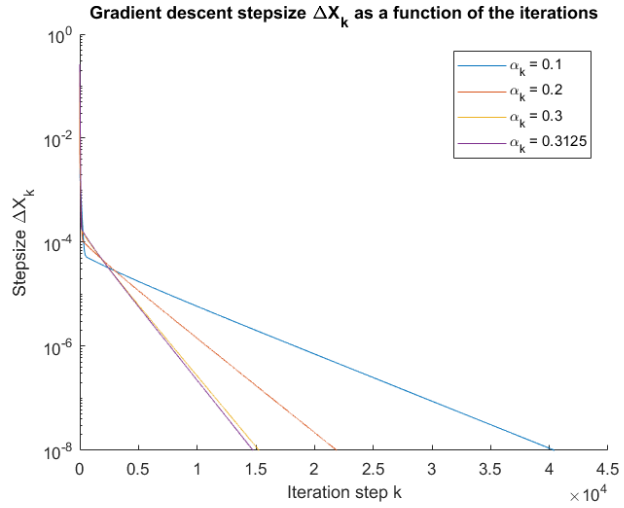

Figure 2: The stepsize ΔX_k decreases exponentially and depends on α_k.

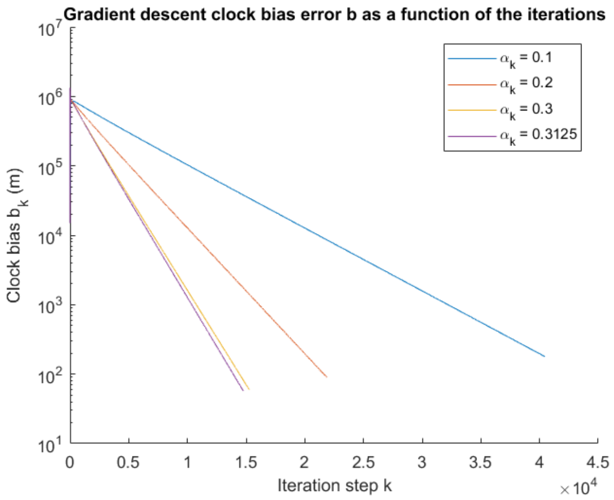

Figure 3: The clock bias error b_k decreases exponentially and depends on α_k.

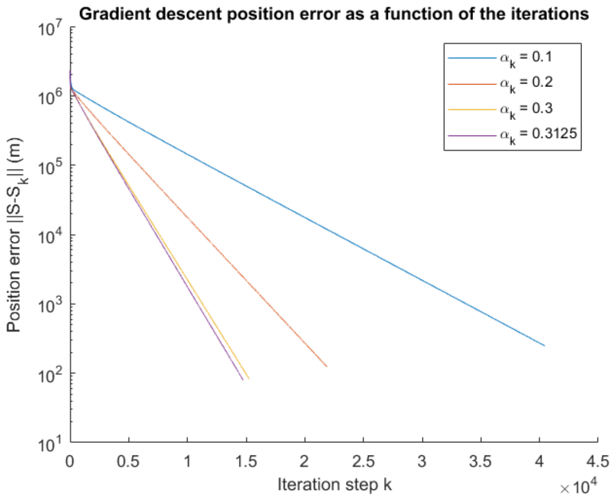

Figure 4: The position error decreases exponentially and depends on α_k.

4.2 Gauss-Newton Method

All plots have been made in a logarithmic scale of the vertical axis to present the data in a clearer way.

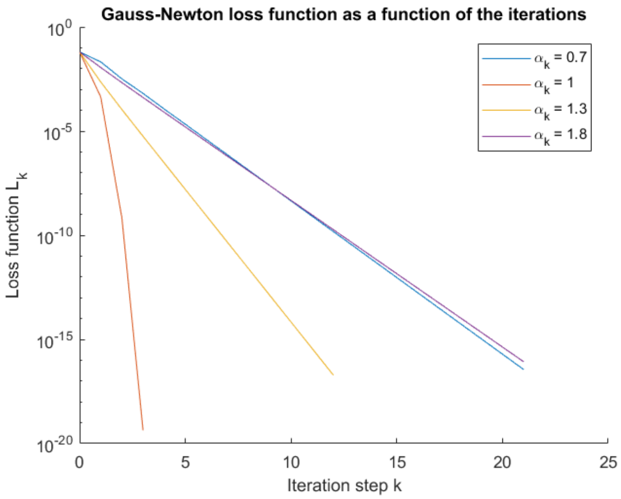

Figure 5: The loss function L_k decreases exponentially and depends on α_k.

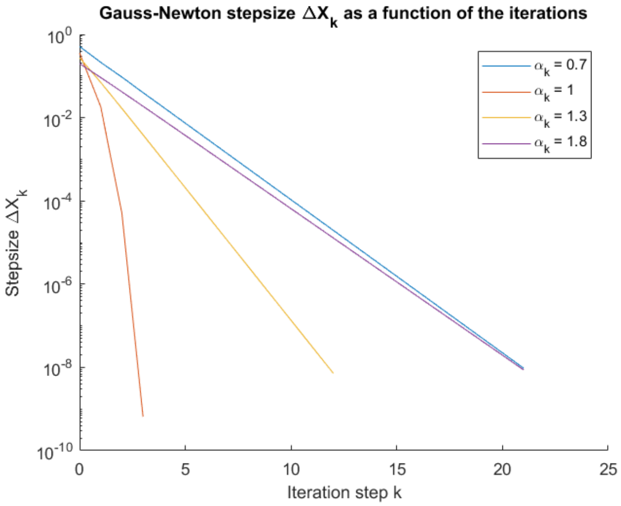

Figure 6: The stepsize ΔX_k decreases exponentially and depends on α_k.

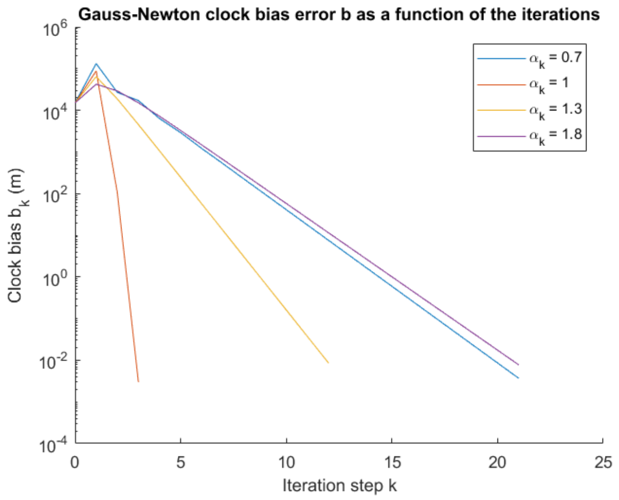

Figure 7: The clock bias error b_k decreases exponentially and depends on α_k.

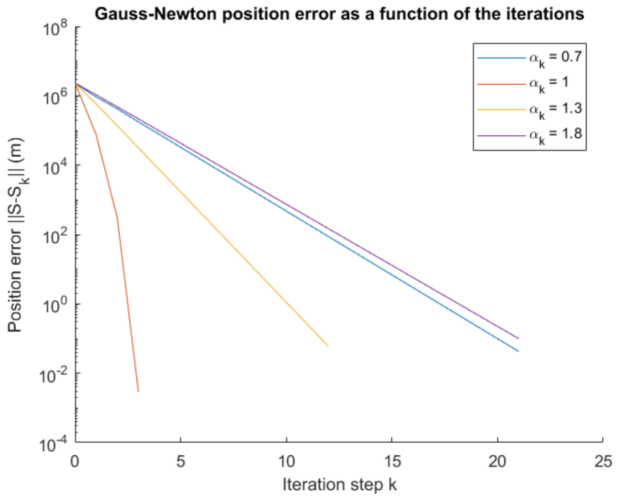

Figure 8: The position error decreases exponentially and depends on α_k.

5 Discussion

5.1 Gradient Descent Method

The loss function is plotted for four values of α_k in figure 1 where $\alpha_k = 0.3125$ was the longest stepsize possible before the method went unstable. As can be seen in figure 1 higher values of the stepsize yields faster convergence towards the true range. This is expected since a bigger stepsize means that fewer steps are needed to get to the minima implying faster convergence, as long as the method is stable. The behavior displayed in figure 1 is consistent with the behavior in figure 2, 3 and 4 where the drop off is exponential for every case. The final results for $\alpha_k = 0.3125$ can be summarized as follows. The final estimate of $\hat{S}_k$ is compared with the assumed position of the receiver S and the distance between the two is calculated to 79.3 m requiring 14,759 iterative steps.

5.2 Gauss-Newton Method

The loss function is plotted for four values of α_k in figure 5 where $\alpha_k = 1$ was choice of α_k that resulted in highest convergence rate. As can be seen in figure 5 higher values of α_k than 1 yields slower convergence towards the true range and smaller values of α_k than 1 also yields slower convergence rate. Contrary to the gradient descent method, the Gauss-Newton method has the ability to utilize very large step sizes without going unstable and the method can therefore overshoot the correct minima meaning that it oscillates towards the correct solution if the step size is chosen too big. The optimal value of α_k is around 1 as can be seen in figuer 5. The behavior displayed in figure 5 is consistent with the behavior in figure 6, 7 and 8 where the drop off is exponential for every case. The final results for $\alpha_k = 1$ can be summarized as follows. The final estimate of $\hat{S}_k$ is be compared with the assumed position of the receiver S and the distance between the two is calculated to 2.87 m requiring only 3 iterative steps. A final comment can be made about why the Gauss-Newton method is so accurate and fast.

It is clear that the Gauss-Newton method is enormously more efficient when it comes to the convergence rate. It does, however, require more computational power for each iterative step, but in comparison with the 4920 times less steps required to reach a better result, it is clear that the Gauss-Newton method is highly preferable.

6 Conclusion

By using a mathematical nonlinear optimization technique, this paper shows that a more accurate description of a receiver's actual position can be found when the receiver clock is inaccurate. It is striking how much the gradient descent method differs from the Gauss-Newton method when it comes to convergence rates which is almost 5000 times smaller. This means that choosing a rational approach when it comes to optimization is highly critical. Note that no consideration has been taken to optimize the choice of stepsize parameter α_k and that the noise has been assumed to be nonexistent. These two factors play a big role with regards to the convergence rate and accuracy respectively of the solution and it is therefore expected that the results are too optimistic for real world applications.