

STAT 31020: Homework 5

Caleb Derrickson

February 14, 2024

Contents

1	Problem 1	2
1	Problem 1, part 1	3
2	Problem 1, part 2	4
3	Problem 1, part 3	5
4	Problem 1, part 4	7
5	Problem 1, part 5	8
6	Problem 1, part 6	9
2	Problem 2	10
3	Problem 3	11
4	Problem 4	13
1	Problem 4, part a	13
2	Problem 4, part b	14

Problem 1

We will aim to prove the first-order optimality conditions Theorem 12.1 and a slightly weaker version of the second-order optimality conditions Theorem 12.5 by assuming the ones for constrained optimization. That is: we will aim to solve the problem (which I will call the ORIGINAL problem)

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1}$$

$$\text{s.t. } c_i(x) = 0, \ i \in \mathcal{E}$$

$$c_i(x) \geq 0, \ i \in \mathcal{I}$$

where the functions f, c_i are smooth (three times continuously differentiable is sufficient, that is what we will assume here), and \mathcal{I}, \mathcal{E} are two indices of sets (typically 1:m, m+1:m+p, but the set notation is much more convenient). A point which only satisfies the constraints is called *feasible*. At such a point some of the inequality constraints may be satisfied with equality, they are said to be *active*. We will denote by the active set at a feasible point x , $\mathcal{A}(x) = \{i \in \mathcal{E} \cup \mathcal{I} : c_i(x) = 0\}$. observe that $\mathcal{E} \subset \mathcal{A}(x)$.

We aim to characterize the properties of a local minimum x^* , an extension of the first-order and second-order optimality conditions of unconstrained minimization. An important quantity used to express these conditions is the Lagrangian

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x). \tag{2}$$

Here λ_i are called the Lagrange multipliers. We will assume that at the solution x^* we satisfy the linear independence constraint qualification (LICQ). That is, we assume the at the set of vectors $\nabla_i c_i(x^*)$, $i \in \mathcal{A}(x^*)$ is linearly independent. I call the matrix J , whose rows are these vectors, the Jacobian (compare with 12.37 that defines the same quantity). Therefore, we will assume J to be full row rank. Observe one standard awkwardness optimization notation standard: gradients of individual functions are thought of as columns but if you look at the definition of the Jacobian they are thought of as rows.

Problem 1, part 1

Read theorem 12.1 and 12.5 and specialize them for the case of equality constraints only. In particular observe that the critical cone and tangent cone coincide in this case (when it comes to interpreting 12.5).

Solution:

When transforming Theorem 12.1 and Theorem 12.5 into the desired form, we can assume $\mathcal{I} = \phi$, that is, the empty set. Therefore, $i \in \mathcal{E} \cup \mathcal{I} = \mathcal{E}$. Thus, when writing Theorem 12.1, which I will call the KKT conditions, we have the following:

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0,$$

$$c_i(x^*) = 0, \quad \text{for all } i \in \mathcal{E},$$

$$\lambda_i^* c_i(x^*) = 0, \quad \text{for all } i \in \mathcal{E}.$$

We will also assume the conditions for KKT hold as well, which are continuous differentiation of f, c_i and that LICQ holds at x^* . I will denote these conditions as Equality KKT. Theorem 12.5 does not include any additional constraints on the problem, except for KKT to be satisfied. We can rewrite Theorem 12.5 in the same manner, that is

$$w^\top \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w \geq 0, \quad \text{for all } w \in C(x^*, \lambda^*),$$

but with the qualifier that our of w will be taken from a simplified set. In Theorem 12.5 in the book, we are saying that $\nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*)$ is positive semi-definite for all elements in the critical cone, defined as

$$C(x^*, \lambda^*) = \{w \in \mathcal{F}(x^*) : \nabla_i c_i(x^*)^\top w = 0, \text{ for all } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* > 0\}.$$

Note that the index set from which we are choosing our i 's from, $\mathcal{A}(x^*) \cap \mathcal{I}$ is the empty set. Thus, we have a “false positive”¹, which implies that this condition will automatically hold true for all choices of i . Thus the critical cone degenerates to the set of linearized feasible directions,

$$\mathcal{F}(x) = \{d : d^\top \nabla c_i(x) = 0, \quad \text{for all } i \in \mathcal{E}\}.$$

By Lemma 12.2, since the LICQ condition is satisfied at x^* , we have that $\mathcal{F}(x^*)$ is equivalent to the tangent cone, $T_\Omega(x^*)$. Therefore, the critical cone is equivalent to the tangent cone for the case of only equality constraints.

¹This isn't exactly the correct terminology: if I wanted to be more explicit, I would say the condition suffers from false implication.

Problem 1, part 2

Rewrite now the optimization problem using the square slack tricks. That is, introduce variables $z_i, i \in \mathcal{I}$ and show that the constraint $c_i(x) \geq 0$ is equivalent to $c_i(x) - z_i^2 = 0$, in the sense of the x for which it holds. Therefore, the problem when minimizing over x, z is equivalent with one with only equality constraints in the sense of the optimal x .

Solution:

Let $y = \begin{bmatrix} x & z \end{bmatrix}^\top$. We wish to show the original problem, **1**, is equivalent to

$$\min_{y \in \mathbb{R}^{n+|\mathcal{I}|}} f(y) \tag{3}$$

$$\text{s.t} \quad c_i(x) = 0, \quad i \in \mathcal{E}$$

$$c_i(x) - z_i^2 = 0, \quad i \in \mathcal{I}$$

For the original problem, **1**, we have that $c_i(x) \geq 0$, therefore setting $z_i = \sqrt{c_i(x)}$ is well defined over the real numbers. Therefore,

$$c_i(x) - z_i^2 = c_i(x) - \left(\sqrt{c_i(x)} \right)^2 = c_i(x) - c_i(x) = 0.$$

This implies that the set of x which satisfy $c_i(x), i \in \mathcal{I}$ in the original problem satisfy $c_i(x) - z_i^2 = 0, i \in \mathcal{I}$. Therefore, this modified problem is equivalent to the original problem upon choice of z_i .

Problem 1, part 3

Prove that the original problem, 1, satisfies the linear independence constraint qualification if and only if the problem with equality constraints obtained in part 2 satisfies it as well at the point x^*, z^* , where $(z_i^*)^2 = c_i(x^*)$, $i \in \mathcal{I}$. Therefore Theorem 12.1 and 12.5 are applicable to the problem in part 2 under the assumptions of this problem.

Solution:

• \implies :

Here we will prove the forward implication - that is, if the first statement holds, then the second one holds. Assume this is not the case, that is, the original problem satisfies LICQ, but the modified problem does not. Therefore, we assume that

$$\{\nabla_y (c_i(x) - (z_i^*)^2)\}_{i \in \mathcal{I}} \cup \{\nabla_y c_j(x^*)\}_{j \in \mathcal{E}}$$

is NOT linearly independent. Thus, there exists some constants a_i, b_j for $i, j \in \mathcal{I}, \mathcal{E}$ respectively, where

$$\sum_{i \in \mathcal{I}} a_i \begin{bmatrix} \nabla_x c_i(x^*) \\ -2z_i^* \vec{e}_i \end{bmatrix} + \sum_{j \in \mathcal{E}} b_j \begin{bmatrix} \nabla_x c_j(x^*) \\ \vec{0} \end{bmatrix} = \vec{0},$$

where we will assume the appropriate dimensions for the zero vectors. The vector \vec{e}_i denote the canonical unit basis for \mathbb{R}^n . Since for $j \in \mathcal{E}$, we have the bottom part zero, we have that

$$\sum_{i \in \mathcal{I}} 2a_i z_i^* \vec{e}_i = \vec{0}.$$

Since all \vec{e}_i 's are linearly independent, we have that $a_i z_i^* = 0$ for all $i \in \mathcal{I}$. This should hold for all $i \in \mathcal{I}$, in particular, when we look at the indices not included within the active set, we have that $c_i(x) > 0$. This implies that $z_i^* > 0$. Thus, $a_i = 0$ for $i \in \mathcal{I} \setminus \mathcal{A}(x^*)$, so

$$\vec{0} = \sum_{i \in \mathcal{I} \cap \mathcal{A}(x^*)} a_i \begin{bmatrix} \nabla_x c_i(x^*) \\ -2z_i^* \vec{e}_i \end{bmatrix} + \sum_{j \in \mathcal{E}} b_j \begin{bmatrix} \nabla_x c_j(x^*) \\ \vec{0} \end{bmatrix}$$

Next we examine the top parts of each vector. This implies

$$\vec{0} = \sum_{i \in \mathcal{I} \cap \mathcal{A}(x^*)} a_i \nabla_x c_i(x^*) + \sum_{j \in \mathcal{E}} b_j \nabla_x c_j(x^*)$$

Note that the indices we are summing over are all within the active set $\mathcal{A}(x^*)$. By the original problem, we had that all $\nabla c_i(x^*)$, $i \in \mathcal{A}(x^*)$ were linearly independent. Thus, $a_i = b_i = 0$. Therefore, we have a contradiction, so the modified problem satisfied LICQ.

- \Leftarrow :

Now we will assume that the modified problem, [3](#), satisfies LICQ. We will then show that the original problem, [1](#), also satisfies LICQ. Thus, we have

$$\nabla_y c_i(x^*) = \left\{ \begin{bmatrix} \nabla_x c_j(x^*) \\ -2z_j^* \vec{e}_j \end{bmatrix}, \begin{bmatrix} \nabla_x c_k(x^*) \\ \vec{0} \end{bmatrix} \right\}_{i \in \mathcal{I} \cup \mathcal{E}, j \in \mathcal{I}, k \in \mathcal{E}}$$

is a linearly independent set. Clearly, any subset of the above set is linearly independent. That is, for $j \in \mathcal{I} \cap \mathcal{A}(x^*)$, we have that the set is linearly independent. We are choosing indices i which are only within the active set, therefore, we have that

$$\left\{ \begin{bmatrix} \nabla_x c_i(x^*) \\ -2z_i^* \vec{e}_i \end{bmatrix}, \begin{bmatrix} \nabla_x c_i(x^*) \\ \vec{0} \end{bmatrix} \right\}_{i \in \mathcal{A}(x^*)}$$

are linearly independent. Since $i \in \mathcal{A}(x^*)$, we have that $c_i(x^*) = 0, i \in \mathcal{I} \cap \mathcal{A}(x^*)$. By the relation found in part 2, $z_i^* = 0$. Therefore, the set

$$\left\{ \begin{bmatrix} \nabla_x c_i(x^*) \\ \vec{0} \end{bmatrix}, \begin{bmatrix} \nabla_x c_i(x^*) \\ \vec{0} \end{bmatrix} \right\}_{i \in \mathcal{A}(x^*)}$$

is linearly independent. Clearly, this implies that $\{\nabla c_i(x^*)\}$ is linearly independent for $i \in \mathcal{A}(x^*)$. Therefore, the original problem satisfies LICQ as well.

Problem 1, part 4

Write down the first-order optimality conditions from part 1 for the problem in part 2, and infer the results of Theorem 12.1 for the original problem EXCEPT for the sign of the multiplier for which we need second order conditions.

Solution:

From Theorem 12.1, the optimality conditions for the modified problem are the following: f and c_i are continuously differentiable and LICQ holds at x^* . That is,

$$\nabla_y c_i(x^*) = \left\{ \begin{bmatrix} \nabla_x c_j(x^*) \\ -2z_j^* \vec{e}_j \end{bmatrix}, \begin{bmatrix} \nabla_x c_k(x^*) \\ \vec{0} \end{bmatrix} \right\}_{i \in \mathcal{I} \cup \mathcal{E}, j \in \mathcal{I}, k \in \mathcal{E}}$$

are linearly independent. We saw that in part 3, the statement that this is a linearly independent set is equivalent to the original problem's gradient constraints being linearly independent. Furthermore, since all c_i 's are continuously differentiable, (the only indices changed are those in $i \in \mathcal{I}$, which has the addition of a continuously differentiable function, $(z_i)^2$), these conditions are satisfied. Therefore, we can infer the results of Theorem 12.1 as there existing a $\lambda_i^*, i \in \mathcal{E} \cup \mathcal{I}$ such that

$$\nabla_y \mathcal{L}(y^*, \lambda^*) = 0,$$

$$c_i(x^*) = 0, \quad \text{for all } i \in \mathcal{E},$$

$$c_i(x^*) - (z_i^*)^2 = 0, \quad \text{for all } i \in \mathcal{I},$$

$$\lambda_i^* c_i(x^*) = 0, \quad \text{for all } i \in \mathcal{E},$$

$$\lambda_i^* (c_i(x^*) - (z_i^*)^2) = 0, \quad \text{for all } i \in \mathcal{I}.$$

Note that instead of writing $c_i(y^*)$, I have written $c_i(x^*) - (z_i^*)^2$. These are equivalent.

Problem 1, part 5

Write down the second-order optimality conditions from part 1 for the problem in part 2 and infer something that looks *almost* like the result of Theorem 12.5 for the original problem.

Solution:

The conditions a problem must satisfy for Theorem 12.5 to hold are that x^* being a local solution of the problem and that LICQ conditions hold. Note that our modified problem is written in terms of $y = \begin{bmatrix} x & z \end{bmatrix}^\top$, thus we assume there is a local solution y^* . Furthermore, we have that LICQ holds when the original problem satisfies LICQ. Assuming this holds, we have that the Lagrange multiplier λ^* from the KKT conditions exists, for which

$$w^\top \nabla_{xx}^2 \mathcal{L}(y^*, \lambda^*) w \geq 0, \quad \text{for all } w \in C(y^*, \lambda^*).$$

Here, C denotes the convex cone of the solution (y^*, λ^*) , written as

$$C(y^*, \lambda^*) = \{w \in \mathcal{F}(y^*) : \nabla_i c_i(y^*)^\top w = 0, \text{ for all } i \in \mathcal{A}(y^*) \cap \mathcal{I} \text{ with } \lambda_i^* > 0\}.$$

For $i \in \mathcal{A}(y^*) \cap \mathcal{I}$, we have that $c_i(y^*) = 0$, per the constraints written above. Then, we have that $\lambda_i^* > 0$ for all $i \in \mathcal{I}$. So all $i \in \mathcal{A}(y^*) \cap \mathcal{I}$ satisfy $\lambda_i^* > 0$. We can then condense this down to just saying

$$C(y^*, \lambda^*) = \{w \in \mathcal{F}(y^*)\}$$

Which implies the convex cone is the same as the feasible set, $C(y^*, \lambda^*) = \mathcal{F}(y^*)$. Thus by Lemma 12.2, when the original problem satisfies LICQ, we have that the tangent cone is equivalent to the convex cone. Thus the results of the modified problem are similar to the results in the original problem.

Problem 1, part 6

Explain under what circumstances the conditions you obtained at 5 are identical to the ones at Theorem 12.5 and likely is for that to occur.

Solution:

An obvious instance when the two results would coincide is when we do not have any inequality constraints. This would of course depend on the problem, and not entirely illustrative to the modified problem. Another choice is when $\mathcal{I} \setminus (\mathcal{A}(y^*) \cap \mathcal{I}) = \phi$, that is, all inequality constraints lead to having an equality constraint in the modified problem. In this case, the results should coincide. The likeliness of this happening is honestly unknown to me.

Problem 2

Formalize and prove the argument I gave in class that the truncated CG from the line-search Newton CG algorithm (Algorithm 7.1) always produces a descent direction at step j provided $d_j^\top B_k d_j > 0$ (and this has, of course, to hold for all iteration before l .) Recall in this setup we do not assume B_k be positive definite, only symmetric.

Solution:

This will be rationalized via induction on j . Note that the algorithm will have its inner loop terminate when $d_{j+1}^\top B_k d_{j+1} \leq 0$, or when the residual r_{j+1} has its norm smaller than some tolerance. Thus we will investigate the $d_{j+1}^\top B_k d_{j+1}$ term in each induction step, since the latter condition returns a descent direction.

- Base case: $j = 0$

If $j = 0$, then we have two cases for the algorithm's termination, which depends on the size of $\|r_{j+1}\|$. If $\|r_1\| \geq \varepsilon_k$, then we consider the expression $d_{j+1}^\top B_k d_{j+1}$. Since we discussed that $d_{j+1}^\top B_k d_{j+1} \leq 0$, then we will return z_1 , which is simply $z_0 + \alpha_0 d_0$. By initial conditions, we have $z_0 = 0$, $d_0 = -\nabla f_k$, $r_0 = \nabla f_k$. Therefore, we have

$$z_1 = \alpha_0 d_0 = \left(\frac{r_0^\top r_0}{d_0^\top B_k d_0} \right) (-\nabla f_k)$$

Since $d_0^\top B_k d_0 > 0$ by the given assumptions, we have that the returned search direction, z_1 is just a scaled steepest descent, which is certainly a descent direction. If $\|r_1\| < \varepsilon_k$, we will similarly return z_1 , which was shown to be a descent direction. Therefore, we will return a descent direction for $j = 0$.

- Induction Hypothesis:

We will now assume this holds for some $l < j$, and we will show that the $l + 1$ case holds. Note in this case, the algorithm will terminate at the $l + 2$ step, so we will similarly have two choices to consider. If $\|r_{l+2}\| < \varepsilon_k$, we will return z_{l+2} , else we return z_{l+2} in the next iteration of the inner loop, since $d_{l+2}^\top B_k d_{l+2} \leq 0$ is assumed. Therefore, the inner loop will return z_{l+2} . Note that

$$z_{l+2} = z_{l+1} + \alpha_{l+1} d_{l+1} = z_{l+1} + \left(\frac{r_{l+1}^\top r_{l+1}}{d_{l+1}^\top B_k d_{l+1}} \right) d_{l+1}.$$

By the above inference, z_{l+1} will be the descent direction produced by case l . Therefore, by the induction hypothesis, it is a descent direction. Furthermore, we have that $d_{l+1}^\top B_k d_{l+1} > 0$, since the algorithm will terminate on step $l + 2$. Therefore, the scalar in parenthesis is ≥ 0 . If $r_{l+1}^\top r_{l+1} = 0$, then $z_{l+2} = z_{l+1}$, which is a descent direction. Else, we increase z_{l+1} by a scaled d_{l+1} , which is a descent direction by the previous iteration of the loop. Therefore, we return a descent direction, proving the statement by induction.

Problem 3

Assume that, when applying the conjugate gradient method, your algorithm runs to completion taking all possible n steps. Note that the residuals r_0, \dots, r_{n-1} form a complete orthogonal basis. Prove that, in this basis, the matrix A of the original problem has a tridiagonal shape. When using this to recursively compute r_{i+1} from r_i, r_{i-1} , this is called the Lanczos iteration, and it is used to create a smaller matrix that approximated the spectrum of A .

Solution:

Let R be the matrix whose columns are those of the residuals, that is

$$R = \begin{bmatrix} | & & | \\ r_0 & \cdots & r_{n-1} \\ | & & | \end{bmatrix}.$$

Since $r_i^\top r_j = 0$ for $i \neq j$, R is an orthogonal matrix. Therefore, $R^{-1} = R^\top$ ². We can then take R to be a change of basis matrix, so that

$$A_R = R^\top A_E R,$$

where A_E is the matrix A in the standard basis and A_R is the matrix with respect to the residuals. Therefore, we seek to investigate the problem

$$r_i^\top A r_j, \quad i, j \leq n$$

Our goal is to then show that for $|i - j| \leq 1$, $r_i^\top A r_j \neq 0$. Since i is arbitrary (for $i = 1, n$ end terms will not be defined, but wouldn't matter in the end), we see that the only nonzero terms are those which hug the diagonal of width 1 - implying tridiagonality. For the case $i = j$, we obviously have that $r_i^\top A r_i \neq 0$, since A is positive definite. We will then investigate the elements for which $(A_R)_{i,j}$, $j = i \pm 1$.

- Case $j = i + 1$:

We will be using the conditions below for both parts:

$$p_k = -r_k + \beta_{k-1} p_{k-1} \tag{4}$$

$$r_{k+1} = r_k + \alpha_k A p_k \tag{5}$$

$$\alpha_k = -\frac{r_0^\top p_k}{p_k^\top A p_k} \tag{6}$$

$$\beta_k = \frac{p_{k-1}^\top A r_k}{p_{k-1}^\top A p_{k-1}} \tag{7}$$

²Note that this is not exactly true, since $\|r_i\|$ is not necessarily 1. This can be taken without loss of generality, as the proof will not rely on this fact.

We can then write the following:³

$$\begin{aligned}
 r_k^\top A r_{k+1} &= (p_k^\top - \beta_{k-1} p_{k-1}^\top) A (p_k - \beta_{k-1} p_{k-1}) && \text{(By above.)} \\
 &= p_k^\top A p_k - \beta_{k-1} p_k^\top A p_{k-1} - \beta_{k-1} p_{k-1}^\top A p_k + \beta_{k-1}^2 p_{k-1}^\top A p_{k-1} \\
 &= p_k^\top A p_k + \beta_{k-1}^2 p_{k-1}^\top A p_{k-1} && \text{(Orthogonality.)} \\
 &\neq 0 && \text{(A is PD.)}
 \end{aligned}$$

- $j = i - 1$:

$$\begin{aligned}
 r_k^\top A r_{k-1} &= (p_k^\top - \beta_{k-1} p_{k-1}^\top) A (p_{k-1} - \beta_{k-2} p_{k-2}) && \text{(By above.)} \\
 &= p_k^\top A p_{k-1} - \beta_{k-2} p_k^\top A p_{k-2} - \beta_{k-1} p_{k-1}^\top A p_{k-1} + \beta_{k-1} \beta_{k-2} p_{k-1}^\top A p_{k-2} \\
 &= -\beta_{k-1} p_{k-1}^\top A p_{k-1} && \text{(Orthogonality.)} \\
 &= -p_{k-2}^\top A r_{k-1} && \text{(By above.)} \\
 &\neq 0 && \text{(If not, } \beta = 0 \text{.)}
 \end{aligned}$$

- $j = i - 2$:

$$\begin{aligned}
 r_k^\top A r_{k-2} &= (p_k^\top - \beta_{k-1} p_{k-1}^\top) A (p_{k-2} - \beta_{k-3} p_{k-3}) && \text{(By above.)} \\
 &= 0 && \text{(No } p_k \text{ terms are alike.)}
 \end{aligned}$$

- $j = i + 2$:

$$\begin{aligned}
 r_k^\top A r_{k+2} &= (p_k^\top - \beta_{k-1} p_{k-1}^\top) A (p_{k+2} - \beta_{k+1} p_{k+1}) && \text{(By above.)} \\
 &= 0 && \text{(No } p_k \text{ terms are alike.)}
 \end{aligned}$$

Note that for the terms where $j > i + 1$, no terms of p_k will be alike. Since the p_k terms increase with j , then this will hold for all greater j . The same argument holds for $j < i - 1$, just in the opposite direction.

³Note that I will swap to the convention used in the book and in lectures, where the i index is replaced with k .

Problem 4

Write a program that implements the BFGS algorithm using backtracking line search with damping safeguarding.

Problem 4, part a

Apply this algorithm to the Rosenbrock problem from homework 4, for increasing dimension of the problem and report the number of function evaluations.

Solution:

I have implemented the BFGS algorithm as stated, and have reported the number of function evaluations in Figure 1. Note that the initial point was set to $x_0 = 2[1, \dots, 1]$, the scaled ones vector, for appropriate dimensions. The tolerance on the normed gradient is set to $1e - 8$. Note that for BFGS, we require a line search to find the maximal decrease for each search, therefore the number of function evaluations needed is variable per iteration. NOTE that the maximum number of iterations for each BFGS call is set to 200 for all dimensions. Because of this, we see a plateau on the number of function evaluation calls. If we set the max number of iterations such that this is not a problem, the plot should be linear. I tried doing this, but Matlab crashed at the exact moment of saving the plot (this is what actually happened) and so I decided to explain it rather than wait another 30 minutes for my code to run.

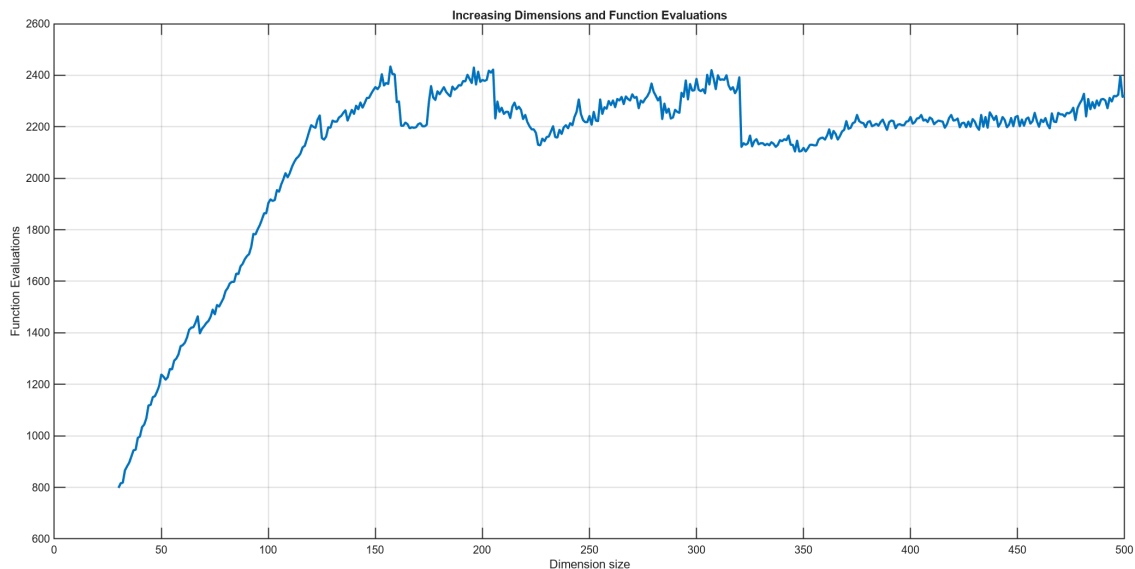


Figure 1: BFGS function evaluation for increasing dimensions.

Problem 4, part b

Suggest an empirical method to decide whether you observe superlinear convergence and apply it.

Solution:

Because we are in a numerical scenario, it isn't feasible to take ratios of increasingly small numbers. I have then two figures to convince us that there is superlinear convergence. Note that, for a general formula to find the power of convergence, we have

$$q \approx \frac{\log\left(\frac{e_{n+1}}{e_n}\right)}{\log\left(\frac{e_n}{e_{n-1}}\right)}$$

We can also just look at the value of the sequence $\|x_k - x^*\|$, if it converges faster than linear in a log scale, then we can reasonably say we have superlinear convergence. Figure 2 shows this for the initial dimensions and conditions described above. It *seems* that we have superlinear convergence in this scenario, as this curve decreases faster than linear in the log scale. We also have the q ratio test described above: It's not the most convincing since the plot jumps around too much to determine superlinear convergence $q \in (1, 2)$. This is shown in Figure 3. The value of q usually stays between -1 and 2 , which is odd, but is approximately correct. Apologies for the formatting on the second plot, it was a last minute inclusion.

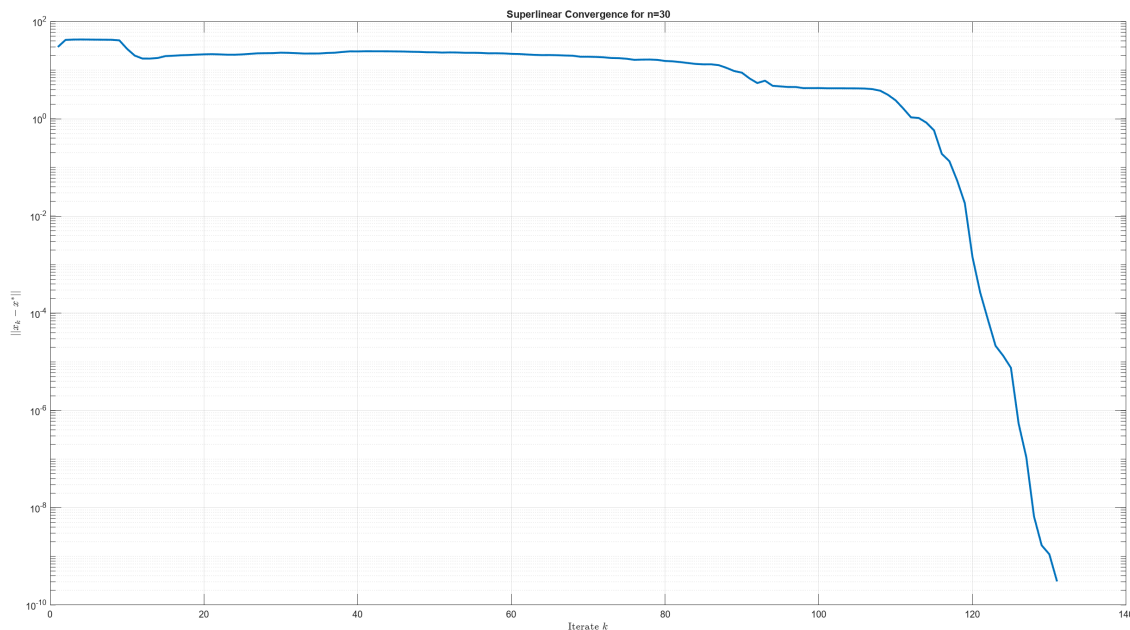


Figure 2: Value of the sequence of iterates $\|x_k - x^*\|$.

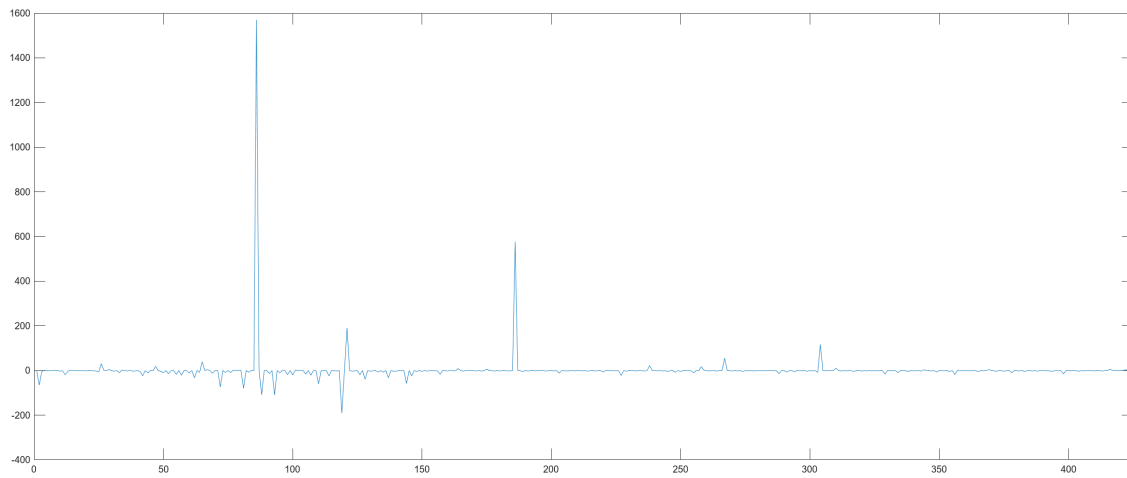


Figure 3: Ratio of convergence.