Final Internship Report

A project submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
INFORMATION TECHNOLOGY

Under Supervision of
Dr. Sathya Peri
Associate Professor, CSE Department
IIT Hyderabad

Submitted by:
Dheeraj
Roll No.:11912020
Bracnch: IT

Submitted To:
Dr. Sourabh Jain
Assistant Professor, IIIT Sonepat

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SONEPAT -131201, HARYANA, INDIA

ACKNOWLEDGEMENT

First, I would like to thank **Dr. Sathya Peri**, Faculty-in-Charge, PDCRL CSE Department, IIT Hyderabad for giving me the opportunity to do an internship within the organization.

I also would like all the people that worked along with me **Mr. Rahul Utkoor, Mr. Soumyajit Chatterjee** with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Director **Dr**. **M.N Doja** and Faculties **Dr. Sourabh Jain** and **Dr. Mukesh Mann**, for the facilities provided to accomplish this internship.

I am extremely great full to my department staff members and friends who helped me in successful completion of this internship.

I also owe a sense of gratitude to my parents for their encouragement and support throughout the project.

Name: Dheeraj

Roll No.: 11912020

Branch: IT

4th Year

CERTIFICATE

DEPARTMENT OF INFORMATION TECHNOLOGY
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SONEPAT

This is to certify that the "**Internship report"** submitted by **Dheeraj (Roll No.: 11912020)** is work done by her and submitted during 2022-2023 academic year, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology,** at **to IITH, Parallel & Distributed Computing Research Lab (PDCRL) internship program.**

Name: Dheeraj

Roll No.: 11912020

Branch: IT

4th Year

Joining Report

Format of Joining Report

Indian Institute of information Technology, Sonepat

Internship

JOINING REPORT

Date of Joining The Internship Station 5|01|2022

Period of Internship	From	To	Total Months
	5|01|2022	5|05|2022	4 months
Student Information	Name	Roll No	Branch
	Dheeraj	11912020	IT
	Student's Signature with Date Dheeraj 29|01|2022		
Name and Address of the Internship Station	IITH, Parallel & Distributed Computing Research Lab (PDCRL) internship program Kandi-502285, Sangareddy, Telangana		
Location of the Project	Kandi-502285, Sangareddy, Telangana INDIA (Work from home)		
Name and Designation of the Industry Guide/ Industry Mentor for the Project	Dr. Sathya Peri, Faculty-in-charge, PDCRL CSE Department, IIT Hyderabad		
	Signature of Industry Mentor		
Industry Mentor Contact No.	+917032397980		
Industry Mentor E-mail Address	sathya_p@cse.iith.ac.in		

17

Certificate of Internship

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

17th Aug 2022

TO WHOM IT MAY CONCERN

This is to certify that **Dheeraj** has done his internship in IITH, Parallel & Distributed Computing Research Lab (PDCRL) internship program from 5th Jan 2022 to 5 May 2022 (Duration 4 months).

He had worked on a project titled **Numerical Methods**. This project was aimed to work on optimizing Linear and Non-Linear Programming Problems in **SQP (Sequential Quadratic Programming)** and in other Algorithms. As part of the project, he contributed well in optimizing Algorithms.

During the internship he demonstrated good performance with a self-motivated attitude to learn new things. His performance during the internship was good and was able to complete the project successfully on time. We wish him all the best for his future endeavors.

Warm regards,

Dr. Sathya Peri
Associate Professor
CSE Department
IIT Hyderabad

sathya_p@cse.iith.ac.in
https://iith.ac.in/
Indian Institute of Technology Hyderabad
Kandi – 502285, Sangareddy, Telangana, INDIA

Objective

The main objective of my project is to create a gauss seidel based SQP Algorithms. Sequential quadratic programming (SQP) is an iterative method for constrained nonlinear optimization. SQP methods are used on mathematical problems for which the objective function and the constraints are twice continuously differentiable.

SQP methods solve a sequence of optimization subproblems, each of which optimizes a quadratic model of the objective subject to a linearization of the constraints. If the problem is unconstrained, then the method reduces to Newton's method for finding a point where the gradient of the objective vanishes. If the problem has only equality constraints, then the method is equivalent to applying Newton's method to the first-order optimality conditions.

Problem Statement

The nlopt library is using SQP which is using Jacobian style of iteration in order to find out solution which is slow method to update variables, also it uses inefficient data structures, also does not use parallel way of doing things and contains some redundant operations. My main objective of my project is to identify Jacobian style of iteration in SQP algorithm and convert it to gauss seidel to make the algorithm efficient in terms of time and number of iterations.

SQP: Sequential quadratic programming (SQP) is an iterative method for constrained nonlinear optimization. SQP methods are used on mathematical problems for which the objective function and the constraints are twice continuously differentiable.

NLopt is a free/open-source library for **nonlinear optimization**, providing a common interface for a number of different free optimization routines available online as well as original implementations of various other algorithms. Its features include:

- Callable from C, C++, Fortran, Matlab or GNU Octave, Python, GNU Guile, Julia, GNU R, Lua, OCaml and Rust.
- A common interface for many different algorithms—try a different algorithm just by changing one parameter.
- Support for large-scale optimization (some algorithms scalable to millions of parameters and thousands of constraints).
- Both global and local optimization algorithms.
- Algorithms using function values only (derivative-free) and also algorithms exploiting user-supplied gradients.
- Algorithms for unconstrained optimization, bound-constrained optimization, and general nonlinear inequality/equality constraints.
- Free/open-source software under the GNU LGPL (and looser licenses for some portions of NLopt).
- https://nlopt.readthedocs.io/en/latest/

Company Profile

Indian Institute of Technology, Hyderabad (abbreviated **IIT Hyderabad** or **IITH**) is a public technical university located in Kandi village of Sanga Reddy district in the Indian state of Telangana.

IITH was founded in 2008, among the eight young Indian Institutes of Technology (IITs). It has a total of 3,903 students (1,553 Undergraduate, 1,221 Masters and 1,129 PhD students) with 255 full-time faculty members as Jan 15, 2022.

The IITH campus is on a land area of 576 acres (234 ha). The academic building is designed by New Delhi based ARCOP and hostel by Pune-based acclaimed American architect, Prof. Christopher Charles Benninger. This organic campus is divided into clusters of buildings being completed in phases starting in 2011. The campus is one of India's best examples of energy efficient, carbon neutral and sustainable architecture. The design grew out of local weather conditions and utmost care to enhance learning. The graduate and post-graduate programs are separated, student and teacher housing is divided, and girls and boys hostels are segregated for a pluralistic environment.

The 25 lakh square feet of buildings in Phase 1A; 3 academic blocks and 10 functioning hostel buildings (each with a capacity of 200) were completed in March 2019.

IIT Hyderabad Research Park

In 2018, Government of India sanctioned Research Park to some IITs in which IIT Hyderabad got its place. IIT Hyderabad Research Park is a self-reliant team endorsed by IIT Hyderabad and its alumni. The IIT Hyderabad Research Park promotes the betterment of research and development by the institute through friendship with industry, helping in the advancement of modern ventures, and build-up economic development. The IIT Hyderabad Research Park assists organizations with a research target to set up a infrastructure in the park and advantage the expertise available at IIT Hyderabad.

IIT Hyderabad Technology Research Park (TRP) with a total Built-up area is 19,560 square meters was recently (On 4th of February 2022) inaugurated by Srivari Chandrasekhar, secretary, Department of Science & Technology, Government of India (GoI).

LIST OF ABBREVIATIONS

SQP	Sequential Quadratic Programming
S/W	Software
PDCRL	Parallel & Distributed Computing Research Lab
BFGS	Broyden-Fletcher-Goldfarb-Shanno method

LIST OF FIGURES

Table of Contents

1. Introduction

This report is a short description of my four-month internship carried out as compulsory component of the Bachelor of Technology in Information Technology. The internship was carried out with IIT Hyderabad. IIT Madras was also involved in this research Since my I am interested in Programming and Mathematics, the work was concentrated on the Linear Programming and Non-Linear Optimization, the main objective of my internship is to design gauss seidel based SQP Algorithms

The nlopt library is using SQP which is using Jacobian style of iteration in order to find out solution which is slow method to update variables, also it uses inefficient data structures, also does not use parallel way of doing things and contains some redundant operations. My main objective of my project is to identify Jacobian style of iteration in SQP algorithm and convert it to gauss seidel to make the algorithm efficient in terms of time and number of iterations.

At the beginning of the internship, I formulated several learning goals, which I wanted to achieve:

- to understand the functioning and working conditions of a governmental organization
- to see what is like to work in a professional environment
- to see if this kind of work is a possibility for my future career
- to use my gained skills and knowledge
- to see what skills and knowledge I still need to work in a professional environment
- to learn about the organizing of a research project (planning, preparation, permissions etc.)
- to learn about research methodologies
- to get fieldwork experience in an environment unknown for me
- to get experience in working in another state with persons from another culture
- to enhance my communication skills;
- to build a network.

This internship report contains my activities that have contributed to achieve a number of my stated goals. In the following chapter a description of the organization IIT Hyderabad, Parallel & Distributed Computing Research Lab (PDCRL) and the activities is given. After this a reflection on my functioning, the unexpected circumstances and the learning goals achieved during the internship are described. Finally, I give a conclusion on the internship experience according to my learning goals.

2. Description of the internship

2.1 The organization IIT Hyderabad, Parallel & Distributed Computing Research Lab (PDCRL): Indian Institute of Technology, Hyderabad (abbreviated **IIT Hyderabad** or **IITH**) is a public technical university located in Kandi village of Sanga Reddy district in the Indian state of Telangana.

IITH was founded in 2008, among the eight young Indian Institutes of Technology (IITs). It has a total of 3,903 students (1,553 Undergraduate, 1,221 Masters and 1,129 PhD students) with 255 full-time faculty members as Jan 15, 2022.

The IITH campus is on a land area of 576 acres (234 ha). The academic building is designed by New Delhi based ARCOP and hostel by Pune-based acclaimed American architect, Prof. Christopher Charles Benninger. This organic campus is divided into clusters of buildings being completed in phases starting in 2011. The campus is one of India's best examples of energy efficient, carbon neutral and sustainable architecture. The design grew out of local weather conditions and utmost care to enhance learning. The graduate and post-graduate programs are separated, student and teacher housing is divided, and girls and boys hostels are segregated for a pluralistic environment.

The 25 lakh square feet of buildings in Phase 1A; 3 academic blocks and 10 functioning hostel buildings (each with a capacity of 200) were completed in March 2019.

IIT Hyderabad Research Park: In 2018, Government of India sanctioned Research Park to some IITs in which IIT Hyderabad got its place. IIT Hyderabad Research Park is a self-reliant team endorsed by IIT Hyderabad and its alumni. The IIT Hyderabad Research Park promotes the betterment of research and development by the institute through friendship with industry, helping in the advancement of modern ventures, and build-up economic development. The IIT Hyderabad Research Park assists organizations with a research target to set up a infrastructure in the park and advantage the expertise available at IIT Hyderabad.

IIT Hyderabad Technology Research Park (TRP) with a total Built-up area is 19,560 square meters was recently (On 4th of February 2022) inaugurated by Srivari Chandrasekhar, secretary, Department of Science & Technology, Government of India (GoI).

2.2 Internship activities: the main objective of my internship is to design gauss seidel based SQP Algorithms The nlopt library is using SQP which is using Jacobian style of iteration in order to find out solution which is slow method to update variables, also it uses inefficient data structures, also does not use parallel way of doing things and contains some redundant operations. My main objective of my project is to identify Jacobian style of iteration in SQP algorithm and convert it to gauss seidel to make the algorithm efficient in terms of time and number of iterations.

2.2.1 Perquisite Knowledge

2.2.1.1 Necessary condition for obtaining Minimum

First Order Necessary condition

- Let $f : R^n \rightarrow R$, $f \in C^1$
  - If x* is a local minimum of f, then g(x*) = 0.

Second Order Necessary condition

- Let $f : R^n \rightarrow R$, $f \in C^2$
- If x* is a local minimum of f
  - g(x*) = 0 and H(x*) is positive semi definite.

Second Order Sufficient condition

- Let $f : R^n \rightarrow R$, $f \in C^2$
- If g(x*) = 0 and H(x*) is positive definite
  - x* is a strict local minimum of f.

2.2.1.2 What is descent direction?

- Let $f \in C^1$, $x \in R^n$ and $g(x^*) = \nabla f(x^*)$.
  - If $g(x^*)^T d<0$
    - d is a descent direction.
- At that point x*
  - d should make obtuse angle with the gradient direction

Suppose we that is surface plot of any function f(x) below is the contours of the function f(x)

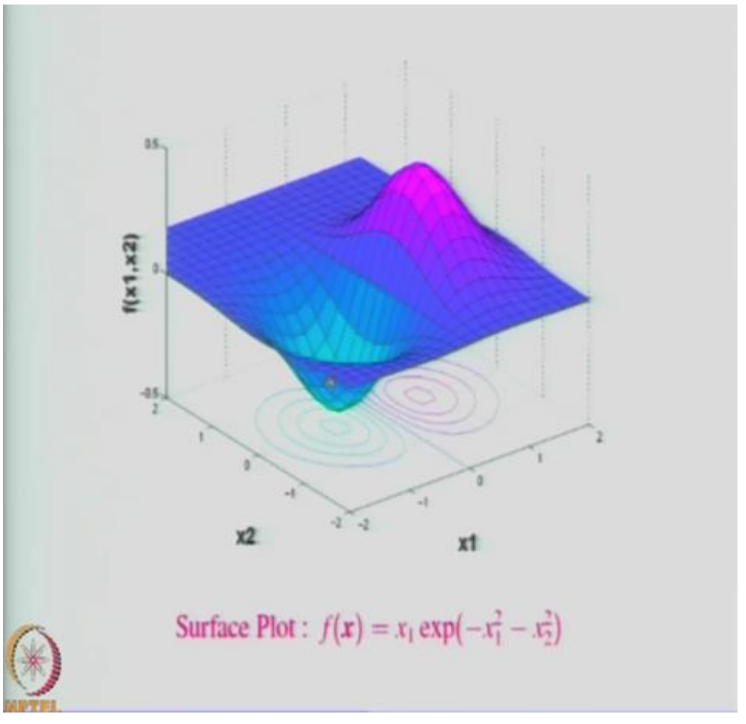


Fig 1

First image is contour plot of f(x),

We can see here that in the first image value of the function decreases from -0.1 to -0.2 then -0.3 then -0.4. and at the center there is local minimum because if we move away from this in any direction function values increases.

The center of 2nd contour is local maximum because if we move away from the center then our function value will decrease.

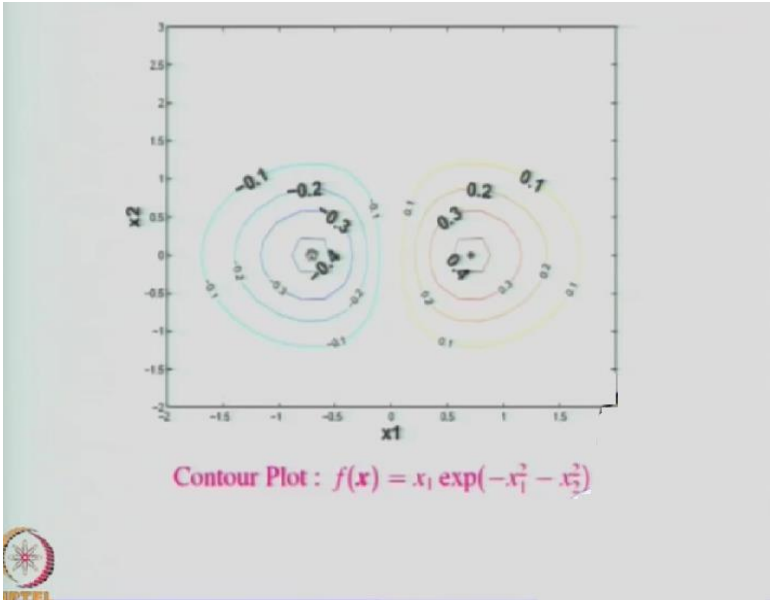


Fig 2

Now let's observe the descent direction in figure 2nd

Here we can see that this is surface plot of function f(x) at point $x = x_0$.

Gradient of f(x) at x_0 is shown by sky-blue line. Now descent direction is set of all direction that make obtuse angle with gradient direction.

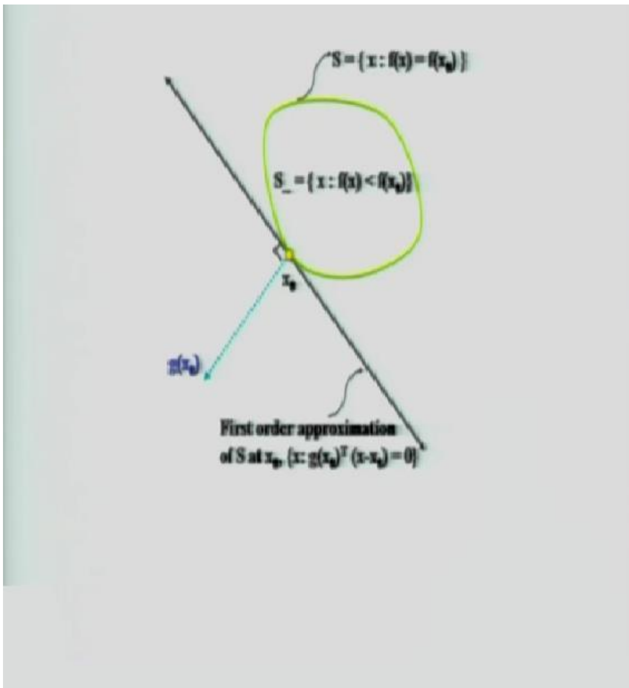


Fig 3

Let us Now compare Jacobi iteration method and Gauss seidel method

In order to compare Jacobi iteration method and Gauss seidel method, I implemented a program of Jacobi iteration method and gauss seidel method which can also take n variable equation. That program automatically stops when we find our exact solution.

I take simple mathematical equation in order to compare these two methods. After Testing the Gauss seidel and Jacobi method these are the results we obtain

2.2.1.3 What is Armijo-Wolfe or Armijo Goldstein condition and why they required

- Sometimes the steps we take to converge to a solution are very small.
- Another problem is there is small decrease in function value.
- These two problems may cause for the algorithm to never converge or converge in very long time
- To tackle these problems, we use Armijo's Goldstein or Armijo's Wolfe condition.

2.2.1.4 What is Armijo condition

- In Armijo condition we choose αk such that there is sufficient decrease in function value.

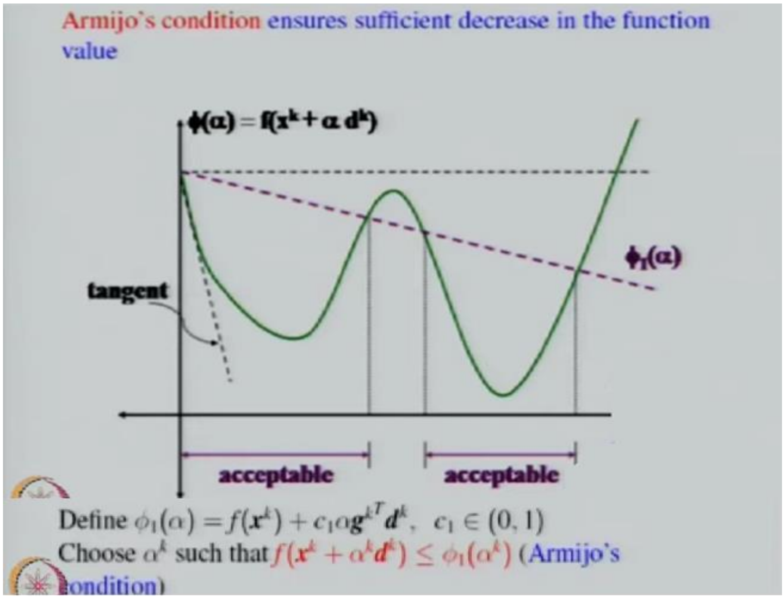


Fig 4

2.2.1.5 Goldstein condition

- In Goldstein condition we choose αk such that step lengths are not too small

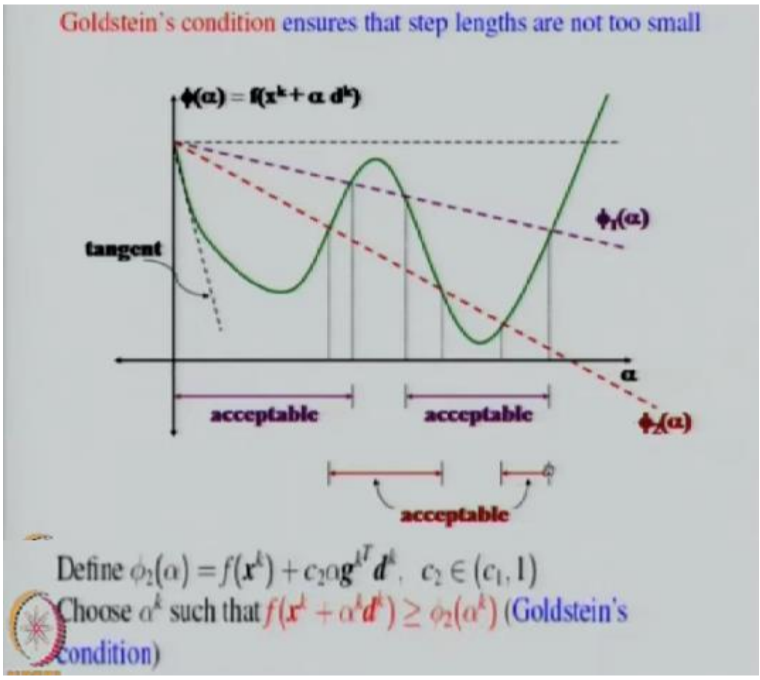


Fig 5

2.2.1.6 Armijo’s Goldstein condition

In Armijo Goldstein conditions we choose αk such that there is sufficient decrease in function value and choose αk such that step lengths are not too small

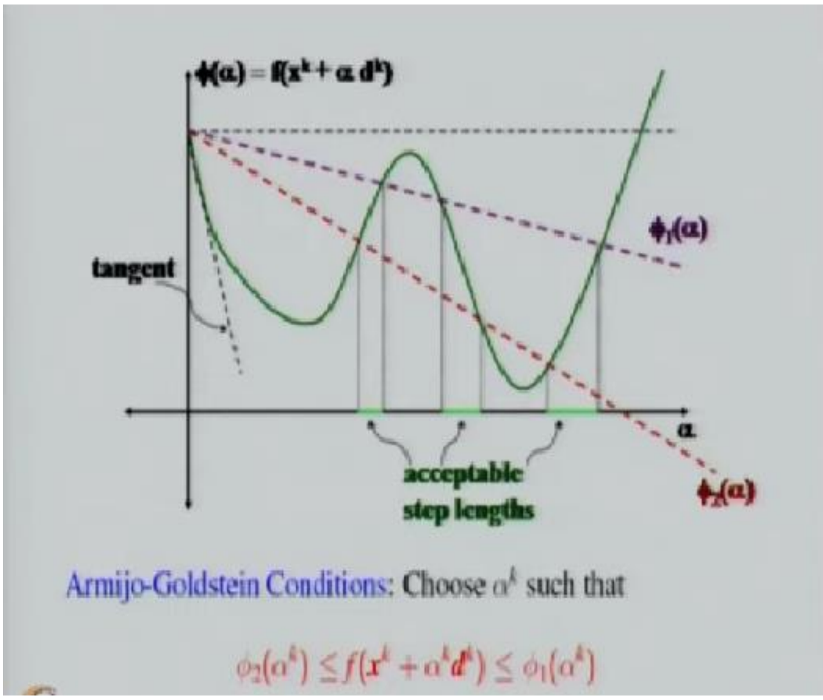


Fig 6

2.2.1.7 If we have Armijo’s Goldstein condition then why we require Wolfe condition

Because one local minimum is missed by Goldstein condition.

2.2.1.8 Wolfe condition

In Wolfe condition we choose αk such that there is sufficient rate of decrease of function value in the given direction

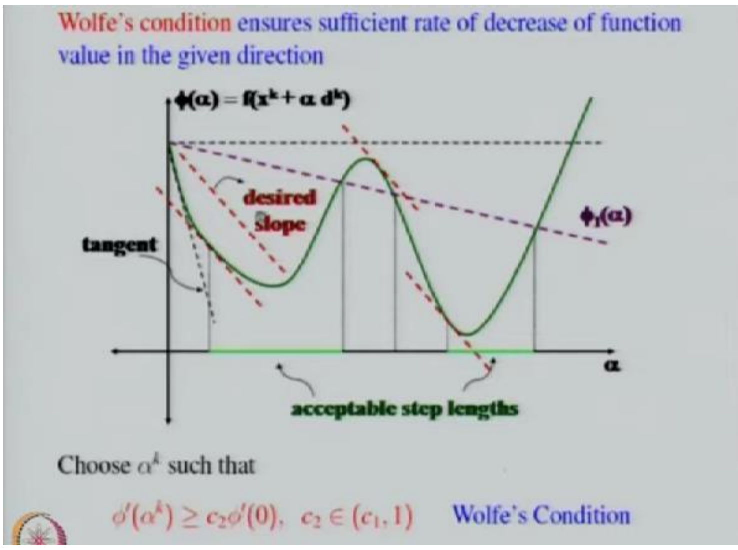
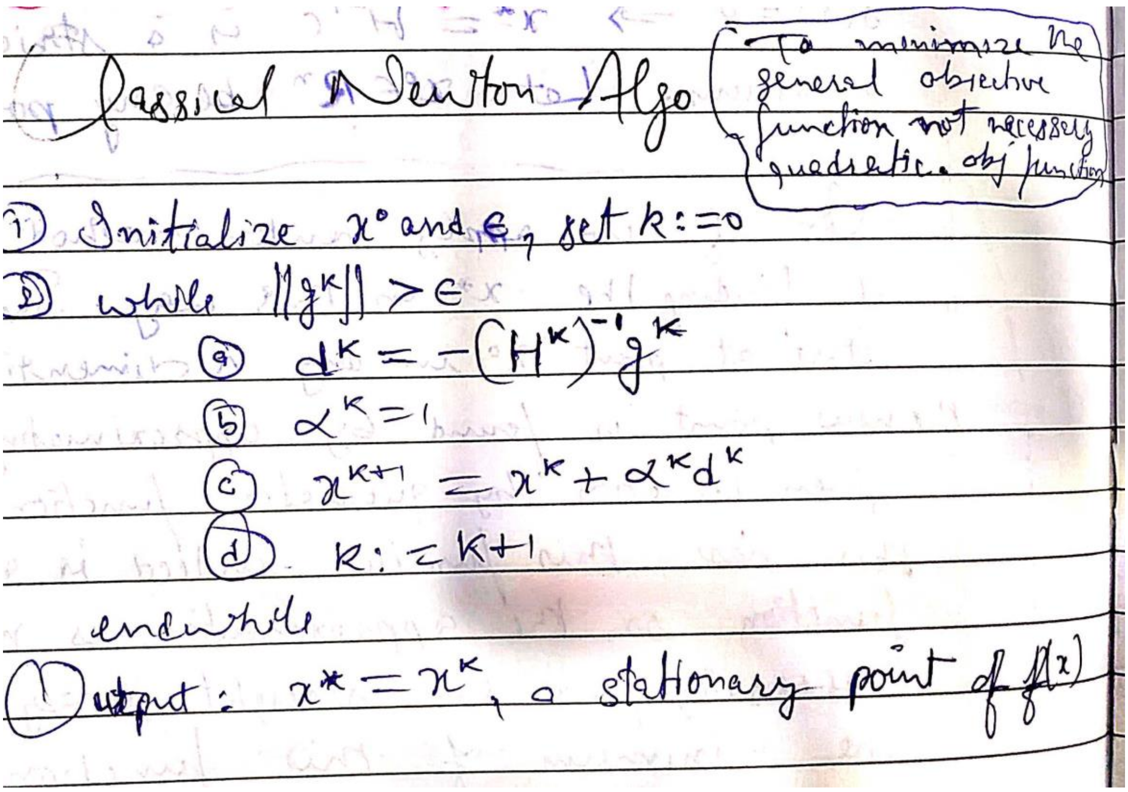


Fig 7

2.2.2 Classical Newton Algorithm


Classical Newton Algo

To minimize the general objective function not necessarily quadratic obj function

1. Initialize x^0 and ϵ, set $k := 0$
2. while $\|g^k\| > \epsilon$
   - (a) $d^k = -(H^k)^{-1} g^k$
   - (b) $\alpha^k = 1$
   - (c) $x^{k+1} = x^k + \alpha^k d^k$
   - (d) $k := k+1$

endwhile

Output: $x^* = x^k$, a stationary point of $f(x)$

Fig 8

In classical newton method We don't use any condition that was satisfied by alpha that is step length.

In this algo First we initialize x_0 that is initial point, epsilon that is our stopping condition of the algo, and set iteration k =0

The while loop will work until gradient of the function is greater than epsilon

Then in the while loop we find our descent direction $d^k = -H^{k-1}g^k$.

Here g^k is the gradient and H^k is the hessian matrix. Hessian matrix contain 2nd order differential info of f(x).

Then we set step length alpha-k to 1

Then we find our new point x^{k+1} by using formula $x^{k+1} = x^k + \alpha.\ d^k$

then we increase our iteration counter.

Then at last we find our local minimum.

Drawbacks of classical Newton Algorithm

- We require 2nd order info & inversion of matrix that is computationally expensive.
- No guarantee that d^k is descent direction.
- Inversion of a matrix which is close to singular matrix will give rise to numerical difficulties.
- No guarantee that $f(x^{k+1})<f(x^k)$
- Sensitive to initial point (Sometimes may not converge to solution)

2.2.3 Quasi Newton Algorithm

Given f is continuously differentiable, form a quadratic model of f at x^k.

$y_k(x)= f(x^k) + g^{kT}(x-x^k) + ½(x-x^k)^T B^{k-1}(x-x^k)$.

Where B^k is symmetric matrix.

Quasi Newton direction $d^k_{QN}=-B^k g^k$

$x^{k} = x^k + \alpha^k d^k_{QN} = x^k-\alpha B^k g^k$.

Given x^k, x^k, g^k , g^k and B^k we can get B^{k+1} by using formula

Quasi newton condition $B^{k+1}\Upsilon^k= \delta^k$. (to satisfy $\nabla y_{k+1}(x) = \nabla f(x^k)$ && $\nabla y_{k+1}(x^{k+1}) = \nabla f(x^{k+1}) = g^{k+1}$)

This condition is already satisfied in B^{k+1} formula.

$$B_{SR1}^{k+1} = B^k + \frac{(\delta^k - B^k\gamma^k)(\delta^k - B^k\gamma^k)^T}{(\delta^k - B^k\gamma^k)^T\gamma^k}$$

Where $\gamma^k = g^{k+1} - g^k$, $\delta^k = x^{k+1} - x^k$

Fig 9

Quasi-Newton Algo (rank one correction)

1) Initialize x^0, ϵ and symmetric +ve definite B^0, set $k := 0$ (can use Identity matrix as B^0)

2) while $\|g^k\| > \epsilon$

(a) $d^k = -B^k g^k$

(b) Find $\alpha^k (>0)$ along d^k such that

(i) $f(x^k + \alpha^k d^k) < f(x^k)$

(ii) α^k satisfies Armijo-Wolfe (or Armijo-Goldstein) conditions.

(c) $x^{k+1} = x^k + \alpha^k d^k$

(d) Find B^{k+1} using rank one correction

(e) $k := k+1$

endwhile

Output: $x^* = x^k$, a stationary point of $f(x)$

Fig 10

First we initialize x0 that is initial point, epsilon that is our stopping condition of the algo, symmetric matrix B^0 and set iteration k =0

The while loop will work until gradient of the function is greater than epsilon

Then in the while loop we find our descent direction $d^k = -B^k g^k$.

Here g^k is the gradient and B^k is the symmetric matrix which approximate hessian matrix.

Then we find out step length alpha-k (that is >0) along d^k such that

The function value at current point is less than function value at previous point and alpha-k satisfies Armijo-Goldstein or Armijo's Wolfe condition.

Then we find our new point x^{k+1} by using formula $x^{k+1} = x^k + \alpha^k . d^k$

then we increase our iteration counter.

Then at last we find our local minimum.

Drawbacks of Quasi Newton Algorithm

- B^{k+1} is +ve definite if $(\delta^k - B^k \Upsilon^k)^T \Upsilon^k > 0$
- Numerical difficulties if $(\delta^k - B^k \Upsilon^k)^T \Upsilon^k = 0$

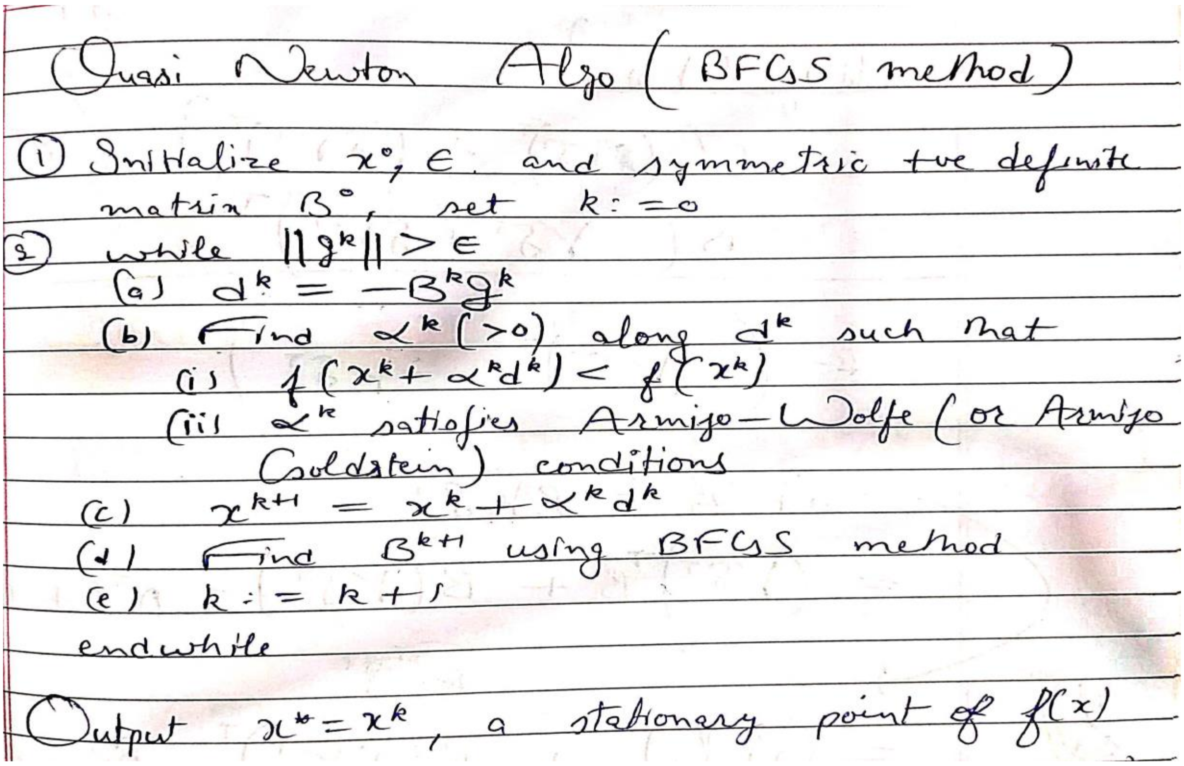
2.2.4 Rank two correction Methods or Algorithms

Davidon-Fletcher Power (DFP) method.

Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. (Good results and perform better)

BFGS algorithm was used in SQP and also it perform better and shown good results.

2.2.5 BFGS Algorithm


(Quasi Newton Algo (BFGS method)

① Initialize x^0, ϵ and symmetric +ve definite matrix B^0, set $k := 0$

② while $\|g^k\| > \epsilon$

(a) $d^k = -B^k g^k$

(b) Find $\alpha^k (>0)$ along d^k such that

(i) $f(x^k + \alpha^k d^k) < f(x^k)$

(ii) α^k satisfies Armijo–Wolfe (or Armijo Goldstein) conditions

(c) $x^{k+1} = x^k + \alpha^k d^k$

(d) Find B^{k+1} using BFGS method

(e) $k := k+1$

endwhile

Output $x^* = x^k$, a stationary point of $f(x)$

Fig 11

First, we initialize x0 that is initial point, epsilon that is our stopping condition of the algo, symmetric +ve definite matrix B^0 and set iteration k =0

The while loop will work until gradient of the function is greater than epsilon

Then in the while loop we find our descent direction $d^k = -B^k g^k$.

Here g^k is the gradient and B^k is the symmetric matrix.

Then we find out step length alpha-k (that is >0) along d^k such that

The function value at current point is less than function value at previous point and alpha-k satisfies Armijo-Goldstein and Armijo's Wolfe condition.

Then we find our new point x^{k+1} by using formula $x^{k+1} = x^k + \alpha^k . d^k$

Then we find our B^{k+1} matrix using BFGS method.

then we increase our iteration counter.

Then at last we find our local minimum.

What is B^{k+1}?

- B^{k+1} is symmetric positive definite matrix. That is obtained using adding two symmetric matrix of rank one. That why BFGS method fall in the category of rank two correction method.

$$B^{k+1}_{BFGS} = B + \left(1 + \frac{\gamma^T B \gamma}{\delta^T \gamma}\right)\frac{\delta \delta^T}{\delta^T \gamma} - \left(\frac{\delta \gamma^T B + B \gamma \delta^T}{\delta^T \gamma}\right)$$

Fig 12

- Where B is symmetric matrix at kth iteration
- $\Upsilon^k = g^{k+1} - g^k$, $\delta^k = x^{k+1} - x^k$.

2.2.6 Jacobi Iteration method:

Main idea of Jacobi

To begin, solve the 1st equation for x_1, the 2nd equation for x_2 and so on to obtain the rewritten equations:

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \cdots a_{1n}x_n)$$
$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \cdots a_{2n}x_n)$$
$$\vdots$$
$$x_n = \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots a_{n,n-1}x_{n-1})$$

Then make an initial guess of the solution $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots x_n^{(0)})$. Substitute these values into the right hand side the of the rewritten equations to obtain the *first approximation,* $\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots x_n^{(1)}\right)$.

This accomplishes one **iteration**.

In the same way, the *second approximation* $\left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \dots x_n^{(2)}\right)$ is computed by substituting the first approximation's x-vales into the right hand side of the rewritten equations.

By repeated iterations, we form a sequence of approximations $\boldsymbol{x}^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots x_n^{(k)}\right)^t, \quad k = 1,2,3, \dots$

Use the Jacobi method to approximate the solution of the following system of linear equations.

$$\begin{aligned} 5x_1 - 2x_2 + 3x_3 &= -1 \\ -3x_1 + 9x_2 + x_3 &= 2 \\ 2x_1 - x_2 - 7x_3 &= 3 \end{aligned}$$

Continue the iterations until two successive approximations are identical when rounded to three significant digits.

To begin, write the system in the form

$$\begin{aligned} x_1 &= -\tfrac{1}{5} + \tfrac{2}{5}x_2 - \tfrac{3}{5}x_3 \\ x_2 &= \tfrac{2}{9} + \tfrac{3}{9}x_1 - \tfrac{1}{9}x_3 \\ x_3 &= -\tfrac{3}{7} + \tfrac{2}{7}x_1 - \tfrac{1}{7}x_2. \end{aligned}$$

Because you do not know the actual solution, choose

$$x_1 = 0, \qquad x_2 = 0, \qquad x_3 = 0$$ **Initial approximation**

as a convenient initial approximation. So, the first approximation is

$$\begin{aligned} x_1 &= -\tfrac{1}{5} + \tfrac{2}{5}(0) - \tfrac{3}{5}(0) = -0.200 \\ x_2 &= \tfrac{2}{9} + \tfrac{3}{9}(0) - \tfrac{1}{9}(0) \approx 0.222 \\ x_3 &= -\tfrac{3}{7} + \tfrac{2}{7}(0) - \tfrac{1}{7}(0) \approx -0.429. \end{aligned}$$

Continuing this procedure, you obtain the sequence of approximations shown in Table 10.1.

TABLE 10.1

n	0	1	2	3	4	5	6	7
x_1	0.000	−0.200	0.146	0.192	0.181	0.185	0.186	0.186
x_2	0.000	0.222	0.203	0.328	0.332	0.329	0.331	0.331
x_3	0.000	−0.429	−0.517	−0.416	−0.421	−0.424	−0.423	−0.423

Fig 13

2.2.7 Gauss seidel method:

- With the Jacobi method, the values of x_i^k obtained in the kth iteration remain unchanged until the entire (k+1)th iteration has been calculated. With the Gauss-Seidel method, we use the new values $x_i^{(k+1)}$ as soon as they are known.

Use the Gauss-Seidel iteration method to approximate the solution to the system of equations given in Example 1.

The first computation is identical to that given in Example 1. That is, using $(x_1, x_2, x_3) = (0, 0, 0)$ as the initial approximation, you obtain the following new value for x_1.

$$x_1 = -\tfrac{1}{5} + \tfrac{2}{5}(0) - \tfrac{3}{5}(0) = -0.200$$

Now that you have a new value for x_1, however, use it to compute a new value for x_2. That is,

$$x_2 = \tfrac{2}{9} + \tfrac{3}{9}(-0.200) - \tfrac{1}{9}(0) \approx 0.156.$$

Similarly, use $x_1 = -0.200$ and $x_2 = 0.156$ to compute a new value for x_3. That is,

$$x_3 = -\tfrac{3}{7} + \tfrac{2}{7}(-0.200) - \tfrac{1}{7}(0.156) \approx -0.508.$$

So the first approximation is $x_1 = -0.200$, $x_2 = 0.156$, and $x_3 = -0.508$. Continued iterations produce the sequence of approximations shown in Table 10.2.

TABLE 10.2

n	0	1	2	3	4	5
x_1	0.000	−0.200	0.167	0.191	0.186	0.186
x_2	0.000	0.156	0.334	0.333	0.331	0.331
x_3	0.000	−0.508	−0.429	−0.422	−0.423	−0.423

Note that after only five iterations of the Gauss-Seidel method, you achieved the same accuracy as was obtained with seven iterations of the Jacobi method in Example 1.

Neither of the iterative methods presented in this section always converges. That is, it is possible to apply the Jacobi method or the Gauss-Seidel method to a system of linear equations and obtain a divergent sequence of approximations. In such cases, it is said that the method **diverges.**

Fig 14

2.2.8 Compare Jacobi iteration and gauss seidel method

Q1
$$\begin{aligned} 2x_1 - x_2 + 0\cdot x_3 &= 7 \\ -x_1 + 2x_2 - x_3 &= 1 \\ 0\cdot x_1 - x_2 + 2x_3 &= 1 \end{aligned}$$

Q2
$$\begin{aligned} 3x_1 - x_2 &= 2 \\ x_1 + 4x_2 &= 5 \end{aligned}$$

Q3
$$\begin{aligned} 4x_1 + x_2 - x_3 &= 3 \\ x_1 + 6x_2 - 2x_3 + x_4 - x_5 &= -6 \\ x_2 + 5x_3 - x_5 + x_6 &= -5 \\ 2x_2 + 5x_4 - x_5 - x_7 - x_8 &= 0 \\ -x_3 - x_4 + 6x_5 - x_6 - x_8 &= 12 \\ -x_3 - x_5 + 5x_6 &= -12 \\ -x_4 + 4x_7 - x_8 &= -2 \\ -x_4 - x_5 - x_7 + 5x_8 &= 2 \end{aligned}$$

Q4
$$\begin{aligned} 4x_1 - x_2 - x_3 &= 3 \\ -x_1 + 4x_2 - x_3 - x_4 &= -6 \\ -x_2 + 4x_3 - x_4 - x_5 &= -5 \\ -x_3 + 4x_4 - x_5 - x_6 &= 0 \\ -x_4 + 4x_5 - x_6 - x_7 &= 12 \\ -x_5 + 4x_6 - x_7 - x_8 &= -12 \\ -x_6 + 4x_7 - x_8 &= -2 \\ -x_7 + 4x_8 &= 2 \end{aligned}$$

Q5
$$\begin{aligned} 4x_1 - x_2 - x_3 &= 18 \\ -x_1 + 4x_2 - x_3 - x_4 &= 18 \\ -x_2 + 4x_3 - x_4 - x_5 &= 4 \\ -x_3 + 4x_4 - x_5 - x_6 &= 4 \\ -x_4 + 4x_5 - x_6 - x_7 &= 26 \\ -x_5 + 4x_6 - x_7 - x_8 &= 16 \\ -x_6 + 4x_7 - x_8 &= 10 \\ -x_7 + 4x_8 &= 32 \end{aligned}$$

Fig 15

Results

Method	Question	Iteration	Time	Variable
Jacobi	**Q1**	**47**	0.104	**3**
Gauss		**25**	0.053	
Jacobi	**Q2**	**13**	0.036	**2**
Gauss		**8**	0.013	
Jacobi	**Q3**	**27**	0.160	**8**
Gauss		**14**	0.085	
Jacobi	**Q4**	**31**	0.186	**8**
Gauss		**16**	0.099	
Jacobi	**Q5**	**34**	0.143	**8**
Gauss		**17**		

Fig 16

You can see Gauss seidel is taking almost half the iteration and half the time as compared to Jacobi method. That means converting SQP from Jacobian style method to gauss seidel method make the algorithm very efficient.

2.2.9 Gauss seidel approach in SQP

- min f (X)
  b (X) ≥ 0
  c (X) = 0

- Lagrange Multipliers:
  To incorporate the constraints we multiply each constraint with a variable and modify the original minimization problem as:
  $$Y = f(X) + \lambda.b(X) + \mu.c(X)$$

- We can see that the this equation is of type
  $y = f(x) + \lambda.b(x) + \mu.c(x)$

y= m1x1 + m2x2 + m3x3

$$x^{k+1} = x^k - \alpha B^k g^k.$$
$$\lambda^{k+1} = \lambda^k - \alpha B^k g^k.$$
$$\mu^{k+1} = \mu^k - \alpha B^k g^k.$$

- We can see that x, λ & μ are sensitive to gradient of the function y
- When you get a new or updated value of x, λ & μ update the function y in order to iterate in a faster way. If Possible, also do it for sub values of x, λ & μ

2.2.10 Conclusion: The research is still going on and continued by IIT H and IIT M faculties and Students. They are also trying to apply the gauss seidel method to other modules. And also work on parallelization of this algorithm and in reducing redundant operation and use efficient data structures.

2.2.11 Summary of internship work

- Work on SQP based gauss seidel algorithms
- Reading documentation of libraries.
- Reading Research Papers.
- Sometimes I have to Develop Algo’s from scratch
- Understanding, explaining and give ppt about Algorithms & Mathematical concepts to IIT Hyderabad and IIT Madras Faculties. Explain code to them.
- Taking feedback of any improvement in Algo.
- Online Meeting Regularly
- Make Survey on Applying Gauss seidel Algorithms to Gradient Descent for Linear Regression and Multi-Linear Regression.

3. Reflection on the internship

In this chapter I reflect on the internship. Regarding my learning goals I shortly discuss my experiences.

3.1 Learning goals

- **The functioning and working conditions of a governmental organization** At the beginning I did not have any experience of working within a **governmental organization**. Although I have seen one, I understand better the functioning like the organization structure and setting up projects.

- **Enhancing communication skills** As I had expected I experienced small language difficulties. I thought that I could communicate well in Hindi and with my good knowledge of English, however the majority of persons I worked with talked in English. Therefore, I was reserved in communication at the beginning, but in the course of months it went better. My stay has contributed to my communication skills, but I would like to pay more attention to it in the future. To contribute more to projects and to progress faster, I want to learn to make a more confident impression and to express my ideas and opinions more certain.

- **The use of skills and knowledge gained in the university:** I gain a lot of knowledge and experience from my research internship. I get knowledge about Linear Programming and Non-Linear Optimization, how to go ahead in research, how to improve time complexity of Algorithms and Gain Mathematics knowledge. I also gain knowledge about how to work in government organization.

- **Organizing projects:** Within the internship I did a lot of fieldwork. Because of this I have seen of what aspects you have to think while organising a project (equipment, boat navigators, and permissions). Furthermore, I have learned how an education program can be set up and what things have to be taken into account. It is important determine the knowledge present and to adjust the program to each group. It is of importance to convey an objective and supported message taking the viewpoints of people into account. I became also aware that local people have a lot of knowledge that could help in research and conservation. Before the internship I did not have any experience in Linear Programming and Non-Linear optimizations and I had no idea if it could work. In the future I would like to do some more problems in Linear Programming and Non-Linear optimizations, because I have seen now that people can be reached and that you can receive a lot of new insights.

- **Working in another state with persons from another culture:** The internship was an opportunity to get immersed on a deeper level in another culture. It was really experiencing to see how other people live. It also helped to look at things from a different perspective. Beside language, I did not experience difficulties caused by different cultural backgrounds. An interesting and open attitude of the people has helped. An internship abroad was also a good way for me to see whether I could work south India. I hardly experienced problems and I got really fast used to the different way of living. Also, the fieldwork came easy to me and I felt a full member of the project teams.

- **The influence on future career plans:** Before my internship in IIT Hyderabad, I had some doubts about my future career. I was excited to continue in research for one more semester but due to limited time I am unable to continue. I also did not know what type of research I would like to do. Through this internship, I have seen what elements of my career I like and I got enthusiastic again to continue in research. I have found out that part of the research should contain fieldwork as I did in the internship.

4. Conclusion

On the whole, this internship was a useful experience. I have gained new knowledge, skills and met many new people. I achieved several of my learning goals, however for some the conditions did not permit. I got insight into professional practice. I learned the different facets of working within a Government Organization. I experienced that financing, as in many organisations, is an important factor for the progress of projects. Related to my study I learned more about the **Linear Programming and Non-Linear Optimizations**. There is still a lot to discover and to improve. The methods and processes used at the moment are standardized, professional and a consistent method in optimizations Algorithms. Furthermore, I experienced that it is of importance that the education is objective and that you have to be aware of the view of other people. Education is not one sided, but it is a way of sharing knowledge, ideas and opinions. The internship was also good to find out what my strengths and weaknesses are. This helped me to define what skills and knowledge I have to improve in the coming time. It would be better that the knowledge level of the language is sufficient to contribute fully to projects. After my completion of the research internship, I gain a lot of knowledge about Linear Programming and Non-Linear Optimization. I did my best in this research-based work at IIT Hyderabad, Parallel & Distributed Computing Research Lab (PDCRL) internship program and contribute in optimizing Algorithms. It would also be better if I can present and express myself more confidently. At last, this internship has given me new insights and motivation to pursue a career in future.

References

1. *Jacobi Iterative Method*. (n.d.). Byjus. Retrieved September 1, 2022, from https://byjus.com/maths/jacobian-method/
2. *The Jacobi and Gauss-Seidel Iterative Methods*. (n.d.). Nd.Edu. Retrieved September 1, 2022, from https://www3.nd.edu/~zxu2/acms40390F12/Lec-7.3.pdf
3. *The Jacobi and Gauss-Seidel Iterative Methods*. (n.d.). Nd.Edu. Retrieved September 1, 2022, from https://www3.nd.edu/~zxu2/acms40390F12/Lec-7.3.pdf
4. *Iterative Methods Jacobi and Gauss-Seidel*. (n.d.). BYJUS. Retrieved September 1, 2022, from https://byjus.com/maths/iterative-methods-gauss-seidel-and-jacobi/
5. *NLopt - NLopt Documentation*. (n.d.). Nlopt.Readthedocs.Io. Retrieved September 1, 2022, from https://nlopt.readthedocs.io/en/latest/
6. *Nonlinear programming*. (2022, June 15). Wikipedia. Retrieved September 1, 2022, from https://en.wikipedia.org/wiki/Nonlinear_programming
7. Shi, A. (2022, February 26). *How To Define A Neural Network as A Mathematical Function*. Medium. Retrieved September 1, 2022, 1.from https://towardsdatascience.com/how-to-define-a-neural-network-as-a-mathematical-function-f7b820cde3f
8. GeeksforGeeks. (2022, January 18). *How to implement a gradient descent in Python to find a local minimum?* Retrieved September 1, 2022, from https://www.geeksforgeeks.org/how-to-implement-a-gradient-descent-in-python-to-find-a-local-minimum/
9. *SQP*. (2022, March 23). Wikipedia. Retrieved September 1, 2022, from https://en.wikipedia.org/wiki/SQP

10. Lithmee, B. (2019, January 3). *What is the Difference Between Linear and Nonlinear Programming*. Pediaa.Com. Retrieved September 1, 2022, from https://pediaa.com/what-is-the-difference-between-linear-and-nonlinear-programming/

11. Niu, F., Recht, B., R´E, C., & J. Wright, S. (2011, June). *Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent*. Arxiv.Org. https://arxiv.org/pdf/1106.5730.pdf