

DATA FITTING
AND LEAST-SQUARE ESTIMATION OF
DIELECTRIC DATA

DATA FITTING AND LEAST-SQUARE
ESTIMATION OF NONLINEAR PARAMETERS FOR
MODELS OF
DIELECTRIC RELAXATION DATA

By

HAI JUN ZOU, B.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Science

McMaster University

©Copyright by HAI JUN ZOU, June 2007

MASTER DEGREE OF SCIENCE (2007) McMaster University
COMPUTING AND SOFTWARE Hamilton, Ontario

TITLE: DATA FITTING AND LEAST-SQUARE ESTIMATION
OF NONLINEAR PARAMETERS FOR
MODELS OF DIELECTRIC RELAXATION DATA

AUTHOR: Hai Jun Zou, B.Sc. (McMaster University)

SUPERVISOR: Dr. Sanzheng Qiao

NUMBER OF PAGES: 16, 133

Abstract

The work in this thesis is to develop a tool for calculating the parameters corresponding to certain theoretical model of dielectric relaxation phenomena and then doing the curve fitting using the result after fetching the data from the user. To our best knowledge, this the first such tool to calculate the parameters corresponding to certain theoretical model of dielectric relaxation phenomena while the user only need to provide the experimental data. The parameters are calculated by using a nonlinear least square algorithm implemented in Matlab and a nonlinear function minimizer available in Matlab. The way to do the curve fitting is not by the traditional way such as cubic spline but by calculating the simulated data using the chosen model and the calculated result for the parameters.

The available mathematical models include all of popular theoretical models, the Cole-Davidson (DC), the Kohlrausch-Williams-Watts (KWW), the Havriliak-Negami (HN) and the model proposed by R. Hilfer (FD).

There are two ways to calculate the parameters for each model as mentioned before. The result returned by this system may not be unique. Especially if the frequency range of data is not wide enough, the result would most likely be non-unique. Since the iterative method is used in the system, it is suggested that the user provides the initial values for the system with his best knowledge or background for the data and the tested sample related to dielectric relaxation process.

It is normal if there is a part having worse fitting than the other parts.

One of reasons could be the mathematical model's defect, which the model does not work for that part. For the further information, please contact me by email at [zouhaijun at yahoo.com](mailto:zouhaijun@yahoo.com).

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Dr. Sanzheng Qiao for the guidance, the valuable comments and patience which helped me to complete this work. Without his support, this work could not have been done. I also thank him for his very useful suggestion when I developed the system, very careful reading and correcting this thesis, and also his thoughtful guidance anytime when I need the help by changing his own schedule. All of those I learned from Dr. Qiao including the way how to overcome an issue and how to pick even single word or a sentence in my thesis, are very helpful and useful not only for my graduate study but also for my career in the future.

I acknowledge Dr. Spencer Smith and Dr. Emil Sekerinski for their careful reading of, and helpful suggestions on my thesis.

Finally and foremost, I thank my wife, my family for their love, patience and support, my colleagues and my friends for their support and encouragement during my work, which let me be able to concentrate to my work, finally finish my work.

Contents

List of Figures	11
List of Tables	13
1 Introduction and Background Information	17
1.1 Introduction	18
1.1.1 Dielectric Relaxation	18
1.1.2 Symbols, Concepts and Definitions	21
1.2 Literature Review	28
1.3 Introduction of Our Method	31
1.4 Thesis Structure	32
2 Theorems	33
2.1 Theories of Relaxation	34
2.1.1 Dielectric Relaxation	35
2.1.2 Mechanical Relaxation	36
2.2 Introduction of Four Models	37
2.2.1 Model of KWW, High Frequency Part, when $ \mu\tau_B \gg 0$	37

2.2.2	Model of KWW, Low Frequency Part, when $ \mu\tau_\beta \approx 0$	38
2.2.3	Model of DC	38
2.2.4	Model of HN	38
2.2.5	Model of FD	39
2.3	Least Squares Data Fitting	39
3	Numerical Software and System Design	41
3.1	Algorithms	42
3.1.1	Background	42
3.1.2	Algorithm	43
3.1.3	Method I and Method II	44
3.2	System Design	46
3.2.1	Scope of the Software Product	46
3.2.2	Software Requirements	46
3.2.3	System Architecture	48
3.3	Data Generation and Checking	51
3.4	Accuracy, Efficiency and Robustness	55
3.4.1	Introduction of Gamma Function	57
3.4.2	Scaling and Gamma Function Decomposition	57
3.4.3	Horner's Rule, Series Handling and Truncations	64
3.4.4	Models Switch Method	68
3.4.5	Improving Correctness and Robustness	69
4	Experiments	75

4.1	Procedure to Carry Out an Experiment	76
4.2	Method to Choose the Initial Values	78
4.2.1	Method to Choose the Initial Value of τ	78
4.2.2	Method to Choose the Initial Value of α , β and γ	79
4.2.3	Method to Choose the Initial Value of ϵ_0 and ϵ_∞	79
4.3	Data Generation and Test Plan	81
4.3.1	Data Generation	81
4.3.2	Test Plan	83
4.4	Discussion and Results	85
4.4.1	Discussion of Testing Using Simulated Data without Noise	85
4.4.2	Discussion of Testing Using Simulated Data with Artificial Noise	88
4.5	Testing the System with the Method I and the Model of KWW	93
4.5.1	Using Data Generated from N.B.S Data Tables	93
4.5.2	Using Data Generated from the Model of KWW	96
4.6	Testing the System with the Method I and the Model of DC	98
4.7	Testing the System with the Method I and the Model of HN	101
4.8	Testing the System with the Method I and the Model of FD	103
4.9	Testing the System with the Method II and the Model of KWW	106
4.9.1	Using Data Generated from N.B.S Data Tables	106
4.9.2	Using Data Generated from the Model of KWW	110
4.10	Testing the System with the Method II and the Model of DC	112
4.11	Testing the System with the Method II and the Model of HN	114

4.12	Testing the System with the Method II and the Model of FD .	117
4.13	Testing the System Using the Data with Some Artificial Noise	119
4.13.1	Using Data 1 with Some Artificial Noise	120
4.13.2	Using Data 5 with Some Artificial Noise	121
4.13.3	Using Data 9 with Some Artificial Noise	122
4.13.4	Using Data 14 with Some Artificial Noise and Method I	123
4.13.5	Using Data 19 with Some Artificial Noise and method II	124
5	Conclusion and Future Work	129
5.1	Conclusion	130
5.2	Future Work	132

List of Figures

1.1	The Sample Curve	26
3.1	The Curve of Gamma Function $\Gamma(x)$	58
4.1	The Fitting Curve for the Table 4.1	88
4.2	The Program Script for the Table 4.1	89
4.3	The Fitting Curve for the Table 4.61	120
4.4	The Fitting Curve for the Table 4.62	121
4.5	The Fitting Curve for the Table 4.63	122
4.6	The Fitting Curve for the Table 4.64	123
4.7	The Fitting Curve for the Table 4.65	124

List of Tables

1.1	Abbreviations and Acronyms	21
1.2	Symbol Table	22
3.1	Special Gamma Function Values	58
4.1	Computation for Data 1 with Method I	94
4.2	Computation for Data with Different Values of τ to Data 1 . .	94
4.3	Computation for Data with Different Values of ϵ_0 to Data 1 .	94
4.4	Computation for Data with Different Values of ϵ_∞ to Data 1 .	95
4.5	Computation for Data 5 with Method I	95
4.6	Computation for Data with Different Values of τ to Data 5 . .	95
4.7	Computation for Data with Different Values of ϵ_0 to Data 5 .	96
4.8	Computation for Data with Different Values of ϵ_∞ to Data 5 .	96
4.9	Computation for Data 9 with Method I	97
4.10	Computation for Data with Different Values of τ to Data 9 . .	97
4.11	Computation for Data with Different Values of ϵ_0 to Data 9 .	97
4.12	Computation for Data with Different Values of ϵ_∞ to Data 9 .	98
4.13	Computation for Data with Different Values of β to Data 9 . .	98

4.14	Computation for Data 14 with Method I	99
4.15	Computation for Data with Different Values of τ to Data 14 .	99
4.16	Computation for Data with Different Values of ϵ_0 to Data 14 .	100
4.17	Computation for Data with Different Values of ϵ_∞ to Data 14	100
4.18	Computation for Data with Different Values of γ to Data 14 .	100
4.19	Computation for Data 19 with Method I	101
4.20	Computation for Data with Different Values of τ to Data 19 .	102
4.21	Computation for Data with Different Values of ϵ_0 to Data 19 .	102
4.22	Computation for Data with Different Values of ϵ_∞ to Data 19	102
4.23	Computation for Data with Different Values of α to Data 19 .	103
4.24	Computation for Data with Different Values of γ to Data 19 .	103
4.25	Computation for Data 25 with Method I	104
4.26	Computation for Data with Different Values of τ to Data 25 .	104
4.27	Computation for Data with Different Values of ϵ_0 to Data 25 .	105
4.28	Computation for Data with Different Values of ϵ_∞ to Data 25	105
4.29	Computation for Data with Different Values of α to Data 25 .	105
4.30	Computation for Data with Different Values of τ' to Data 25 .	106
4.31	Computation for Data 1 with Method II	107
4.32	Computation for Data with Different Values of τ to Data 1 . .	107
4.33	Computation for Data with Different Values of ϵ_0 to Data 1 .	108
4.34	Computation for Data with Different Values of ϵ_∞ to Data 1 .	108
4.35	Computation for Data 5 with Method II	108
4.36	Computation for Data with Different Values of τ to Data 5 . .	109

4.37	Computation for Data with Different Values of ϵ_0 to Data 5	109
4.38	Computation for Data with Different Values of ϵ_∞ to Data 5	109
4.39	Computation for Data 9 with Method II	110
4.40	Computation for Data with Different Values of τ to Data 9	110
4.41	Computation for Data with Different Values of ϵ_0 to Data 9	111
4.42	Computation for Data with Different Values of ϵ_∞ to Data 9	111
4.43	Computation for Data with Different Values of β to Data 9	111
4.44	Computation for Data 14 with Method II	112
4.45	Computation for Data with Different Values of τ to Data 14	113
4.46	Computation for Data with Different Values of ϵ_0 to Data 14	113
4.47	Computation for Data with Different Values of ϵ_∞ to Data 14	113
4.48	Computation for Data with Different Values of γ to Data 14	114
4.49	Computation for Data 19 with Method II	115
4.50	Computation for Data with Different Values of τ to Data 19	115
4.51	Computation for Data with Different Values of ϵ_0 to Data 19	115
4.52	Computation for Data with Different Values of ϵ_∞ to Data 19	116
4.53	Computation for Data with Different Values of α to Data 19	116
4.54	Computation for Data with Different Values of γ to Data 19	116
4.55	Computation for Data 25 with Method II	117
4.56	Computation for Data with Different Values of τ to Data 25	118
4.57	Computation for Data with Different Values of ϵ_0 to Data 25	118
4.58	Computation for Data with Different Values of ϵ_∞ to Data 25	118
4.59	Computation for Data with Different Values of α to Data 25	119

4.60	Computation for Data with Different Values of τ' to Data 25	119
4.61	Computation for Data 1 with Some Artificial Noise and Method I	120
4.62	Computation for Data 5 with Some Artificial Noise and Method II	121
4.63	Computation for Data 9 with Some Artificial Noise and Method I	122
4.64	Computation for Data 14 with Some Artificial Noise and Method I	123
4.65	Computation for Data 19 with Some Artificial Noise and Method II	124
4.66	The Table Illustrating How Many Sets of Data Required	125
4.67	Comparison of <i>iterativetimes</i> between Method I and Method II	126
4.68	Comparison of <i>duration</i> [seconds] between Method I and II	127

Chapter 1

Introduction and Background

Information

This chapter introduces the basic knowledge necessary for reading this thesis, including a summary of the dielectric relaxation process, a list of symbols, and a description of important concepts and definitions. Following this a brief literature review is presented about the dielectric relaxation process and its data processing. The main part of this chapter is to discuss the issues of current methods about parameters calculation and data fitting, and the traditional way to do curve fitting. Finally, the chapter introduces a new method for data fitting and parameters calculation developed in the thesis.

1.1 Introduction

1.1.1 Dielectric Relaxation

Dielectric relaxation is the momentary delay in the dielectric constant of a material. This is usually caused by the delay in molecular polarization when a changing electric field is applied to a dielectric medium inside capacitors or between two large conducting surfaces. Dielectric is an adjective that does not conduct electricity or is a noun representing a dielectric substance usable for insulating things referred from the Oxford paperback dictionary [1]. The dielectric relaxation process is widely used in materials science, chemistry, physics, electrical engineering, mechanical science and health science. To study the micro-structure of the tested material, which may include solids, liquids or some other insulator such as patient's tissue, many scientists use a dielectric relaxation process to get some data that reflect the micro-structure of the tested material by conducting experiment measure with some equipment such as radio frequency (RF) Impedance analyzer. The scientists may do some analysis on the data to get some characteristic parameters while the material's property is characterized by these parameters so that they can compare several different materials, same kind of materials with different sample preparation procedure. It is interesting and important how to analyze the dielectric relaxation data by doing the data fitting and calculating the parameters corresponding to certain model. An efficient algorithm or program for the data fitting or the calculating of parameters is very helpful for the research

in the fields above. Before we discuss that, let us briefly discuss some terms of dielectric relaxation borrowed from materials science.

The measurement or the experiment procedure may vary from different scientists to different equipments. However, there are only two kinds of data returned by experiments: time domain and frequency domain. For example, the relaxation consists in the recovery of strain on removal of stress, which implies therefore a time dependence, typically for the case by suddenly applying and then after a while by suddenly removing a steady stress. This is the basis for time domain, which leads to data in time domain. Here is another example in dielectric relaxation. Relaxation also consists in the recovery of the shape of molecules of an insulator, which a sinusoidal electric field is applied to. This example implies therefore a frequency dependence. This is the basis for frequency domain leading to the data in frequency domain. The relationship between the data in time domain and frequency domain will be discussed in Chapter Two. The work in this thesis only focuses in the data in frequency domain.

Now let us briefly introduce some characteristic parameters of the data in frequency domain borrowed from materials science. They include the following:

- The fractional stretched exponential parameter for time-invariant
- The relaxation time of a time-dependent property
- The limited low frequency permittivity or 'instantaneous' susceptibility

- The limited high frequency permittivity or static susceptibility
- The isothermal susceptibility, which is difference between static susceptibility and instantaneous susceptibility.

where one model may have more fractional stretched exponential parameters while another model may have more parameters for relaxation time. Different models may have different characteristic parameters. Because the physical meaning of these terms does not affect the system, we do not explain them in detail here.

1.1.2 Symbols, Concepts and Definitions

Abbreviations and Acronyms

DC	The model discussed by Cole Davidson [2]
HN	The model discussed by Havriliak-Negami [2]
KWW	The model discussed by Kohlrausch-Williams-Watts [2]
FD	The model obtained from the theory of fractional dynamics [2]
fre	The list of frequencies as the independent variables from the input
TD	The time domain measurement
FrD	The frequency domain measurement

Table 1.1: Abbreviations and Acronyms

Table of Symbols

Her is the table for some symbols used in this thesis at Table 1.2.

ω	Circular frequency of the alternated voltage applied to the tested material
$\chi^*(\omega)$	Dynamic susceptibility normalized by corresponding isothermal susceptibility
χ_o	Static susceptibility
χ_∞	Instantaneous response from the material
$\varepsilon^*(\omega)$	Complex frequency-dependent dielectric susceptibility
ε_o	Static susceptibility for dielectric relaxation
ε_∞	Instantaneous response from the material for dielectric relaxation
β	Single stretching exponent for the model of KWW, $0 < \beta < 1$
τ_β	Relaxation time for the model of KWW
τ_γ	Relaxation time for the model of DC
α	One single stretching exponent for the model of HN or FD, $0 < \alpha < 1$
τ_H	Relaxation time for the model of HN
γ	One single stretching exponent for the models of DC or HN, $0 < \gamma < 1$
τ_α	Relaxation time for the model of FD
τ'_α	Second relaxation time for the model of FD
μ	$-i * \omega, i^2 = -1$
χ'	The real part of the input data
χ''	The imaginary part of the input data
ε'	The real part of the input data for dielectric relaxation
ε''	The imaginary part of the input data for dielectric relaxation

Table 1.2: Symbol Table

Concepts, Definitions and Goal

This section introduces some concepts and definitions used in this thesis.

- The input (experimental) data is a matrix having three columns. In this thesis, we only discuss the input data for dielectric relaxation. The first column is a list of frequencies, which are independent variables of dielectric relaxation data. The second column is a list of real part of dielectric relaxation data, which are obtained through the measurement. The third column is a list of imaginary part of dielectric relaxation data, which are also obtained through the measurement.
- The simulated (generated) data is a matrix having three columns. It is a special input data with an extra header. The extra header is the first two rows of the matrix, where the information about the parameter's values and the model that is used to generate the data is stored.
- The data fitting in this work: The data fitting includes two parts. One part is to do the data fitting for the real part of the data, i.e. the second column of the input data, with the independent variables, which are frequencies. The other part is to do the data fitting for the imaginary part of the data, i.e. the third column of the input data, with the independent variables, which are frequencies. The data fittings for two parts are carried out simultaneously in this work rather than separately.
- The least-square estimation of nonlinear parameters for a model: The values of nonlinear parameters for a theoretical model are estimated in

the least-square manner so that the value returned by the theoretical model is the closest to the experimental data with respect to the same independent variable, i.e. the sum of squares of differences between experimental data and their approximation is the least.

- The fractional stretched exponential parameter for time-invariant: is one of parameters for a model. There might be only one fractional stretched exponential factor for one model or two fractional stretched exponential factors for the other model. However, its function is same for all models, which is to justify the data returned by a model. Simply speaking, its function is just to modify the shape of curves including the curve for the real part of the data and the curve for the imaginary part of the data.
- The relaxation time of a time-dependent property: is also one of parameters for a model. In most cases, there is one relaxation time for a model. Sometimes, there are two relaxation times for a model such as FN model. This parameter plays very important role in a model. This parameter affects the shape of the curve and the location of the peak for the curve of the imaginary data.
- The limited high frequency permittivity or 'instantaneous' susceptibility, ε_∞ , is the minimum value of the real part of the data if there existed an equipment without limitation. It is usually unknown due to the limitation of the equipment.
- The limited low frequency permittivity or static susceptibility, ε_0 , is the

maximum value of the real part of the data if there existed an equipment without limitation. It is also usually unknown due to the limitation of the equipment.

- The curve: The curve we discuss here includes two parts. One part is the curve of the real part of the data plotted as a function of frequency. The first column of data, frequency, as the abscissa and the real part of the data, the second column of data, as the ordinate. This curve should have many data points having the value of ε_o , and then monotonously decrease, finally reach the value of ε_∞ . Due to the limitation of equipment, a limited number of data points and the limited frequency range of measurement, this curve most likely only has the part that decreases monotonously. The other part of the curve is the curve of the imaginary part of the data plotted as a function of frequency. The first column of data, frequency, as the abscissa and the imaginary part of the data, the third column of data, as the ordinate. This curve should look like normal distribution curve but usually the curve is not symmetrical. Due to the same reason as the curve of the real part, the curve for some tested material may not include the peak but only have monotonously-increased part or only have monotonously-decreased part. Here is an example of the curve Figure 1.1.

- The goal: Since the dielectric relaxation process is widely used in materials science, chemistry, physics, electrical, mechanical science and even

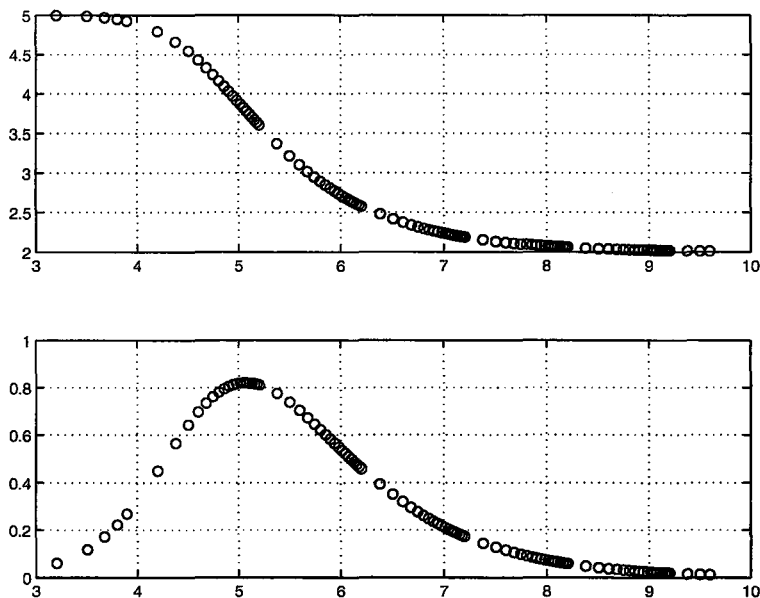


Figure 1.1: The Sample Curve

health science. It is interesting and important how to analyze the dielectric relaxation data by doing the data fitting and calculating the parameters corresponding to certain model. An efficient algorithm or program for data fitting or calculation of parameters is very helpful for research in the fields above. The goal of this work is to develop a tool to eliminate the limitation of testing equipment such as the limited frequency range of measurement as much as possible by doing the data fitting to show the other part of curve that cannot be acquired by doing the measurement and return the characteristic parameters for the tested material, which are very helpful for researchers to compare different materials. This tool will help researchers to save time to handle

their experimental data because the current methods for data fitting and parameters acquiring cost a great amount of time, which you can see in the following section of literature review.

1.2 Literature Review

There are many papers discussing how to do data fitting of dielectric relaxation.

D. L. Misell and R. J. Sheppard discussed the method using de-convolution techniques to dielectric data. Although the de-convolution procedures can be used to analyze the dielectric data by deriving a distribution function representing the molecular process, it was not discussed how to fit the data and return the parameter's values because the method even did not require a model for the number of dispersions [3]. It is not a method to get the parameter's values and see the fitting result.

C. T. Moynihan, L. P. Boesch and N. L. Laberge presented a method using an approximate expression of the Fourier transform for the decay function. The criteria used for the fit is that the calculated curve should match the targeted curve in maximum height, and they should also coincide at the two half height's positions. The method had to consider the width of the imaginary curve at half-height and the half height position on the low frequency. However, the method can only handle a special input data, which have to contain a peak in the imaginary part [4].

Paul K. Dixon, Lei Wu, and Sidney R. Nagel found that the primary relaxation for all the liquids, over the entire frequency and temperature range, can be collapsed into a single scaling curve [7]. Some forms were introduced to scale the dielectric data. However, the scaling required some parameter's

values, which usually are unknown, and also required that the imaginary part of the input data has to contain a peak.

S. Gabriel, R. W. Lau and C. Gabriel programmed the model in terms of multiple Cole-Cole dispersion into a Microsoft Excel spreadsheet (The Cole-Cole dispersion is also a model to describe dielectric relaxation. However, we will not do the fitting for this model simply because it is not so popular). A systematic procedure was followed in which the main parameters of the model were fitted. It was suitable when the data to be fitted span several orders of magnitude, creating a bias towards fitting the low frequency data and rendering the fit insensitive to the high frequency parameters. However, the whole fitting process was visual and requires an understanding of the Cole-Cole function and appreciation of the correlation between parameters [8].

E. Tuncer and S. M. Gubanski used the Monte Carlo Method to obtain relaxation time distribution and compare non-linear spectral function fits. They used a pre-distribution of relaxation times, reconstructed the original data by an ideal model called single Debye relaxation to describe dielectric relaxation and used a box constraint, least squares algorithm. However, single Debye relaxation is not a practical but an ideal model for dielectric relaxation [9].

In the paper of dielectric spectroscopy data treatment: I. Frequency domain, N Axelrod et al. introduced a method based on a penalized maximum likelihood approach. They also suggested using the fast Fourier transforms and a suitable interpolation technique. They used much mathematical techniques.

For example, they had to solve the mathematically ill-posed problem. To get smooth parameter estimation, they need to introduce some penalty term to the cost function in the parameter minimization problem, together with a constraint condition. They just transfer dielectric spectroscopy data analysis problem to the continuous optimization problem by setting up an objective function, introducing penalty function and some constraints [10].

D. Wasylyshyn introduced a way to manually calculate a set of parameters for KWW model. Specifically, Wasylyshyn normalized the dielectric relaxation data by the factor of $\omega\tau_0$ such that the peak of imaginary part of the curve, seen at the second part of Figure 1.1, corresponded with the peak of the tabulated values described in the data tables in the paper [13]. The height of peak was normalized by $N'' = \epsilon''/(\epsilon_o - \epsilon_\infty)$ until the N'' peak height matched the peak height of the tabulated data. Then a suitable value of β was selected so that the N'' peak matched the peak of the tabulated values. Finally after values of τ_0 , β and $\delta\epsilon$ were known, the value of ϵ_∞ was adjusted to let the curve of N' match the tabulated values. It is obvious that there is much work by following the method to get the values of a set of parameters for KWW model [12]. And also it requires the experimental data including a peak too.

Now let us introduce our method, which is easy to use with some background of dielectric relaxation. It does the least square fitting and parameters calculation together and shows the fitting result by plotting the curve.

1.3 Introduction of Our Method

Our method is an iterative method. The system first reads the data from the user and asks the user to choose the model. The system lets the user choose the part of the data that is required to do the best fitting. The user provides the initial values for a set of parameters. Then the system will automatically return a set of parameters and the fitting error. The developed system also provides the visual fitting result of image and the file of the fitting result. The system provides two different ways to fit the data. The first way uses the algorithm introduced in [15]. The second way uses a Matlab built in function as a minimizer by constructing an objective function. Since the system returns a set of parameters and the fitting result is unlikely unique [4], the initial values for a set parameters, the frequency range of data and the selected data are critical to get the best fit. The user may select a suitable way to do the data fitting based on the factors above and the chosen model. It will be discussed in detail how to choose a suitable way in Chapter Three.

Compared with those methods briefly introduced in the section of literature review, from a user's point of view, our method does not need to worry about whether the input data have the peak or not. It does not need to get the position of half-height of the curve. It does not need to normalize the data. There is not any manual computation before doing data fitting. The user does not need to reconstruct data. For example, the user does not need to build an objective function or penalty function. The user does not need to

scale the data. The system is able to handle very large range of data and cover data having several decade's range. And the user does not need to learn many mathematical skills to fit data but some background of dielectric relaxation property to choose the initial values for the unknown parameters.

1.4 Thesis Structure

Chapter 2 discusses the theorems including dielectric relaxation, mechanical relaxation, and their models. It also discusses the fitting techniques and least squares method. Chapter 3 will present our numerical software and system design. It emphasizes the applied algorithm, system design and its design principle, data generation and the applied techniques. Chapter 4 discusses some experiments and testing for the system including two different ways of generating the simulated data, testing four subsystems based on different models and the experimental result. Finally, we talk about the conclusion and the future work for the system, such as how to do the more complicated data fitting and the condition for choosing the initial values.

Chapter 2

Theorems

This chapter presents some theorems regarding dielectric relaxation, mechanical relaxation and their mathematical models, KWW, DC, HN and FD. Since the work done here is about data fitting and estimation of nonlinear parameters, the data fitting techniques, linear least square and nonlinear least square methods will also be discussed in this chapter.

2.1 Theories of Relaxation

Relaxation is a property that reflects the recovery of strain after the applied stress is removed. There are two ways to do the measurement. If a steady stress is applied to the tested material suddenly and then is removed suddenly after a while, this is the time domain measurement. The other way, which is used more often, is called the frequency domain measurement. It is a measurement by applying to the tested material a harmonically varying stress of an angular frequency ω such as a sinusoidal electric field rather than a steady stress. However, we will not discuss how to do the measurement. We assume the data are ready to use when using this tool. There is a relation between the tested material's response under the time domain(TD) measurement and the tested materials's response under the frequency domain(FrD) measurement, i.e. TD responses $f(t)$ and FrD responses $\chi'(\omega), \chi''(\omega)$ are Fourier transforms of one another seen in the following mathematical equations:

$$\begin{aligned}\chi'(\omega) &= \int_0^{\infty} f(t)\cos(\omega t)dt \\ \chi''(\omega) &= \int_0^{\infty} f(t)\sin(\omega t)dt \\ f(t) &= (2/\pi) \int_0^{\infty} \chi'(\omega)\cos(\omega t)d\omega \\ f(t) &= (2/\pi) \int_0^{\infty} \chi''(\omega)\sin(\omega t)d\omega\end{aligned}$$

The subject of relaxation covers all types of stress relief including dielectric, mechanical, photo-conductive, chemical stress and so on [18]. We only talk about dielectric relaxation and mechanical relaxation.

2.1.1 Dielectric Relaxation

The science of dielectrics is multidisciplinary. To fully understand it, knowledge of physics, chemistry, materials science and electrical engineering is required. Different researchers may be interested in different part of dielectric relaxation. For example, some engineers may be interested in the dielectric loss and electrical strength. Some engineers may be interested in the movement of charge. Chemists may use dielectric relaxation as a handle on molecular dynamics by doing TD measurement or FrD measurement [18].

There are five different principal dielectric functions currently presenting dielectric information.

- The complex permittivity.
- The dielectric modulus.
- The complex capacitance.
- The admittance of the sample.
- The impedance of the sample.

There are two most common ways to present the data.

- Plots of the real and imaginary parts either in logarithmic or in linear coordinates as a function of frequency on a logarithmic presentation.
- Plots of the imaginary part against the real component.

As mentioned in Chapter One, the complex permittivity and plots of the real and imaginary parts in linear coordinates as a function of frequency on a logarithmic presentation are selected in this thesis.

One of measurements to obtain the data is to place the sample between two capacitor plates. A electric field is applied to the two capacitor plates [12]. Then the tested sample will be polarized and cause instantaneous and time dependent charge build up on the capacitor plates. We can obtain the data by measuring the the build up of surface charge. If the electric field is not constant, the Boltzmann superposition principle is applied. After a long series of formula reduction with some new definition by treating the superposition principle in the dynamic manner, we can end up with the following equation:

$$N_{\epsilon}^* = \frac{(\epsilon^* - \epsilon_{\infty})}{(\epsilon_s - \epsilon_{\infty})} = \mathcal{L}\left(-\frac{\partial\Phi_D(t)}{\partial t}\right) \quad (2.1)$$

where $\mathcal{L}(x)$ is the Laplace transform of x , and N_{ϵ}^* is the normalized complex dielectric permittivity.

2.1.2 Mechanical Relaxation

The reason we discuss mechanical relaxation here is because it is analogous to the dielectric case. For example, there are very similar principal functions currently presenting mechanical information: the complex mechanical compliance and the shear modulus, with respective to the complex permittivity and the dielectric modulus. There are also two most common ways to present the data just like what the dielectric data. In this thesis, however, we only focus

on the dielectric relaxation.

Moreover, we also can end up with very similar equation by treating the superposition principle in the dynamic manner:

$$N_J^* = \frac{(J^* - J_U)}{(J_R - J_U)} = \mathcal{L}\left(-\frac{\partial\Phi_M(t)}{\partial t}\right) \quad (2.2)$$

where $\mathcal{L}(x)$ is the Laplace transform of x , J_U and J_R are analogue to ε_o and ε_∞ , and N_J^* is the normalized complex mechanical compliance.

After we briefly introduced the mechanical relaxation, now we introduce the principal types of dielectric functions or models, which also apply to mechanical relaxation, in the next section.

2.2 Introduction of Four Models

The left-handed side of all of the coming equations is:

$$\hat{\chi}(\mu) = \frac{(\varepsilon^* - \varepsilon_\infty)}{(\varepsilon_o - \varepsilon_\infty)} \quad (2.3)$$

2.2.1 Model of KWW, High Frequency Part, when |

$$\mu\tau_\beta \gg 0$$

Introduced By R.Hilfer in [2]

$$\hat{\chi}(\mu) = 1 - H_\beta(-(\mu\tau_\beta)^\beta); \quad 0 < \beta \leq 1 \quad (2.4)$$

where $H_\beta(x)$ is defined by the series

$$H_\beta(x) = \sum_{k=0}^{\infty} \frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)} x^{-k} \quad (2.5)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

2.2.2 Model of KWW, Low Frequency Part, when $|\mu\tau_\beta| \approx 0$

$$|\mu\tau_\beta| \approx 0$$

Introduced By R.Hilfer in [20]

$$\hat{\chi}(\mu) = 1 + H_\beta(-(\mu\tau_\beta)) \quad (2.6)$$

where $H_\beta(x)$ is defined by the series

$$H_\beta(x) = \sum_{k=0}^{\infty} \frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)} x^{k+1}; \quad 0 < \beta \leq 1 \quad (2.7)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

2.2.3 Model of DC

Introduced By R.Hilfer in [2]

$$\hat{\chi}(\mu) = \frac{1}{(1 + \mu\tau_\gamma)^\gamma}; \quad 0 < \gamma \leq 1 \quad (2.8)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

2.2.4 Model of HN

Introduced By R.Hilfer In [2]

$$\hat{\chi}(\mu) = \frac{1}{(1 + (\mu\tau_H)^\alpha)^\gamma}; \quad 0 < \gamma \leq 1; \quad 0 < \alpha \leq 1 \quad (2.9)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

2.2.5 Model of FD

Introduced By R.Hilfer in [2]

$$\hat{\chi}(\mu) = \frac{1 + (\tau_\alpha \mu)^\alpha}{1 + (\mu \tau_\alpha)^\alpha + \mu \tau_\alpha^\alpha}; \quad 0 < \alpha \leq 1 \quad (2.10)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

2.3 Least Squares Data Fitting

The used data fitting techniques are actually a technique in least square manner to minimize simulated data and the experimental data. The work done here first meant implementing four different models in Matlab so that those implemented models are able to return accurate data with respect to the provided parameters. Then we constructed an objective function by combining each of four models with the experimental data in least squares manner. Finally we used the nonlinear least square algorithm introduced in [15] to return approximated estimation of parameters based on the experimental data. We also used the Matlab build-in function `fminsearch()` as a minimizer to call the objective function. These two ways are both able to return approximated estimation of parameters.

Chapter 3

Numerical Software and System Design

This chapter presents numerical software design of the developed software. It will cover the algorithms used in the system, software requirements, system architecture, design principle, data generation and finally numerical techniques applied to the software to improve its accuracy, efficiency and robustness. It is the main part of the thesis.

3.1 Algorithms

3.1.1 Background

Usually in data fitting, a theoretical model is existing and the analyst is attempting to determine either how well the input data match the theoretical model in order to test how good the theoretical model is or to get the values for the parameters of the models such that the results produced by the theoretical model fit the input data. In our case, we need to determine the values of the parameters of the theoretical models. We know that four different models are in the complex number field and the values for the parameters of the theoretical models are in the real number field. If we work on the complex number field, the results returned by the system are most likely also complex numbers which are not what we expect. Some scientists [2] do the fitting on the real part and the imaginary part separately. This method returns two sets of values for the parameters of the model for the real part of data and the imaginary part of data respectively and may not fit two parts of data with the same set of values for the parameters of the model at the same time. Our method takes the consideration of both the real part and the imaginary part and fits the two parts of data at the same time. Since the input of the system is a set of data and the number of unknown parameters are 4 or 5 depending on different models, the problem of finding the values for the parameters of the model is over-determined. So we consider using the most commonly used algorithm for the least square regression.

3.1.2 Algorithm

We know that four available models are nonlinear models introduced in the last chapter. We use nonlinear least squares approximation. The main idea of the least squares algorithm is that we are seeking some values for the parameters of the chosen model so that the chosen model will return values as close as possible to the experimental data with respect to the same independent variables.

Suppose we have the experimental data consist of N triples as the following:

$$Data = (x_0, y_0, z_0), (x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_{N-1}, y_{N-1}, z_{N-1})$$

where x_i , y_i and z_i are representing frequency, ϵ' and ϵ'' . For the meaning of ϵ' and ϵ'' , see the Table 1.2.

To evaluate how close the simulated data returned by the chosen model under certain values of the parameters are to the experimental data, the residual for each data point is defined as follows: Suppose f is the chosen model, a and b are values returned by the chosen model f , which are respectively real part and imaginary part of simulated data point corresponding to the experimental data point (x_k, y_k, z_k) and i is the symbol of imaginary unit of complex number, then we have:

$$a + bi = f(x_k)$$

$$Realresidual_k = | y_k - a |$$

$$Imagresidual_k = | z_k - b |$$

As we mentioned in section 3.1.1, our method takes the consideration of both the real part and the imaginary part and fits the two parts of data at the same time. We defined the sum of the squares of the residuals as the following:

$$sumofsquare = \sum_{k=0}^{N-1} (Realresidual_k^2 + Imagresidual_k^2) \quad (3.1)$$

Then the system will minimize the value of *sumofsquare* using the least squares technique or some available minimizers. When the system minimizes the value of *sumofsquare*, the system returns the approximate values for the parameters of the chosen model which the system does the best to return.

3.1.3 Method I and Method II

The developed system uses two different methods to obtain the estimation of unknown parameters. One method is a least squares algorithm, the other is a method using the Matlab built-in function `fminsearch()`. The first method runs faster than the second method while the second method is more robust than the first one, which is seen in the next chapter of testing. The comparison between these two methods is seen at the Table 4.67 and the Table 4.68 in Chapter 4.

Algorithm for Least Squares Estimation of Nonlinear Parameters

This algorithm is introduced by Donald W. Marquardt [15]. This algorithm gets the least squares estimation of nonlinear parameters by calculating the derivatives with respect to some unknown parameters. These derivatives are used to direct the converging direction. This algorithm uses the Taylor series

through the linear terms to expand the chosen model. This algorithm performs an optimum interpolation between the Taylor series method and steepest descent method. So each iterative step the algorithm calculates the derivatives to make sure they are less than zero. The iteration will stop when the derivatives are zero. Because we know in order to minimize *sumofsquare* (3.1), the derivatives with respect to all of unknown parameters in the model must be zero. It is well known that the gradient method is not a method with a scale invariant [14]. This algorithm uses the scaling technique to improve the numerical aspects of computing procedure. The developed system utilizes the finite differentiation method to estimate the derivatives at each data point for all of unknown parameters. Since it is an iterative method, the initial values for all of unknown parameters are required. For more detail, see [15].

Method Calling the Matlab built-in Function `Fminsearch()`

This Matlab built-in function `fminsearch()` is a method of multidimensional unconstrained nonlinear minimization (Nelder-Mead). It is available in Matlab 7.0. It is used to minimize the *sumofsquare* of the difference between simulated data and the experimental data. For more detail of Nelder-Mead algorithm, see [22].

3.2 System Design

This section discusses the scope of the software product, software requirements and system architecture.

3.2.1 Scope of the Software Product

The main usage of the software product is to discover the dielectric relaxation properties of a tested material while a material properties are characterized by the parameters. The developed software does the least squares fitting for the available experimental data of the tested material and calculates the unknown parameters for the tested material. This software could be a tool to predict the behavior of the tested material by getting the material properties from the experimental data.

3.2.2 Software Requirements

Here are several software requirements for the developed system:

- The user provides the experimental data to the system.
- The experimental data consist of three parts, the independent variable is frequency or logarithm of frequency only, while the dependent variables are real components and imaginary components of dielectric relaxation property.
- The system allows the user to decide which part data is the most important or the most accurate and then select the required data by providing

the lower bound and upper bound of the frequency and visible curve.

- The system allows a user to choose a model.
- The system also asks the user to provide the initial values for the set of unknown parameters by observing the visible curve. In order to input the initial values easily for the user, the system sets the unit of relaxation time τ as microsecond [msec], 10^{-6} second.
- After doing the calculation, the system returns the output including values for the unknown five or four parameters of the tested material and a report.
- The report records the following information: the chosen model, the values of parameters, the values of terminated criteria— the fitting error, and the simulated data with the corresponding experimental data.
- The system terminates if the system reaches the maximum iterative times.
- Finally the system draws the fitting curve to show how well the visualization of fitting is.
- The system is implemented using Matlab.

3.2.3 System Architecture

Based on the requirements in the last section, we designed the following modules: main module to start the system, model selection module, data storage module, data preprocessing module, data selection module, data processing module, curve drawing module and output module. Our design principle is behavior hiding. The concept of behavior hiding fetches from Mesh generator modules guide introduced by Dr. Smith in 2006 [16]. Here are the detailed information including module name, service, secret, expected changes and the prefix for all modules.

Main Module

Module name	Main module
Module service	Main control, starting the whole system
Module secret	The way how to call modules
Expected changes	Which modules should be called
Prefix	Dr

Model Selection, Data Reading and Preprocessing

1. Model selection module:

Module name	Model selection module
Module service	Let the user select the suitable model
Module secret	The way how to select a model
Expected changes	The number of models available
Prefix	Msel

2. Data storage module:

Module name	Data storage module
Module service	Store data, change formalism of data get the lower and upper bound of selected data
Module secret	The way how to provide services
Expected changes	How to implement these services using different languages
Prefix	Dat

3. Data preprocessing module:

Module name	Data preprocessing module
Module service	Transfer the input data to the data accepted by the model, i.e. the independent variable is frequency only
Module secret	The way how to transfer the data to ones accepted by the system
Expected changes	The way how to transfer the data
Prefix	Dpre

4. Data selection module:

Module name	Data selection module
Module service	Select part of the data
Module secret	How data is selected from the sequence of data
Expected changes	How many parts of the data can be selected
Prefix	Dsel

Data Processing

Module name	Data processing module
Module service	Handle data to calculate the output parameters
Module secret	The algorithm how to do it
Expected changes	The algorithm
Prefix	Dpm

Report System and Fitting Curve Drawing

1. Curve drawing module:

Module name	Curve drawing module
Module service	Draw the fitting curve
Module secret	The method how to draw a curve
Expected changes	Which curve drawing functions will be called
Prefix	Cdra

2. Output module:

Module name	Output module
Module service	Write results to output file
Module secret	The method how to do it
Expected changes	More models available, right now only KWW, DC, HN and FD
Prefix	Out

3.3 Data Generation and Checking

Apart from the developed system, we also developed another system to generate data to test the program. The system has two modules as follows:

1. Data generation module:

Module name	Data generation module
Module service	Generate the data based on the selected model and values of parameters
Module secret	The way how to generate the simulated data
Expected changes	Available methods to generate the simulated data
Prefix	Gen

2. Data checking module:

Module name	Data checking module
Module service	Return values of parameters of the simulated data
Module secret	How to return the values
Expected changes	How to store values for those parameters
Prefix	Che

The system first reads the input file from the current directory. The input file is obtained from Journal of Research of the National Bureau of Standards [13]. Each input file is corresponding to a particular value for β . The system generates the simulated data with respect to the values of parameters τ_β , ϵ_0 , ϵ_∞ , a list of frequencies provided by the user and the particular value for β according to the input file. The explanation of the parameters is

seen in Table 1.2. The simulated data generated from the Journal of Research of the National Bureau of Standards are based on the model of KWW and will be the simulated data to test the program since we know the result returned by the tested system should be. There are the other four different ways to generate the data by using four available models introduced in Chapter Two and providing the values for parameters to the model.

This system records the values of parameters, and which the chosen model is used to generate the data, at the first two rows of data for the generated data.

The system also provides the way to see which model is used and the values of the parameters the model uses to generate the data by displaying the first two rows of data.

For example, if we choose KWW model to generate data, suppose KWW is represented as -1, the values of parameters β , τ_β , ϵ_0 , and ϵ_∞ and a list of frequencies are 0.5, 1[msec], 5 and 2 respectively, and the list of frequencies is the vector, which the values of elements are monotonously increased from $1.591549430918953 \times 10^3$ to $6.366197723675814 \times 10^4$ and usually available from the experimental data, then the first two rows of the generated data are the following:

```
5.000000000000000e-01  1.000000000000000e+00  0.000000000000000e+00
5.000000000000000e+00  2.000000000000000e+00  -1.000000000000000e+00
```

And the generated data are the following:

Frequencies	Real part, ϵ'	Imaginary part ϵ''
5.000000000000000e-01	1.000000000000000e+00	0.000000000000000e+00
5.000000000000000e+00	2.000000000000000e+00	-1.000000000000000e+00
1.591549430918953e+03	4.996450400000000e+00	5.964000000000001e-02
3.183098861837907e+03	4.986318519173119e+00	1.173438824448000e-01
4.774648292756860e+03	4.970754836777365e+00	1.718324729076224e-01
6.366197723675814e+03	4.951095272939271e+00	2.224458886823872e-01
7.957747154594767e+03	4.928465813386232e+00	2.690537564749919e-01
1.591549430918953e+04	4.795921987532930e+00	4.490299797170239e-01
2.387324146378430e+04	4.661437259759429e+00	5.653195462743827e-01
3.183098861837907e+04	4.538418922469724e+00	6.430251652619502e-01
3.978873577297384e+04	4.428576445242227e+00	6.965981701657942e-01
4.774648292756860e+04	4.330826245416617e+00	7.343737895185591e-01
5.570423008216338e+04	4.243555245822341e+00	7.613874532696657e-01
6.366197723675814e+04	4.165222885950977e+00	7.808252886739697e-01

From the generated data above, we can get the values of parameters and the chosen model because the first two rows record the values of parameters, 0.5, 1[msec], 5, 2 for β , τ_β , ϵ_0 , ϵ_∞ and the chosen model since -1 represents KWW model (The slot of the first row and the third column is reserved for the theoretical model having five parameters). The system just simply displays the parameter's values and the chosen model by getting the information from

the first two rows. The remain part, which is from the third row to the last row, is the generated data where the first column is a list of frequencies, the second column is a list of ϵ' and the third column is a list of ϵ'' (for the meaning of symbols, see the table 1.2). If we draw a curve for the data, we will get the half part of curve in the Figure 1.1.

3.4 Accuracy, Efficiency and Robustness

There are four available models, DC, HN, FD and KWW models discussed in chapter Two. DC, HN, and FD models are all in the form of one term. It is not too difficult to implement them in Matlab. However, KWW model is a sum of series seen in the next page. Because it is impossible to require infinite terms, a care need to be taken to implement this important model. Before we discuss that, let us review this important model.

Here is the model of KWW introduced in chapter Two. It includes two parts:

- High frequency part when $|\mu\tau_\beta| \gg 0$:

$$\hat{\chi}(\mu) = 1 - H_\beta(-(\mu\tau_\beta)^\beta); \quad 0 < \beta \leq 1$$

where $H_\beta(x)$ is defined by the series

$$H_\beta(x) = \sum_{k=0}^{\infty} \frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)} x^{-k} \quad (2.5)$$

- Low frequency part when $|\mu\tau_\beta| \approx 0$:

$$\hat{\chi}(\mu) = 1 + H_\beta(-(\mu\tau_\beta))$$

where $H_\beta(x)$ is defined by the series

$$H_\beta(x) = \sum_{k=0}^{\infty} \frac{\Gamma((k + 1)/\beta)}{\beta\Gamma(k + 1)} x^{k+1}; \quad 0 < \beta \leq 1 \quad (2.7)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

From this model, we know that it includes two parts, low frequency part and high frequency part. When $|\mu\tau_\beta| \approx 0$, the system should use low

frequency part to do the calculation while the system should use high frequency part to do the calculation when $|\mu\tau_\beta| \gg 0$. Unfortunately, the condition of $|\mu\tau_\beta| \approx 0$ is ambiguous. It does not tell us what the threshold is exactly since the threshold is the place or point of beginning according to American Heritage Dictionary [17]. After we did many experiments, we know that, with a very small frequency value closed to zero, the model of KWW, low frequency part, the equation (2.5), works and reflects the dielectric relaxation property correctly. But when the value of frequency increases and exceeds the threshold, the model of KWW, low frequency part, the equation (2.5) diverges, returns very large value and does not reflect dielectric relaxation property any more. Similarly, with very large frequency value, the model of KWW, high frequency part, the equation (2.7) works and reflects the dielectric relaxation property correctly. When the value of frequency decreases and exceeds threshold, the model of KWW, high frequency part, the equation (2.5) diverges, returns very large value and does not reflect dielectric relaxation property any more. It is also turned out by the experiments that the threshold in term of frequency for the model of KWW, high frequency part is less than that for the model of KWW, low frequency part, which means there is overlap of frequency range or a common frequency range where both parts work and reflect the dielectric relaxation property correctly and simultaneously. So we need to take care of not only how to implement the model of KWW as a sum of series, but also how to switch two parts of the model precisely and efficiently. Before we do that, let us discuss one of components in KWW model, gamma function.

3.4.1 Introduction of Gamma Function

The definition of gamma function for the argument in real number is the following:

$$\begin{aligned}
 \Gamma(x) &= \int_0^{\infty} t^{x-1} e^{-t} dt \\
 &= (x-1) \int_0^{\infty} t^{x-2} e^{-t} dt \\
 &= (x-1)\Gamma(x-1)
 \end{aligned} \tag{3.2}$$

However, for the equation (3.2), x cannot be equal to one.

If x is an integer $n = 1, 2, 3, \dots$, then

$$\Gamma(n) = (n-1)! \tag{3.3}$$

When x changes from 0 to 1.4616, the gamma function monotonously decreases from Inf to the minimum value of 0.8856. Then the gamma function becomes monotonously increased and the slope of the curve becomes greater and greater. Here is the Table 3.1 with some particular values returned by gamma function with respective to some particular values of x . Here is the Figure 3.1 for Gamma function $\Gamma(x)$.

3.4.2 Scaling and Gamma Function Decomposition

From the previous subsection, we know when x is greater than or equal to 172, $\Gamma(x)$ will overflow in double precision. Since we use Matlab to implement the system, $\Gamma(k+1)$ overflows when k is greater than or equal to 171. From

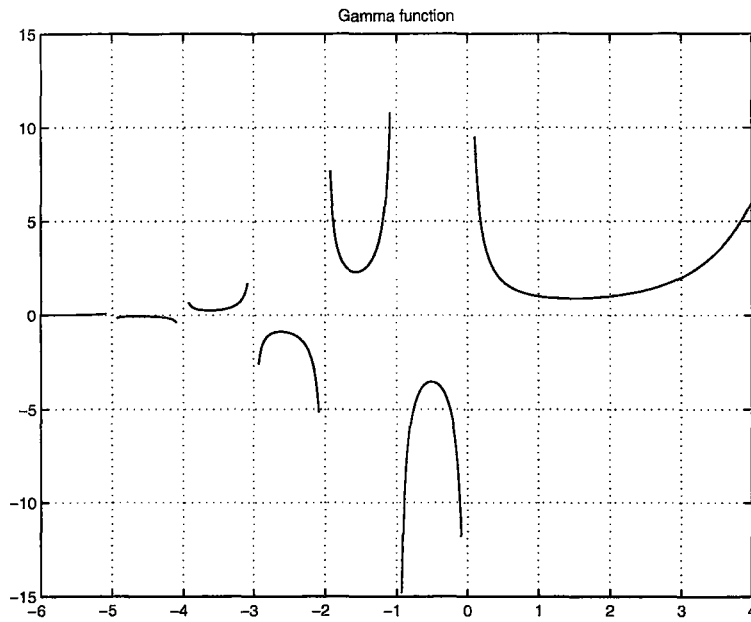


Figure 3.1: The Curve of Gamma Function $\Gamma(x)$

x	$\Gamma(x)$
0	<i>Inf</i>
1	1
1.4616	0.8856
2	1
3	2
171	7.2574×10^{306}
172	overflow in double precision

Table 3.1: Special Gamma Function Values

the equation (2.5) and the equation (2.7) we know $\Gamma(k + 1)$ is part of the divider. So the value of each term in the equation (2.5) and the equation (2.7) will become zero when k is greater than or equal to 171 if we calculate each term by calling the Matlab built-in function $\Gamma(x)$ directly. After we did many experiments, it turned out that when frequency becomes smaller and smaller the equation (2.5) converges more and more slowly. And the required value for k to let the (2.5) return accurate values will exceed 171 when frequency is small enough. So we should not implement equation (2.5) by calling the Matlab built-in function $\Gamma(x)$ directly. To let the value of k exceed 171 and also avoid the large intermediate value, we apply scaling technique and decompose $\Gamma(x)$ function to implement the model of KWW. Before we explore the relationship between two parts of the model of KWW, we need to discuss how to apply scaling technique and decomposition of $\Gamma(x)$ to the model of KWW, high frequency part and low frequency part.

- Here is the technique or solution the system takes to calculate the coefficients for the model of KWW, high frequency part, equation (2.5):

$$\hat{\chi}(\mu) = 1 - H_{\beta}(-(\mu\tau_{\beta})^{\beta}); \quad 0 < \beta \leq 1 \quad \text{high frequency part}$$

where $H_{\beta}(x)$ is defined by the series

$$H_{\beta}(x) = \sum_{k=0}^{\infty} \frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)} x^{-k} \quad (2.5)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

Because

$$\Gamma(\beta k + 1) = \Gamma(k + 1), \text{ where, } k = 0, 0 < \beta \leq 1$$

$$\Gamma(\beta k + 1) \leq \Gamma(k + 1), \text{ where, } k = 1, 0 < \beta \leq 1$$

$$\Gamma(\beta k + 1) \leq \Gamma(k + 1), \text{ where, } k > 1, 0 < \beta \leq 1$$

so we have

$$\Gamma(\beta k + 1) \leq \Gamma(k + 1), \text{ where, } k = 0, 1, 2, \dots, 0 < \beta \leq 1$$

the system does the following scaling and decomposition according to the equation (3.2):

$$\frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)} = \left(\prod_{n=0}^{m-1} \frac{\beta k - n}{k - n} \right) \left(\frac{\Gamma(\beta k - (m - 1))}{\Gamma(k - (m - 1))} \right) \quad (3.4)$$

where $m = \lfloor \beta k \rfloor$

For example, suppose $\beta = 0.3$ and $k = 180$, if we calculate the following term directly

$$\frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)} = \frac{\Gamma(0.3 \times 180 + 1)}{\Gamma(181)} = \frac{2.3084 \times 10^{71}}{Inf} = 0$$

But if we apply the technique (3.4), we have the following:

$$\frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)} = \left(\prod_{n=0}^{53} \frac{54 - n}{180 - n} \right) \frac{\Gamma(54 - 53)}{\Gamma(180 - 53)} = 1.1491 \times 10^{-258}$$

Here is the detailed algorithm to calculate the left hand sided of the equation (3.4):

The inputs are β and k

1. $\text{betak} = \beta k$.

2. $m = \lfloor \text{betak} \rfloor$.
3. if $k - (m - 1) < 172$, $\text{result} = \left(\prod_{n=0}^{m-1} \frac{\beta k - n}{k - n} \right) \left(\frac{\Gamma(\beta k - (m-1))}{\Gamma(k - (m-1))} \right)$.
4. else $\text{result} = 0$.
5. return result .

The reason decomposing the gamma function until $n = (m - 1)$ is to avoid the term:

$$\Gamma(\beta k - m)$$

being *Inf*. Because we know it is possible for $\beta k = m$ and $\Gamma(0) = \text{Inf}$.

Before the system does the decomposition above, the system will see whether the value of $k - (m - 1) > 172$ or not. If it is the case, the system will return the value of the coefficient with zero because anyway $\Gamma(k - (m - 1)) = \text{Inf}$, which is a part of the divider. Although $\left(\frac{\Gamma(\beta k - (m-1))}{\Gamma(k - (m-1))} \right)$ can be in the range, we do not do further decomposition. The reason we stop decomposing $\left(\frac{\Gamma(\beta k - (m-1))}{\Gamma(k - (m-1))} \right)$ is that $\Gamma(x)$ becomes very unstable and it is not continuous any more when $x \leq 0$ and $\beta k - (m - 1) \leq 0$ if we do the decomposition for $\left(\frac{\Gamma(\beta k - (m-1))}{\Gamma(k - (m-1))} \right)$ seen at the Figure 3.1.

Why this scaling and decomposition can solve the problem? It is because all of intermediate terms including $\left(\prod_{n=0}^{m-1} \frac{\beta k - n}{k - n} \right)$ and $\left(\frac{\Gamma(\beta k - (m-1))}{\Gamma(k - (m-1))} \right)$ are less than one when $\beta < 1$.

However, when $\beta > 1$, the equation (3.4) is not correct any more because one of the dividers, $k - n$, may be equal to zero in the equation (3.4). Because when $\beta > 1$, then $k \leq m$, $k - n$ may be equal to zero when

n changes from 0 to $m - 1$. We will have the detailed in the coming subsection 3.4.5.

- Here is the technique or solution the system takes to calculate each term for the model of KWW, low frequency part, equation (2.7):

$$\hat{\chi}(\mu) = 1 + H_{\beta}(-(\mu\tau_{\beta})) \quad \text{low frequency part}$$

where $H_{\beta}(x)$ is defined by the series

$$H_{\beta}(x) = \sum_{k=0}^{\infty} \frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)} x^{k+1}, \quad 0 < \beta \leq 1 \quad (2.7)$$

where $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable.

Since when $x \geq 2$, $\Gamma(x)$ increases monotonously, then

$$\Gamma\left(\frac{k+1}{\beta}\right) > \beta\Gamma(k+1), \text{ where, } k \geq 1, 0 < \beta \leq 1$$

and by drawing the curve for the function

$$y = \Gamma\left(\frac{1}{\beta}\right) - \beta\Gamma(1), 0 < \beta \leq 1, \text{ where, } \Gamma(1) = 1,$$

it is easy to see

$$\Gamma\left(\frac{1}{\beta}\right) \geq \beta\Gamma(1), 0 < \beta \leq 1, \text{ where, } \Gamma(1) = 1.$$

So we know from the property of gamma function above,

$$\Gamma\left(\frac{k+1}{\beta}\right) \geq \beta\Gamma(k+1), \text{ where, } k = 0, 1, 2, \dots, 0 < \beta \leq 1,$$

and only when $\beta = 1$, they are equal.

According to this, the system does the following scaling and decomposition according to the equation (3.2):

$$\frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)}x^{k+1}(-1)^k = \left(\prod_{n=1}^k \frac{(-x)((k+1)/\beta - n)}{k+1-n}\right)\Gamma((k+1)/\beta - k)x \quad (3.5)$$

of course, when $k = 0$, the system calculated (3.5) directly as follow:

$$\frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)}x^{k+1}(-1)^k = \Gamma(1/\beta)x/\beta$$

For example, suppose $\beta = 0.5$, $k = 86$ and $x = 1$, if we calculate the following term directly

$$\begin{aligned} \frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)}x^{k+1}(-1)^k &= \frac{\Gamma((86+1)/0.5)}{0.5\Gamma(87)} \\ &= \frac{2\Gamma(174)}{\Gamma(87)} = \frac{Inf}{2.4227 \times 10^{130}} = Inf \end{aligned}$$

But if we apply the technique (3.5), we have the following:

$$\begin{aligned} \frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)}x^{k+1}(-1)^k &= \left(\prod_{n=1}^{86} \frac{((86+1)/0.5 - n)}{86+1-n}\right)\Gamma((86+1)/0.5 - 86) = \\ &= 7.2315 \times 10^{50} \times \Gamma(88) = 1.5242 \times 10^{183} \end{aligned}$$

The example for when $\beta \geq 1$ is similar to the example above. But in this case the system returns a very small number rather than 0 since $\Gamma((k+1)/\beta)$ is less than $\Gamma(k+1)$.

3.4.3 Horner's Rule, Series Handling and Truncations

To speed up the system, we also apply the Horner's rule when we implement the theoretical model. Now let us discuss how to apply the Horner's rule and how to truncate a infinite series such that the model will return accurate values.

Horner's Rule

The Horner's rule is a rule for polynomial computation. It is a rule which both reduces the number of necessary multiplications and results in less numerical instability due to potential subtraction of one large number from another. The rule simply factors out powers of x , giving [23]

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 = ((a_n x + a_{n-1})x + \dots)x + a_0$$

Horner's Rule and Series Handling for the Model of KWW, High Frequency Part, Equation (2.5)

We know that the equation (2.5) is a polynomial and the equation (3.4) is used to calculate coefficients for the polynomial more precisely according to the analysis in the previous section. We also know that coefficients for the polynomial (2.5) are always same as long as the value of β is same. And when the value of k increases, the coefficient for the polynomial $\frac{\Gamma(\beta k + 1)}{\Gamma(k + 1)}$ decreases. So we calculate all of coefficients of the polynomial for a particular β until the new calculated coefficient is equal to zero. We only calculate them one time. And then construct a vector which elements are coefficients for the polynomial. We

just use these coefficients to calculate the simulated data in the range of high frequency part. When we calculate the simulated data in the range of high frequency part, we apply the Horner's rule and the scalar-vector multiplication to improve the performance while the scalar is one of the coefficients and the vector is the $-(\mu\tau_\beta)^\beta$. This way by calculating all of coefficients at the same time will improve the accuracy for the high frequency part and also save much execution time from comparison to the value of intermediate term and the program will not be composed of many if statements.

Here is a piece of algorithm:

1. $m = \text{length}(\text{coeff})$;
2. $\text{kwwterm} = \text{coeff}(m)$;
3. for $k=m-1:-1:1$
4. $\dots\text{kwwterm} = \text{kwwterm}./\text{xm} + \text{coeff}(k)$;
5. end;

where coeff is the vector which elements are coefficients for the polynomial, xm is the $-(\mu\tau_\beta)^\beta$, τ_β is the relaxation time, $\mu = -2\pi i \times \text{frequency}$ and frequency is the independent variable in the range of high frequency part.

The reason that such design can save the execution time is because most of data points are in the range of high frequency part and they have the same vector which elements are coefficients of the polynomial for a common β .

Series Handling and Truncations for the Model of KWW, Low Frequency Part, Equation (2.7)

We know that the experimental data are most likely in high frequency part if using the model of KWW. We know how to let the model return accurate result in the range of high frequency part. However, we can not tell whether or not the intermediate simulated data corresponding to the same independent variable frequency as the experimental data have are also most likely in the range of high frequency part in each single iterative step. The intermediate simulated data we discuss here is the simulated data generated by the chosen model using the intermediate values for the parameters in the iterative steps in the nonlinear least squares algorithm. We have to make sure they are also accurate if where they are located is changed from the range of high frequency part to the range of low frequency part. Otherwise, we cannot confirm that the result returned by the system is correct or the system may not work correctly and only return the wrong result because it does not follow the correct path to converge the result. We have to make sure the model of KWW, low frequency part is also able to return accurate result though it is not called all the time but occasionally. Before we do the design for it, we do the following analysis. Firstly, we do the following approximation,

$$\frac{\Gamma((k+1)/\beta)}{\beta\Gamma(k+1)}x^{k+1} \approx \frac{1}{\beta}\left(\frac{(k+1)}{\beta} - 1\right)\left(\frac{(k+1)}{\beta} - 2\right)\cdots(k+1)(x)^{k+1}. \quad (3.6)$$

And suppose we have the following geometric series,

$$b_1, b_2, b_3, \dots, b_i(\text{currentTerm}), \dots, b_{n-1}, b_n(\text{verylastTerm}),$$

in term of the truncated tail from the last term to the current term, we have

$$a_1(= b_n = \text{verylastTerm}), a_2(= b_{n-1}), \dots, a_{n-(i-1)}(= b_i = \text{currentTerm}), \dots,$$

since it is a geometric series, we have

$$\sum_{j=1}^{n-(i-1)} a_j = \frac{a_1(1 - q^{(n-(i-1)+1)})}{1 - q} = \frac{\text{verylastTerm} - \text{currentTerm}}{1 - q}, \quad a_{j+1} = a_j q. \quad (3.7)$$

So if we set

$$\left| \frac{1}{\beta} \left(\frac{(k+1)}{b} - 1 \right) \left(\frac{(k+1)}{b} - 2 \right) \dots (k+1) (x)^{k+1} \right| \leq | (1 - (x)) \text{tolerance} |$$

or

$$| \text{currentTerm} | \leq | (1 - (x)) \text{tolerance} |,$$

since the series eventually converges, we suppose *verylastTerm* closed to zero,

and in our case $q = x$, then we have,

$$\begin{aligned} \sum_{i=1}^n a_i &= \frac{\text{verylastTerm} - \text{currentTerm}}{1 - q} = \\ &= \left| \frac{\text{verylastTerm} - \text{currentTerm}}{1 - x} \right| \approx \left| \frac{\text{currentTerm}}{1 - x} \right| \leq \text{tolerance} \end{aligned}$$

then we will know the truncation error for the particular data point will be less than *tolerance* according to the formula (3.7) since the sum of the truncated tail is less than *tolerance*.

After doing the truncation above, the system solves the issue of ∞ terms for the model of KWW, low frequency part, equation (2.7) and the model of KWW, low frequency part is able to return accurate result. The chosen default truncation error, *tolerance*, in the system is 10^{-6} .

However, since the theoretical model of KWW, low frequency part and high frequency part are different, we have to provide a robust way for the system to find threshold such that the system can switch between two parts of KWW model automatically and precisely. Now let us describe the model switch method.

3.4.4 Models Switch Method

Although we know the experimental data are most likely in high frequency part if using the model of KWW, we are still not sure the simulated data corresponding to the initial values for unknown parameters are in high frequency part or not. Based on the fact that the model of KWW, low frequency part, equation (2.7) only covers really small portion of frequency range, we did the following design in the system.

The design here is based on the assumption that the experimental data are sorted in frequency. At the beginning, the system will use the model of KWW, low frequency part, equation (2.7) to calculate the simulated data corresponding to the values for unknown parameters. The system will automatically determine whether the current data point is threshold or not. The method to find threshold from the low frequency part to the high frequency part in the system is to let system use the model of KWW, low frequency part, equation (2.7) to calculate the simulated data point for each data point until the real part of simulated data point returned by the system is any of *Inf, NaN*, less than zero or greater than 1. If the real part of simulated data

point is any of *Inf*, *NaN*, less than zero or greater than 1, the system will treat it as the threshold and then switch the model of KWW from low frequency part to high frequency part and also recalculate the simulated data point at the threshold, the current data point. When the system uses the model of KWW, low frequency part, equation (2.7) to calculate the simulated data corresponding to the values for unknown parameters, the system will see whether the current data point is threshold or not for each data point by the means of lazy evaluation. Fortunately, it does not consume too much execution time for this step since the low frequency part is the small portion of the full range of frequency. The system also detects the following case and return the invalid value to signal the program to switch model if the system use the model of KWW, low frequency part, equation (2.7) to calculate the simulated value for current data point, which the system should use the model KWW, high frequency part, equation (2.5). When the system used the model of KWW, high frequency part, equation (2.5), the system uses scalar and vector multiplication to calculate the remainder of simulated data, the whole high frequency part together as we described in the section 3.4.3.

3.4.5 Improving Correctness and Robustness

As we mentioned in the section 3.4.2, the system may encounter doing the calculation by dividing some term by zero. So it is also important to apply some techniques to improve the correctness and robustness for the model of KWW and the algorithm. So let us discuss some applied techniques in the

system to improve correctness and robustness here.

Techniques Applied to the Model of KWW, High Frequency Part

1. When $\beta > 1$, the decomposition of (3.4) may lead to being divided by zero. In order to make the program more robust, the system will check whether the factor is zero or not. If the factor is zero, the system will let the factor become the minimum value that the Matlab can present precisely to keep the computation on going. The system takes 10^{-16} . At this point, we do not consider the precision much here. No matter what the system returns, the system just discards the result since it is not what the user want. The purpose to replace zero with the minimum value is to keep the system doing the calculation and improving the system's robustness. we do not care the result any more. If the system eventually converges the satisfied result, that is good and the user will take the result. Otherwise the user can run the system again with better initial values.
2. When $m < 2$, where $m = \lfloor \beta k \rfloor$, the decomposition of (3.4) is also not correct because there is not common factor and if $k=0$, $\beta * k/k \neq \beta$. When $m < 2$, the system will calculate the (2.5) directly or if $k \geq 172$ let the returned value be zero.
3. When calculating the coefficient of polynomial using the decomposition of (3.4), if the value for $\beta > 1$, the coefficient using the decomposition (3.4) will not converge to zero leading to a infinity loop. The system will

force the coefficient become zero to avoid the infinite loop by detecting whether the absolute value of difference between k and m is larger than 171 and using the gamma function's property where the gamma function $\Gamma(x)$ overflows when $x > 171$. In this way, the system can share the same code for the calculation for the case either when $\beta \geq 1$ or $\beta < 1$.

Techniques Applied to the Model of KWW, Low Frequency Part

After we perform many experiments, it is found that the threshold in term of frequency for the model of KWW, from low frequency part to high frequency part is different from threshold in term of frequency for the model of KWW, from high frequency part to low frequency part. The previous one is greater than the latter one in term of frequency, which means there is overlap between the model of KWW, low frequency part and high frequency part, or we can say, there is a common frequency range where both parts of the model of KWW work. Moreover it is more precise or accurate using low frequency part than using high frequency part in the overlap area. So the system uses the model of KWW, low frequency part until it does not work any more, which was described in the section 3.4.4.

Techniques Applied to the Algorithm for Least Squares Estimation of Nonlinear Parameters

1. In the algorithm introduced by Donald W. Marquardt [15], a special matrix is required as follows:

$$P^{n \times k} = \left(\frac{\partial f_i}{\partial b_j} \right), \quad i = 1, 2, 3, \dots, n, \quad j = 1, 2, 3, \dots, k$$

where n is the number of data points available, k is the number of unknown parameters, f is the function or model and b is the set of unknown parameters.

However it is not easy to get the derivative for the functions of models available. To get the special matrix above, the system took the finite differentiation method to approximate the derivative by calculating the slope of each data point with its perturbed data point.

2. We know that it is the complex number operation when applying the algorithm [15] to our particular topic. So the returned vector of δ introduced in the algorithm [15] is most likely the vector with some complex number elements, which are supposed to be real numbers. So the system called the Matlab built-in function `real.m` to ignore their imaginary parts such that the returned vector of δ is in the real number field. The reason we can ignore the imaginary parts is that the imaginary parts are less than one percent of the corresponding real parts we found when we perform the experiments. Moreover, even if the imaginary parts are not less than one percent of the corresponding real parts, according to the

gradient methods, any proper method must result in a correction vector whose direction is within 90 degree of the negative gradient of θ [15], the angle of real parts and complex vector must be within 90 degree. So we can ignore the imaginary part by sacrificing the speed of convergence to make sure the returned values for those unknown parameters are real number.

3. There are two different ways to terminate the program: the first is reaching a local optimal solution when the derivative becomes zero and the other is reaching the maximum iteration times. If the system reaches the maximum iteration times, the system will notice the user that the maximum iteration times is reached and suggest the user the answer may not be correct and the user can choose new initial values to recalculate them again.

Chapter 4

Experiments

This chapter discusses the testing using thirty sets of data and different methods. The testing results will be shown as sixty tables. The contents for each table include initial values of unknown parameters, true values of unknown parameters, the parameter values returned by the system, the fitting error and the iterative times. This chapter firstly discusses the procedure how to carry out an experiment, how to choose the initial values, how to generate data and then does the comparison between the true values and the parameter values returned by the system. Finally there are discussion and result.

4.1 Procedure to Carry Out an Experiment

The procedure to carry out an experiment is the following:

1. Type the name of the input file containing the experimental data.
2. Type the name of the output file where the result will be written.
3. Tell the system that the format of the independent variable, frequency of the experimental data is logarithmic value of frequency or frequency. The developed system can only handle these two kinds of the independent variables, logarithmic value of frequency and frequency.
4. Check the drawn curve and get some valuable information from it such as the location of the peak. It is helpful to choose the initial values.
5. Determine which part of the curve that need the best fit by providing lower bound and upper bound.
6. Choose a model to fit the data. The available models are KWW, DC, HN and FD.
7. Choose a method to fit the data. The available methods are by using the algorithm introduced in the paper [15], named after method I and by calling the Matlab built-in function `fminsearch()`, named after method II.
8. Provide the initial values for the unknown parameters after observing the curve drawn by the system. We will have the discussion how to choose

the initial values for the unknown parameters in the coming section.

9. Observe the intermediate results and final result on screen shown by the system.
10. Wait for the system writes the result into the output file and draws the fitting curve to finish the whole calculation.

When we follow the procedure above, the most difficult step is how to guess the initial values for the unknown parameters with the user's background and knowledge of the experimental data and the hidden information from the curve for the experimental data drawn by the system. The curve drawn by the system here which is discussed in the section 1.1.2 includes two parts. One part is the curve of the real part of the experimental data plotted as a function of frequency. The first column of the experimental data, frequency, as the abscissa and the real part of the experimental data, the second column of experimental data, as the ordinate. The other part of the curve is the curve of the imaginary part of the experimental data plotted as a function of frequency. The first column of the experimental data, frequency, as the abscissa and the imaginary part of the experimental data, the third column of the experimental data, as the ordinate. There is an example of a fitting curve, Figure 4.1. Now let us discuss how to choose the initial values for the unknown parameters.

4.2 Method to Choose the Initial Values

It is important to choose the initial values when using an iterative method. Better initial values lead to the correct result with less execution time. The system may not return the correct result with the weak initial values. Here is the discussion how to choose the initial values for τ , α , β , γ , ϵ_0 and ϵ_∞ . Let us start to discuss how to choose the initial value for the parameter τ . Its value is usually the most difficult to guess.

4.2.1 Method to Choose the Initial Value of τ

From [13], we know that:

$$\tau = k * \frac{1}{f_{max}};$$

where k is a constant depending on the testing material and f_{max} is the coordinate value of frequency where the peak of the curve of the imaginary part of the experimental data plotted as a function of frequency is located. When the value of f_{max} is greater, the value of τ is less.

For the model only having one parameter τ , such as KWW, DC and HN, if the logarithmic value of frequency is around 5 to 6, then the value of τ would probably be around 1 microsecond [msec]. If the logarithmic value of frequency is around 6 to 7, then the value of τ would probably be around 0.1 microsecond [msec]. If the logarithmic value of frequency is around 4 to 5, then the value of τ would probably be around 10 microsecond [msec]. If no peak occurs, the user may guess the initial value of τ by combining two parts

of the curve to determine where the peak would be.

For the model having two parameters of τ , such as FD, we know one τ means one peak in the curve of the imaginary part of the experimental data plotted as a function of frequency and two τ s mean two peaks in the curve. The latter does not mean the overlap of these two peaks also includes two peaks. The curve may only show one peak. Then we need to guess two values of τ based on the location of the only one peak using the equation above. It is more difficult to guess the initial values for the model having two parameters of τ than for those models only having one parameter τ .

4.2.2 Method to Choose the Initial Value of α , β and γ

There is not a robust or efficient way to guess the initial values of α , β and γ . The range of α , β and γ theoretically is from 0 to 1. And usually most of time the practical range of α , β and γ is from 0.2 to 0.75. The system will provide the default value for them, which is 0.5, if the user does not know how to choose them. If there are two parameters among α , β and γ in the model, such as HN model, the user may choose the values, 0.4 and 0.6 such that the range from 0.2 to 0.75 may be divided into three approximately equal spaces.

4.2.3 Method to Choose the Initial Value of ϵ_0 and ϵ_∞

There is also not a robust or efficient way to guess the initial values of ϵ_0 and ϵ_∞ . The system will provide the default values for them, which are the maximum value of the real part data and minimum value of the real part data,

if the user does not know how to choose them.

However, the default values are usually far away from the true values for ϵ_0 and ϵ_∞ due to the limitation of experimental equipment. Here is the second way to choose the initial values for ϵ_0 and ϵ_∞ : the default value of ϵ_0 added by some value for ϵ_0 and the default value ϵ_∞ reduced by some value for ϵ_∞ .

And usually the system will always return some approximate values for these two parameters if the user tries any values first. Then the user may use the values returned by the system as the initial values. This is the third way to choose the initial values for ϵ_0 and ϵ_∞ .

4.3 Data Generation and Test Plan

First thing we need to do is to prepare the input data to test the system. After the input data are ready, we just follow the procedure to do the testing. When we do the testing, we do not cut any part of the input data or we may say we ask the system to do the fitting for the whole data. The testing results are shown as tables. The table contents include initial values for the unknown parameters, true values of the unknown parameters, the returned values from the system, the fitting error and the iterative times. The fitting error is defined as the *sum of square*, introduced in the equation 3.1, divided by how many data points. The simulated data in the equation 3.1 used to calculate *sum of square* is generated by using the values for the unknown parameters returned by the system such that we know how good the data fitting is. And the true values of the unknown parameters are the true values for the parameters with respect to the input data that the system is supposed to return. We will have discuss about how we get these true values for the unknown parameters in the following section.

4.3.1 Data Generation

To prepare the input data for testing, we need to provide the values for the parameters of the chosen model. Then we can use the developed system, which is used to generate data, to generate the input data to test the system. The provided values for parameters of the chosen model and the chosen model will

be recorded at first two rows of the generated data as we discuss in chapter 3. We treat the provided values for parameters of the chosen model as the true values for the parameters with respect to the generated data. If the data point are enough and the system works properly, the system is supposed to return the provides values for parameters. However, like what we discuss before, the answer or the result to the same input data may not be unique. The true values for the parameters are just references to see how close to them the values for the parameters returned by the system is. There are totally five ways to generate the input data. One is the way from N.B.S data tables while the other four ways are from four available models. We will generate thirty sets of the input data to test the system.

Data Generation from N.B.S Data Tables

There are eight sets of data generated from N.B.S data tables introduced by Menachem Dishon, George H. Weiss and John T. Bendler [13]. The reason why we generate eight sets of data is the following: we choose two typical values for the parameter β and then choose two different values for each of other three parameters $\tau[msec]$, ϵ_0 and ϵ_∞ such that there are always at least two sets of data where only one parameter has different value between them. Then there should be only five sets of the input data. Here is the table as an example at Table 4.66.

However, after we check the N.B.S data tables, we found that each of N.B.S data table is corresponding to one β value. We decide not to use only

one N.B.S data table to generate data. So we decide generate four sets of the input data for both of N.B.S data tables by using two different data tables from N.B.S.

Data Generation from Available Models

There are four models available. They are KWW model, DC model, HN model and FD model. We chose different sets of values of parameters for each model such that there are always at least two sets of data where only one parameter has different value between them. Similar to the case of N.B.S data tables, if there are four parameters in the model such as the models of KWW and DC, then there are five sets of data for each of them. If there are five parameters in the model such as the models of HN and FD, then there are six sets of data for each of them. We used the implemented theoretical models to generate the data using the different sets of values of parameters.

4.3.2 Test Plan

After thirty sets of the input data are ready to use, we use the model of KWW to fit the input data generated from N.B.S data tables and from the model of KWW and we use the model of DC, HN and FD to fit the input data generated from the model of DC, HN and FD respectively. We use two different methods, which are the method by using the algorithm introduced in the paper[15], named after method I, and the method by calling the Matlab built-in function `fminsearch()`, named after method II, to fit the input data.

The way how to choose the initial values is just following the section 4.2.2 about how to choose the initial values. The testing results as sixty tables are shown at the end of this chapter.

4.4 Discussion and Results

This discussion includes two parts. One is the discussion about the testing using the simulated data without noise, the other is the discussion about the testing using the simulated data with some artificial noise.

4.4.1 Discussion of Testing Using Simulated Data without Noise

We know that there are eight sets of data from N.B.S data tables, respecting to Tables 4.1–4.8 and 4.31–4.38, five sets of data from KWW model, respecting to Tables 4.9–4.13 and 4.39–4.43, five sets of data from DC model, respecting to Tables 4.14–4.18 and 4.44–4.48, six sets of data from HN model, respecting to Tables 4.19–4.24 and 4.39–4.54, and six sets of data from FD model respecting to Tables 4.25–4.30 and 4.55–4.60. From sixty tables at the end of this chapter, we can see that there are eight calculations which error is 10^{-6} when using the method I. They are Table 4.1, Table 4.8, Table 4.21, Table 4.23, Table 4.25, Table 4.27, Table 4.28 and Table 4.30. The errors of other twenty two calculations are less than 10^{-6} . Among eight calculations which error is 10^{-6} , there are two cases which data were generated from N.B.S data tables and used KWW model to fit the data, two cases which data were generated from HN model and used HN model to fit the data and four cases which data were generated from FD model and used FD model to fit the data. Firstly, we may say that the simulated data generated from FD model are most difficult to

fit even using the FD model itself to fit because there are four cases which the fitting error is 10^{-6} using the simulated data generated from FD model. Secondly, it should be HN data model since there are two cases which fitting error is 10^{-6} using the simulated data generated from HN model. The reasons for these may be the following:

- There are six parameters for both HN model and FD model, which increase more difficulty to fit the data and lead to multi-fitting results when the range of data is not enough.
- There are two relaxation times τ in the FD model, which means the data generated from FD model have two peaks and overlap into one peak when the data only have one peak. This increases the difficulty for the user to choose suitable initial values for these relaxation times τ .
- In either HN model or FD model, there are two parameters which play similar role used to modify the shape of the curve in each of model. In HN model, they are α and γ . In FD model, they are τ and τ' . These increase the difficulty to choose initial values for them. And the initial value for relaxation time τ , which magnitude is around microsecond, 10^{-6} , is more difficult to decide than the initial value for α and γ , which range is only from 0.2 to 0.7 practically.

And then we may say, the data generated from N.B.S data tables, KWW model and DC model are easier for a user to fit than those of HN model and FD model by observing the results from sixty tables at the end of

this chapter since there are only two cases which error is 10^{-6} when using the algorithm in [15], method I.

There is no calculation which fitting error is greater or equal to 10^{-6} when using the method II. This means the subsystem using the method II works better than the subsystem using the method I in term of the fitting error. However, it is shown by the experiments that the subsystem using the method II runs more slowly than the subsystem using the method I seen at Table 4.67 and Table 4.68. The results returned by the system for the data generated from N.B.S data tables, KWW model and DC model are very closed to the true values. Although the fitting errors are all less than 10^{-6} for HN model and FD model, the results showed in the Tables 4.50, 4.51, 4.54, 4.57 and 4.60 are not so close to the true values. The result shown in the Table 4.51 even worse and far away from the true values. However, the simulated data returned by the system still matched the input data very well in the fitting error by observing the curves drawn by the system. This meant they were good fittings in term of fitting error. This showed that the result to the data fitting for KWW model and DC model, which have only four parameters, is more likely unique than those of HN model and FD model having five parameters.

Comparison between Method I and Method II is seen at Table 4.67 and Table 4.68.

Although there are so many cases, which fitting error is 10^{-6} when using method I, the curves drawn by the system showed that the simulated data returned by the system matched the input data in the fitting error very

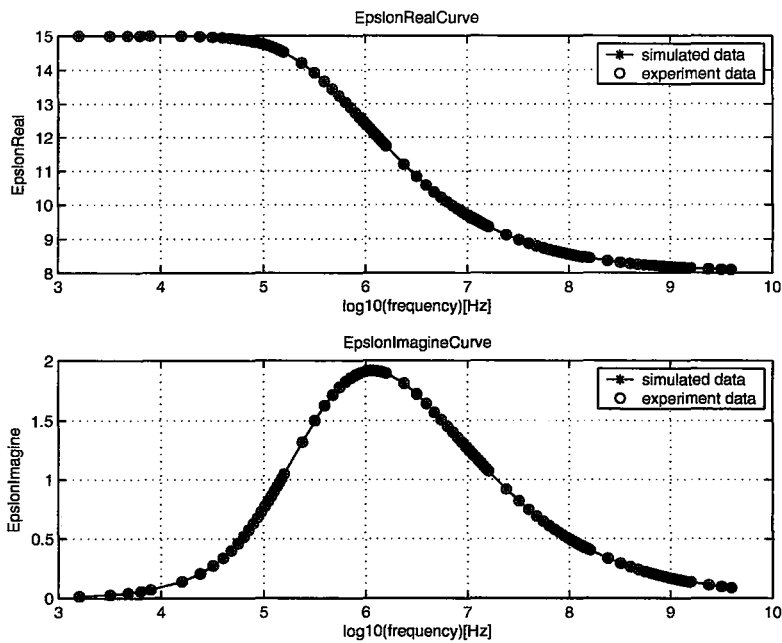


Figure 4.1: The Fitting Curve for the Table 4.1

well, which meant they were good fittings, too. Here is the fitting curve for the Table 4.1 at Figure 4.1, which fitting error is 10^{-6} .

Here is the program script Figure 4.2 for the experiment regarding to the Table 4.1.

4.4.2 Discussion of Testing Using Simulated Data with Artificial Noise

This section is the discussion about the testing using some previous generated data with some artificial noise generated with Matlab function `randn()` to test the system for remedying the shortage of experimental data. The results are

Figure 4.2: The Program Script for the Table 4.1



available at Tables 4.61 to Table 4.65 with the corresponding fitting curves.

Firstly, Let us discuss how to quantify the relative error adding to the simulated data through the artificial noise.

For each column of the input data to the system, we have the original data x and the data with noise xx . To quantify the relative error, first find the vector of the difference between x and xx ; that is,

$$e = (x - xx),$$

then we reduce this vector to one scalar value using the norm of the vector.

To make this a relative error, we divide it by the norm of x ; that is,

$$error = || e || / || x ||$$

We add the artificial noise to the real part of the simulated data and the imaginary part of the simulated data separately. Here are what we add:

For the data used for Table 4.61, we add 0.79% noise and 3.77% noise into the real part and imaginary part of the simulated data respectively.

For the data used for Table 4.62, we add 2.21% noise and 9.90% noise into the real part and imaginary part of the simulated data respectively.

For the data used for Table 4.63, we add 2.69% noise and 8.00% noise into the real part and imaginary part of the simulated data respectively.

For the data used for Table 4.64, we add 1.67% noise and 10.65% noise into the real part and imaginary part of the simulated data respectively.

Secondly, Let us see how well the results are returned by the system.

From the Table 4.61, we can see that returned values are 0.499604, 0.101681, 15.014866 and 7.989935 respectively to the true values with 0.5, 0.1, 15 and 8. The relative error for unknown parameters is 0.07%, 1.68%, 0.1% and 0.12% respectively.

From the Table 4.62, we can see that returned values are 0.305919, 1.002758, 4.999919 and 2.020259 respectively to the true values with 0.3, 1, 5 and 2. The relative error for unknown parameters is 1.97%, 0.27%, 0.001% and 1.01% respectively.

From the Table 4.63, we can see that returned values are 0.501162, 0.998743, 4.948217 and 2.000990 respectively to the true values with 0.5, 1, 5 and 2. The relative error for unknown parameters is 0.23%, 0.12%, 1.04% and 0.05% respectively.

From the Table 4.64, we can see that returned values are 0.490221, 0.991036, 5.980375 and 3.988693 respectively to the true values with 0.5, 1, 6 and 4. The relative error for unknown parameters is 2.0%, 0.9%, 0.33% and 0.28% respectively.

From the previous discussion, we know that the percentage change in the fitted parameters relative to the percentage noise added into the simulated data is acceptable. The fitted parameters are not changed very much in the presence of reasonable noise, which suggests that the algorithm does not suffer from sensitivity problems, at least for the testing here.

Combining the discussion of testing using the simulated data without artificial noise with the discussion of testing using the simulated data with

artificial noise, we may say the algorithm used by the developed system here does not suffer from sensitivity problems.

4.5 Testing the System with the Method I and the Model of KWW

There are 13 testing results using Method I for this section. Eight of thirteen testings used data generated from N.B.S tables, the other five testings used data generated from the model of KWW.

4.5.1 Using Data Generated from N.B.S Data Tables

There are eight tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of KWW. The table also records the fitting error and the iterative times for the testing. There are eight sets of data with respective to each of eight tables. They are Data 1, 2, 3, 4, 5, 6, 7 and 8, generated from N.B.S tables. Each of data 2, 3 and 4 only has one parameter except β with different value compared to data 1 respectively. Each of data 6, 7 and 8 only has one parameter except β with different value compared to data 5 respectively. There are only two different values for β . One is for data 1, 2, 3 and 4, the other is for data 5, 6, 7 and 8.

Table 4.1: Computation for Data 1 with Method I

Data 1	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.4	0.2	13	7
True values	0.5	0.1	15	8
Returned values	0.499918	0.100010	14.999365	8.001801
Error	0.000001		Iterative times	4

Table 4.2: Computation for Data with Different Values of τ to Data 1

Data 2	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.39	0.0092	14.7	7.56
True values	0.5	0.01	15	8
Returned values	0.500209	0.010002	14.999521	8.000790
Error	0.000000		Iterative times	2

Table 4.3: Computation for Data with Different Values of ϵ_0 to Data 1

Data 3	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	0.4	11	7
True values	0.5	0.1	10	8
Returned values	0.499944	0.099989	10.000058	8.000065
Error	0.000000		Iterative times	4

Table 4.4: Computation for Data with Different Values of ϵ_∞ to Data 1

Data 4	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.8	0.5	18	7
True values	0.5	0.1	15	11
Returned values	0.500156	0.100053	14.999706	11.000598
Error	0.000000		Iterative times	6

Table 4.5: Computation for Data 5 with Method I

Data 5	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	0.5	default	default
True values	0.3	1	5	2
Returned values	0.299847	0.999290	4.999534	2.000298
Error	0.000000		Iterative times	3

Table 4.6: Computation for Data with Different Values of τ to Data 5

Data 6	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.6	0.7	default	default
True values	0.3	2	5	2
Returned values	0.300016	2.000039	4.999987	2.000064
Error	0.000000		Iterative times	4

Table 4.7: Computation for Data with Different Values of ϵ_0 to Data 5

Data 7	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	1	7.1	1.9
True values	0.3	1	7	2
Returned values	0.299979	0.999659	6.999852	2.000446
Error	0.000000		Iterative times	3

Table 4.8: Computation for Data with Different Values of ϵ_∞ to Data 5

Data 8	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	3	8	4
True values	0.3	1	5	3
Returned values	0.300467	0.996654	5.000134	3.002701
Error	0.000001		Iterative times	3

4.5.2 Using Data Generated from the Model of KWW

There are five tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of KWW. The table also records the fitting error and the iterative times for each testing. There are five sets of data with respective to each table. They are Data 9, 10, 11, 12 and 13, generated by the model of KWW. Each of data 10,

11, 12 and 13 only has one parameter with different value compared to data 9 respectively.

Table 4.9: Computation for Data 9 with Method I

Data 9	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	1.05	5.1	default
True values	0.5	1	5	2
Returned values	0.499858	1.012288	5.006555	2.001018
Error	0.000000		Iterative times	1

Table 4.10: Computation for Data with Different Values of τ to Data 9

Data 10	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	3	default	default
True values	0.5	2	5	2
Returned values	0.499999	1.999636	4.999966	2.000003
Error	0.000000		Iterative times	3

Table 4.11: Computation for Data with Different Values of ϵ_0 to Data 9

Data 11	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.75	1.0	6	1.6
True values	0.5	1	7	2
Returned values	0.500000	0.999938	6.999963	2.000000
Error	0.000000		Iterative times	3

Table 4.12: Computation for Data with Different Values of ϵ_∞ to Data 9

Data 12	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.66	2.4	default	default
True values	0.5	1	5	1.7
Returned values	0.500426	0.995416	4.997588	1.699923
Error	0.000000		Iterative times	3

Table 4.13: Computation for Data with Different Values of β to Data 9

Data 13	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	3	default	default
True values	0.3	1	5	2
Returned values	0.299891	0.998384	5.000047	2.000126
Error	0.000000		Iterative times	4

4.6 Testing the System with the Method I and the Model of DC

There are five tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of DC. The table also records the fitting error and the iterative times for each

testing. There are five sets of data with respect to each table. They are Data 14, 15, 16, 17 and 18, generated by the model of DC. Each of data 15, 16, 17 and 18 only has one parameter with different value compared to data 14 respectively.

Table 4.14: Computation for Data 14 with Method I

Data 14	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	default	1.5	default	default
True values	0.5	1	6	4
Returned values	0.499356	1.000384	5.999862	4.000128
Error	0.000000		Iterative times	2

Table 4.15: Computation for Data with Different Values of τ to Data 14

Data 15	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.25	0.5	7.1	2.89
True values	0.5	0.1	6	4
Returned values	0.499671	0.100193	6.000011	4.000166
Error	0.000000		Iterative times	4

Table 4.16: Computation for Data with Different Values of ϵ_0 to Data 14

Data 16	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	2.3	8.5	3.5
True values	0.5	1	8	4
Returned values	0.499995	0.999973	8.000005	4.000005
Error	0.000000		Iterative times	5

Table 4.17: Computation for Data with Different Values of ϵ_∞ to Data 14

Data 17	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	1.8	default	default
True values	0.5	1	6	2
Returned values	0.499827	1.000563	5.999996	2.000079
Error	0.000000		Iterative times	4

Table 4.18: Computation for Data with Different Values of γ to Data 14

Data 18	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	2	7.2	2.8
True values	0.3	1	6	4
Returned values	0.299906	0.998026	6.000013	4.000160
Error	0.000000		Iterative times	4

4.7 Testing the System with the Method I and the Model of HN

There are six tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of HN. The table also records the fitting error and the iterative times for each testing. There are six sets of data with respective to each table. They are Data 19, 20, 21, 22, 23 and 24, generated by the model of HN. Each of data 20, 21, 22, 23 and 24 only has one parameter with different value compared to data 19 respectively.

Table 4.19: Computation for Data 19 with Method I

Data 19	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.6	1.5	default	default
True values	0.3	0.7	1	6	4
Returned values	0.300084	0.699280	1.001741	5.999853	3.999923
Error	0.000000		Iterative times	5	

Table 4.20: Computation for Data with Different Values of τ to Data 19

Data 20	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.33	0.66	0.23	6.3	4
True values	0.3	0.7	0.1	6	4
Returned values	0.305677	0.641133	0.133702	5.995452	3.968593
Error	0.000000		Iterative times	4	

Table 4.21: Computation for Data with Different Values of ϵ_0 to Data 19

Data 21	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.3	0.8	2	8	3
True values	0.3	0.7	1	7	4
Returned values	0.288612	0.761091	0.762138	7.028075	4.016047
Error	0.000001		Iterative times	5	

Table 4.22: Computation for Data with Different Values of ϵ_∞ to Data 19

Data 22	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.25	0.8	0.1	7	2
True values	0.3	0.7	1	6	2
Returned values	0.298102	0.710974	0.946816	6.004842	2.004730
Error	0.000000		Iterative times	5	

Table 4.23: Computation for Data with Different Values of α to Data 19

Data 23	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.25	0.68	1.5	default	default
True values	0.5	0.7	1	6	4
Returned values	0.506941	0.681826	1.055397	5.993776	3.998985
Error	0.000001		Iterative times	3	

Table 4.24: Computation for Data with Different Values of γ to Data 19

Data 24	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.4	0.6	1.5	default	default
True values	0.3	0.6	1	6	4
Returned values	0.304805	0.577715	1.120708	5.993001	3.993315
Error	0.000000		Iterative times	3	

4.8 Testing the System with the Method I and the Model of FD

There are six tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of FD. The table also records the fitting error and the iterative times for each

testing. There are six sets of data with respect to each table. They are Data 25, 26, 27, 28, 29 and 30, generated by the model of FD. Each of data 26, 27, 28, 29 and 30 only has one parameter with different value compared to data 25 respectively.

Table 4.25: Computation for Data 25 with Method I

Data 25	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.55	0.7	1.3	6.1	3.9
True values	0.3	0.3	0.8	6	4
Returned values	0.301950	0.276007	0.792622	5.999996	4.000041
Error	0.000001		Iterative times	6	

Table 4.26: Computation for Data with Different Values of τ to Data 25

Data 26	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.6	0.5	1.3	6.1	3.9
True values	0.3	0.5	0.8	6	4
Returned values	0.294137	0.617030	0.824889	6.000267	4.000039
Error	0.000000		Iterative times	5	

Table 4.27: Computation for Data with Different Values of ϵ_0 to Data 25

Data 27	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	default	0.5	0.9	default	default
True values	0.3	0.3	0.8	7	4
Returned values	0.295490	0.361339	0.820414	7.000186	4.000237
Error	0.000001		Iterative times	2	

Table 4.28: Computation for Data with Different Values of ϵ_∞ to Data 25

Data 28	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	default	0.5	0.9	default	default
True values	0.3	0.3	0.8	6	3
Returned values	0.295489	0.361341	0.820415	6.000186	3.000237
Error	0.000001		Iterative times	2	

Table 4.29: Computation for Data with Different Values of α to Data 25

Data 29	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	default	0.5	0.9	default	default
True values	0.6	0.3	0.8	6	4
Returned values	0.600284	0.299861	0.799894	6.000009	4.000043
Error	0.000000		Iterative times	3	

Table 4.30: Computation for Data with Different Values of τ' to Data 25

Data 30	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	0.2	0.5	default	default
True values	0.3	0.3	0.6	6	4
Returned values	0.300381	0.287523	0.598196	5.999973	3.999996
Error	0.000001	Iterative times		4	

4.9 Testing the System with the Method II and the Model of KWW

There are 13 testing results using Method II for this section. Eight of thirteen testings used data generated from N.B.S tables, the other five testings used data generated from the model of KWW.

4.9.1 Using Data Generated from N.B.S Data Tables

There are eight tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of KWW. The table also records the fitting error and the iterative times for the testing. There are eight sets of data with respective to each of eight tables. They are Data 1, 2, 3, 4, 5, 6, 7 and 8, generated from N.B.S tables. Each of data 2, 3 and 4 only has one parameter except β with different value compared

to data 1 respectively. Each of data 6, 7 and 8 only has one parameter except β with different value compared to data 5 respectively. There are only two different values for β . One is for data 1, 2, 3 and 4, the other is for data 5, 6, 7 and 8.

Table 4.31: Computation for Data 1 with Method II

Data 1	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	default	1.5	default	default
True values	0.5	0.1	15	8
Returned values	0.499985	0.100000	15.000036	7.999954
Error	0.000000		Iterative times	335

Table 4.32: Computation for Data with Different Values of τ to Data 1

Data 2	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	default	1.5	default	default
True values	0.5	0.01	15	8
Returned values	0.499980	0.010000	15.000039	7.999954
Error	0.000000		Iterative times	300

Table 4.33: Computation for Data with Different Values of ϵ_0 to Data 1

Data 3	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	1.5	12	6
True values	0.5	0.1	10	8
Returned values	0.500015	0.099982	9.999980	8.000004
Error	0.000000		Iterative times	193

Table 4.34: Computation for Data with Different Values of ϵ_∞ to Data 1

Data 4	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	1	15.5	10
True values	0.5	0.1	15	11
Returned values	0.499977	0.100003	15.000032	10.999973
Error	0.000000		Iterative times	272

Table 4.35: Computation for Data 5 with Method II

Data 5	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	3	7	1
True values	0.3	1	5	2
Returned values	0.300006	1.000052	5.000002	2.000025
Error	0.000000		Iterative times	244

Table 4.36: Computation for Data with Different Values of τ to Data 5

Data 6	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	5	7	1
True values	0.3	2	5	2
Returned values	0.300004	2.000007	4.999992	2.000028
Error	0.000000		Iterative times	221

Table 4.37: Computation for Data with Different Values of ϵ_0 to Data 5

Data 7	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	4	9	0.5
True values	0.3	1	7	2
Returned values	0.300011	1.000048	6.999992	2.000070
Error	0.000000		Iterative times	311

Table 4.38: Computation for Data with Different Values of ϵ_∞ to Data 5

Data 8	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	5	7	1
True values	0.3	1	5	3
Returned values	0.300008	1.000002	4.999987	3.000024
Error	0.000000		Iterative times	433

4.9.2 Using Data Generated from the Model of KWW

There are five tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of KWW. The table also records the fitting error and the iterative times for each testing. There are five sets of data with respective to each table. They are Data 9, 10, 11, 12 and 13, generated by the model of KWW. Each of data 10, 11, 12 and 13 only has one parameter with different value compared to data 9 respectively.

Table 4.39: Computation for Data 9 with Method II

Data 9	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.1	8	0.51
True values	0.5	1	5	2
Returned values	0.499987	1.000021	5.000010	2.000004
Error	0.000000		Iterative times	291

Table 4.40: Computation for Data with Different Values of τ to Data 9

Data 10	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.9	9	1
True values	0.5	2	5	2
Returned values	0.500001	1.999956	4.999990	1.999996
Error	0.000000		Iterative times	242

Table 4.41: Computation for Data with Different Values of ϵ_0 to Data 9

Data 11	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.1	5	10	0.1
True values	0.5	1	7	2
Returned values	0.499994	1.000006	7.000074	1.999980
Error	0.000000		Iterative times	472

Table 4.42: Computation for Data with Different Values of ϵ_∞ to Data 9

Data 12	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	5	default	default
True values	0.5	1	5	1.7
Returned values	0.500008	0.999960	4.999962	1.699998
Error	0.000000		Iterative times	143

Table 4.43: Computation for Data with Different Values of β to Data 9

Data 13	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	5	7	1
True values	0.3	1	5	2
Returned values	0.299994	1.000026	5.000021	1.999983
Error	0.000000		Iterative times	196

4.10 Testing the System with the Method II and the Model of DC

There are five tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of DC. The table also records the fitting error and the iterative times for each testing. There are five sets of data with respective to each table. They are Data 14, 15, 16, 17 and 18, generated by the model of DC. Each of data 15, 16, 17 and 18 only has one parameter with different value compared to data 14 respectively.

Table 4.44: Computation for Data 14 with Method II

Data 14	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	default	3	default	default
True values	0.5	1	6	4
Returned values	0.499993	0.999973	6.000021	3.999984
Error	0.000000		Iterative times	160

Table 4.45: Computation for Data with Different Values of τ to Data 14

Data 15	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	1.5	default	default
True values	0.5	0.1	6	4
Returned values	0.499984	0.099999	5.999999	3.999960
Error	0.000000		Iterative times	226

Table 4.46: Computation for Data with Different Values of ϵ_0 to Data 14

Data 16	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	default	5	default	default
True values	0.5	1	8	4
Returned values	0.500000	0.999966	8.000000	3.999990
Error	0.000000		Iterative times	193

Table 4.47: Computation for Data with Different Values of ϵ_∞ to Data 14

Data 17	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.1	2.5	7	0.8
True values	0.5	1	6	2
Returned values	0.500008	0.999958	6.000025	2.000012
Error	0.000000		Iterative times	214

Table 4.48: Computation for Data with Different Values of γ to Data 14

Data 18	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	2.5	7.2	3.1
True values	0.3	1	6	4
Returned values	0.300014	0.999944	5.999995	4.000033
Error	0.000000		Iterative times	191

4.11 Testing the System with the Method II and the Model of HN

There are six tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of HN. The table also records the fitting error and the iterative times for each testing. There are six sets of data with respective to each table. They are Data 19, 20, 21, 22, 23 and 24, generated by the model of HN. Each of data 20, 21, 22, 23 and 24 only has one parameter with different value compared to data 19 respectively.

Table 4.49: Computation for Data 19 with Method II

Data 19	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.6	2.5	default	default
True values	0.3	0.7	1	6	4
Returned values	0.300001	0.699993	1.000030	5.999997	3.999999
Error	0.000000		Iterative times	501	

Table 4.50: Computation for Data with Different Values of τ to Data 19

Data 20	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.4	0.8	0.25	6.5	3.8
True values	0.3	0.7	0.1	6	4
Returned values	0.300984	0.691938	0.103732	5.998659	3.996584
Error	0.000000		Iterative times	638	

Table 4.51: Computation for Data with Different Values of ϵ_0 to Data 19

Data 21	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.6	1.5	8.1	2.7
True values	0.3	0.7	1	7	4
Returned values	0.303756	0.679036	1.100330	6.991092	3.992120
Error	0.000000		Iterative times	198	

Table 4.52: Computation for Data with Different Values of ϵ_∞ to Data 19

Data 22	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.22	0.56	1.5	6.5	1.5
True values	0.3	0.7	1	6	2
Returned values	0.300001	0.699997	1.000008	5.999995	2.000000
Error	0.000000		Iterative times	486	

Table 4.53: Computation for Data with Different Values of α to Data 19

Data 23	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.3	0.6	2.5	7.2	3.8
True values	0.5	0.7	1	6	4
Returned values	0.499995	0.700009	0.999977	6.000005	3.999999
Error	0.000000		Iterative times	363	

Table 4.54: Computation for Data with Different Values of γ to Data 19

Data 24	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.6	2.2	7	3.5
True values	0.3	0.6	1	6	4
Returned values	0.301965	0.589376	1.058080	5.997468	3.995791
Error	0.000000		Iterative times	644	

4.12 Testing the System with the Method II and the Model of FD

There are six tables. Each table is corresponding to a set of data and the testing using this set of data. Each table records initial values, true values and returned values by the system for all of unknown parameters for the model of FD. The table also records the fitting error and the iterative times for each testing. There are six sets of data with respective to each table. They are Data 25, 26, 27, 28, 29 and 30, generated by the model of FD. Each of data 26, 27, 28, 29 and 30 only has one parameter with different value compared to data 25 respectively.

Table 4.55: Computation for Data 25 with Method II

Data 25	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.4	0.9	default	default
True values	0.3	0.3	0.8	6	4
Returned values	0.300002	0.300005	0.800000	6.000000	3.999999
Error	0.000000		Iterative times	221	

Table 4.56: Computation for Data with Different Values of τ to Data 25

Data 26	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.45	0.3	0.9	6.5	3.8
True values	0.3	0.5	0.8	6	4
Returned values	0.299995	0.500057	0.800015	6.000000	4.000001
Error	0.000000		Iterative times	393	

Table 4.57: Computation for Data with Different Values of ϵ_0 to Data 25

Data 27	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	0.2	1	7.2	3.85
True values	0.3	0.3	0.8	7	4
Returned values	0.300929	0.293719	0.797305	6.999725	3.999916
Error	0.000000		Iterative times	121	

Table 4.58: Computation for Data with Different Values of ϵ_∞ to Data 25

Data 28	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.1	0.5	1	7.2	2.1
True values	0.3	0.3	0.8	6	3
Returned values	0.299999	0.300017	0.800007	6.000003	2.999999
Error	0.000000		Iterative times	568	

Table 4.59: Computation for Data with Different Values of α to Data 25

Data 29	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	0.2	0.6	7	2
True values	0.6	0.3	0.8	6	4
Returned values	0.599996	0.300014	0.800002	5.999991	3.999997
Error	0.000000		Iterative times	308	

Table 4.60: Computation for Data with Different Values of τ' to Data 25

Data 30	α	$\tau[msec]$	$\tau'[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	0.2	0.7	7.2	2.5
True values	0.3	0.3	0.6	6	4
Returned values	0.298957	0.308176	0.602587	6.000213	4.000076
Error	0.000000		Iterative times	162	

4.13 Testing the System Using the Data with Some Artificial Noise

This section uses some previous generated data with some artificial noise generated with Matlab function `randn()` to test the system for remedying the shortage of experimental data.

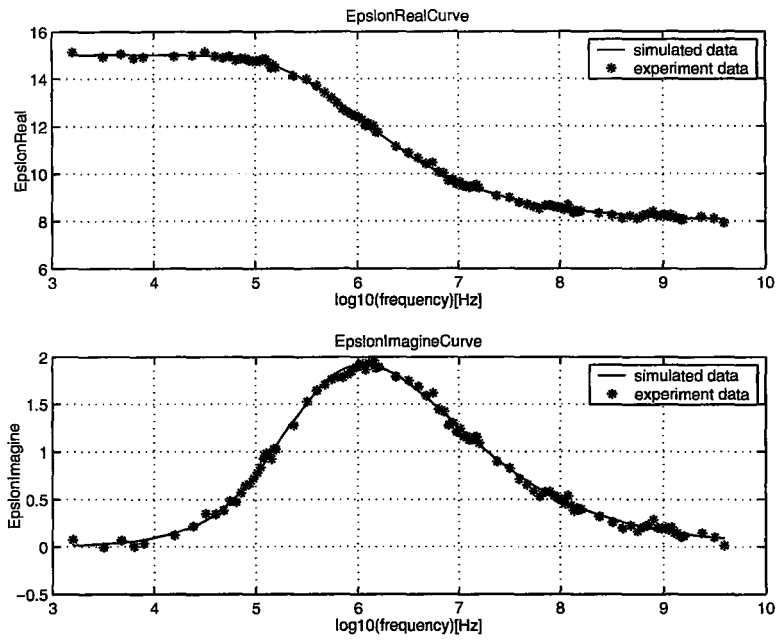


Figure 4.3: The Fitting Curve for the Table 4.61

4.13.1 Using Data 1 with Some Artificial Noise

Table 4.61: Computation for Data 1 with Some Artificial Noise and Method I

	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.4	0.2	13	7
True values	0.5	0.1	15	8
Returned values	0.499604	0.101681	15.014866	7.989935
Error	0.004698		Iterative times	28

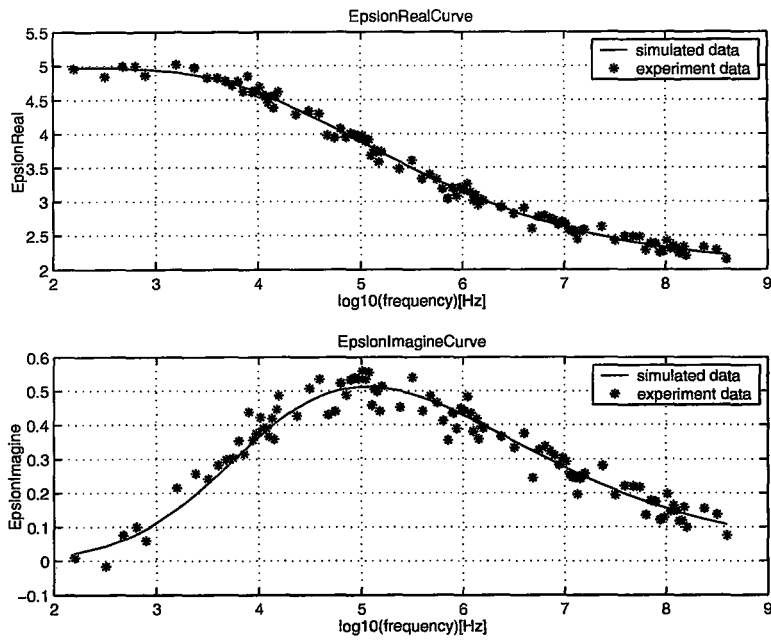


Figure 4.4: The Fitting Curve for the Table 4.62

4.13.2 Using Data 5 with Some Artificial Noise

Table 4.62: Computation for Data 5 with Some Artificial Noise and Method

II

	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.7	3	7	1
True values	0.3	1	5	2
Returned values	0.305919	1.002758	4.999919	2.020259
Error	0.003633		Iterative times	190

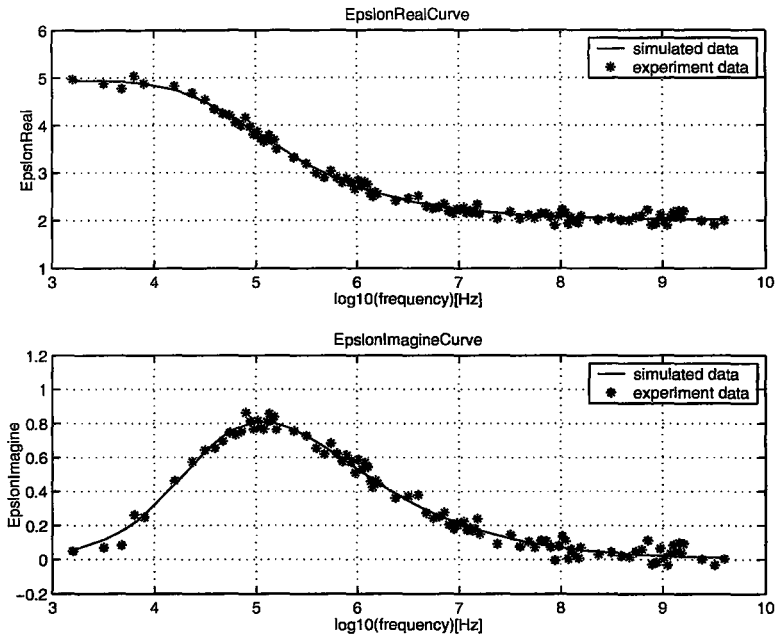


Figure 4.5: The Fitting Curve for the Table 4.63

4.13.3 Using Data 9 with Some Artificial Noise

Table 4.63: Computation for Data 9 with Some Artificial Noise and Method I

Data 9	β	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.5	1.05	5.1	default
True values	0.5	1	5	2
Returned values	0.501162	0.998743	4.948217	2.000990
Error	0.004090		Iterative times	28

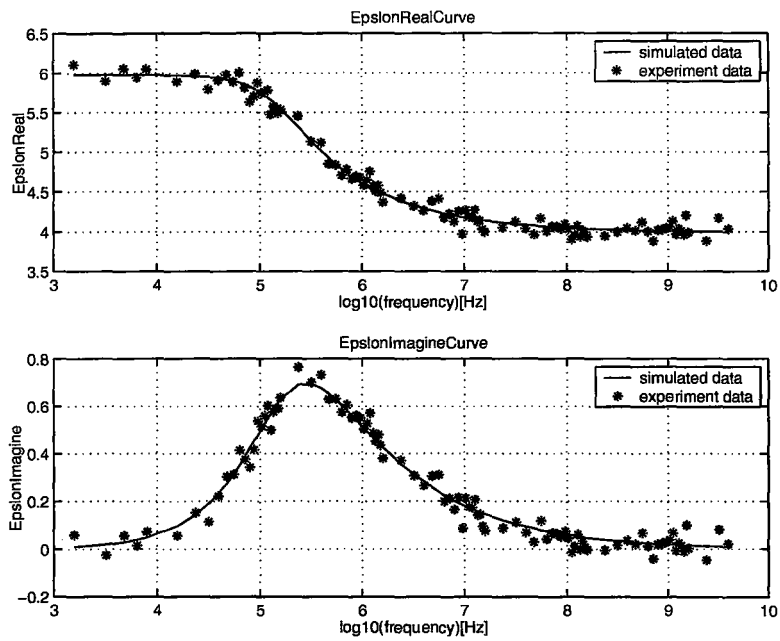


Figure 4.6: The Fitting Curve for the Table 4.64

4.13.4 Using Data 14 with Some Artificial Noise and Method I

Table 4.64: Computation for Data 14 with Some Artificial Noise and Method

I

Data 14	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	default	1.5	default	default
True values	0.5	1	6	4
Returned values	0.490221	0.991036	5.980375	3.988693
Error	0.004318		Iterative times	32

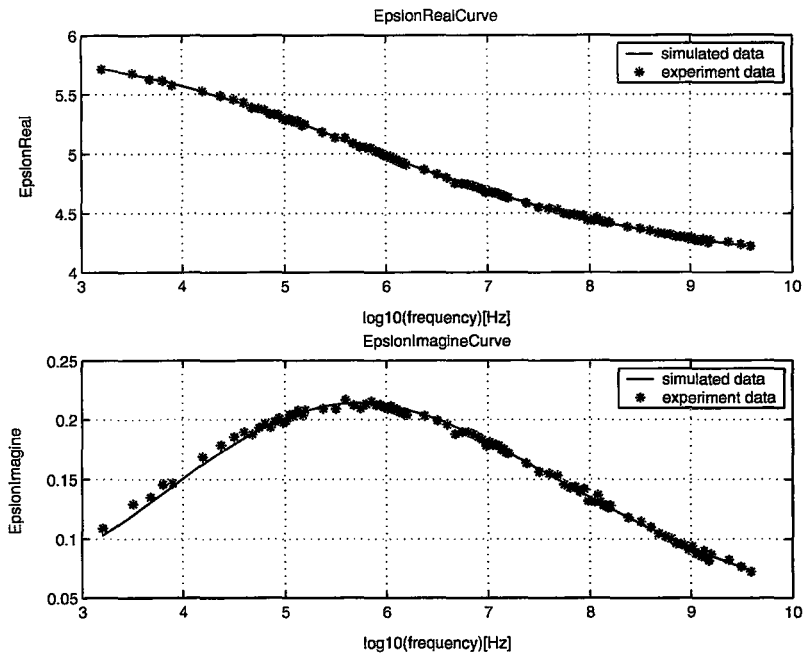


Figure 4.7: The Fitting Curve for the Table 4.65

4.13.5 Using Data 19 with Some Artificial Noise and method II

Table 4.65: Computation for Data 19 with Some Artificial Noise and Method

II

Data 19	α	γ	$\tau[msec]$	ϵ_0	ϵ_∞
Initial values	0.2	0.6	2.5	default	default
True values	0.3	0.7	1	6	4
Returned values	0.318903	0.627752	1.287524	5.960400	3.988909
Error	0.000040		Iterative times	182	

β	$\tau[msec]$	ϵ_0	ϵ_∞
0.5	1	8	4
0.3	1	8	4
0.5	0.1	8	4
0.5	1	6	4
0.5	1	8	3

Table 4.66: The Table Illustrating How Many Sets of Data Required

Data	Method I	Method II	Data	Method I	Method II
Data 1	4	335	Data 16	5	193
Data 2	2	300	Data 17	4	214
Data 3	4	193	Data 18	4	191
Data 4	6	272	Data 19	5	501
Data 5	3	244	Data 20	4	638
Data 6	4	221	Data 21	5	198
Data 7	3	311	Data 22	5	486
Data 8	3	433	Data 23	3	363
Data 9	1	291	Data 24	3	644
Data 10	3	242	Data 25	6	221
Data 11	3	472	Data 26	5	393
Data 12	3	143	Data 27	2	121
Data 13	4	196	Data 28	2	568
Data 14	2	160	Data 29	3	308
Data 15	4	226	Data 30	4	162

Table 4.67: Comparison of *iterativetimes* between Method I and Method II

Data	Method I	Method II	Data	Method I	Method II
Data 1	10.6200	143.6900	Data 16	0.1200	0.6900
Data 2	05.8500	121.2600	Data 17	0.0900	0.7800
Data 3	12.5500	108.8100	Data 18	0.1200	0.6800
Data 4	20.7100	134.7200	Data 19	0.4300	1.9900
Data 5	06.9200	103.0800	Data 20	0.1600	2.4000
Data 6	09.9800	095.8900	Data 21	0.1600	0.8400
Data 7	06.6700	114.4100	Data 22	0.1400	1.8500
Data 8	09.1000	151.1100	Data 23	0.1000	1.4200
Data 9	03.0100	118.1400	Data 24	0.0900	2.4300
Data 10	09.5500	127.7500	Data 25	0.4700	0.8700
Data 11	10.6700	178.8300	Data 26	0.1500	1.3800
Data 12	08.9000	070.3800	Data 27	0.0900	0.4700
Data 13	07.9300	077.0700	Data 28	0.0600	1.9500
Data 14	00.3200	000.6700	Data 29	0.0800	1.0800
Data 15	00.1100	000.7800	Data 30	0.1100	0.6000

Table 4.68: Comparison of *duration*[seconds] between Method I and II

Chapter 5

Conclusion and Future Work

This chapter will talk about the conclusion we can draw and some future work worth to do after the system is developed.

5.1 Conclusion

In this thesis, we have developed a software for data fitting and least square estimation of nonlinear parameters for models of dielectric relaxation data. The first thing we have done is developing four available theoretical models using Matlab software package. We utilize several techniques including scaling technique and decomposition technique to develop them precisely especially for the model of KWW including two parts, low frequency part and high frequency part, whose format is a sum of series.

After these four theoretical models are implemented, we implement two different methods to do the calculation for data fitting and least square estimation of nonlinear parameters by utilizing these four available models.

In order for a user to use the software, we also implement some modules as parts of the system to preprocess the input data, show the intermediate results, show the fitting curve and write the fitting result into a file.

To test the system, we implement another system to generate some input data which result is known such that we know how well the result returned by the system is when we do the testing using these data. We provide some hint in the thesis to tell the user how to choose the initial values for the unknown parameters. We also give the instruction or the procedure how to carry out an experiment for the user to use this software.

And we have tested the system using thirty sets of the input data, which results are known. The testing results are shown as sixty tables since

we used two different methods to do the calculation. It is shown that both of methods work very well with the appropriate initial values because both of methods are iterative methods. The fitting error for most of them is less than 10^{-6} .

The developed software eliminates the limitation of testing equipment such as the limited frequency range by doing the data fitting and returns values for some unknown parameters. The software is very helpful for researchers doing the dielectric relaxation research. It saves much time for them to gain the values of some unknown parameters. This developed software can be used anywhere where four available theoretical models, the model of DC, the model of KWW, the model of HN and the model of FD, are applied to.

As far as I know, the developed tool will be the first tool for researchers to get the values of parameters for the chosen model for dielectric relaxation phenomena and also for the other phenomena where four available theoretical models are applied to such as mechanical relaxation phenomena.

The idea in this thesis not only could be applied to dielectric relaxation and mechanical relaxation but also could be applied to the other mathematical models where several unknown parameters need to solve in the research. This will save a little time for researchers to handle their data. The first thing to apply this idea is to implement the mathematical models as accurately as possible while the other thing to use some minimizers to calculate the unknown parameters by applying least square algorithm.

5.2 Future Work

In chapter 4, we only used the simulated data to test the system due to the lack of real data. One of future works that we may consider is to use the real data to test the system before the system is deployed although some of simulated data are generated from National Bureau of Standards. It is clear that, in an experimental context in science or engineering, almost all measured quantities have an error because a perfect experimental apparatus does not exist [24]. If the system is used to handle the experimental data with error, how to handle these errors or noise may become future work. For example, one way to handle them is to use these errors to weight each term in the sum of the squares. It should be a very interesting and also important part for future works.

Another work is about the issue regarding to the local minimum. There is not much discussed how to choose the initial values in this work. It might be worth to do further research so that the user know what kind of criteria that initial values should meet to let the system work otherwise the system would not work.

The least squares method is the most commonly used algorithm. However, we used the sum of squares to evaluate the goodness of fit. Although a smaller sum of squares means a better fit, in many cases, it may be more popular that confidence intervals would be used to characterize the goodness of fit. It is worth to search better way to evaluate the goodness combining with visualization such as fitting curves.

We used Matlab to implement the system. It is also worth to implement a stand alone system such that it can be used in any machine no matter whether there is Matlab package or not. This may need to know how to use Matlab compiler and C compiler and how to let the system be available on the Internet.

It is also a part of future works how to re-weight data if necessary. The developed system is to do nonlinear fits. We should not force the system to re-weight data because it can cause the value of the parameters to which we are fitting to be changed. How to provide the option and when the user should use the option may be interesting.

It can be very interesting topic to explore the relationship of parameters of different models such that the user can get the values of parameters for the other model if the user knows the values of parameters of one model by providing mapping curves.

That is all for the future work that I can imagine for this developed system.

Bibliography

- [1] *The Oxford Paperback Dictionary* fourth edition, edited by Elaine Pollard in 1994

- [2] R.Hilfer *Fitting the excess wing in the dielectric α -relaxation of propylene carbonate* Journal of Physics: Condensed Matter 14 (2002) 2297-2301

- [3] D.L Misell, R.J. Sheppard *the application of deconvolution techniques to dielectric data* J.Phys.D: Appl.Phys., Vol.6,1973

- [4] C.T. Moynihan, L.P. Boesch and N.L.Laberge *Decay function for the electric field relaxation in vitreous ionic conductors* Physics and Chemistry of Glasses Vol. 14 No. 6 December 1973

- [5] Michael F. Shelesinger and Elliott W. Montroll *On the Williams-Watts function of dielectric relaxation* Proc. Natl. Acad. Sci. USA Vol.81. pp. 1280-1283, February 1984

- [6] Andrzej J. Surpwiec, Stanislaw S. Stuchly, J. Robin Barr and Arvind *Dielectric Properties of breast carcinoma and the surrounding tissues* IEEE transactions on biomedical engineering. Vol. 35, No. 4 April 1988

- [7] Pal K. Dixon, Lei Wu, and Sidney R. Nagel, Bruce D. Williams and John P. Carini *Scaling in the Relaxation of Supercooled Liquids* Volume 65, Number 9, Physical review letters 27 August 1990
- [8] S Gabriel, R W Lau and C Gabriel *The dielectric properties of biological tissues: III. parametric models for the dielectric spectrum of tissues* Phys. Med. Bill. 41 (1996) 2271-2293
- [9] E. Tuncer and S. M. Gubański *On Dielectric Data Analysis Using the Monte Carlo Method to Obtain Relaxation Time Distribution and Comparing Non-linear Spectral Function Fits* IEEE Transactions on Dielectrics and Electrical Insulation Vol. 8 No.3, June 2001
- [10] N Axelrod, E Axelrod, A Gutina, A Puzenko, P Ben Ishai and Yu Feldman *dielectric spectroscopy data treatment: I. Frequency domain* Institute of Physics publishing, Measurement science and technology, Meas. Sci. Technol. 15 (2004) 755-764, PH: S0957-0233(04)69473-2
- [11] Enis Tuncer and J. Ross Macdonald *Comparison of methods for estimating continuous distributions of relaxation times* arXiv:cond-mat/0504428 v1 18 Apr 2005
- [12] Dwayne Andrew Wasylshyn *The synthesis and properties of three new network polymers studied by calorimetric, dielectric and mechanical relaxation methods* McMaster University. Dept. of Materials Science and Engineering. 1995

- [13] Menachem Dishon, George H. Weiss and John T. Bendler *Stable Law Densities and Linear Relaxation Phenomena* Journal of Research of the National Bureau of Standards Volume 90, Number 1, January-February 1985
- [14] B. Curry, Quart *Appl. Math.*, 2 (1994), pp. 258-261
- [15] Donald W. Marquardt *an algorithm for least squares estimation of non-linear parameters* J. Soc. Indust. Appl. Math. Vol. 11, No. 2 June, 1963
- [16] Spencer Smith *MeshGenMG.pdf* CAS781, Advanced Topics in Computer Science, Department of Computing and Software, McMaster University, January, 2006
- [17] *threshold. (n.d.). The American Heritage Dictionary of the English Language* Fourth Edition. Retrieved June 10, 2007
- [18] Andrew K Jonscher *Review article dielectric relaxation in solids* J.Phys. D: Appl. Phys. 32(1999)R57-R70. Printed in the UK
- [19] N.G. McCrum, B.E.Read and G.Williams *Anelastic and Dielectric Effects in Polymeric Solids* John Wiley and Sons Ltd. 1967
- [20] R.Hilfer *Analytical representations for relaxation functions of glasses* Journal of Non-Crystalline Solids 305 (2002) 122 -126
- [21] <http://netlib3.cs.utk.edu/opt/dqed.f>
- [22] http://www.cas.mcmaster.ca/~cs4te3/notes/s_intro.pdf

[23] <http://mathworld.wolfram.com/HornersRule.html>

[24] *PHY307F/407F - computational Physics*