INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
RICE UNIVERSITY

Optimization for Parameter Estimation with Application to Transmission Electron Microscopy

by

Olena Sinkevich

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Richard A. Tapia, Chairman
Noah Harding Professor of Computational and Applied Mathematics

Yin Zhang, Co-Chairman
Associate Professor of Computational and Applied Mathematics

George N. Phillips, Jr.
Professor of Biochemistry and Cell Biology

Steven J. Ludtke
Research Associate at Baylor College of Medicine

Houston, Texas
July, 2000
Abstract

Optimization for Parameter Estimation with Application to Transmission Electron Microscopy

by

Olena Sinkevich

We consider a parameter estimation problem for an important model in structural molecular biology. We propose two new mathematical formulations for the problem as constrained nonlinear least-squares problems, develop a numerical algorithm for solving this problem using interior-point methodology, and prove the convergence results for nonlinear least-squares problems with general constraints. Through numerical experimentation we show that our approach to the parameter determination problem is more effective than previous methods.
Acknowledgments

This thesis is dedicated to my parents, Dr. Ludmila Bychkova and Dr. Oleg Sinkevich, who have always given me their love and support.

I would like to express my most sincere and heartfelt acknowledgments to my advisors, Dr. Richard A. Tapia and Dr. Yin Zhang. I would like to thank Dr. Zhang for his willingness to help, for making himself available, and for his kindness and support. I am most grateful to Dr. Tapia for supporting me professionally, emotionally, and in many other ways. Dr. Tapia’s wisdom, experience, and willingness to share them is very much appreciated. I also thank him for his understanding and support of my interests in psychology, art, and other areas outside mathematics. I feel very privileged to work with two such prominent scientists and wonderful, warm, and supportive human beings. I could not have wished for better advisors.

I would like to thank Dr. George Phillips and Steve Lutdke for serving on my committee.

I would like to thank many people in the Department of the Computational and Applied Mathematics. I thank Dr. Amr El-Bakry for numerous discussions we had and for the time that he gave me, no matter how busy he was. Michael Pearlman, Daria Lawrence, Fran Moshiri, and Linda Neyra have been invaluable. I would like to thank Keith Berrier, Erica Zimmer, Diane Jamrog, and many other present and former students in the department.

The last but not the least I would like to thank my host families: Chuck and Doris Wolter, Waldo and Irene Asp, Pat and Jerry Adams, and Jane and Cleon Dunham. Without Chuck Wolter I probably would have never come to the US.
I have appreciated and enjoyed the opportunities that I had in this country (I mean the US, not just Texas), in Houston, at Rice, and in the Department of Computational and Applied Mathematics.
## Contents

Abstract ii
Acknowledgments iii
List of Illustrations viii
List of Tables ix

1 Introduction 1

2 Electron Microscopy 3
   2.1 Image Formation ........................................... 7
   2.2 The Contrast Transfer Function (CTF) .................. 10

3 CTF and Total Intensity Model 12
   3.1 Contrast Transfer Function Model ....................... 12
   3.2 Noise And Total Intensity Model ....................... 14

4 Previous Work on the CTF Parameter Estimation

   Problem 16
   4.1 "Manual" Curve-Fitting Procedure .................... 16
   4.2 Problems with the Current Approach .................. 17

5 Mathematical Formulation of the Problem 20
   5.1 The Nonlinear Least-Squares Problem .................. 21
   5.2 Mathematical Formulations ............................. 22
### 6 Interior-Point Algorithm

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>The Nonlinear Programming Problem and Lagrange Multiplier Background</td>
<td>29</td>
</tr>
<tr>
<td>6.2</td>
<td>The Karush-Kuhn-Tucker (KKT) Conditions</td>
<td>30</td>
</tr>
<tr>
<td>6.3</td>
<td>The Interior-Point Method</td>
<td>31</td>
</tr>
<tr>
<td>6.4</td>
<td>Slack-Variable Form of the KKT Conditions</td>
<td>33</td>
</tr>
<tr>
<td>6.5</td>
<td>An Interior-Point Algorithm</td>
<td>34</td>
</tr>
</tbody>
</table>

### 7 Theoretical Convergence Analysis

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Previous Local Convergence Results</td>
<td>38</td>
</tr>
<tr>
<td>7.2</td>
<td>Convergence Rates</td>
<td>39</td>
</tr>
<tr>
<td>7.3</td>
<td>Local Convergence Properties</td>
<td>40</td>
</tr>
</tbody>
</table>

### 8 Numerical Experiments

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Implementation</td>
<td>51</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Parameters</td>
<td>52</td>
</tr>
<tr>
<td>8.1.2</td>
<td>User-Supplied Information</td>
<td>53</td>
</tr>
<tr>
<td>8.1.3</td>
<td>Bounds</td>
<td>54</td>
</tr>
<tr>
<td>8.1.4</td>
<td>Scaling</td>
<td>54</td>
</tr>
<tr>
<td>8.2</td>
<td>Singularity</td>
<td>54</td>
</tr>
<tr>
<td>8.3</td>
<td>Initial Guess</td>
<td>56</td>
</tr>
<tr>
<td>8.4</td>
<td>Other Approaches</td>
<td>57</td>
</tr>
<tr>
<td>8.5</td>
<td>Data</td>
<td>58</td>
</tr>
<tr>
<td>8.5.1</td>
<td>Human Fatty Acid Synthase</td>
<td>59</td>
</tr>
<tr>
<td>8.5.2</td>
<td>Herpes Virus Particles</td>
<td>59</td>
</tr>
<tr>
<td>8.5.3</td>
<td>α-crystallin</td>
<td>59</td>
</tr>
</tbody>
</table>
8.6 Numerical Results ........................................... 61
8.7 Summary ..................................................... 62

9 Conclusions .................................................. 71

Bibliography ................................................... 73

A Matlab Implementation of the Algorithm ............... 77
# Illustrations

2.1 Comparison of light and transmission electron microscopes. .................. 6
2.2 Electron-specimen interaction. .................................................. 8
2.3 An object before and after application of CTF. .............................. 11

4.1 "Manual" data fit with the known structure factor. ......................... 18

8.1 Structure of the Jacobian matrix $J$. ........................................... 55
8.2 Data fit for human fatty acid synthase. ...................................... 64
8.3 Comparison of data fits for human fatty acid synthase. .................. 65
8.4 Data fit for the herpes simplex virus (HSV-1). ............................. 66
8.5 Comparison of data fits for the herpes simplex virus (HSV-1). .......... 67
8.6 Data fit for $\alpha$-crystallin A. ................................................ 68
8.7 Data fit for $\alpha$-crystallin AB. ................................................ 69
8.8 Data fit for $\alpha$-crystallin B. .................................................. 70
Tables

8.1 Computational results for the sample particles. 61
Chapter 1

Introduction

Knowledge of protein structure is essential in many areas of biology. A few specific examples of applications include virus studies, understanding the operation of muscles on a molecular level, and studies of a protein contained in the eye lens, whose aggregation causes a cataract.

Transmission Electron Microscopy is one of the primary techniques used to determine macromolecular structure (such as for proteins and viruses). Since the microscope is not a perfect optical system, the projected images are not exact representations of the sample particle. This effect is quantified as the contrast transfer function (CTF) of the microscope. In order to allow optimal image correction it is necessary to approximate the parameters of both the CTF and a noise function, as well as the so-called structure factor* values of the molecule. The density information recovered from this corrected image is thus improved. Currently, the parameter identification process performed in the laboratories is tedious, time consuming and uses ad-hoc procedures.

We present a method for determining the CTF and noise function by estimating their parameters while simultaneously determining the structure factor. We propose two new mathematically sound formulations based on two key ideas: (i) the use of multiple data sets and (ii) the explicit imposition of bounds on the parameters. We treat the problem as a constrained nonlinear least-squares minimization problem where we demand that the corrected structure factor be the same in different micrographs of the same molecule. Combined with a functional form for the CTF and

*Structure factor contains the electron density information.
background noise, this provides sufficient information to simultaneously determine the CTF parameters and the corrected structure factor of the molecule.

We construct a numerical algorithm for solving this problem using a new interior-point methodology, which deals efficiently with the bounds on the variables. Theoretical convergence results are proved and supporting numerical results are provided.

The thesis is organized as follows: the next chapter explains electron microscopy and the phenomena involved; the CTF is also introduced. In Chapter 3, the CTF, noise, and total intensity models are discussed. Chapter 4 gives the overview of the previous work on the subject. The new proposed mathematical formulations are introduced and discussed in Chapter 5. Chapter 6 explains the background for the interior-point methods and introduces the interior-point algorithm developed for the parameter estimation problem. Our convergence results are proved in Chapter 7, and the numerical results are discussed in Chapter 8. The last chapter summarizes the work done.
Chapter 2

Electron Microscopy

Proteins and protein assemblies are the subjects of the single particle analysis. Protein is one of the major functional constituents of the cell, along with DNA, RNA, lipids (which form cellular membranes), and small messenger molecules. DNA contains information about how to make individual proteins. DNA can be viewed as a string formed from a four letter alphabet, where each segment of 3 letters describes an amino-acid and has the information about the beginning and the end of each protein. Proteins consist of combinations of 20 amino-acids joined in a linear chain, which is then folded into a very complex, but reproducible, structure. Primary structure of a protein describes which amino-acids are attached to each other. Secondary structure describes the local structural organization of groups of 5 or 6 amino acids. Tertiary and quaternary structures describe the overall shape of the molecule.

The size of the particles being examined can vary from less than 100Å to about 1μm. Proteins are generally less than 300Å. For instance, the size of the protein that causes cataracts is 140Å. Assemblies, such as viruses, may be as large as 0.1-0.2μm.

Transmission electron microscopy, along with X-ray crystallography, is one of the primary techniques used to study molecular structure, i.e., the structure of individual components within the cell (see, for example, Chiu [6]). Light-based microscopy, which is used to study cellular structure, cannot resolve such small structures. All methods consist of data collection and application of reconstruction techniques to retrieve the structural information about the particles under study (see, for example,

---

1Each amino-acid is a molecule.
1Å=10^{-10}m, 1μm=10^{-6}m.
Kam [15]). The data for electron microscopy is two-dimensional density projections of individual particles, so called micrographs. For X-ray crystallography, the data is the intensity of diffracted X-rays. Each step of data collection and reconstruction is a very complicated and involved process.

In X-ray crystallography, first, the particle is crystallized, X-rays are diffracted through the particle and the diffraction pattern is recorded. The scattering process effectively performs a Fourier transform of the density. The diffraction pattern provides the intensity of the Fourier transform of the density function. The phase of the X-ray wave cannot be determined directly, therefore the image in real space is impossible to regenerate because of the absence of phase information necessary for the inverse Fourier transform. The resolution provided by X-ray crystallography is currently better than the resolution achieved by electron microscopy. However, it is difficult to crystallize large molecules. Determining the correct crystallization process for a new protein may take months or even years.

Unlike X-ray crystallography in electron microscopy, “an optical system is needed to transform the information about the specimen that the electrons have acquired into a visible image” (Frank [10]). The biggest advantage of electron microscopy is the collection of the image in real space. Scientists working in X-ray crystallography are facing the challenge of finding ways to get the lost information, which is not an issue in electron microscopy.

“Three-dimensional electron microscopy refers to the combination of the instrument and the necessary computational tools to obtain a three-dimensional image of the object’s interior”[10]. Electron cryo\textsuperscript{5} microscopy works by rapidly freezing the particle in order to avoid movement and to capture the native state of protein or

\textsuperscript{5}Cryo comes from the Greek word meaning “cold".
virus. Different functional states can easily be acquired. The particle does not need to possess any kind of symmetry or order.

Transmission electron microscopy produces two-dimensional images which represent projections of the density of the three-dimensional object. Electron microscopy is similar to light microscopy, but the electrons rather than photons are scattered by the specimen. For comparison of light and transmission electron microscopes see Figure 2.1 [19].

The resolution limit of electron microscope is about a 1000 times better than that of a light microscope. The electron microscope resolution needed for atomic reconstruction (i.e. the finest level of detail) is about 1.5Å. In reality, a typical initial reconstruction will be performed at about 25Å resolution. This is then gradually improved to about 8Å.

The size and intensity of the electron beam can be varied using a sequence of lenses before the sample. The beam is directed through the sample which scatters the electrons. The resulting beam is sent through several additional lenses to generate a real space image of the specimen.

The data collection time for electron microscopy is short, on the order of a second for a single micrograph. Each micrograph corresponds to a different settings of microscope focus. Data consists of several hundred particles.

Current experimental limitations include structural inhomogeneity of the molecules in question, electron radiation damage, and specimen movement under the beam. The main resolution limiting factors are beam coherence, sample motion/charging and imperfections in the lens.

When reconstructing the image there are several factors that must be considered: geometrical location of the data, extraction of signal from noise, image correction, sufficient data sampling, and reliability of reconstruction.
Figure 2.1 Comparison of light and transmission electron microscopes.
2.1 Image Formation

Image formation is a complex process. The raw images are not faithful representations of the particle [10]. The microscope induces artifacts that need to be corrected. It is important to know the conditions that lead to best image as well as the types of computational correction that are needed to recover as much of the original information as is possible.

The basis of image formation is the interaction of the electrons with the object (see Figure 2.2 [19]). The theory of image formation states that the wave function at the diffraction plane of an electron lens is the Fourier transform of the object's projected potential function (electron density function); and that the image intensity at the image plane of an electron lens is linearly related to the inverse Fourier transform of the wave function at the diffraction plane convoluted with a Contrast Transfer Function (CTF) of the microscope.

We distinguish between elastic and inelastic scattering. The former involves no transfer of energy, has a fairly wide angular distribution, and gives rise to high resolution information. The latter involves transfer of energy, its angular distribution is narrow, and it produces an undesired background term in the image. Because this term occurs at low resolution, it is normally tolerated. If an electron experiences both elastic and inelastic interactions, one can say that only the inelastic interaction took place. The probability of two elastic interactions is very small, so we are not concerned with the overlapping effect of multiple elastic interactions.

In the wave-optical picture, the elastic scattering of the electron with the object is depicted as a phase shift $\Phi(r)$ of the incoming wave traveling in the $z$ direction by

$$\Phi(r) = \int_{-\infty}^{\infty} \Phi_3D(r, z)dz,$$  \hspace{1cm} (2.1)
Figure 2.2  Electron-specimen interaction.
where \( r \) is a two-dimensional vector \( r = [x, y]^T \), and \( \Phi_{3D}(r, z) \) is the three-dimensional Coulomb potential distribution within the object. Thus, the incoming plane wave \( \psi = e^{ikz} \) is modified according to

\[
\psi' = \psi e^{i\Phi(r)}.
\]  

(2.2)

The weak phase approximation assumes that \( \Phi(r) \ll 1 \), enabling the expansion

\[
\psi' = \psi[1 + i\Phi(r) - \frac{1}{2}\Phi(r)^2 + \cdots],
\]

(2.3)

which is usually truncated after the second term. This form implies a decomposition of the wave behind the object into an "unmodified" or "unscattered wave" (the first term) and a "scattered wave" (the term \( i\Phi(r) \) and the following terms). The wave function past the objective lens is - in the ideal case - the Fourier transform of the modified equation or its weak phase (phase shift is very small) approximation.

However, the lens aberrations and the defocusing have the effect of shifting the phase of the scattered wave by the term dependent upon the scattering angle and the spatial frequency, \( s = \{s_x, s_y\} \) [1]. An ideal lens will transform an incoming plane wave into a spherical wavefront converging into a single point on the back focal plane. The wave aberration has the effect of deforming the spherical wavefront.

In summary, the wave front function in the back focal plane can be written as the Fourier transform of the wave function immediately behind the object times a term that represents the phase shift due to lens aberrations.

Next, the wave function in the image plane is obtained from the wave in the back focal plane, after an aperture modification, through another Fourier transform. Finally, the observed intensity distribution in the image plane is the product of the above function with its conjugate. In the weak phase approximation, the image intensity is dominated by a term representing the phase contrast that is caused by
elastic scattering (interference of the "unmodified wave" with the "scattered wave"). Amplitude contrast between points in the sample is caused by the absorption of electrons by the sample. Phase contrast provides no information about large features of the object.

Both amplitude and phase information (needed for the inverse Fourier transformation) are available. It leads to image whose contrast is linearly related to the projected object potential.

2.2 The Contrast Transfer Function (CTF)

The electron microscope modifies the true signal. Erickson and Klug (1971) [1] and Hawkes [12] present how these instrumental modifications can be modeled, within certain limitations, in terms of the Contrast Transfer Function (CTF).

CTF correction is the first part of image processing. Each particle image is two-dimensionally Fourier transformed, then the set of images is incoherently averaged (that is, the Fourier intensities are averaged) to reduce the noise. The final average is then rotationally averaged to produce a function of radius only. Each micrograph is characterized by a signal intensity curve which varies with spatial frequency.

CTF describes the relationship between the three-dimensional object and the two-dimensional image obtained from it. In theory, the product of CTF and the Fourier transform of the object is the Fourier transform of the projection. The CTF is the Fourier transform of the point-spread function which is used for reconstruction. Both functions carry the same information but it is easier to work with the CTF in the Fourier space [1]. The real space image is the convolution of projection and point-spread function.

CTF has a small value over an extended range of low spatial frequencies. The effect of this property on the image is that particle as a whole does not stand out
from the background, but its edges are sharply defined by contours, and short-range interior density variations are exaggerated. Other effects include the inversion of contrast of larger areas, and the appearance of fringes with alternating contrast along borders. Figure 3.4 ([10]) shows an object before (left figure) and after (right figure) application of CTF.

![Image of object before and after application of CTF](image)

**Figure 2.3** An object before and after application of CTF.
Chapter 3

CTF and Total Intensity Model

3.1 Contrast Transfer Function Model

The Contrast Transfer Function for microscope is traditionally modeled by

\[ \text{CTF}(s) = \text{Amp} \times e^{-B s^2} (\sqrt{1 - C_A^2} \sin \gamma(s) + C_A \cos \gamma(s)), \]  

where \( s \) is the spatial frequency (measured in Å \(^{-1}\)), \( \text{Amp} \) is the overall amplitude, and \( C_A \) is the amplitude contrast factor. Parameter \( C_A \) is a unitless quantity that describes the relative amount of elastic and inelastic scattering. It depends on the specimen and the microscope in use. The term \( \sqrt{1 - C_A^2} \sin \gamma(s) \) describes the phase contrast and the \( C_A \cos \gamma(s) \) term describes the absorption (i.e. amplitude contrast). The term \( e^{-B s^2} \) is an envelope function.

Theoretically, information is transferred in a wide range of spatial frequencies. The increasingly rapid oscillations of CTF make it difficult to exploit all of the high frequency information. In practice, as we go toward higher spatial frequencies, the CTF is damped, limiting the resolution. This effect can be described by the envelope function \( e^{-B s^2} \). Various sources contributing to the envelope function fall-off include electron microscope factors, specimen movement, photographic emulsion, digital scanner, the specimen itself. Even the skills of the microscope operator (e.g. waiting for too long to take a picture or not getting the defocus right) affect the envelope function fall-off. The known theoretical form for the envelope function is fairly complex. All of the parameters for the full envelope function cannot be determined unambiguously. Empirically, \( e^{-B s^2} \) functional form fits the data within the bound of error in many cases. A simple envelope function is not sufficient for very clear data because
it is difficult to fit both the second peak and right-hand part of the signal intensity curve. A better, though more complex envelope function is \( e^{-\left(k\sqrt{B_0^2 + (1-k)B^2}\right)} \), where \( k \) is a mixing factor between an exponential envelope function and a Gaussian envelope function. This function fits data taken from a wide range of microscopes.

Parameter \( B \) is a microscope dependent factor, whereas it is defined as a temperature factor in X-ray crystallography. It is weakly dependent on the defocus but in the current model it is considered to be independent.

Technically, the definition of the CTF is just

\[
\text{CTF}(s) = \sqrt{1 - C_A^2} \sin \gamma(s) + C_A \cos \gamma(s),
\]

with the amplitude \( Amp \) and the envelope function \( e^{-Bs^2} \) being separate entities. However, since the problem of finding \( C_A, \Delta z, Amp, B, \) and the noise function parameters, is referred to as the determination of the CTF parameters, we will use CTF to refer to (3.1) and not (3.2).

The factor \( \gamma(s) \) is the phase shift in reciprocal (Fourier) space for phase contrast microscopes:

\[
\gamma(s) = -2\pi \left( \frac{C_s 10^7 \lambda^3 s^4}{4} + 5000 \times \Delta z \lambda s^2 \right),
\]

where \( C_s \) is the spherical aberration of the microscope, \( \lambda = \frac{he}{\sqrt{V^2 + 2E_0 V}} \) is the wavelength of the electrons in the beam, \( V \) is the microscope voltage (in kilovolts)\(^4\), \( E_0 V \) is the rest mass of the electron, \( h \) is Plank’s constant, and \( c \) is the speed of light; \( \Delta z \) is the defocus value (in microns). Defocus is the distance between the focal plane and the sample plane. The defocus value is never known precisely for an experiment even though a specific focus is chosen. The defocus parameter varies between micrographs (recall, that micrographs are recorded two-dimensional density projections of individual particles). Actually, in an experiment perfect focus is undesirable. It

\(^4\)The value for \( V \) is either 100 kV or 400 kV for the microscopes in use.
is desirable to record images with various levels of defocus, since each contributes information corresponding to different resolutions. With perfect focus, corresponding to zero defocus, low frequency/resolution information is lost.

This dependence of the phase shift on spatial frequency (3.3) has long been known to give a good agreement with data from the microscope in the absence of astigmatism. In practice, if astigmatism is present, or if the micrograph is corrupted otherwise (such as by specimen drift), the micrograph is undesirable for the purpose of the CTF parameter determination. Therefore, we can assume there is no astigmatism present and we can use the above expression for the phase shift (3.3).

One can rewrite the CTF as:

\[
\text{CTF}(s) = A \text{mp} e^{-B s^2} \left( \sqrt{1 - C_A^2} \sin(\gamma(s)) + C_A \cos(\gamma(s)) \right)
\]
\[
= A \text{mp} e^{-B s^2} (\sin(\gamma(s) + \theta))
\]

where \( \sin(\theta) = C_A \). The parameter used in our computations is \( \theta = \arcsin(C_A) \).

### 3.2 Noise And Total Intensity Model

The functional form of the noise intensity is empirical. It includes effects of many different sources and it fits data obtained from several different microscopes well.

\[
\text{Noise Intensity}(s) = n_3 e^{-((\pi/2)n_4 s)^2 - n_1 \sqrt{s} - n_2 s},
\]

where \( n_3 \) is the squared amplitude of the noise. Since the noise is additive and (empirically) incoherent with regards to the true signal, the total intensity of the signal observed on the micrograph may be expressed by:

\[
\text{TI}(s) = \text{CTF}(s)^2 F(s) + \text{Noise Intensity}(s),
\]
where Noise Intensity is the squared average of white random noise, and \( F(s) \) is the unknown structure factor of the particles (squared Fourier transform of the projected density).

There are four unknown parameters which characterize the CTF: the microscope dependent parameter \( B \), the amplitude contrast \( C_A \), the defocus \( \Delta z \), and the overall amplitude of the signal \( Amp \). Also there are four unknown parameters describing the noise \( n_i, i = 1 \ldots 4 \). These model parameters need to be determined, along with the vector of the structure factor values \( F(s) \).

The following requirements bound the parameters. All the parameters are non-negative, with the exception of the defocus \( \Delta z \). Parameter \( B \) is practically strictly positive; otherwise, the envelope function is equal to one and there is no damping effect. Parameter \( C_A \) varies between zero and one, typically \( 0 \leq C_A \leq 0.2 \). There are soft bounds on \( \Delta z \): \(-5 \leq \Delta z \leq 5\).

The determination of \( F(s) \) as a function of spatial frequency is a difficult problem. In case \( F(s) \) is unknown, it must be determined by demanding that it is consistent between different defocus settings.
Chapter 4

Previous Work on the CTF Parameter Estimation Problem

In this chapter we describe the current approach to the problem of the CTF and noise function parameter estimation and the determination of the structure factor values and its shortcomings.

4.1 "Manual" Curve-Fitting Procedure

Currently, the parameter identification process is done by "manually" fitting the model to the measured data through trial and error (see, for example, [3]). First, one assumes that the structure factor values are all ones. Then, one guesses the parameter values based on the physical characteristics of the CTF and noise functions and applies an unconstrained local optimization method to obtain a fit. The current CTF and two noise parameters fitting procedure goes as follows. For large wave-numbers the intensity is entirely comprised of noise. By plotting the intensity, the coefficients \( n_3 \) and \( n_2 \) of \( \text{Noise Intensity} = n_3 e^{-n_2 s} \) are fitted. To manually fit the data, noise is fitted first from the tail of the curve and the zero points. When one determines the CTF "by hand", the positions of the zeroes of the CTF are measured and fitted to a chart of the CTF characteristics. The four noise parameters can easily be used up to fit the peak but one should avoid using the degrees of freedom for this purpose. If the data is not very clear, the noise parameters can be defined from the tail of the curve. Otherwise, "manual" fitting becomes almost impossible. Currently, the boundaries are enforced by manually restricting the parameter choice to sensible values.
If the fit does not look right or the parameters are out of range, then the guessing and fitting procedures are repeated again until a fit is acceptable. From the fitted parameter values, one computes a set of corresponding structure factor values. Since the structure factor values should be the same for different data sets for the same particle, the fitting procedure requires going back and forth many times between different data sets and parameter values in order to obtain consistent structure factor values.

4.2 Problems with the Current Approach

Other disadvantages of the "manual fitting" approach include the lack of a sound and precise measure for the quality of obtained parameter values and the quality of fit, the excessive sensitivity to the noise intensity parameters, no effective enforcement of parameter bounds, and the inflexibility of accommodating new parameters in the model. Previous attempts (Zhu et al. [30]) at automating the parameter estimation process mimic the "manual fitting" approach, hence sharing similar flaws.

Software developed by Steve Ludtke from the National Center for Macromolecular Imaging at Baylor College of Medicine represents the two-dimensional Fourier transform as an image, where white represents high pixel values and black represents low pixel values. The software also graphs the total intensity as a function of spatial frequency. The current data fitting tool in the Baylor software uses a local optimization method. Currently, in the Baylor fitting routine the residual is calculated in the $l_1$-norm. Some points are excluded from the calculation. The fitting routine does not work well if the envelope function is narrow, or if the defocus and/or other parameters fall out of a smaller subset of the parameter space. It also fails in the presence of distinguished peaks in the high frequency area.
If the structure factor is available from X-ray scattering, it is used in the fitting. Even then, the fit might not be very accurate. Figure 4.1 illustrates the "manual" data fit for the Human Fatty Acid Synthase when the structure factor known from X-ray scattering is used. When the X-ray data is absent, the fitting becomes even harder. The approach that we propose in the next chapter will not only allow us to perform the data fit with the structure factor being unknown, but often allows to have a better fit. One of the reasons for a more accurate fit is that there are more degrees of freedom when the structure factor is not fixed (even to a 'good' value).

**Figure 4.1** "Manual" data fit with the known structure factor.
Other attempts of fitting include the "intelligent" (similar to manual) fitting algorithm and a gradient-descent like method.

Zhu, Penczek, Scroder, and Frank [30] proposed an "intelligent" fitting method, which uses zeros of the CTF to determine the defocus and amplitude contrast parameters. The method uses the information from the peaks of the signal intensity curve to define amplitude. The authors do not specify the details of determining the envelope function and noise parameters.
Chapter 5

Mathematical Formulation of the Problem

We propose two new formulations of the problem. Both are constrained nonlinear least squares that simultaneously determine the structure factor and the Contrast Transfer Function parameters. This new approach requires multiple data sets and finds all the unknown parameters corresponding to all the involved data sets at once.

We consider the functional form of the Contrast Transfer Function and Noise Intensity function as defined in (3.4) and (3.5), respectively. Corresponding to each data set there are eight unknown parameters in the CTF and noise function and an unknown vector of structure factor values whose size is that of the data set. If only one data set is used, then the system is underdetermined. This is why it is the current practice, where a single data set is used at a time, to fix the structure factor vector to some value while fitting the other model parameters.

A key observation is that if one uses multiple data sets, then it becomes possible to determine the unknown model parameters, along with the vector of structure factor values which is invariant among different data sets that are generated for the same particle. This approach is feasible because multiple date sets for the same particle are easy to generate in electron microscopy (and actually are already available for other purposes).

This approach offers considerable advantages and it will be discussed in detail in this chapter. However, we have discussed it in preliminary form at meetings and conferences, e.g. [24], (see also [18]).

We propose to minimize the sum of the squares of the residuals arising from fitting the model to multiple data sets concurrently. The data sets are of the same size, and
the minimization is subject to the appropriate bounds on all variables. Thus we estimate the vector of the structure factor values for the sample particles, as well as the parameters of the Contrast Transfer Function and incoherent background (noise) to fit the experimental data.

5.1 The Nonlinear Least-Squares Problem

The nonlinear least-squares problem is

\[
\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} R(x)^T R(x) = \frac{1}{2} \sum_{i=1}^{m} r_i(x)^2
\]  

where \( m > n \), the *residual function* \( R: \mathbb{R}^n \rightarrow \mathbb{R}^m \) is nonlinear in \( x \), and \( r_i(x) \) denotes the \( i \)th component of \( R(x) \). If \( R(x) \) is linear, (5.1) is a linear least-squares problem. If one is attempting to fit the data \((s_i, y_i), i = 1, 2, \ldots, m\), with a model \( M(x,s) \) that is nonlinear in \( x \), then the nonlinear least-squares problem consists of choosing \( x \) so that the fit is as close as possible in the sense that the sum of squares of the residuals \( r_i(x) = M(x,s_i) - y_i \) is minimized. Typically, \( m \) is much larger than \( n \). The choice of the sum-of-squares measure for data fitting is justified by statistical considerations (see, for example, [8]).

The CTF and structure factor determination problem can be viewed as a constrained (because of the bounds on the variables) nonlinear least-squares problem. In this problem we fit the total intensity model which is nonlinear in the parameters, as defined in (3.6), to the total intensity of the signal observed on the micrograph. Usually the data set contains several hundred points.
We chose to formulate the problem as a nonlinear least-squares problem. It is also possible to use the $l_1$ norm

$$f_1(x) = \sum_{i=1}^{m} |r_i|$$

or the $l_\infty$ norm

$$f_\infty(x) = \max_{1 \leq i \leq m} |r_i|$$

to evaluate the quality of the fit. However, we prefer the least-squares formulation because of statistical considerations and because the objective function is differentiable only when using the $l_2$ norm and not in the other two cases.

Note that the objective function is highly nonlinear due to the physics involved in creating the model.

It is possible that the constrained solution is not at the bounds of the feasible region, however, an optimum may be outside the constrained region and removing the constraints may lead to a solution outside of the range of meaningful values.

### 5.2 Mathematical Formulations

It is convenient to use the following notation for the unknown parameters:

$$x_i = n_i, \; i = 1, 2, \ldots, 4,$$

where $n_i$ are the parameters of the noise intensity function (3.5), and the rest are the parameters of the CTF (3.4):

$$x_5 = Amp, \; x_6 = -\Delta z + 5, \; x_7 = \arcsin(C_A), \; x_8 = B.$$

In contrast with the "manual" approach to parameter estimation, parameters $\Delta z$, $B$, and $Amp$ cannot be viewed as unconstrained and the values cannot be simply
discarded if they are out of the range of meaningful values. Therefore we introduce
a vector of upper bounds $b$, corresponding to the vector of unknown parameters.
The upper bound values are soft. They are determined empirically. The defocus
parameter $\Delta z$ is the only parameter that can have negative values. For convenience,
we can shift it to ensure nonnegativity without affecting the formulation. Therefore
we consider all the parameters to be nonnegative.

Let us introduce the following notation.

- $p$ is the number of different defocus data sets available for the same particle,
  usually 5 or 6;

- $m$ is the number of points in each data set, usually several hundred;

- $n$ is the number of unknown CTF and noise parameters for each data set; in
  the current model $n = 8$;

- $x_j$ is the unknown vector of CTF and noise parameters for $j$th data set, $x_j \in \mathbb{R}^n$;

- $x$ is a vector composed of vectors $x_j$, $x = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, x \in \mathbb{R}^{mn}$;

- $b_j$ is a given vector of upper bounds for the CTF and noise parameters $x_j$, $b_j \in \mathbb{R}^n$; upper bounds are the same for each data set, i.e., $b_j = b_i \forall i, j \in 1, 2, \ldots, p$;

- $b$ is a vector composed of vectors $b_j$, $b = \begin{bmatrix} b_1 \\ \vdots \\ b_p \end{bmatrix}, b \in \mathbb{R}^{mn}$;

- $y$ is the unknown vector of structure factors, $y \in \mathbb{R}^m$;
• $ctf(x_j)$ is the unknown CTF vector for $j$th data set, $ctf(x_j) \in \mathbb{R}^m$;

• $noise(x_j)$ is the unknown noise intensity vector for $j$th data set, $noise(x_j) \in \mathbb{R}^m$;

• $t_j$ is the vector of total intensity of the signal observed on the micrograph; $t_j \in \mathbb{R}^m$.

Based on the model of total intensity of the signal observed on the micrograph (3.6), we can now formulate our problem as

$$
\min_{x,y} \frac{1}{2} \sum_{j=1}^{p} \| \text{diag}(ctf(x_j))^2 y + noise(x_j) - t_j \|^2_2
$$

$$
0 \leq x \leq b, \quad x \in \mathbb{R}^m
$$

$$
y \geq 0, \quad y \in \mathbb{R}^m,
$$

(5.2)

where $noise(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$, $ctf(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$, $t_j \in \mathbb{R}^m$, $j = 1, \ldots, p$.

Let us denote

$$
D(x_j) = \text{diag}(ctf(x_j))^2, D(x_j) \in \mathbb{R}^{m \times m}, j = 1, \ldots, p
$$

$$
c(x_j) = -noise(x_j) + t_j, c(x_j) \in \mathbb{R}^m, j = 1, \ldots, p.
$$

We want to show that $c(x_j) \geq 0$. Recall that

$$
noise(x_j) = x_{j3} e^{-((\tau/2) x_{j4} s)^2} - x_{j1} \sqrt{s} - x_{j2} s
$$

and

$$
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
x_{j1} \\
x_{j2} \\
x_{j3} \\
x_{j4}
\end{bmatrix}
\leq
\begin{bmatrix}
b_{j1} \\
b_{j2} \\
b_{j3} \\
b_{j4}
\end{bmatrix}.
$$
Then

\[ \text{noise}(x_j) = x_{j3} e^{-(\pi/2) x_{j1} s^2 - x_{j1} \sqrt{s} - x_{j2} s} \leq b_{j3} e^{-(\pi/2) 0 s^2 - 0 \sqrt{s} - 0 s} = b_{j3} e^0 = b_{j3}. \]

Let us introduce the new bound $\bar{b}_{j3}$ on $x_{j3}$: $b_{j3} \leq \min(b_{j3}, t_j)$, then $c(x_j) \geq 0$. The restriction $b_{j3} \leq t_j$ is justified because the noise intensity is one of the nonnegative addendums that add up to total intensity:

\[ \text{Total Intensity} = \text{CTF}^2 y + \text{Noise Intensity}, \]

where $y \geq 0, \text{noise}(x_j) \geq 0$.

Omitting the multiplier constant ($\frac{1}{2}$) from the objective function, (5.2) can then be rewritten as

\[
\min_{x, y} \sum_{j=1}^{p} \| D(x_j) y - c(x_j) \|_2^2 \\
0 \leq x \leq b, \ x \in \mathbb{R}^m \\
y \geq 0, \ y \in \mathbb{R}^m
\]

or as

\[
\min_{x, y} \left\| \begin{bmatrix} D(x_1) \\ \vdots \\ D(x_p) \end{bmatrix} y - \begin{bmatrix} c(x_1) \\ \vdots \\ c(x_p) \end{bmatrix} \right\|_2^2 \\
0 \leq x \leq b, \ x \in \mathbb{R}^m \\
y \geq 0, \ y \in \mathbb{R}^m
\]

Let $A(x) = \begin{bmatrix} D(x_1) \\ \vdots \\ D(x_p) \end{bmatrix}$, $A(x) \in \mathbb{R}^{m \times m}$, and $c(x) = \begin{bmatrix} c(x_1) \\ \vdots \\ c(x_p) \end{bmatrix}$, $c(x) \in \mathbb{R}^m$, then
(5.4) can be viewed as a linear least-squares problem in $y$ for a given $x$ that satisfies the bound constraints $0 \leq x \leq b$

$$\min_y \|A(x)y - c(x)\|_2^2$$

$$y \geq 0, \ y \in \mathbb{R}^m.$$  

Next we would like to demonstrate that the nonnegativity constraint on $y$ can be removed. Now consider problem (5.5) without the nonnegativity constraint on $y$

$$\min_{y \in \mathbb{R}^m} \|A(x)y - c(x)\|_2^2.$$  

(5.6)

The solution $y^*$ of (5.6) satisfies

$$A^T(x)A(x)y = A^T(x)c(x),$$

(5.7)

in other words

$$(\sum_{j=1}^{p} D(x_j)^2)y = \sum_{j=1}^{p} D(x_j)c(x_j).$$

(5.8)

Consider diagonal matrix $\sum_{j=1}^{p} D(x_j)^2$. A zero entry on the diagonal

$$(\sum_{j=1}^{p} D(x_j)^2)_i \equiv (\sum_{j=1}^{p} \text{diag}(ctf(x_j)^4))_i = 0$$

(5.9)

can occur for some $i \in 1, 2, \ldots, m$ only if for some $s$, $ctf(x_j) = 0$ in all $p$ data sets. In practice, this situation is highly unlikely because we use different defocus micrographs and it is improbable that the zeros of the CTF functions will coincide. However, one has to ensure that (5.9) is not equal to zero. It is possible that the values of (5.9) are very small, causing numerical problems when dividing by (5.9). The solution $y^*$ of (5.8) can be expressed in terms of $x$:

$$y^*(x) = (\sum_{j=1}^{p} D(x_j)^2)^{-1}(\sum_{j=1}^{p} D(x_j)c(x_j))$$

(5.10)
or, component-wise,

$$y^*_i(x) = \frac{\sum_{j=1}^{p} d_i(x_j)c_i(x_j)}{\sum_{j=1}^{p} d_i(x_j)^2},$$  \hspace{1cm} (5.11)$$

where $d_i(x_j)$ is the $i$th diagonal element of $D(x_j)$ and $c_i(x_j)$ is the $i$th element of $c(x_j)$. The non-negativity condition on $y^*(x)$ no longer needs to be enforced since $d_i(x_j) \geq 0, c(x_j) \geq 0$.

Therefore, above, we have proven the following lemma that says that the nonnegativity constraint on $y$ can be removed from the formulation of the problem:

**Lemma 5.1** For $b_{j3} \leq t_j$, the solution of (5.6) is the solution of (5.5), and vice versa.

Substituting the expression for the solution $y^*(x)$ (5.10) into the original formulation of the problem (5.3), we have a minimization problem in the $x$ variables only:

$$\min_{\mathbf{x}} \sum_{j=1}^{p} \|D(x_j)y^*(x) - c(x_j)\|_2^2$$

$$0 \leq x \leq b, \ x \in \mathbb{R}^{pn}$$

or

$$\min_{\mathbf{x}} \sum_{j=1}^{p} \|D(x_j)((\sum_{k=1}^{p} D(x_k)^2))^{-1}(\sum_{k=1}^{p} D(x_k)c(x_k)) - c(x_j)\|_2^2$$

$$0 \leq x \leq b, \ x \in \mathbb{R}^{pn}.$$  \hspace{1cm} (5.13)

**Theorem 5.1** If (5.3) has a local minimizer $(x^*, y^*)$ and

$$(\sum_{j=1}^{p} D(x_j)^2)_i \neq 0 \quad \forall \ i \in 1, 2, \ldots, m$$

then $y^*$ satisfies (5.10), namely

$$y^* = ((\sum_{j=1}^{p} D(x_j)^2))^{-1}(\sum_{j=1}^{p} D(x_j^*)c(x_j^*))$$  \hspace{1cm} (5.14)
**Proof** Suppose $y^* \neq y^*$ does not satisfy (5.14). Then it is not a solution for (5.6):

$$\min_{y \in \mathbb{R}^m} \|A(x)y - c(x)\|_2^2$$

Therefore there exists $\bar{y} \neq y^*$ that is the solution for (5.6) and satisfies (5.10). For a fixed $x^*$, we have that the solution to (5.3) $(x^*, \bar{y})$ is better than $(x^*, y^*)$, i.e.

$$\|A(x^*)\bar{y} - c(x^*)\|_2^2 < \|A(x^*)y^* - c(x^*)\|_2^2$$

We can see that $(x^*, y^*)$ can not be minimizer of the original problem (5.3) but $(x^*, \bar{y})$ is, because the objective function $\|A(x)y - c(x)\|_2^2$ is quadratic in $y$ for a fixed $x$. We have a contradiction, therefore a minimizer $(x^*, y^*)$ of (5.3) must satisfy (5.14).

□

As a consequence of this result we have an optimality check for a solution candidate $(x^*, y^*)$.

We demonstrated that we can treat the structure factor vector $y$ and the CTF and noise parameters $x$ together, as independent variables, or, $y$ can be treated as a variable dependent on $x$, so that the explicit variable $y$ can be eliminated, leaving fewer variables in the problem formulation. An additional advantage of the second formulation is that no initial guess for the of linear variables $y$ is required. Another observation is that the Hessian approximation matrix is nearly singular when we use the first formulation and the Gauss-Newton method to solve it. See more detailed discussion about singularity in Chapter 8. A disadvantage of the second formulation is that the function, its Jacobian and Hessian calculations become very cumbersome.
Chapter 6

Interior-Point Algorithm

We have formulated the problem as nonlinear least squares with simple bound constraints. With these more sophisticated formulations, we need more sophisticated numerical algorithms. An important issue is how to handle the bound constraints. To address this, we will employ interior-point methods, a class of relatively new methods that have enjoyed great success as constrained optimization algorithms in the last decade, in terms of both their theoretical properties and practical performance. The strength of the interior-point methods lies in their ability to treat inequality constraints effectively (see, for example, [9, 26, 28]).

We investigate specific primal-dual interior-point methods that best exploit the special structure of the problem at hand. Major issues include Hessian calculations or approximations, line search, merit functions, and linear algebra treatment of ill-conditioned linear systems for search step calculation. We conduct extensive numerical experiments to find the best combination of the many factors that leads to a reliable and efficient algorithm. The results appear in Chapter 8.

6.1 The Nonlinear Programming Problem and Lagrange Multiplier Background

The nonlinear programming problem (NLP) in its most general form is the optimization of an objective function $f(x)$ over some space $x \in X$, subject to constraints on the variable $x$. The subject of nonlinear programming is covered extensively in the literature (see, for example, Zangwill [29] and Bertsekas [4]). Our problem has the
following form:

\[
\min_x f(x) \quad \text{(6.1)}
\]

s.t. \( g(x) \geq 0, \)

where

\[
f(x) = \frac{1}{2} \|A(x)y(x) - c(x)\|^2_2 = \frac{1}{2} R(x)^TR(x), f(\cdot) : \mathbb{R}^n \to \mathbb{R}, \quad \text{(6.2)}
\]

\[
R(x) = A(x)y(x) - c(x), R(\cdot) : \mathbb{R}^n \to \mathbb{R}^{p \times m}, \quad \text{(6.3)}
\]

\[
g(x) = \begin{bmatrix} x \\ b - x \end{bmatrix}, g(\cdot) : \mathbb{R}^n \to \mathbb{R}^{2n}, \quad \text{(6.4)}
\]

where \( n \) is the dimension of the problem.

Let us introduce the Lagrange multipliers \((v, w) \in \mathbb{R}^{n \times n}, (v, w) \geq 0,\) and the
Lagrangian function for our problem \( L(\cdot) : \mathbb{R}^{3n} \to \mathbb{R} \) defined thus:

\[
L(x, v, w) = f(x) - v^T x - w^T (b - x). \quad \text{(6.5)}
\]

### 6.2 The Karush-Kuhn-Tucker (KKT) Conditions

Under the reasonable assumption that the constraint qualification holds, the Karush-Kuhn-Tucker, or KKT, conditions [16] must necessarily hold at an optimal point for (6.1). Using the notation \( X \) to denote a diagonal matrix with the vector \( x \) on its diagonal, these conditions can be written:

\[
\nabla_x L(x, v, w) = 0, \quad \text{(6.6)}
\]

\[
Xv = 0, \quad \text{(6.7)}
\]

\[
(B - X)w = 0, \quad \text{(6.8)}
\]

\[
x \geq 0, \quad \text{(6.9)}
\]

\[
b - x \geq 0, \quad \text{(6.10)}
\]

\[
(v, w) \geq 0, \quad \text{(6.11)}
\]
where $X = \text{diag}(x)$, $B = \text{diag}(b)$. Condition (6.6) is the optimality condition, conditions (6.7), (6.8) are the complementarity conditions, and conditions (6.9), (6.10) are the feasibility conditions for the candidate point.

For convenience, we express the KKT conditions more compactly as

\[
\begin{align*}
F(x, v, w) &= 0, \quad \text{(6.12)} \\
(x, v, w) &\geq 0, \quad \text{(6.13)} \\
x &\leq b, \quad \text{(6.14)}
\end{align*}
\]

with

\[
F(x, v, w) = \begin{pmatrix}
\nabla_x L(x, v, w) \\
xv \\
(B - X)w
\end{pmatrix} = \begin{pmatrix}
\nabla_x f(x) - v + w \\
xv \\
(B - X)w
\end{pmatrix}.
\quad \text{(6.15)}
\]

The nonlinear system $F(x, v, w) = 0$ has dimension $3n$. Let us denote the unknowns $(x, v, w)$ as $u \in \mathbb{R}^{3n}$.

### 6.3 The Interior-Point Method

The interior-point method uses the perturbed KKT conditions $F_\mu$ which result from perturbing the last $2n$ components of $F$ by a perturbation parameter $\mu \geq 0$. The resulting expression is

\[
\begin{align*}
F_\mu(x, v, w) &= 0, \\
(x, v, w) &\leq 0, \\
x &\leq b,
\end{align*}
\]

where

\[
F_\mu(x, v, w) = F(x, v, w) - \mu \hat{e}, \quad \text{(6.16)}
\]
and the vector \( \hat{e} = (0, \ldots, 0, 1, \ldots, 1)^T \) has \( n \) zeroes and \( 2n \) ones, i.e., the perturbation only affects the complementarity conditions.

In order to obtain a solution to the perturbed KKT conditions, we consider the Newton's method which consists of the iterative solution of the linear systems

\[
F'(u_k) \Delta u_k = -F'_\mu(u_k)
\]

\[
u_{k+1} = u_k + \Delta u_k,
\]

given some initial guess \( u_0 \). Here \( F'_{i,j}(u) \) denotes the Jacobian of \( F_\mu \), i.e., the matrix of partial derivatives of \( F \) with the respect to the independent variables,

\[
F'(u)_{i,j} = \frac{\partial F_i(u)}{\partial u_j}.
\] (6.17)

The Jacobian for our problem has the form

\[
F'(u) = \begin{pmatrix}
\nabla^2_{x,x} L(x, v, w) & -I & I \\
V & X & 0 \\
-W & 0 & B - X
\end{pmatrix},
\] (6.18)

where \( I \) is the identity matrix. The Hessian of the Lagrangian is given by

\[
\nabla^2_{x,x} L(x, v, w) = \nabla^2_{x,x} f(x) = J^T(x)J(x) + S(x),
\]

where \( J(x) \) is the Jacobian matrix of the residual \( R(x) \)

\[
J(x) = \frac{\partial r_i(x)}{\partial x_j},
\]

and

\[
S(x) \triangleq \sum_{i=1}^{pm} r_i(x) \nabla^2 r_i(x)
\]

denotes the second-order information in \( \nabla^2 f(x) \).
The right-hand side is

\[ F_\mu(x, v, w) = \begin{pmatrix} \nabla_x f(x) - v + w \\ Xv - \mu e \\ (B - X)w - \mu e \end{pmatrix}, \quad (6.19) \]

where \( e \) is a vector of \( n \) ones.

### 6.4 Slack-Variable Form of the KKT Conditions

Often in solving the KKT system of equations (6.6) – (6.8) the most difficult task is to satisfy the nonlinear equation (6.6). Since the upper-bound constraints on the \( x \) variables need to be satisfied, there is often less flexibility in choosing \( x \). The step length can become very small and adversely affect the convergence rate. Therefore for practicality reasons, let us introduce a set of slack variables \( z \in \mathbb{R}^a \) and eliminate the need to be feasible with respect to a set of upper-bound constraints at the expense of solving a larger system, instead. The resulting equivalent slack-variable form of the KKT conditions is:

\[ \nabla_z L(x, z, v, w) = 0 \quad (6.20) \]
\[ x + z - b = 0 \quad (6.21) \]
\[ Xv = 0 \quad (6.22) \]
\[ Zw = 0 \quad (6.23) \]
\[ (x, z) \geq 0 \quad (6.24) \]
\[ (v, w) \geq 0. \quad (6.25) \]

Note that the upper-bound constraints can be violated during the solution process but will be satisfied at the solution, since it is easier to satisfy the linear equation (6.21) than the nonlinear one (6.20).
Then

\[
F_{\mu}(x, z, v, w) = \begin{pmatrix}
\nabla_x f(x) + w - v \\
x + z - b \\
Xv - \mu e \\
Zw - \mu e
\end{pmatrix} = \begin{pmatrix}
F_1 \\
F_2 \\
F_3 \\
F_4
\end{pmatrix} = 0,
\]

\[
F'(x, z, v, w) = \begin{pmatrix}
\nabla^2_{xx} f(x) & 0 & -I & I \\
I & I & 0 & 0 \\
V & 0 & X & 0 \\
0 & W & 0 & Z
\end{pmatrix}.
\]

### 6.5 An Interior-Point Algorithm

Here we will state a primal-dual Newton interior-point algorithm. We will apply it to our problem with the nonlinear least-squares objective function and simple-bound constraints. We will use the symbol \( \| \cdot \| \) to denote the Euclidean \( l_2 \) norm of a vector, and the subscript \( k \) to denote an iteration counter. A question with any interior-point method for nonconvex nonlinear programming is which merit function to use. For treatment on this issue see, for example, Vanderbei [25] or Argaez [2].

For global convergence we will use the merit function \( \phi(u) \) that indicates the progress toward the solution \( u^* = (x^*, z^*, v^*, w^*) \). This globalized algorithm is a perturbed and damped Newton’s method with a backtracking line search. There are several possible choices for the merit function to be used in line search. We choose to use the \( l_2 \)-norm of the residual of the KKT conditions,

\[
\phi(u) = \| F(u) \|.
\]

The parameters \( \lambda_k, \alpha_k \) are real numbers satisfying \( 0 \leq \lambda_k, \alpha_k \leq 1 \).

**The Newton Interior-Point Algorithm**
• Choose an initial guess $u_0 = (x_0, z_0, u_0, w_0)$ satisfying $u_0 > 0$.

For $k = 0, 1, 2, \ldots$ do

• STEP 1. Test for convergence. If a convergence criterion is satisfied then exit.

• STEP 2. Update perturbation parameter $\mu_k > 0$.

• STEP 3. Solve for perturbed Newton step, $\Delta u_k = -(F'(u_k))^{-1}F_{\mu_k}(u_k)$.

• STEP 4. Adjust step-length to ensure $u_k > 0, x_k < b$.

$$\Delta u_k \leftarrow \lambda_k \Delta u_k$$

• STEP 5. Adjust step-length for globalization $\Delta u_k \leftarrow \alpha_k \Delta u_k$

• STEP 6. Update unknowns: $u_{k+1} = u_k + \Delta u_k$.

$$k \leftarrow k + 1$$

EndFor.

We now give a more detailed description of the steps of the algorithm.

In STEP 1 there are several possible stopping criteria that we can choose from to use in the interior-point algorithm. One choice is $F(u) = 0$. Because of the globalization strategy of the algorithm and because of the Hessian perturbation to make it positive definite we can guarantee that the algorithm will reach a minimum and avoid maxima and saddle points (see [8]). In the implementation, a parameter $\epsilon$ needs to be chosen, so we can use a test $\|F(u)\| < \epsilon$.

In updating the perturbation parameter in STEP 2, there are several issues that need to be addressed. First of all, the point $u_k$ has to lie near the central path in order to stay away from the bounds until the algorithm is fairly near the solution. Therefore, the perturbation might need to be fixed for several iterations (instead of being decreased at every step) until all the components of the perturbed KKT system have displayed sufficient decrease.
In STEP 3 of the algorithm, exploring the sparsity and structure of $F'(u)$, we can solve for

\[
\begin{align*}
\Delta x_k &= -(J^TJ + S) + WZ^{-1} + VX^{-1})^{-1}(J^TR - X^{-1}μe + Z^{-1}μe + Z^{-1}WF_2) \\
\Delta z_k &= -\Delta x_k - x - z + b = -\Delta x_k - F_2 \\
\Delta v_k &= -v + X^{-1}μe - VX^{-1}Δx_k = -X^{-1}(F_3 + VΔx_k) \\
\Delta w_k &= -w + Z^{-1}μe - WZ^{-1}Δz_k = -Z^{-1}(F_4 + WΔz_k).
\end{align*}
\]

If the usual (not slack-variable) form of the KKT conditions is used then in STEP 3 of the Newton interior-point algorithm we solve for

\[
\begin{align*}
\Delta x_k &= (J^TJ + S) + W(B - X)^{-1} + VX^{-1})^{-1}(-J^TR - (B - X)^{-1}μe + μe), \\
\Delta v_k &= -v + X^{-1}μe - VX^{-1}Δx_k, \\
\Delta w_k &= -w + (B - X)^{-1}μe + W(B - X)^{-1}Δx_k.
\end{align*}
\]

Alternatively, $\Delta v_k$ can be calculated as

\[
\begin{align*}
\Delta v_k &= J^TR + w - v + (J^TJ + S)Δx_k + Δw_k \\
&= J^TR - v + (B - X)^{-1}μe + ((J^TJ + S) + W(B - X)^{-1})Δx_k.
\end{align*}
\]

Note that $W(B - X)^{-1} + VX^{-1}$ is a diagonal matrix.

In STEP 4 we choose $τ_k ∈ (0, 1]$ and adjust the step length to ensure the non-negativity of the variables at the new iteration level and the condition. We compute the quantities

\[
\begin{align*}
\hat{α}_x &= \frac{-1}{\min((X_k)^{-1}Δx_k, -1)}, & \hat{α}_z &= \frac{-1}{\min((Z_k)^{-1}Δz_k, -1)}, \\
\hat{α}_v &= \frac{-1}{\min((V_k)^{-1}Δv_k, -1)}, & \hat{α}_w &= \frac{-1}{\min((W_k)^{-1}Δw_k, -1)}, \\
α_x &= \min(1, τ_k\hat{α}_x), & α_z &= \min(1, τ_k\hat{α}_z), & α_v &= \min(1, τ_k\hat{α}_v), & α_w &= \min(1, τ_k\hat{α}_w).
\end{align*}
\]
In STEP 5 we choose $\alpha_k \in (0,1]$ satisfying

$$\phi(u_k + \alpha_k \Delta u_k) \leq \phi(u_k) + \beta \alpha_k \nabla \phi(u_k)^T \Delta u_k,$$

(6.26)

for some fixed $\beta \in (0,1)$. In fact, we let $\alpha_k \leftarrow \rho^t \alpha_k$, where $t$ is the smallest nonnegative integer such that $\alpha_k$ satisfies the above condition.

In this chapter we have reviewed the Lagrange Multiplier and KKT conditions background. The slack-variable form of the KKT conditions for the specific problem was described as a useful alternative to the standard KKT conditions. We chose to base the algorithm for the parameter estimation problem on the interior-point methodology, known for its strength to deal successfully with inequality constraints. We proposed a Newton interior-point algorithm that can be easily modified to Gauss-Newton or Levenberg-Marquardt interior-point algorithm.
Chapter 7

Theoretical Convergence Analysis

In this chapter we describe previous theoretical work related to proving local convergence for nonlinear least-squares problems. Then we introduce the necessary definitions and prove theoretical local convergence results for nonlinear least squares with general constraints.

7.1 Previous Local Convergence Results

There has been interest in extending Gauss-Newton convergence theory from unconstrained nonlinear least squares to include either general constraints, or at least special case, such as linear constraints (including simple bounds). Special cases for the constraints have been considered in the following literature. Wright and Holt [27] prove a global convergence result for nonlinear least squares with linear inequality constraints. Sagara and Fukushima [23] also consider just linear inequality constraints. Holt and Fletcher [14] consider a class of nonlinear least squares with the constraints being upper bounds, lower bounds, and a set $x_1 \leq x_2 \leq \ldots \leq x_k$. Gulliksson et al. [11] prove convergence results for the equality constrained case.

In 1983, Dennis and Schnabel [8] considered unconstrained nonlinear least-squares problems. They demonstrated local convergence for the Gauss-Newton method, stating that for zero-residual or linear least-squares problems, the convergence is $q$-quadratic. For problems that are not too nonlinear and have reasonably small residual the convergence is $q$-linear. We would like to extend these results to include general constraints for nonlinear least-squares problems.
In 1992, El-Bakry, Tapia, Tsuchiya, and Zhang [9] proved convergence results for Newton interior-point method for nonlinear programming. That is they consider the case when second-order information is available for the objective function and the Hessian matrices for both equality and inequality constraints are exact. We are interested in proving results for modifications of Newton's method, when the Hessian matrices for the objective and constraints are perturbed.

In 1998, Cores and Tapia [7] studied the convergence of the perturbation of the Newton method for solving a nonlinear system of equations. They prove results for nonlinear equality constrained optimization problem. We would like to add inequality constraints to the problem.

Lindström and Wedin [17] present the results for Gauss-Newton based algorithms for either inequality or just equality constrained nonlinear least-squares problems.

We prove results regarding local convergence and rate of convergence for generally constrained nonlinear least squares when the Hessian matrices for the objective and constraints are perturbed. We establish $q$-linear convergence for the general nonlinear least-squares constrained problem, as well as $q$-quadratic or $q$-superlinear rates of convergence for certain conditions.

### 7.2 Convergence Rates

We will need the following standard definitions of the rates of convergence (see, for example, Dennis and Schnabel [8] or Ortega and Rheinboldt [20]).

**Definition 7.1** Let $x_* \in \mathbb{R}, x_k \in \mathbb{R}, k = 0, 1, 2, \ldots$. Then the sequence

$\{x_k\} = \{x_0, x_1, x_2, \ldots\}$

is said to converge to $x_*$ if

$$\lim_{k \to \infty} \|x_k - x_*\| = 0.$$
If in addition, there exists a constant \( c \in [0,1) \) and an integer \( \hat{k} \geq 0 \) such that for all \( k \geq \hat{k} \),

\[
\|x_{k+1} - x_*\| \leq c \|x_k - x_*\|
\]

then \( \{x_k\} \) is said to be \textit{q-linearly convergent} to \( x_* \). If for some sequence \( \{c_k\} \) that converges to 0,

\[
\|x_{k+1} - x_*\| \leq c_k \|x_k - x_*\|
\]

then \( \{x_k\} \) is said to converge \textit{q-superlinearly} to \( x_* \). If there exist constants \( p > 1, c \geq 0 \), and \( \hat{k} \geq 0 \) such that \( \{x_k\} \) converges to \( x_* \) and for all \( k \geq \hat{k} \),

\[
\|x_{k+1} - x_*\| \leq c \|x_k - x_*\|^p
\]

then \( \{x_k\} \) is said to converge to \( x_* \) with \textit{q-order at least} \( p \). If \( p = 2 \), the convergence is said to be \textit{q-quadratic}.

### 7.3 Local Convergence Properties

Consider the general constrained least squares problem

\[
\begin{align*}
\min_x & \quad f(x) \\
\text{s.t.} & \quad h(x) = 0, \\
& \quad g(x) \geq 0,
\end{align*}
\]  

(7.1)

where \( f(x) = \frac{1}{2}R(x)^T R(x) \).

The Lagrangian associated with the problem (7.1) is

\[
L(x, v, w) = f(x) + v^T h(x) - w^T g(x).
\]  

(7.2)
The KKT conditions for problem (7.1) are

\[ \nabla_x L(x, v, w) = 0, \quad h(x) = 0, \quad g(x) \geq 0, \quad Wg(x) = 0, \quad w \geq 0, \tag{7.3} \]

where

\[ W = \text{diag}(w), \]

\[ \nabla_x L(x, v, w) = \nabla_x f(x) + \nabla_x h(x)v - \nabla_x g(x)w. \]

The KKT conditions (7.3)-(7.7) can be written as

\[ F(x, v, w) = \begin{pmatrix} \nabla_x L(x, v, w) \\ h(x) \\ Wg(x) \end{pmatrix} = 0, w, g(x) \geq 0. \tag{7.8} \]

Then

\[ F'(x, v, w) = \begin{pmatrix} \nabla_x^2 L(x, v, w) & \nabla_x h(x) & -\nabla_x g(x) \\ \nabla_x h(x)^T & 0 & 0 \\ W\nabla_x g(x) & 0 & \text{diag}(g(x)) \end{pmatrix}. \tag{7.9} \]

The Hessian of the Lagrangian is given by

\[ \nabla_x^2 L(x, v, w) = \nabla_x^2 f(x) + \nabla_x^2 h(x)v - \nabla_x^2 g(x)w \]

\[ = J^T(x)J(x) + S(x) + \nabla_x^2 h(x)v - \nabla_x^2 g(x)w, \]

where \( J(x) \) is the Jacobian matrix of the residual \( R(x) \)

\[ J(x) = \frac{\partial r_i(x)}{\partial x_j} \]
and

\[ S(x) \triangleq \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x) \]  \hspace{1cm} (7.10)

denotes the second-order information in \( \nabla^2 f(x) \).

The slack-variable form of the KKT conditions for the problem (7.1) is

\[
F(x, u, z, w) = \begin{pmatrix}
\nabla_x L(x, u, z, w) \\
h(x) \\
g(x) - z \\
Wz
\end{pmatrix} = 0, \quad (w, z) \geq 0, \hspace{1cm} (7.11)
\]

where \( z \) are the slack variables.

Recall that a function \( g \) is called 
\textit{Lipschitz continuous} with constant \( \gamma \) in a set \( X \), written \( g \in \text{Lip}_\gamma(X) \), if \( \forall \ x, \ y \in X \),

\[ ||g(x) - g(y)|| \leq \gamma ||x - y||. \]

The standard Newton's method assumptions for problem (7.1) are as follows:

(A1) Existence. There exists \((x_*, u_*, w_*)\), solution to problem (7.1) and associated multipliers, satisfying the KKT conditions (7.8).

(A2) Smoothness. The Hessian matrices \( \nabla^2 f(x) \), \( \nabla^2 h_i(x) \), \( \nabla^2 g_i(x) \) for all \( i \) exist and are locally Lipschitz continuous at \( x_* \).

(A3) Regularity. The set \( \{ \nabla h_1(x_*), \ldots, \nabla h_m(x_*) \} \cup \{ \nabla g_i(x_*): g_i(x_*) = 0 \} \) is linearly independent.

(A4) Second-Order Sufficiency. For all \( \eta \neq 0 \) satisfying \( \nabla h_i(x_*)\eta = 0, \ i = 1, \ldots, m \), and \( \nabla g_i(x_*)\eta = 0, \ i : g_i(x_*) = 0 \), we have \( \eta^T \nabla^2_x L(x_*) \eta > 0 \).

(A5) Strict Complementarity. For all \( i, w_{*i} + g_i(x_*) > 0 \).
In dealing with our problem we would like to take advantage of the structure of the least-squares problem. Gauss-Newton and Levenberg-Marquardt methods are techniques that are most commonly used for this purpose (see, for example, Björk [5] and Dennis and Schnabel [8]). Gauss-Newton method is an iterative method that updates the current solution at every step as follows (assuming that \( J(x_k) \) has full column rank):

\[
x_{k+1} = x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T R(x_k).
\]

In the Levenberg-Marquardt method the next iterate is chosen as follows:

\[
u_{k+1} = u_k - (J(u_k)^T J(u_k) + \rho_k I)^{-1} J(u_k)^T R(u_k).
\] (7.12)

By the *perturbed and damped interior-point Gauss-Newton method* for the problem

\[ F(u) = 0, \] (7.13)

we mean the construction of the iteration sequence

\[
u_{k+1} = u_k - \alpha_k (F'(u_k) - B_k)^{-1} (F(u_k) - \mu_k \beta), \quad k = 0, 1, 2, \ldots,
\] (7.14)

where \( 0 < \alpha_k \leq 1 \) is the damping parameter, \( \mu_k > 0 \) is the function perturbation parameter, and \( \beta \) is a fixed vector in \( \mathbb{R}^n \). Matrix \( B_k \) is the perturbation of the Jacobian matrix \( F'(u_k) \). In case when \( F(u) \) represents the KKT conditions (7.8) for the general constrained least-squares problem (7.1), and the constraints are linear

\[
B_k = B(x_k) = \begin{pmatrix}
S(x_k) & 0 \\
0 & 0 
\end{pmatrix},
\] (7.15)

where \( S(x) \) is defined as second-order information on the least-squares objective function (7.10).

Now we are ready to state and prove the theorem about linear convergence of the iterates generated by the perturbed and damped interior-point Gauss-Newton
method (7.14) applied to the problem \( F(u) = 0 \). We will also demonstrate that the desirable quadratic convergence of Newton’s method in case when perturbation of \( F \) is zero at the solution and a reasonable additional assumption that the perturbation matrix \( B \) is Lipschitz continuous.

**Theorem 7.1** Let \( F \) be defined as (7.11) and let \( F \) be continuously differentiable in an open convex set \( D \subset \mathbb{R}^n \). Let \( u_k \to u_* \in D \) such that

\[
F(u_*) = 0, \quad (7.16)
\]

\[
\|F'(u_*)^{-1}\| \leq \beta, \quad (7.17)
\]

\[
F'(u) \in \text{Lip}_q(D), \quad (7.18)
\]

and

\[
\|B(x_*)\| < \frac{1}{5\beta}. \quad (7.19)
\]

There exists \( \epsilon > 0 \) such that for all \( u_0 \) in the neighborhood \( N(u_*, \epsilon) \) and the sequence generated by the perturbed and damped interior point Gauss-Newton method (7.14) the following hold:

(i) if \( \alpha_k \to 1 \) and \( \mu_k = o(\|F(u_k)\|) \), then the sequence \( \{u_k\} \) converges to \( u_* \) \( q \)-linearly;

(ii) if \( \alpha_k = 1 + O(\|F(u_k)\|), \mu_k = O(\|F(u_k)\|^2) \), and \( B(x_*) = 0 \), then the sequence \( \{u_k\} \) converges to \( u_* \) \( q \)-superlinearly. If, in addition, \( B \) is Lipschitz continuous then the sequence \( \{u_k\} \) converges to \( u_* \) \( q \)-quadratically.

Note that (7.16) - (7.18) are the standard Newton method assumptions for problem (7.13). Also, note that in case when \( F(u) \) represents the KKT conditions (7.8) for the general constrained least-squares problem (7.1), and the constraints are linear, we have that \( \|S(x_*)\| = \|B(x_*)\| \).
Proof. The proof is by induction. Let iteration index \( k = 0 \). Let us abbreviate \( F(u_0), F'(u_0), F(u_k), F'(u_k), B(x_0), B(x_k), \) and \( B(x_*) \) by \( F_0, F'_0, F_k, F'_k, F_*, B_0, B_k, \) and \( B_* \), respectively, and let \( \| \cdot \| \) denote the vector or matrix \( l_2 \) norm.

It can be shown (see, for example, [8]), that there exists \( \epsilon_1 > 0 \) such that \( (F'_0)^{-1} \) is nonsingular and \( \| (F'_0)^{-1} \| \leq 2\beta \) for \( \| u_0 - u_* \| \leq \epsilon_1 \) because of the Lipschitz continuity of \( F'_* \) and choice of \( \epsilon_1 \). There also exists \( \epsilon_2 > 0 \) such that

\[
\| B_0 \| = \| S_0 \| \leq \frac{1}{4\beta}.
\]

Let \( \epsilon = \min(\epsilon_1, \epsilon_2) \). Then

\[
\rho = \| (F'_0)^{-1}[(F'_0 - B_0) - F'_0]\| \leq \| (F'_0)^{-1}\| \| B_0 \| < 2\beta \cdot \frac{1}{4\beta} = \frac{1}{2},
\]

where \( 0 < \rho < \frac{1}{2} \). It can be shown (see [8]) that, since \( F'_0 \) is nonsingular and \( \| (F'_0)^{-1}[(F'_0 - B_0) - F'_0]\| < 1 \), then \( F'_0 - B_0 \) is nonsingular and

\[
\| (F'_0 - B_0)^{-1} \| \leq \frac{\| (F'_0)^{-1} \|}{1 - \| (F'_0)^{-1}[(F'_0 - B_0) - F'_0]\|} = \frac{\| (F'_0)^{-1} \|}{1 - \rho} \leq \frac{2\beta}{1 - \rho}.
\]  \hspace{1cm} (7.20)

Recall that \( F_* = 0 \). Then at the first step, \( u_1 \) is well defined and

\[
u_1 - u_* = u_0 - u_* - \alpha_0(F'_0 - B_0)^{-1}[F_0 - \mu_0 \hat{p}] = (1 - \alpha_0)(u_0 - u_*) + \alpha_0(u_0 - u_*) - \alpha_0(F'_0 - B_0)^{-1}[F_0 - \mu_0 \hat{p}] = (1 - \alpha_0)(u_0 - u_*) + \alpha_0(F'_0 - B_0)^{-1}[(F'_0 - B_0)(u_0 - u_*) - F_0 + \mu_0 \hat{p} + F_*] = (1 - \alpha_0)(u_0 - u_*) + \alpha_0(F'_0 - B_0)^{-1}[F_* - F_0 - (F'_0 - B_0)(u_* - u_0) + \mu_0 \hat{p}] = (1 - \alpha_0)(u_0 - u_*) + \alpha_0(F'_0 - B_0)^{-1}[F_* - F_0 - F'_0(u_* - u_0) + B_0(u_* - u_0) + \mu_0 \hat{p}].
\]

Since \( F \) is continuously differentiable in an open convex set \( D \) and \( F' \) is Lipschitz continuous at \( u \) in the neighborhood \( N \) with the constant \( \gamma \), it can be shown (see [8]) that

\[
\| F_* - F_0 - F'_0(u_* - u_0) \| \leq \frac{\gamma}{2} \| u_0 - u_* \|^2 \]  \hspace{1cm} (7.21)
Combining (7.20) and (7.21) gives

\[
\|u_1 - u_*\| \leq (1 - \alpha_0)\|u_0 - u_*\| + \frac{2\beta}{1 - \rho} \frac{\gamma}{2} (\|u_0 - u_*\|^2) \\
+ \alpha_0\|(F'_0 - B_0)^{-1}\|\|B_0\|\|u_0 - u_*\| \\
+ \alpha_0\mu_0\|(F'_0 - B_0)^{-1}\|\|
\] (7.22)

If \(B_* = 0\) and \(B\) is Lipschitz continuous with constant \(\tilde{\gamma}\), then

\[
\|B_0\| = \|B_0 - B_*\| \leq \tilde{\gamma}\|u_0 - u_*\| = O(\|u_0 - u_*\|).
\] (7.23)

Next, we will establish that

\[
\|F_0\| = O(\|u_0 - u_*\|).
\] (7.24)

Consider the definition of the Jacobian (6.17) and a component-by-component application of Newton's formula for \(f : \mathbb{R}^n \to \mathbb{R}\) which is continuously differentiable in an open convex set, and points \(x\) and \(x + p\) in this set

\[
f(x + p) = f(x) + \int_0^1 \nabla f(x + tp)^T p \, dt.
\]

We can prove that for \(F : \mathbb{R}^n \to \mathbb{R}^m\) continuously differentiable in an open set \(D \subset \mathbb{R}^n\) and any \(x, x + p \in D\),

\[
F(x + p) - F(x) = \int_0^1 J(x + tp)p \, dt.
\]

Therefore if \(p = u_* - u_0\) then

\[
\|F_0\| = \|F_0 - F_*\| = \|\int_0^1 J(u_0 + tp)p \, dt\| = O(\|u_0 - u_*\|),
\]

because \(\|F'(z)\|\) is bounded above.

The induction step now proceeds identically. We obtain the following estimates,

\[
\|(F'_k - B_k)^{-1}\| \leq \frac{2\beta}{1 - \rho}, \quad 0 < \rho < \frac{1}{2},
\] (7.25)
\[ \|u_{k+1} - u_*\| \leq (1 - \alpha_k)\|u_k - u_*\| + \frac{2\beta}{1 - \rho} \frac{\gamma}{2} \|u_k - u_*\|^2 + \alpha_k\|(F'_k - B_k)^{-1}\|\|B_k\|\|u_k - u_*\| \]
\[ + \quad \alpha_k\mu_k\|(F'_k - B_k)^{-1}\hat{p}\|, \]
\[ \|B_k\| = O(\|u_k - u_*\|), \text{ if } B_* = 0 \text{ and } B \text{ is Lipschitz continuous}, \quad (7.27) \]
\[ \|F_k\| = O(\|u_k - u_*\|), \quad (7.28) \]

corresponding to (7.20), (7.22), (7.23), and (7.24), respectively.

To prove (i) and (ii) consider the estimates (7.25) - (7.28).

If \(\alpha_k \to 1\) and \(\mu_k = o(\|F(u_k)\|)\), then the first term in (7.26) disappears, the second term is \(O(\|u_k - u_*\|^2)\), and the third and forth terms are \(O(\|u_k - u_*\|)\). Therefore, we have established linear convergence.

If \(\alpha_k = 1 + O(\|F(u_k)\|), \mu_k = O(\|F(u_k)\|^2)\), \(B(x_*) = 0\), and \(B\) is Lipschitz continuous, i.e. (7.27) holds, we establish q-quadratic convergence. If \(B\) is not Lipschitz continuous but \(B(u_k) \to 0\) as \(u_k \to u_*\), we establish q-superlinear convergence.

\[ \square \]

In the following, it will be convenient to write
\[ \mu_k = \sigma_k \frac{x^Tv + z^Tw}{2n} \]
and state the conditions in terms of \(\sigma\).

**Theorem 7.2** Consider problem (7.1) and a solution \(u_*\) such that the standard assumptions (A1)-(A5) hold at \(u_*\). Given \(\hat{\tau} \in (0, 1)\), there exists a neighborhood \(N\) of \(u_*\) and a constant \(\hat{\sigma} > 0\) such that, for any \(u_0 \in N\) and any choice of the algorithmic parameters \(\tau_k \in [\hat{\tau}, 1]\) and \(\sigma_k \in [0, \hat{\sigma}]\), perturbed and damped interior-point Gauss-Newton method (7.14) is well defined and the iteration sequence converges q-linearly to \(u_*\).
Proof. First we establish the following relationship for $\alpha_k$:

$$\alpha_k = \min(1, \frac{\tau_k}{1 + O(||F_k||) + O(\sigma_k)})$$

Since

$$\Delta u = -(F_k' - B_k)^{-1}(F_k - \mu_k \hat{e})$$

where $\hat{e}$ is the vector $\hat{e} = (0, \ldots, 0, 1, \ldots, 1)$ with $p$ ones, and $||(F_k' - B_k)^{-1}||$ is bounded above (7.25), we see that

$$||\Delta z_k|| = O(||F_k||) + O(\mu_k), \quad (7.29)$$
$$||\Delta w_k|| = O(||F_k||) + O(\mu_k). \quad (7.30)$$

From linearized perturbed complementarity, we have

$$Z_k^{-1}\Delta z_k + W_k^{-1}\Delta w_k = -e + \mu_k Z_k^{-1}W_k^{-1}e. \quad (7.31)$$

It follows from strict complementarity (A5), (7.29), (7.30), and (7.31) that, if $i$ is an index such that $z_{si} = 0$, then $w_{si}$ is equal to some number and

$$\frac{[\Delta z_k]_i}{[z_k]_i} = -1 + \frac{\mu_k}{z_k w_k} - \frac{[\Delta w_k]_i}{[w_k]_i} = -1 + O(||F_k||) + O(\sigma_k).$$

If $i$ is an index such that $[z_\ast]_i > 0$, then $w_\ast = 0$. Similar relationships hold for the $w$ variables. Therefore,

$$\min(Z_k^{-1}\Delta z_k, W_k^{-1}\Delta w_k) = -1 + O(||F_k||) + O(\sigma_k).$$

We have

$$\alpha_z = \min(1, \tau_k \hat{\alpha}_z), \quad \alpha_w = \min(1, \tau_k \hat{\alpha}_w),$$

where

$$\hat{\alpha}_z = \frac{-1}{\min((Z_k)^{-1}\Delta z_k, -1)}, \quad \hat{\alpha}_w = \frac{-1}{\min((W_k)^{-1}\Delta w_k, -1)}.$$
Then

\[
\alpha_k = \min(1, \frac{\gamma}{1 + O(\|F_k\|) + O(\sigma_k)}).
\]  

(7.32)

Note that the estimates constructed in the proof of Theorem 7.1 did not depend on the fact that we assumed convergence of the iteration sequence. Consider (7.26), (7.28) and the fact that:

\[
\mu_k = \sigma_k O(\|F(u_k)\|) = \sigma_k O(\|u_k - u_*\|).
\]

Based on the estimate (7.26), namely

\[
\|u_{k+1} - u_*\| \leq (1 - \alpha_k)\|u_k - u_*\| + \frac{2\beta}{1 - \rho} \frac{\gamma}{2}\|u_k - u_*\|^2
\]

\[
+ \alpha_k \|(F'_k - B_k)^{-1}\|\|B_k\|\|u_k - u_*\|
\]

\[
+ \alpha_k \mu_k \|(F'_k - B_k)^{-1}\tilde{p}\|
\]

and estimates (7.25) and (7.27), we can derive

\[
\|u_{k+1} - u_*\| \leq (1 - \alpha_k + O(\|u_k - u_*\|) + \alpha_k \frac{2\beta}{1 - \rho} \frac{1}{4\beta} + \alpha_k O(\sigma_k)) \|u_k - u_*\|
\]

\[
= (1 - \alpha_k + O(\|u_k - u_*\|) + \alpha_k \frac{1}{1 - \rho} \frac{1}{2} + O(\sigma_k))\|u_k - u_*\|
\]

Recall that \(0 < \rho < \frac{1}{2}\).

Denote \(\hat{\epsilon} = O(\|u_k - u_*\|) + O(\sigma_k)\). Since \(\sigma_k > 0\), \(\|u_k - u_*\| > 0\), and by definition of big O notation, \(\hat{\epsilon} > 0\).

To establish \(q\)-linear convergence we will show that

\[
1 - \alpha_k + \hat{\epsilon} + \alpha_k \frac{1}{1 - \rho} \frac{1}{2} < 1,
\]

i.e.

\[
\hat{\epsilon} < \alpha_k - \alpha_k \frac{1}{2(1 - \rho)} = \alpha_k (1 - \frac{1}{2(1 - \rho)}).
\]
or
\[ \alpha_k > \frac{\hat{\epsilon}}{1 - \frac{1}{2(1-\rho)}}. \]

Using the expression for \( \alpha_k \) established above (7.32),
\[ \alpha_k = \min(1, \frac{\tau_k}{1 + O(\|F_k\|) + O(\sigma_k)}) > \frac{\hat{\epsilon}}{1 - \frac{1}{2(1-\rho)}}. \]

If \( \alpha_k = \frac{\tau_k}{1 + O(\|F_k\|) + O(\sigma_k)} \) then we need
\[ \tau_k \geq \frac{\hat{\epsilon}(1 + O(\|F_k\|) + O(\sigma_k))}{1 - \frac{1}{2(1-\rho)}} \quad (7.33) \]

Since \( \hat{\epsilon} = O(\|u_k - u_*\|) + O(\sigma_k) \), given \( \hat{\tau} \in (0, 1) \), there exists a neighborhood \( N \) of \( u_* \) and a constant \( \hat{\sigma} > 0 \) such that the quantities \( O(\|u_k - u_*\|) \), \( O(\|F_k\|) \), and \( O(\sigma_k) \) are small enough, so that the expression on the right hand side of (7.33) is less than or equal to \( \hat{\tau} \). Then the inequality (7.33) is satisfied for any \( u_0 \in N \) and any choice of the algorithmic parameters \( \tau_k \in [\hat{\tau}, 1] \) and \( \sigma_k \in [0, \hat{\sigma}] \).

If \( \alpha_k = 1 \) then we need
\[ 1 - \frac{1}{2(1-\rho)} > \hat{\epsilon}, \]
which again can be achieved when \( O(\|F_k\|) \), and \( O(\sigma_k) \) are sufficiently small. 

\[ \square \]
Chapter 8

Numerical Experiments

8.1 Implementation

One of our goals was to develop an efficient algorithm for the numerical solution of the problem, which can be used as a base for an automated routine for the determination of structure factor values and the CTF and noise function parameters. We construct an algorithm based on interior-point methodology in order to treat bound constraints efficiently. In this section we report our numerical experience with a perturbed and damped interior-point Gauss-Newton algorithm. The computational work was done on a SUN Ultrasparc 2 Workstation running SunOS 64 megabytes of memory and with a 200 MHz processor. The programs were written in Matlab and run under Version 5.1.

The function and Jacobian calculations for both formulations of the problem are programmed in Matlab and FORTRAN. Recall that the first formulation deals with the CTF and the noise parameters and the structure factor values as independent variables. In the second formulation the structure factor is implicit. Our implementation can be easily modified to use Newton’s or Levenberg-Marquardt method instead of the Gauss-Newton method. If Newton’s method is desired, the Hessian is calculated symbolically in Matlab which allows for flexibility if the function changes. We should note that a numerical error is introduced by performing $J^TJ$ multiplication. It is preferable to use a QR factorization of the Jacobian $J$. We have also calculated Jacobian and Hessian matrices analytically to check their correctness. Even though there is second derivative information for this problem, the finite difference approxi-
mation to Jacobian and Hessian matrices can be used both to check the correctness of the analytic and symbolic calculations and to use on its own when the analytic calculations become too cumbersome, as in the second formulation of the problem, or when exact computations or other approximations are too expensive.

The function calculations were checked for correctness against the values provided by EMAN software\textsuperscript{\textdagger} calculations. The values agreed to the available digits of accuracy. We observed that the function calculations are not very sensitive to the changes in data.

The code was tested on the Hock-Schittkowski set of problems ([13]). In order to do this we modified the input form and added the file containing the Hessian matrix.

One testing strategy was to set the measured total intensity values equal to the values predicted by the model, i.e. create a zero-residual problem. Then slightly perturb the values of $x$ and $y$ variables used to calculate the function value predicted by the model. The method exhibits quadratic convergence if the step length $\alpha$ approaches one, the perturbation $\mu$ approaches zero, and the Jacobian of the function of KKT residuals is nonsingular (given that all the assumptions are satisfied). Therefore we have another confirmation of the correctness of the code.

\subsection{Parameters}

The algorithm based on the perturbed and damped interior-point Gauss-Newton method was implemented. It is described in Section 6.5. The following parameters were chosen for the algorithm. In STEP 2 we chose

$$
\mu_k = \sigma_k \frac{x^Tv + z^Tw}{2n},
$$

\textsuperscript{\textdagger}EMAN software written by Steve Ludtke is available at http://ncmi.bioch.bcm.tmc.edu
where

\[ \sigma_k = \min(\eta_1, \eta_2(x^Tv + z^Tw)). \]

Following El-Bakry et al. [9] we have chosen the values \( \eta_1 = 0.2 \) and \( \eta_2 = 100 \). In STEP 5 (6.26) \( \beta = 10^{-4} \) and the backtracking factor \( \rho = 0.5 \).

### 8.1.2 User-Supplied Information

The user of the code needs to provide files with the input data: spatial frequency \( s \) and corresponding measured total intensity of the signal \( TI(s) \) for 5 data sets for each particle. Currently these data files are obtained from EMAN software. Spatial frequencies \( s \) (data points) are the same for a given particle from the same microscope for different defocus data sets. The sharp peak of the signal intensity curve near \( s=0 \) represents an artifact of the microscope, which must be ignored in the fitting procedure. Therefore the data corresponding to \( (0,0) \) and points before the first peak are excluded from the data fitting process. The range to be ignored must be determined. If the peak is wide, the choice has to be made as to which part of the peak to disregard. For certain data, peaks do represent valuable information. Generally, the fitting starts at the spatial frequency roughly equal to the inverse of the particle size, because for smaller spatial frequencies the useful information is lost.

The EMAN software currently also provides a file with the values of CTF and noise parameters used as initial guesses for the developed code.

The user may choose to overwrite maximum number of iterations allowed, the vector of upper bounds, the tolerance \( \epsilon \), the line-search parameter \( \alpha \), the backtracking parameter \( \rho \), and the matrix of second derivative information \( S \), when using Newton's or Levenberg-Marquardt method.
8.1.3 Bounds

When we use the formulation of the problem explicitly including the structure factor, we need to introduce an upper bound for it, so that the values are reasonable. The upper bound value is a difficult question and depends on the particle. The bounds on the other variables are empirical and based on the observed values.

8.1.4 Scaling

The problem is badly scaled due to the difference in magnitude of the variables. One of the CTF parameters (amplitude contrast) varies from 0 to 1, but the observed values, available from both electron microscopy and X-ray crystallography, are below 0.15. At the same time the B-factor varies from 0 to 1000. The other parameters have roughly the same magnitude varying from 0 to somewhere around 50. To improve the scaling, we scaled parameter B by, first, the factor of 1000 and, then, by the factor of 100. Numerical experimentation showed that scaling B did not effect the computational results significantly.

8.2 Singularity

Both the Hessian approximation matrix $J^T J$ and the Jacobian of the KKT conditions $F'$ are singular at the solution. The estimates for the reciprocal of the condition of these matrices in the 1-norm obtained by the LINPACK condition estimator were approximately $10^{-28}$.

Some of the noise parameters are very close in value, which contributes to singularity. To fix the linear dependence of the columns containing nearly same values, we can fix the noise parameters (reducing the number of variables), as well as some other parameters that are likely to be similar, such as amplitude contrast parameter $C_A$. 
Even if the noise parameters are different, the Hessian approximation matrix $J^T J$ remains singular. This is due to the small values of the CTF, corresponding to high spatial frequencies. The structure of the Jacobian matrix $J$ is illustrated in Figure 8.1. The left-hand side of the matrix corresponds to the partial derivatives of the residual with respect to the variables representing CTF and noise parameters. The right columns of $J$, corresponding to the partial derivatives of the residual with respect to the structure factor variables, are 5 diagonal sub-matrices, stacked on top of each other. Each such sub-matrix has the CTF, corresponding to the data set, on the diagonal. Since the CTF values are very small in the rightmost columns, we have several nearly zero-valued columns. The problems caused by these columns could

![Figure 8.1 Structure of the Jacobian matrix $J$.](image)

be avoided if the second mathematical formulation is used. Excluding the structure
factor variables, their derivatives (equal to CTF) are excluded, as well. Note that the
trade off when using the second formulation is the complicated computations.

To overcome the numerical problems caused by the singularity of the Hessian ap-
proximation, we add the term \( \rho_k I \), where \( \rho_k \) is a number and \( I \) is the identity matrix.
Thus, we arrive at the Levenberg-Marquardt method (7.12). There are various strate-
gies to choose \( \rho_k \). We chose it to be proportional to the value of unperturbed KKT
residuals \( \| F(u_k) \| \). When \( \rho_k = 10^{-1} \| F(u_k) \| \) we observe very fast convergence on a
zero-residual modification of our problem from an initial guess close to the solution.

8.3 Initial Guess

The issue of finding a good initial guess is an important one. We are concerned with
having good initial guess firstly, for the values of CTF and noise parameters, secondly,
for the vector of structure factor values, and, finally, for the dual variables. While
the biologists might have some feeling for the initial values of the parameters for each
specific data set, we cannot always rely on them for a good initial guess. In order for
the user not to have to provide an initial guess, it has to be produced algorithmically
from the data. It remains an open question as to how it can be generated based on
the data.

In the current implementation, the initial guess is read from a file provided by the
user. The values of CTF and noise parameters produced by manual fit are used as
initial guess for the algorithm.

It is a challenge to come up with a good initial guess for the vector of structure
factor values. There is often no such initial guess provided by biologists. The reason
is that, if there is an information about structure factor from X-ray scattering, these
values are just plugged in to find the values of CTF and noise parameters. One
option is to assume that it is a unit vector, solve the resulting (smaller) nonlinear
least-squares problem and use $\frac{\sqrt{TI-NI}}{CTP^2}$ as an initial guess. Another possibility is to use $e^{-\text{const} \times s}$, since the structure factor vector mapped against spatial frequency roughly has an exponential shape. I have experimented with several heuristic initial guesses for the vector of structure factor values, such as exponential form and different constant values. It seems that independently of the starting estimate the final answers are consistent.

The use of the second formulation of the problem, where the linear variables, i.e. the structure factors, are implicit, allows one to avoid providing an initial guess for the vector of structure factor values.

Another issue that needs to be addressed is a good initial guess for dual variables. Generally it is a difficult question. There are some heuristics that provide initial guess for dual variables. Currently, we use the vector of ones as initial guess. Since the solution should be strictly feasible in our problem, then, by complementarity, the dual variables will all be zero at a solution.

### 8.4 Other Approaches

In this section the experience with other methods used to solve our problem is discussed.

I have tested the nonlinear interior-point solver based on the algorithm described in El Bakry et al. [9] on the Baylor problem, the Hock-Schittkowski problem set ([13]), and other test problems that I have created. The code seems to be sensitive to gradient and function errors. It also appears not to be robust in absence of equality constraints. The code took many iterations to converge from initial guess equal to exact solution, possibly because of a poor initial guess for the dual variables. Another problem encountered was a very short step length that caused the algorithm to stall.
Originally I intended to modify this code to adapt to our problem, however, I decided to implement my own interior-point algorithm, instead.

Another code based on the same algorithm ([22]) was also tested. It is a FORTRAN implementation of the interior-point method for nonlinear programming problems which includes a detailed line search.

The optimizer based on limited memory BFGS (Broyden Fletcher Goldfarb Shanno) method available through NEOS software** was also used to solve the problem. I provided function, gradient, and initial guess routines in FORTRAN.

Neither software provided satisfactory solution of the problem.

8.5 Data

The numerical experiments were performed on data available for the following particles: α-crystallin AB, α-crystallin B, α-crystallin A, human fatty acid synthase, and herpes simplex virus (HSV-1). Usually, there are five or six data sets available for each particle, each containing from around 60 to several hundred data points. Usually each data set has information from 50 to 1000 particles. The pixel size of the images depends on the particle.

The micrographs were recorded at the National Center for Macromolecular Imaging at Baylor College of Medicine at the Texas Medical Center. The Center has microscopes with 3Å, 5Å and 15Å resolution.

8.5.1 Human Fatty Acid Synthase

The micrographs were recorded with an electron cryo-microscope operating at 100kV. The range of the spatial frequency fitted by our algorithm was between 0.003125 to 0.1, i.e. all data points except the origin.

For comparison purposes top left plot in Figure 8.3 illustrates one of the best “manual” data fit for the Human Fatty Acid Synthase. This “manual” data fit was performed with the structure factor know from X-ray scattering.

8.5.2 Herpes Virus Particles

The Herpes Simplex Virus (HSV-1) consists of four distinct compartments. Typical HSV-1 capsids contain seven proteins that form an icosahedron that is 160 Å thick and has a diameter of 1250 Å (see Saad et al. [21]). All micrographs were recorded with a JEOL4000 electron cryo-microscope operating at 400kV, under minimal dose conditions (∼7 electrons/Å²). Forty selected micrographs were digitized on a scanner. 1300 particle images (240x240 pixels) were selected automatically.

The question arising from dealing with all data sets is where to cut off the data to be fitted. Data from herpes virus oscillates to higher resolutions than other data. Size of a herpes particle is about 1200 Å therefore we start fitting the data at 0.001 Å⁻¹. The peak near the origin is from incoherent scattering.

For comparison purposes top left plot in Figure 8.5 illustrates one of the best “manual” data fit for the Herpes Simplex Virus.

8.5.3 α-crystallin

The α-crystallin protein is the predominant lens protein of the human eye. It plays a lot of functions for example, it acts as a molecular chaperone and stabilizes neighboring proteins with respect to heat and UV damage. Despite a great effort, the
protein has not been crystallized and most conjectured models are based on similarity with proteins with similar functions (see Baldwin et al. [3]). α-crystallin appears as isoforms containing two subunits, A and B. We consider the isoforms where only A subunits are present, only B subunits are present, and where the A to B ratio is 3:1. The latter isoform is the predominant protein in the eye. The A and B units appear in numerous places in the body but together only in the eye-lens. They are known to have quite distinct mechanical properties so that the similarity of the structures of the two isoforms should not be assumed too readily ([3]).

• α-crystallin A

The micrographs were recorded with an electron cryo-microscope operating at 100kV. For α-crystalline A data the first peak actually represents data, so only the first point (0,0) and points before the top of the peak should be cut off. The size of the particle is around 140 Å, so the data fit should start at \( s = \frac{1}{140 \text{Å}} = 0.007 \text{Å}^{-1} \).

• α-crystallin AB

We have the data for the α-crystalline AB(3:1). Particle size is around 120 Å, so the data fit starts at \( s = 1/120 = 0.0083 \text{Å}^{-1} \).

• α-crystallin B

The micrographs were recorded with an electron cryo-microscope operating at 400kV. This is a challenging problem. The data for α-crystallin B has higher resolution than others, because 90 particles are used to generate data. It is good to possibly 10 Å or maybe even 9 Å. Therefore the spatial frequency ranges from 0.007 Å\(^{-1}\) (starting at the top of first peak) to 0.0999 Å\(^{-1}\) (100 data points) or to 0.111 Å\(^{-1}\) (111 data points).
8.6 Numerical Results

The results of our numerical experience are summarized in Table 8.1. The first column gives the particle type. The \( n \) column gives the dimension of the problem, which equals to the size of the vector of structure factor values (i.e. the size of one data set) plus 40 (i.e. the number of the unknown CTF and noise function parameters for 5 data sets). The third column gives the final objective function value. The forth column states the value of the KKT residual at the termination of the algorithm. The fifth column provides a relative error, defined as the ratio of the objective function value to the size of the vector of unknowns. Last column gives the total computing time in CPU seconds that algorithm took.

The results in Table 8.1 were produced when the exit criterion for the algorithm was the absolute value of the KKT residual to be less the tolerance \( 10^{-7} \). That criterion was too strict and the algorithm terminated due to short step-length in the linesearch procedure. We introduced the following heuristic to overcome stalling. The exit value of the variables was slightly perturbed and used as an initial guess. Then the algorithm was restarted. The algorithm combined with this simple heuristic achieved very good objective function values.

The Figures 8.2, 8.4, 8.6, 8.7, and 8.8 illustrate the fit obtained with the use of our formulation and algorithm for all the particles mentioned above. The plot in the

| Type of Particle             | \( n \) | \( f \)   | \( ||F|| \) | Rel | CPU  |
|------------------------------|--------|----------|------------|-----|------|
| Human Fatty Acid Synthase    | 103    | 0.53338  | 0.056952   | 0.005 | 22.3 |
| Herpes Virus Particles       | 279    | 3.7729   | 0.30941    | 0.013 | 1405.1 |
| \( \alpha \)-crystallin A   | 107    | 0.057627 | 0.0032962  | 0.0005 | 167.91 |
| \( \alpha \)-crystallin AB   | 104    | 3.5288   | 0.19409    | 0.0339 | 76.860 |
| \( \alpha \)-crystallin B   | 266    | 2.2884   | 1.1472     | 0.008 | 1261.1 |
top left corner of each figure shows 5 distinct data sets that need to be fitted for each particle. The other plots show the data, represented by asterisks (*), and the fit, represented by a solid line. We can see that in most cases the fit is accurate. The quality of the fit increases as the points, corresponding to the low frequency part of the data curve are excluded. The current model does not always capture the phenomena responsible for the peaks if the data curve. Therefore, the user can decide how many points to exclude from the fitting process to get the optimal fit.

8.7 Summary

We have tried several methods and software on our problem. The NEOS software and the other codes described in Section 8.4 did not work successfully. With our algorithm we tried two different mathematical formulations as well as Gauss-Newton and Levenberg-Marquardt techniques for dealing with least squares. The results in Table 8.1 were produced using the first formulation of the problem. Levenberg-Marquardt approach was used in the final calculations since it allows to reduce the numerical errors caused by ill-conditioning.

Our code features a number of advantages.

It simultaneously fits the CTF and noise parameters and the estimates the structure factor values.

The quality of fit can be precisely judged from the objective function value.

Our fitting procedure can successfully fit the model to the data over a greater range of spatial frequencies than the "manual" procedure.

It is easy to incorporate changes in the model. Currently there are 8 unknown parameters for each data set, as well as a vector of structure factor values, whose size equals that of the data set. As was previously discussed (see Section 3.1), there is interest in making the model more complicated to account for the currently under-
represented phenomena. One of the first such changes, would, perhaps, be the more complicated model of the envelope function, currently modeled as a simple Gaussian. There is no conceptual difficulty in adding as many parameters as desired to the model of the envelope function, or the entire total intensity model.

The user can take advantage of Gauss-Newton, Levenberg-Marquardt, or Newton's method, when using the algorithm.

The objective value obtained by algorithm is very low in most cases. In cases, when is was not reduced to a small value, we were attempting to fit the peaks of the intensity curve, that are not described by the current model. Removing points to the left of the peak led to an accurate fit with low objective function values.
Figure 8.2  Data fit for human fatty acid synthase.
Figure 8.3  Comparison of data fits for human fatty acid synthase.
Figure 8.4 Data fit for the herpes simplex virus (HSV-1).
Figure 8.5  Comparison of data fits for the herpes simplex virus (HSV-1).
Figure 8.6 Data fit for α-crystallin A.
Figure 8.7 Data fit for α-crystallin AB.
Figure 8.8 Data fit for α-crystallin B.
Chapter 9

Conclusions

In this thesis, we consider an important problem in molecular biology. The previous attempts to solve this problem were unsatisfactory. We mathematically formulate the problem in two related but different ways as a nonlinear least-squares problem with simple bounds. Our first formulation is a mixed linear-nonlinear least-squares problem, where the residual function is linear in the variables representing structure factors and nonlinear in the others. Taking advantage of the linearity, we propose the second formulation, that reduces the number of variables, perhaps, at the cost of an increase in the non-linearity of the problem. The second formulation does avoid the need for an initial guess of the structure factor values.

Next, we developed an algorithm for this class of problems. It is based on interior-point methodology which deals efficiently with the inequality constraints. We also take advantage of the special structure of our problem. Interior-point approach is combined with the Gauss-Newton method, which is especially successful on zero- and small-residual problems. We would like to point out that our implementation allows for the Levenberg-Marquardt method to be used. Newton’s method also can be used, i.e. we can use the second derivative information, if desired.

The convergence results are proved for the interior-point method applied to a nonlinear least-squares problem with general constraints. It is shown that when perturbation of $F'$ is zero at the solution the highly desirable quadratic convergence of Newton’s method is preserved. When the perturbation matrix is small we establish linear convergence.
Our numerical experimentaion confirms that our approach is successful and demonstrates a good fit of the model to the data. In most cases, the objective function is reduced to a very small value.
Bibliography


Appendix A

Matlab Implementation of the Algorithm

```matlab
% the driver
clear;
global sdata cs V t1 t2 t3 t4 t5 m n R DR DF;
[particle,probnum,m,n,x,b,funcname,gradname,maxiter,S,tau,alpha,rho,
epsilon,z,v,w] = probdata;
[x,z,v,w,obj] = ip(m,n,x,b,S,epsilon,maxiter,tau,alpha,rho,particle,
probnum,gradname,funcname,z,v,w);
```

How to run this code:

In 'getdata' file:

- LOAD correct files;
- figure out how many points to exclude from fit:
  - start from point s=1/size(particle)
  - ex: sdata = plot0620(6:48,1); t1 = plot0620(6:48,11);
- READ in spatial frequency 'sdata' and measured intensities 't1', 't2',
  't3', 't4', 't5' from correct files.

In 'probdata' file:

- PUT correct 'particle';
- PUT correct 'probnum';
- PUT correct initial guess 'x';
PUT correct 'maxiter';
no need to change 'm','n' as they are read from the data;
upper bound 'b', UBSF' is set to standard value;
'funcname', 'gradname' are set to standard calls for the
first (explicit) formulation of the problem.

In 'main' file:

PUT correct 'epsilon', 'tau', 'alpha', 'rho';
'S' is set to zero (Gauss-Newton).

In 'func' file:

PUT correct 'cs', 'V' constants for particle in use.

In 'ip' file:

slack 'z' is set to upper bound 'b' minus 'x';
multipliers 'v','w' are set to the vectors of ones.

% contains all the problem information

function [particle,probnun,m,n,x,b,funcname,gradname,maxiter,S,tau,
alpha,rho,epsilon,z,v,w] = probdata

global sdata cs V t1 t2 t3 t4 t5 m n;

m = size(sdata);

m = m(1); % number of data points

n = m+40; % number of variables

probnun = 1;

particle = 'Herpes.';

[x,cs,V] = getdataHerpes;

x = x';
x(41:n) = .1*ones(m,1);
% yb is the upper bound for the vector of structure factor values
UBSF = 100;
yb = UBSF*ones(m,1);
% b is the upper bound for the parameters
b = [10 80 10 125 12 8 pi/4 1000 10 80 10 125 12 8 pi/4 1000
    10 80 10 125 12 8 pi/4 1000
    10 80 18 125 12 8 pi/4 1000
    10 80 10 125 12 8 pi/4 1000 ];
b = [b'; yb];
funccname = 'func';
gradname = 'grad';
maxiter = 500;
% S is the second-order information in the second derivative of F
S = zeros(n);
% epsilon is the convergence tolerance
epsilon = 1.e-7;
% tau is the damping parameter
tau = .8;
% alpha is the parameter in globalized LINESEARCH
alpha = 1.e-4;
%alpha = 0;
% rho is the stabilizing parameter in J'*J + rho*I
rho = 0;
% CHOOSE AN INITIAL GUESS z>0
z = b - x;
if (min(z)<0)
    disp('Infeasible initial guess!')
    break
end

% INITIALIZE MULTIPLIERS
v = ones(n,1); w = ones(n,1);

% getdataHerpes.m contains the Herpes specific information
function [x,cs,V] = getdataH

global sdata cs V t1 t2 t3 t4 t5;

% load files that contains data sdata and function values ydata
load /home/olena/Research/Baylor/DATA/HERPES/jj0477.dat
load /home/olena/Research/Baylor/DATA/HERPES/jj0507.dat
load /home/olena/Research/Baylor/DATA/HERPES/jj0804.dat
load /home/olena/Research/Baylor/DATA/HERPES/jj1169.dat
load /home/olena/Research/Baylor/DATA/HERPES/jj4047.dat

sdata = jj0477(:,1);
t1 = jj0477(:,11);
t2 = jj0507(:,11);
t3 = jj0804(:,11);
t4 = jj1169(:,11);
t5 = jj4047(:,11);

% initial guess for Herpes
jj0477
x(6) = 1.243269; x(8) = 76.045998; x(5) = 9.756607;
x(7) = asin(8.745200/100); x(1) = sqrt(1.634980); x(3) = 3.650160;
x(2) = 5.247150; x(4) = (4.087450)^2;

%jj0507
x(14) = 0.724813; x(16) = 91.254997; x(13) = 11.769891;
x(15) = asin(12.927800/100); x(9) = 0.1; x(11) = 9.429680;
x(10) = 4.960990; x(12) = (3.136875)^2;

%j0804
x(22) = 0.787911; x(24) = 163.498001; x(21) = 2.034220;
x(23) = asin(7.604600/100); x(17) = sqrt(0.152090); x(19) = 0.0001;
x(18) = 2.927760; x(20) = (4.942975)^2;

%jj1169
x(30) = 0.691331; x(32) = 82.889801; x(29) = 11.432286;
x(31) = asin(8.060840/100); x(25) = 0.1; x(27) = 17.395424;
x(26) = 8.212935; x(28) = (2.155895)^2;

%jj4047
x(38) = x(22); x(40) = x(24); x(37) = x(21); x(39) = x(23);
x(33) = x(25); x(35) = x(19); x(34) = x(18); x(36) = x(20);

% evaluate function values for my problem
% min f(x)  (objective)
% x + s = b

function R = func(x)
global sdata cs V s2 sqs m n t1 t2 t3 t4 t5 y;
global sshift1 sshift2 sshift3 sshift4 sshift5;
global expo1 expo2 expo3 expo4 expo5;
global ctf1 ctf2 ctf3 ctf4 ctf5;
global noise1 noise2 noise3 noise4 noise5;
global ctf12 ctf22 ctf32 ctf42 ctf52;
global exp1 exp2 exp3 exp4 exp5;
global exss1 exss2 exss3 exss4 exss5;
global gamma1 gamma2 gamma3 gamma4 gamma5;
global lambda;
s2 = sdata.^ 2;
s4 = s2.^ 2;
sqs = sqrt(sdata);
y = x(41:n);
lambda = 12.2639/sqrt(1.000.0*.97845*V*V);
const1 = -2*pi*cs * (1.e+7)/4*lambda3 * s4;
const2 = -2*pi*5000 * lambda * s2;
gamma1 = const1 - x(6) * const2;
gamma2 = const1 - x(14) * const2;
gamma3 = const1 - x(22) * const2;
gamma4 = const1 - x(30) * const2;
gamma5 = const1 - x(38) * const2;
sshift1 = sin(gamma1 + x(7));
sshift2 = sin(gamma2 + x(15));
sshift3 = sin(gamma3 + x(23));
sshift4 = sin(gamma4 + x(31));
sshift5 = sin(gamma5 + x(39));
exp1 = exp(-x(8)*s2);
\[
\begin{align*}
\text{exp2} &= \exp(-x(16)s2) \\
\text{exp3} &= \exp(-x(24)s2) \\
\text{exp4} &= \exp(-x(32)s2) \\
\text{exp5} &= \exp(-x(40)s2) \\
\text{exss1} &= \exp1.*\text{sshift1} \\
\text{exss2} &= \exp2.*\text{sshift2} \\
\text{exss3} &= \exp3.*\text{sshift3} \\
\text{exss4} &= \exp4.*\text{sshift4} \\
\text{exss5} &= \exp5.*\text{sshift5} \\
\text{ctf1} &= x(5)\text{exss1} \\
\text{ctf2} &= x(13)\text{exss2} \\
\text{ctf3} &= x(21)\text{exss3} \\
\text{ctf4} &= x(29)\text{exss4} \\
\text{ctf5} &= x(37)\text{exss5} \\
\text{expo1} &= \exp(-\pi/2/4x(4)s2 - x(1)sqs - x(2)sdata) \\
\text{expo2} &= \exp(-\pi/2/4x(12)s2 - x(9)sqs - x(10)sdata) \\
\text{expo3} &= \exp(-\pi/2/4x(20)s2 - x(17)sqs - x(18)sdata) \\
\text{expo4} &= \exp(-\pi/2/4x(28)s2 - x(25)sqs - x(26)sdata) \\
\text{expo5} &= \exp(-\pi/2/4x(36)s2 - x(33)sqs - x(34)sdata) \\
\text{noise1} &= x(3) \times \text{expo1} \\
\text{noise2} &= x(11) \times \text{expo2} \\
\text{noise3} &= x(19) \times \text{expo3} \\
\text{noise4} &= x(27) \times \text{expo4} \\
\text{noise5} &= x(35) \times \text{expo5} \\
\text{ctf12} &= \text{ctf1}.^-2 \\
\text{ctf22} &= \text{ctf2}.^-2
\end{align*}
\]
ctf32 = ctf3.^ 2;
ctf42 = ctf4.^ 2;
ctf52 = ctf5.^ 2;
c1 = t1-noise1;
c2 = t2-noise2;
c3 = t3-noise3;
c4 = t4-noise4;
c5 = t5-noise5;
r1 = ctf12.*y - c1;
r2 = ctf22.*y - c2;
r3 = ctf32.*y - c3;
r4 = ctf42.*y - c4;
r5 = ctf52.*y - c5;
R = [r1; r2; r3; r4; r5];

% evaluate gradient values for my problem
function DR = grad(x)
global sdata s2 sqs m n y;
global sshift1 sshift2 sshift3 sshift4 sshift5;
global expo1 expo2 expo3 expo4 expo5;
global ctf1 ctf2 ctf3 ctf4 ctf5;
global noise1 noise2 noise3 noise4 noise5;
global ctf12 ctf22 ctf32 ctf42 ctf52;
global exp1 exp2 exp3 exp4 exp5;
global exss1 exss2 exss3 exss4 exss5;
global gamma1 gamma2 gamma3 gamma4 gamma5;

global lambda;

excs1 = exp1.*cos(gamma1 + x(7));
excs2 = exp2.*cos(gamma2 + x(15));
excs3 = exp3.*cos(gamma3 + x(23));
excs4 = exp4.*cos(gamma4 + x(31));
excs5 = exp5.*cos(gamma5 + x(39));

pi24 = pi^- 2/4;
dgamma = 2*pi*5000*lambda*s2;

dr1 = zeros(m,n);
dr2 = zeros(m,n);
dr3 = zeros(m,n);
dr4 = zeros(m,n);
dr5 = zeros(m,n);

dr1(:,1) = -sqs.*noise1; % 1st column of dr1

dr1(:,2) = -sdata.*noise1;
dr1(:,3) = expo1;
dr1(:,4) = -pi24*noise1.*s2;
dr1(:,5) = 2*ctf1.*exss1.*y;
dr1(:,7) = 2*x(5)*ctf1.*exp1.*cos(gamma1 + x(7)).*y;
dr1(:,6) = dr1(:,7).*dgamma;
dr1(:,8) = -2*ctf12.*s2.*y;
dr1(:,41:n) = diag(ctf12);
dr2(:,9) = -sqs.*noise2;
dr2(:,10) = -sdata.*noise2;
dr2(:,11) = expo2;
dr2(:,12) = -pi24*noise2.*s2;
dr2(:,13) = 2*ctf2.*exss2.*y;
dr2(:,15) = 2*x(13)*ctf2.*excs2.*y;
dr2(:,14) = dr2(:,15).*dgamma;
dr2(:,16) = -2*ctf22.*s2.*y;
dr2(:,41:n) = diag(ctf22);
dr3(:,17) = -sqs.*noise3;
dr3(:,18) = -sdata.*noise3;
dr3(:,19) = expo3;
dr3(:,20) = -pi24*noise3.*s2;
dr3(:,21) = 2*ctf3.*exss3.*y;
dr3(:,23) = 2*x(21)*ctf3.*excs3.*y;
dr3(:,22) = dr3(:,23).*dgamma;
dr3(:,24) = -2*ctf32.*s2.*y;
dr3(:,41:n) = diag(ctf32);
dr4(:,25) = -sqs.*noise4;
dr4(:,26) = -sdata.*noise4;
dr4(:,27) = expo4;
dr4(:,28) = -pi24*noise4.*s2;
dr4(:,29) = 2*ctf4.*exss4.*y;
dr4(:,31) = 2*x(29)*ctf4.*excs4.*y;
dr4(:,30) = dr4(:,31).*dgamma;
dr4(:,32) = -2*ctf42.*s2.*y;
dr4(:,41:n) = diag(ctf42);
dr5(:,33) = -sqs.*noise5;
dr5(:,34) = -sdata.*noise5;
\texttt{dr5(:,35) = expo5;}  
\texttt{dr5(:,36) = -pi24*noise5.*s2;}  
\texttt{dr5(:,37) = 2*ctf5.*exss5.*y;}  
\texttt{dr5(:,39) = 2*x(29)*ctf5.*excs5.*y;}  
\texttt{dr5(:,38) = dr5(:,39).*dgamma;}  
\texttt{dr5(:,40) = -2*ctf52.*s2.*y;}  
\texttt{dr5(:,41:n) = diag(ctf52);}  
\texttt{DR = [dr1;dr2;dr3;dr4;dr5];}  

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ip.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the interior-point algorithm
% use slack-variable form of KKT
function [x,z,v,w,obj] = ip(m,n,x,b,S,epsilon,maxiter,tau,alpha,
rho,particle,probnunm,gradname,funcname,z,v,w)
global sdata cs V t1 t2 t3 t4 t5 m n R DR DF;
s = num2str(probnunm);
s = strcat(particle,s);
filename = strcat('output.',s);
fid = fopen(filename,'w');
R = feval(funcname, x);
J = feval(gradname, x);
obj = .5*R'*R
x0 = x; z0 = z; v0 = v; w0 = w; u = [x;z;v;w];
f1 = J'*R + w - v;
f2 = x+z-b;
\texttt{cost = f1'*f1 + f2'*f2 + (x.*v)'*(x.*v) + (z.*w)'*(z.*w);}
l2norm = sqrt(cost);
% test for convergence
% convergence criterion is norm of unperturbed KKT
k=1;
search_stopped = 0;
sigma = min(.2, 100*(x'*v+z'*w));
sigma = sigma*(x'*v+z'*w)/(2*n)
tic;
while ( l2norm > epsilon*(1+norm(u)) & k < maxiter )
% (1) select a perturbation parameter sigma
if f1'*f1/((x.*v)'*(x.*v) + (z.*w)'*(z.*w)) < 1.e1
    sigma = min(.2, 100*(x'*v+z'*w));
    sigma = sigma*(x'*v+z'*w)/(2*n)
end
f3 = x.*v-sigma;
f4 = z.*w-sigma;
% obtain current value of cost function
cost = f1'*f1 + f2'*f2 + (x.*v)'*(x.*v) + (z.*w)'*(z.*w);
% (2) obtain new step
rho = 10*l2norm;
[du,dirder] = linear solver(f1,f2,f3,f4,n,R,J,S,b,x,z,v,w,sigma,
rho,fid);
% (3) adjust step-length:
% - perform damping for feasibility
% - perform line-search for descent
tau=max(.8,(1-1.e-2*(x'*v+z'*w)))
alphamin = damping(du,u,n,tau,fid);
[x,z,v,w,R,J,f1,f2,12norm,search_stopped] = lns(sigma,alphamin,
dirder,du,u,n,b,cost,alpha,gradname,funcname,fid);
u = [x;z;v;w]; obj = .5*R'*R
if search_stopped
    x = x + 1.e-3; z = b-x; v = v0; w = w0; u = [x;z;v;w];
    if (min(z)<0)
        x = x-1.e-3 + 1.e-4; z = b-x; v = v0; w = w0; u = [x;z;v;w];
        if (min(z)<0)
            break;
    end
end
end
k=k+1
end
fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% linearsolver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [du,dirder] = linearsolver(f1,f2,f3,f4,n,R,J,S,b,x,z,v,w,
sigma,rho,fid)
xinv = min(1.e14,1./x);
zinv = min(1.e14,1./z);
B= J'*J + rho*eye(n);
C = diag(w.*zinv) + diag(v.*xinv);
A = B+C;
con_num = rcond(A)
\texttt{rhs = J'*R - sigma.*xinv + sigma.*zinv + w.*f2.*zinv;}

\texttt{dx = - A * rhs;}

\texttt{dz = - dx - f2;}

\texttt{dv = - (f3+v.*dx).*xinv;}

\texttt{dw = - (f4+w.*dz).*zinv;}

\texttt{du = [dx;dz;dv;dw];}

\texttt{dirder = [B*f1+f2+diag(v)*f3 ; f2+diag(w)*f4;}

\texttt{-f1+diag(x)*f3; f1+diag(z)*f4];}

\texttt{dirder = dirder' * du;}

\begin{verbatim}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% damping.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function alphamin = damping(du,u,n,tau,fid)

alphamin = -tau/min([-1; du./u]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lns.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x,z,v,w,R,J,f1,f2,l2norm,search_stopped] = lns(sigma,step,

dirder,du,u,n,b,cost,alpha,gradname,funcname,fid)

backt = 0.5;

maxstep = 10;

iter = 0;

while iter <= maxstep

x = u(1:n) + step*du(1:n);

z = u((n+1):2*n) + step*du((n+1):2*n);

v = u((2*n+1):3*n) + step*du((2*n+1):3*n);

w = u((3*n+1):4*n) + step*du((3*n+1):4*n);

R=feval(funcname, x); J=feval(gradname, x);

\end{verbatim}
$f_1 = J'*R + w - v; \ f_2 = x+z-b;$

costplus = $f_1'*f_1 + f_2'*f_2 + (x.*v)'*(x.*v) + (z.*w)'*(z.*w);$

l2norm = sqrt(costplus)

if (.5*costplus <= (.5*cost + dirder*alpha*step))
    break;
end

iter = iter + 1;

step = backt * step;
end

search_stopped = 0;

if iter <= maxstep
    if iter = 0;
        disp([' * LINESEARCH: (Linesearch exited at backtrack ',
            int2str(iter),')]);
    end

else
    disp(' * LINESEARCH: (Linesearch stopped due to short steplength)');
    search_stopped = 1;
end