STAT 31020: Homework 2

Caleb Derrickson

Wednesday, January 17, 2022

Contents

Problem 1

We call an analytical function in the domain $D \subset \mathbb{R}$, D and open set, an infinitely differentiable function for which the Taylor series at x converges at every point $y \in N(x) \subset D$ to the value of the function at y, $f(y)$.

Problem 1, part a

Prove that for an analytical function on $\mathbb{R}$ a necessary and sufficient condition for x to be an isolated local minimum is for $k(x)$ in problem 1, homework 1, to be even, positive, and for f to satisfy $f^{(k(x))}(x) > 0$.

Solution:

I will assume necessary and sufficient conditions are impinged on showing x is an isolated local minimum if and only if $k(x)$ is positive, even, and $f^{(k(x))}(x) > 0$. For simplicity, I will take x^* to be the isolated local minimum, $k(x)$ to be expressed simply as k, and $N(x)$ to be expressed as N.

For the forward direction, we will assume that x^* is an isolated local minimum within a Neighborhood N of x^*. Thus $f(x^*) < f(x)$ for all $x \in N$[1]. By Taylor's Theorem, and since f is given as infinitely differentiable, we can express $f(y)$ for any $y \in N$ as

$$f(y) = f(x^*) + \sum_{j=1}^{\infty} \frac{1}{j!} f^{(j)}(x^*)(y - x^*)^{(j)}$$

Note that k is the first nonzero derivative of $f(x^*)$. Within this neighborhood, we can approximate the value of $f(y)$ by the investigation of two terms,

$$f(y) \approx f(x^*) + \frac{1}{k!} f^{(k)}(x^*)(y - x^*)^{(k)}.$$

Since x^* is a strict local minimum within the neighborhood, $f(y) - f(x^*) > 0$. This means

$$f^{(k)}(x^*)(y - x^*)^{(k)} > 0,$$

where the factor of $1/k!$ was discarded. This statement implies the signs of both terms $f^{(k)}(x^*)$ and $(y - x^*)^{(k)}$ are equal. Thus, we can break this down into two cases:

- Case 1: $f^{(k)}(x^*) < 0$ and $(y - x^*)^{(k)} < 0$.

  Considering the second term, we see that $(y - x^*)^{(k)} < 0$. This implies that k is odd, since if k were even, then $k = 2m$ for $m \in \mathbb{N}$, so $(y - x^*)^{(k)} = (y - x^*)^{(2m)}$, which is strictly positive. Note that $y \neq x^*$ by assumption. Thus k is odd. Note that we are investigating around an open ball of x^*, so the expression

[1] Within the neighborhood, isolated local minimum implies strict local minimum.

$y - x^*$ could be either positive or negative. If we consider $y > x^*$, then $y - x^* > 0$, so $(y - x^*)^{(k)} > 0$. This is a contradiction, since $(y - x^*)^{(k)} < 0$. Thus $f^{(k)}(x^*) < 0$ and $(y - x^*)^{(k)} < 0$ cannot be true.

- Case 2: $f^{(k)}(x^*) > 0$ and $(y - x^*)^{(k)} > 0$.

  We will again only consider the second term. Here, since we are investigating around an open ball of x^* $y - x^*$ could be either positive or negative. In the previous case, we saw that the condition $(y - x^*)^{(k)} < 0$ restricted us to only considering k odd. Since $(y - x^*)^{(k)} > 0$, this restricts to k being even.[2] Therefore, by exhaustion of cases, and consideration of the two terms, we see that $f^{(k)}(x^*) > 0$ and k even. We also have that k is positive, by the Taylor series expansion. Thus the forward direction is shown.

For the backward direction, we have that k is even, positive, and that $f^{(k)}(x^*) > 0$. We wish to show that x^* is an isolated local minimum for N. We will again consider the Taylor series approximation for any y within a local neighborhood N of x^* as

$$f(y) \approx f(x^*) + \frac{1}{k!} f^{(k)}(x^*)(y - x^*)^{(k)}.$$

We are given that $f^{(k)}(x^*) > 0$. Furthermore, note that since k is even (and positive), we note that $(y - x^*)^{(k)} > 0$ for $y \neq x^*$. We then see that

$$f(y) - f(x^*) = \frac{1}{k!} f^{(k)}(x^*)(y - x^*)^{(k)} > 0.$$

Therefore, $f(y) > f(x^*)$ for $y \in N \setminus \{x^*\}$. Therefore, x^* is an isolated local minimum.

[2] Or, we only restrict values for which $y > x^*$. This gives the contradiction from Case 1.

Problem 1, part b

What can you then say about the function that you used as an example at problem 1 homework 1, part 4?

Solution:

The function I used in part 4 was

$$f(x) = \begin{cases} \exp\left(-\frac{1}{x^2}\right), & x \neq 0 \\ 0, & x = 0. \end{cases}$$

This function was used as an example for $k = \infty$ (i.e., all derivatives of f equal zero at the critical point $x = 0$), and the minimum $x = 0$ being a isolated local minimum. From the arguments made above, this would then mean that f is even, positive, and that $f^{(k)}(x^*) > 0$. The last implication however is not satisfied, since $k = \infty$.

Problem 2

Prove Zoutendijk's Theorem (Theorem 3.2 stated in the book) for the case where the step length α_k is computed by backtracking, but starting at fixed $\tilde{\alpha} > 0$. For this to work, assume in the following there exists some $c_3 > 0$ such that the search direction p_k satisfies $\|p_k\| \geq c_3 \|\nabla f(x_k)\|$. Explain how the conclusion may fail if you do not impose this condition.

Solution:

I will provide the Theorem below.

Consider any iteration of the form $x_{k+1} = x_k + \alpha_k p_k$, where p_k is a descent direction and α_k satisfies the Wolfe conditions. Suppose that f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set N containing the level set $L \overset{\text{def}}{=} \{x : f(x) \leq f(x_0)\}$, where x_0 is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on N, that is, there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(\tilde{x})\| \leq L \|x - \tilde{x}\|, \quad \text{for all } x, \tilde{x} \in N.$$

Then

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty.$$

There are a few things I should mention before going into calculations. First, we take the cosine of the angle θ_k as

$$\cos \theta_k = -\frac{\nabla f_k^\intercal p_k}{\|\nabla f_k\| \|p_k\|}$$

Furthermore, we can rewrite the condition on the length of the search direction as

$$\|p_k\| \geq c_3 \|\nabla f_k\| \implies \|p_k\|^2 \geq c_3 \|\nabla f_k\| \|p_k\| \geq c_3 \left\|\nabla f_k^\intercal p_k\right\| \implies \frac{c_3 \left\|\nabla f_k^\intercal p_k\right\|}{\|p_k\|^2} \leq 1.$$

And, since we are given that the stepsize is taken from backtracking, starting at $\tilde{\alpha}$, we can say that $\alpha_k \leq \rho \tilde{\alpha}$ for some fixed $\rho \in (0, 1)$. Then the steps below follow:

$$f_{k+1} \leq f_k + c_1 \alpha_k \nabla f_k^\intercal p_k \qquad \text{(Armijo Condition.)}$$

$$\leq f_k + c_1 \rho \tilde{\alpha} \nabla f_k^\intercal p_k \qquad \text{(Backtracking Results.)}$$

$$\leq f_k - c_1 \rho \tilde{\alpha} \left\|\nabla f_k^\intercal p_k\right\| \qquad \text{(Absolute value definition.)}$$

$$
\begin{aligned}
&\leq f_k - c_1 \rho \tilde{\alpha} \frac{c_3 \left\| \nabla f_k^\intercal p_k \right\|^2}{\|p_k\|^2} && \text{(Search Direction Restriction.)} \\
&\leq f_k - c_1 c_3 \rho \tilde{\alpha} \cos^2 \theta_k \|\nabla f_k\|^2 && \text{(Cosine definition.)} \\
\implies f_{k+1} &\leq f_0 - c_1 c_3 \rho \tilde{\alpha} \sum_{k>0} \cos^2 \theta_k \|\nabla f_k\|^2 && \text{(Iteration to initial.)} \\
\implies \sum_{k>0} \cos^2 \theta_k \|\nabla f_k\|^2 &\leq \frac{1}{\kappa} (f_0 - f_{k+1}) && \text{(Rearranging, κ is all the constants.)}
\end{aligned}
$$

Since f is bounded below, there exists some B such that for any $k > 0$, $f(x_k) > B$. Thus,

$$\kappa \sum_{k>0} \cos^2 \theta_k \|\nabla f_k\|^2 < f(x_0) - B < \infty.$$

The only counter I could provide to give justification to this condition is if you choose your sequence of angles θ_k to be exactly π. This would mean $\cos \theta_k = -1$, so $\left\| \nabla f_k^\intercal p_k \right\| = \|\nabla f_k\| \|p_k\|$. This is incompatible with the results above, since the condition implies there being some angle less than π between p_k and ∇f_k.

Problem 3

Consider an optimization problem $\min_{x\in\mathbb{R}^n} f(x)$ where f is twice continuously differentiable. Assume that x^* is a local solution that satisfies the first order necessary condition $\nabla f(x^*) = 0$ and the second-order sufficient condition $\nabla^2_{xx} f(x^*) \succ 0$.

Problem 3, part a

Prove that, there exists parameters $c_1, c_2 > 0$ such that for all x sufficiently close to the local solution we have that $c_1\|x - x^*\| \leq \|\nabla f(x)\| \leq c_2\|x - x^*\|$

Solution:

We first note that the Hessian $\nabla^2 f(x^*)$, so in a neighborhood D around the critical point x^*, the function $f|_D$ is convex. Therefore, the gradient $\nabla f(x)$ is Lipschitz in D. The right inequality is simple. Since ∇f is lipschitz continuous, for some $L > 0$, we can say for any x, y in a neighborhood of x^*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

Setting $y = x^*$, and noting that $\nabla f(x^*) = 0$, we get

$$\|\nabla f(x)\| \leq L\|x - x^*\|.$$

Therefore, setting $c_2 = L$ satisfies the right inequality for any x in a neighborhood of our local minima. For the left hand side, we can first note by the Fundamental Theorem of Calculus,

$$\int_x^{x^*} \nabla f(x') \cdot \mathbf{dx'} \leq \left|\int_x^{x^*} \nabla f(x') \cdot \mathbf{dx'}\right| \leq \int_x^{x^*} \|\nabla f(x')\|\|\mathbf{dx'}\|.$$

Since $\mathbf{dx'}$ is a unit vector in the direction of integration, the integral is just the value $\|\nabla f(x)\|$. Relating this to the distance from x to x^*, we can take a Taylor expansion of f.

$$f(x) = f(x^*) + \nabla f(x^*)(x - x^*) + \frac{1}{2}(x - x^*)^\intercal \nabla^2 f(x^*)(x - x^*).$$

x^* is a local minimizer, so after simplification and rearranging,

$$f(x) - f(x^*) = \frac{1}{2}(x - x^*)^\intercal \nabla^2 f(x^*)(x - x^*).$$

Therefore,

$$|f(x) - f(x^*)| = \frac{1}{2}|(x - x^*)^\intercal \nabla^2 f(x^*)(x - x^*)| \leq \|x - x^*\|\left\|\nabla^2 f(x)\right\|.$$

Combining this with what we found before, we get

$$\frac{\left\|\nabla^2 f(x^*)\right\|}{2}\|x - x^*\| \leq \|\nabla f(x)\|.$$

Since $\nabla^2 f(x^*)$ is positive definite, its norm is positive. So setting $c_1 = \frac{1}{2}\left\|\nabla^2 f(x^*)\right\|$ will satisify the inequalities.

Problem 3, part b

Use this result to prove that under the sufficient conditions, not only is x^* a strict local minima, but also an isolated local minimum.

Solution:

Suppose there is a neighborhood N around x^* such that the second order Taylor Expansion at $y \in N$ is a fine approximation. That is,

$$f(y) = f(x^*) + \nabla f(x^*)(y - x^*) + \frac{1}{2}(y - x^*)\nabla^2 f(x^*)(y - x^*).$$

Since $\nabla f(x^*) = 0$, the second term is cancelled. After rearranging,

$$f(y) - f(x) = \frac{1}{2}(y - x^*)\nabla^2 f(x^*)(y - x^*)$$

Since $\nabla^2 f(x^*) \succ 0$, we have that for any $z \in N$, $z^T \nabla^2 f(x^*) z > 0$. Thus the right side is greater than zero, meaning $f(y) - f(x^*) > 0$, which means $f(y) > f(x^*)$ for any $y \in N$. Thus, x^* is an isolated local minima.

Problem 3, part c

Assume now that you have an algorithm that produces a sequence $x_k \to x^*$ such that the gradient sequence $\|\nabla f(x_k)\|$ converges R-linearly to zero. What can you say about the convergence of the error sequence $\|x_k - x^*\|$?

Solution:

We have that $\|\nabla f(x_k)\| \to 0$ R-linearly. By definition, there exists a sequence $\{v_k\}$ satisfying $\|\nabla f(x_k)\| \leq v_k$ for all k, where v_k converges Q-linearly to zero. From part a, we have that there exists $c_1 > 0$ such that $c_1\|x_k - x^*\| \leq \|\nabla f(x)\|$. By transitivity, $\|x_k - x^*\| \leq \frac{1}{c_1}v_k$ for all k. [3] Defining a new sequence $w_k = \frac{1}{c_1}v_k$, we can see that w_k converges Q-linearly to zero as well, since

$$\frac{w_{k+1}}{w_k} = \frac{(1/c_1)v_{k+1}}{(1/c_1)v_k)} = \frac{v_{k+1}}{v_k} < r < 1.$$

Therefore, we can say the sequence x_k converges R-linearly to x^*, so $\|x_k - x^*\| \to 0$.

[3] If x_k is does not start in a sufficient neighborhood, extract a subsequence that is then in the neighborhood.

Problem 3, part d

Assume now that you have an algorithm that produces a sequence $x_k \longrightarrow x^*$ such that the gradient sequence $\|\nabla f(x_k)\|$ converges R-superlinearly to zero. What can you say about the convergence of the error sequence $\|x_k - x^*\|$?

Solution:

In the same spirit as the last part, we have that $c_1\|x_k - x^*\| \leq \|\nabla f(x_k)\| \leq v_k$ for some sequence v_k which converges to zero Q-superlinearly. Then x_k converges R-superlinearly as well, since it is bounded by a Q-superlinear sequence.

Problem 3, part e

Assume now that there is an optimization algorithm that is convergent to x^*. Propose a computable test to decide whether the sequence is superlinearly or quadratically convergent.

Solution:

Since we have no knowledge of a limit, I would imagine that we could analyse the forward error of the sequence x_k. That is, we would expect the sequence $\|\nabla f(x_k) - \nabla f(x^*)\|$ to converge to zero. Since x^* would be a local minima, th first order necessary conditions would hold. So $\nabla f(x^*) = 0$, meaning the forward error simplifies to $\|\nabla f(x_k)\|$. We see that this bounds the sequence $\|x_k - x^*\|$, so $\|x_k - x^*\|$ will converge when $\|\nabla f(x_k)\|$ converges to zero. Therefore, the rate of convergence of x_k to x^* depends entirely on the convergence of $\|\nabla f(x_k)\|$. So if $\|\nabla f(x_k)\|$ is R-quadratically convergent, then so is x_k, and the same for R-superlinear convergence.

Problem 4

Problem 4, part 1

Implement the steepest descent algorithm as discussed in the class using the backtracking (Armijo) algorithm 3.1 from the textbook. We will apply it to the quadratic function $f(x) = x_1^2 + cx_2^2, c > 0$ in various circumstances that I will describe below. The solution to its minimization is $(0, 0)$.

Solution:

I have implemented this, where all you need to do is call calebderrickson_hw2p4(x0, c) when you are in the folder. It will output two plots, one showing the normed gradient sequence as described in problem 3, and the ratio of normed gradient sequences. The script will also print out the number of iterations, as well as the estimate on the minima.

Problem 4, part 2

There is a theorem (3.3) that tells you what a bound on the rate of convergence could be, though for a different line search type, but we will use it as a yardstick. What is that bound as a function of c?

Solution:

The bound is given as

$$\left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right),$$

where $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ are the eigenvalues of Q, a symmetric positive definite matrix, where our f is of the form $f(x) = \frac{1}{2}x^\top Qx - b^\top x$. Though not exactly similar, for all intensive purposes, we can assume Q is the Hessian of our function. Therefore, we take Q as

$$Q = \nabla^2 f = \begin{bmatrix} 2 & 0 \\ 0 & 2c \end{bmatrix}$$

Therefore, the eigenvalues of Q are 2 and $2c$. Although not explicitly told, for all parts of this problem, we take $c \geq 1$, so the ordering of eigenvalues are $\lambda_1 = 2,\ \lambda_2 = 2c$. This then gives our bound as

$$\left(\frac{c-1}{c+1}\right).$$

Problem 4, part 3

Propose a method to track the convergence without making use of the information about where the minimum is. Can you think of ways to use this information to check the rate of convergence (and what rate can you hope to obtain)? Discuss your rationale.

Solution:

From Problem 3, we can take convergence to a minimum when our sequence of points x_k has the property that $\|\nabla f(x_k)\| \to 0$ as $k \to \infty$. Checking the rate of convergence, we can take it as the ratio of sequential values of the normed-gradient. We will encounter floating point / cancellation errors as we get closer to the minima, however. This will be ignored for the moment, and be assuaged if necessary.

Problem 4, part 4

Propose a stopping criterion based on 2, that is, when would it seem it does not make sense to keep computing. Implement it in your code and stop when either it was triggered or when you have reached 1000 iterations. Discuss your rationale.

Solution:

In the loop used to create the sequence x_n, we can have our while loop terminate when either condition mentioned above is invalid. I will provide some pseudocode that describes this:

```
MAX_ITER = 1e4
seq = [x0]
epsilon = 1e-8
[f_eval, g_eval] = FunctionEvaluation(x0, c)
i = 1

while norm(g_eval) > epsilon and i < MAX_ITER
    alpha = Backtracking(seq(i, :), c)
    seq(i+1, :) = seq(i, :) - alpha * g_eval
    [f_eval, g_eval] = FunctionEvaluation(seq(i+1, :), c)
    i = i + 1
end while
```

Problem 4, part 5

Apply now your method to $f(x)$ for $c = 3$ from 4 starting points : (1, 1), (-1, 1), (-1, -1), (1, -1). Report your convergence metric from step 3 as well as the true error. Does it match what you got at point 2?

Solution:

I have added the plots below. We can see in Figure 1 that the change in initial conditions does not change the curve significantly. [4] We do see that not only does each sequence of normed gradients converge to zero, it does so in a time well below the maximum allowed iterations. The reason that the plots are very similar is that the backtracking algorithm found a stepsize of 0.25 to be the best in almost all steps. It could be the case that my value for ρ, that is, the iterating factor in backtracking, was chosen too small. I did modify it to an increased value (0.5 $\rightarrow$ 0.9), but found worse results. In Figure 2, we can see that the ratio between sequential normed gradients is flat,[5] with value 0.5. Note that in part 2, we found the bound between sequences to be roughly

$$\left(\frac{c-1}{c+1}\right).$$

When we plug in $c = 3$, we get the bound to be 0.5. This is exactly the value we see as our ratio between sequences.

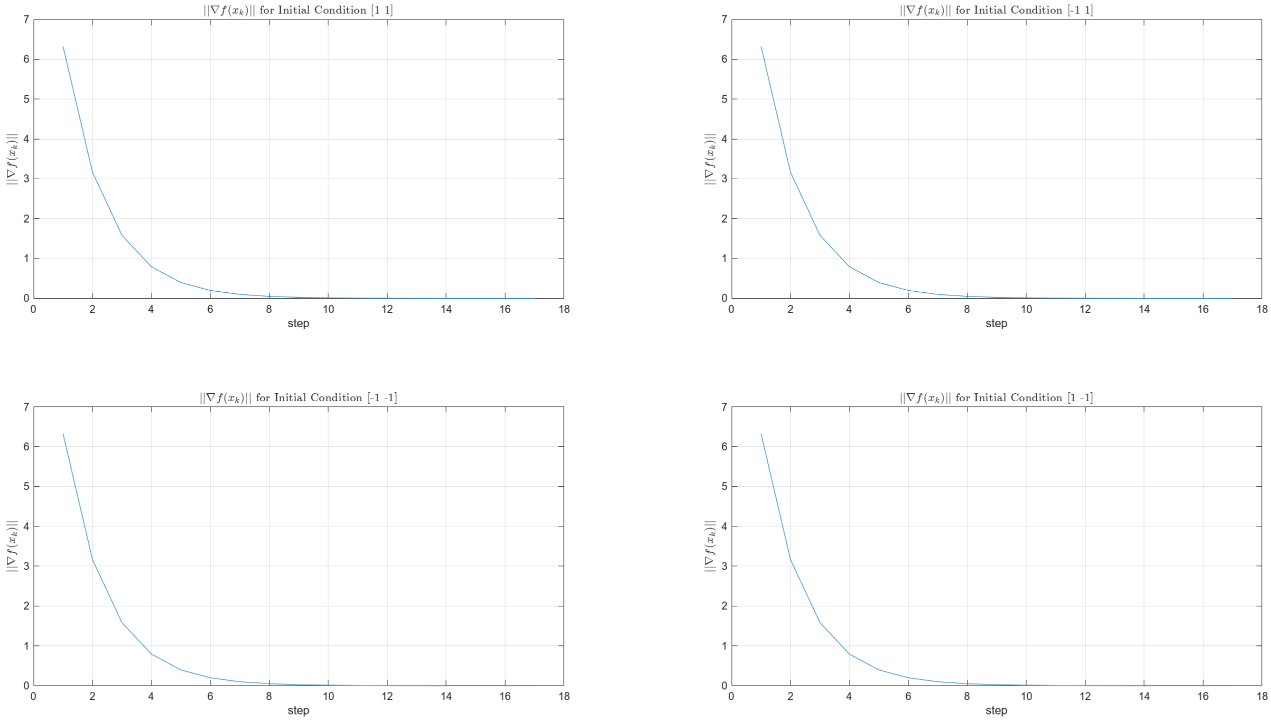

Figure 1: Sequence of Normed Gradients for all initial conditions.

[4] I checked it wasn't the same plot four times.

[5] Aside from the last entry, which I will chalk that up to error.

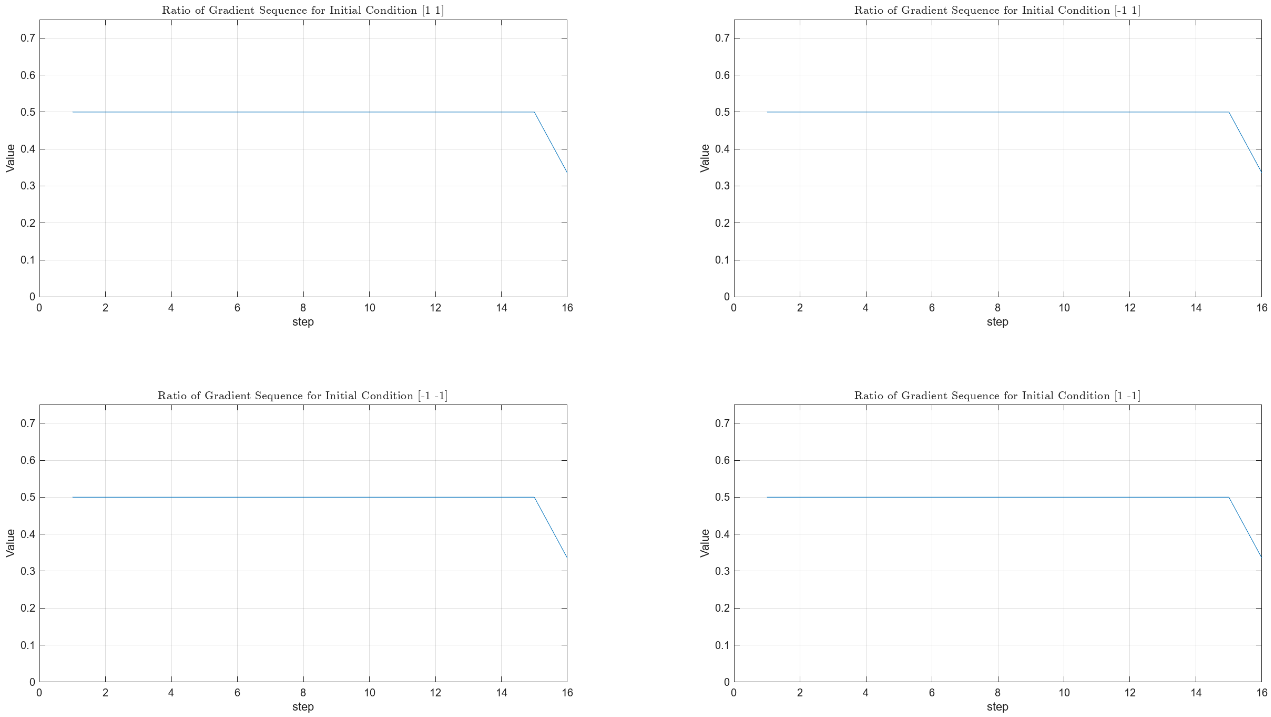

Figure 2: Ratio of Gradient Sequences.

Problem 4, part 6

Repeat part 5 for $c = 1000$.

Solution:

I have included the plots below. It seems like we reach our tolerance in finite time [6] in Figure 3. Figure 4 is somewhat less assuring, however. When zooming in, there is one point that pops up, then it goes back down to a value roughly 0.96. It might be the case that backtracking gives us a stepsize too large, which might get in the way of correctly finding the minima nicely. Figure 5 is a zoom-in on one of the initial conditions. We can see that there is a "zig-zag" pattern in the sequence. It is normal for steepest descent algorithms to have this, as discussed in class. I have also included a log plot of one of the initial conditions, for legibility.

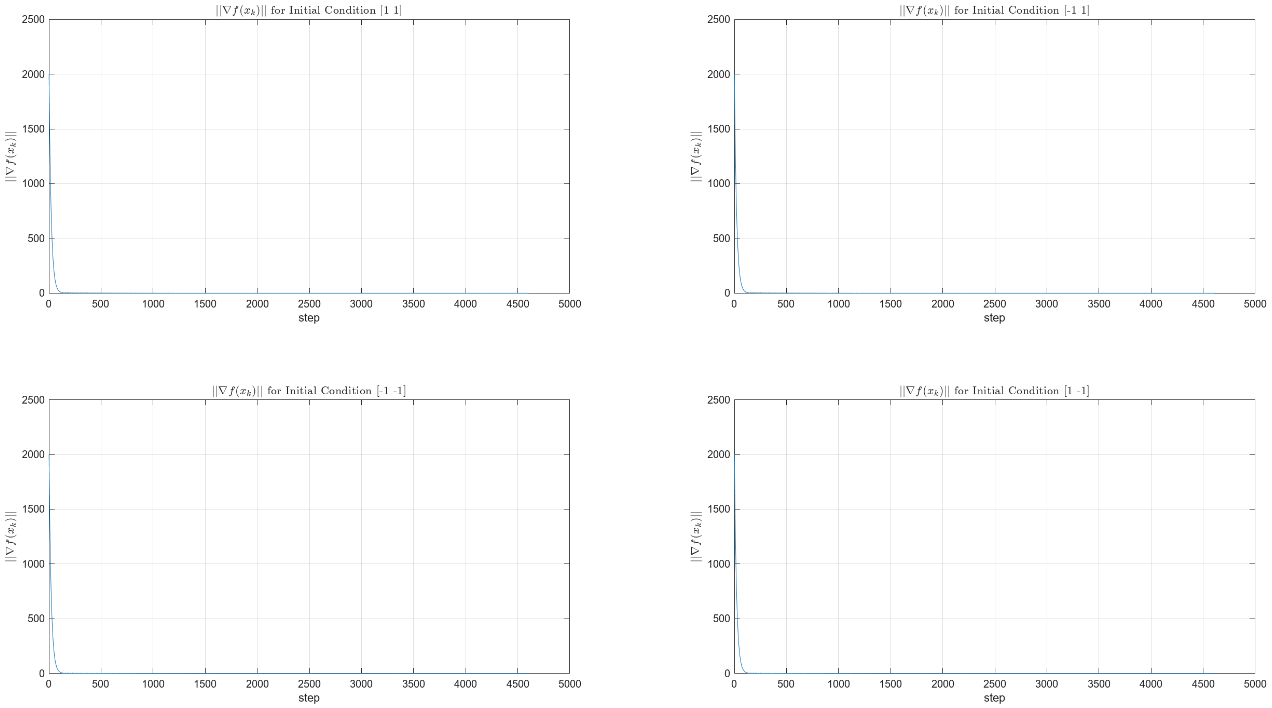

Figure 3: Sequence of Normed Gradients for all initial conditions with $c = 1000$.

[6] This is set to 1 x 10^{-4} for all plots.

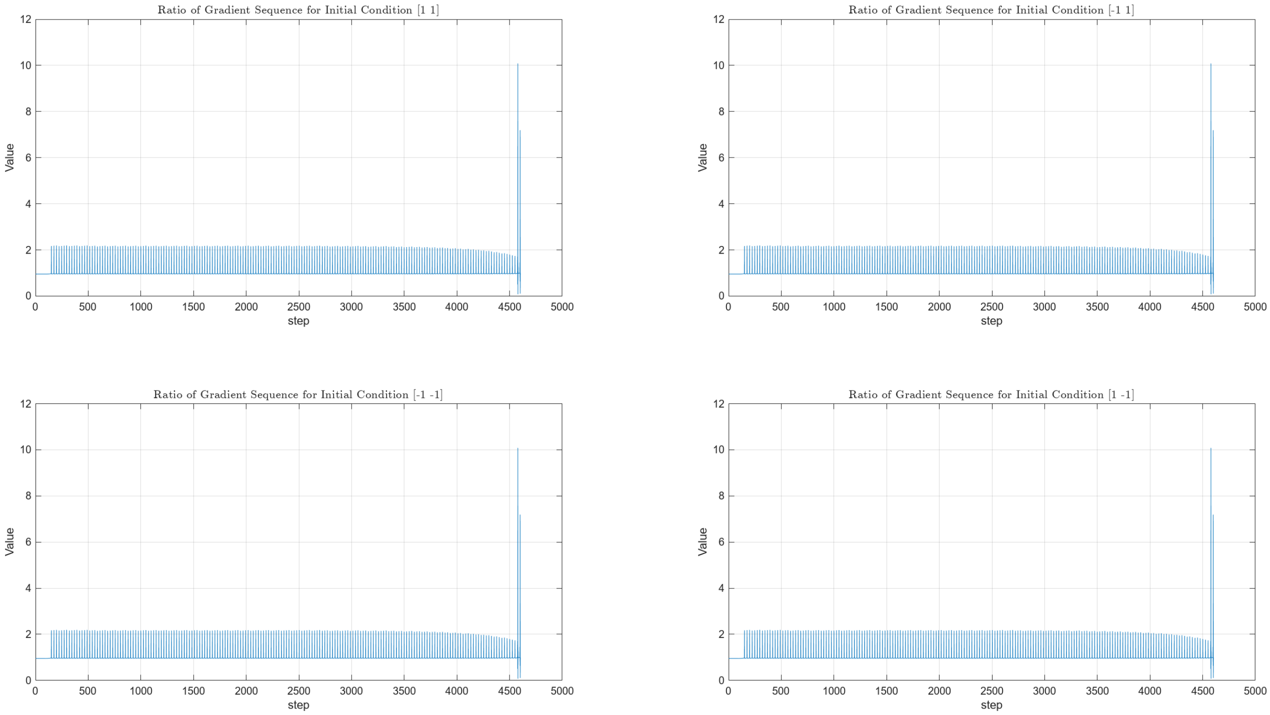

Figure 4: Ratio of Gradient Sequences with $c = 1000$.

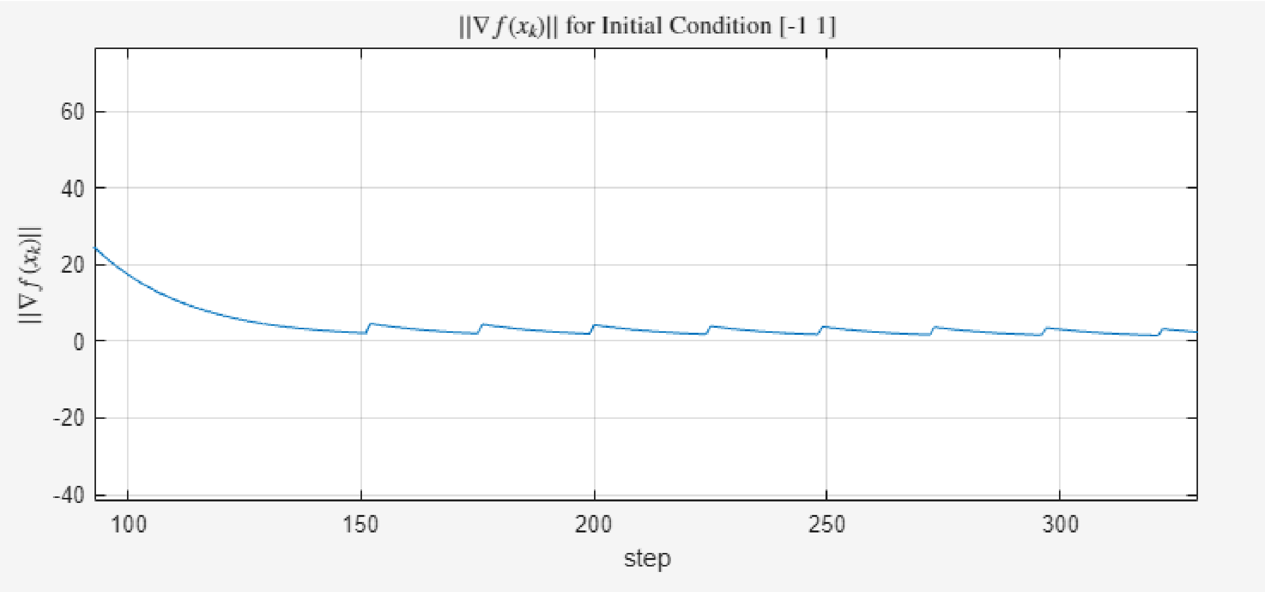

Figure 5: A zoom in on the sequence of normed gradients. We can see the zig-zag pattern that steepest descent is known for.

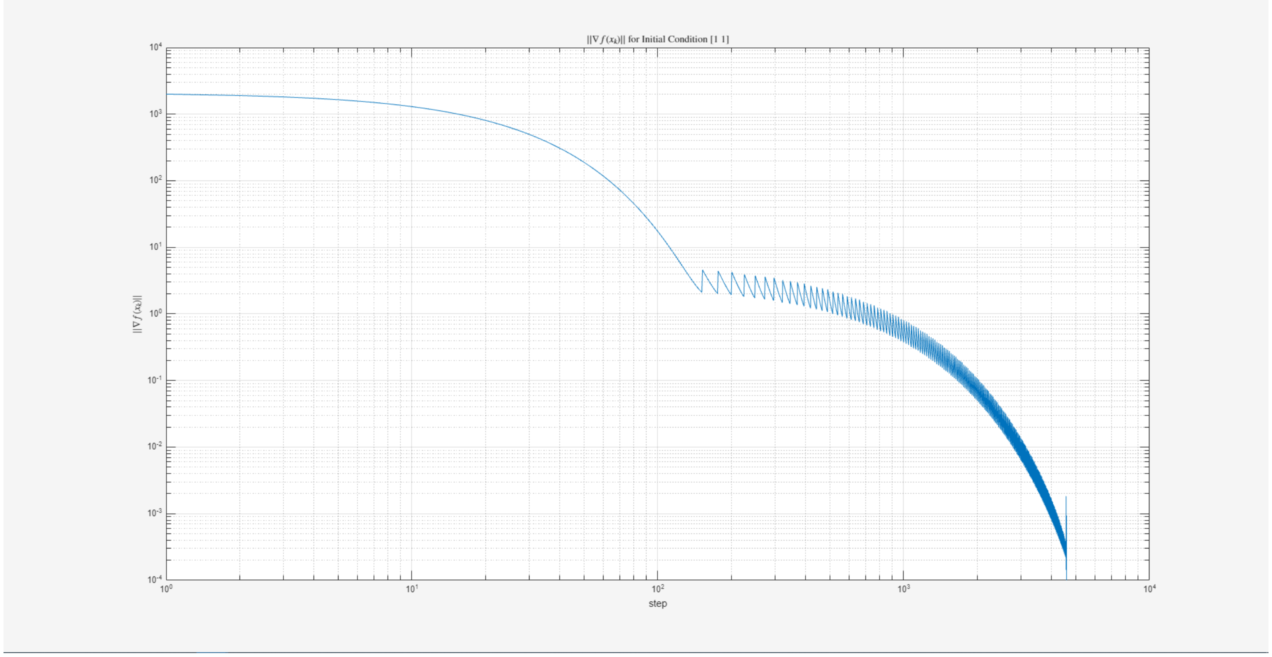

Figure 6: Log Plot of The sequence of normed gradients for $c = 1000$, with initial conditions $[1, 1]$.