John C. Nash
Faculty of Administration
University of Ottawa

Mary Walker-Smith
General Manager
Nash Information Services Inc.

NONLINEAR PARAMETER ESTIMATION:

an Integrated System in BASIC

1975 Bel Air Drive, Ottawa, Ontario, K2C 0X1

ISBN 0-88769-009-2

Originally published in 1987 by Marcel Dekker Inc., New York,

under ISBN 0-8247-7819-7

This edition of **Nonlinear Parameter Estimation: an Integrated System in BASIC** is made available electronically in order to provide the material to interested readers. The text has been kept as close to the original as possible, except for correction of known errors and minor adjustments to fit the dual-page format chosen to minimize paper use.

We can be contacted as follows:

Electronic mail to John C. Nash at jcnash@uottawa.ca
or Mary Nash at mnash@nashinfo.com
Fax: (613) 225 6553
Telephone: (613) 225 3781 [Note: this line has an answering machine. We do NOT return long-distance calls nor accept collect calls except where there are prior arrangements.]

Electronic mail is our preferred method of communication.

John C. Nash, August 1995

TABLE OF CONTENTS

PREFACE

This book and software collection is intended to help scientists, engineers and statisticians in their work. We have collected various software tools for nonlinear parameter estimation, along with representative example problems, and provided sufficient "glue" in the form of procedures, documentation, and auxiliary program code to allow for relatively easy use of the software system for nonlinear parameter estimation.

We use the expression "relatively easy use", because nonlinear parameter estimation has many traps for the unwary. The problems are, in general, an order of magnitude more difficult or effort-consuming to solve than linear parameter estimation problems. Because nonlinear parameter estimation is related to the solution of nonlinear function minimization problems and to the solution of sets of nonlinear equations, the software we present may also be applied in those areas.

In our presentation, we have tried to provide enough background to the methods to allow for at least an intuitive understanding of their working. However, we have avoided mathematical detail and theorem proving. This is a book about mathematical software and about problem solving rather than mathematics. In partial compensation for this deliberate restriction of the scope of our treatment, we include a quite extensive bibliography. [We had originally intended to update this bibliography on a periodic basis and distribute it on diskette for a small charge. The costs of doing this have prevented us from its execution. However, we remain interested in bibliographic information on nonlinear estimation and hope to eventually provide an Internet form at modest cost to users. In the interim, we welcome contact at the electronic mail addresses on the frontespiece.]

The methods included have been developed over the last 25 years by a number of workers. The implementations presented have been modified over the past 15 years for use on "small" computers. In fact, the IBM PC and compatible machines are quite large by comparison with the earlier computing environments for which the programs were tailored. Despite the availability of quite high-speed personal computers with large random access memories, there are many reasons why compact programs continue to be useful, for example:

- less code to debug
- faster consolidation / compilation
- possibly faster execution
- less disk overhead for storage of code
- more memory for user problem code.

We have aimed to make this collection of software easy to use and to adapt to a variety of situations. Therefore functionality has been a major requirement, as has reliability. Speed has been secondary. If computing time is critical, users may wish to modify the codes to tune their

performance. However, it is quite possible that reliability may be sacrificed in this process. As it is, one of our major worries in presenting this code is that there remain undiscovered weaknesses which a variety of test problems and multiple perusals of the code have failed to reveal. We respectfully request users who discover such weaknesses to communicate them to us so that we may compile (and eventually distribute) corrections or warnings.

To extend the already quite wide-ranging set of examples which are presented in the following pages, we also are preparing a collection of nonlinear parameter estimation and function minimization problems, presented with program code and (partial) solutions as in the present work. This collection, entitled "Nonlinear parameter estimation: More Examples", is available from the second author at the address above for $25 (prepaid). Also included are the abbreviated function minimization codes.

In preparing this work, we have a number of people to whom we wish to express our gratitude:

to Stephen Nash, for private communication of various important ideas, and especially a version of the truncated Newton method

to Don Watts and Doug Bates, for sharing ideas which helped us to focus on the software and its use

to Mary Nash and Neil Smith, for reading our drafts and living with our enthusiasm for sequences of numbers appearing on a screen

to Lucy Yang, for a portion of the text entry of the early drafts of the book.

The Corona PC-21 microcomputer and some of the computer supplies used in the research and preparation of this work were paid for from research grants from the Natural Sciences and Engineering Research Council of Canada.

NOTATION

We will write any function to be minimized as f(B), where the vector B represents the n parameters

$$(0\text{-}3\text{-}1) \qquad \underline{B} = (B(1), B(2), ..., B(n))^T$$

though in most real-world situations there will be exogenous data, which for convenience will be stored in a matrix Y. We could write our function (and constraints) using this matrix i.e. f(B,Y). However, this tends to clutter the page and lead to confusion wherever the ",Y" is inadvertently omitted, so we will mostly use the simpler form. Furthermore, we will assume users have taken care to choose units for data so that the function evaluation is reasonably scaled, as we shall discuss in Chapter 3.

We shall strive to be consistent in our use of labels for variables and arrays within the book. Moreover, we shall try to carry over the same notations to computer programs so that the usage is readily accessible for modification and application.

1

OVERVIEW AND PURPOSE

1-0. Beginnings

This introduction is intended to set out the background and history of nonlinear modeling, to provide a statement of the goals and purposes of this book, and to present a notational framework for describing nonlinear estimation problems and solution methods.

1-1. History and Motivations

The development of science and technology has chronicled improvements in our description of natural phenomena, particularly description in the form of mathematical models -- equations and inequalities which allow us to predict the evolution and properties of the phenomena of interest. Such mathematical models relate diverse observable or conceptual variables which together translate into a description of the phenomenon itself. The mathematical forms of the relationships are called functions, with a function being the (many-to-one) mapping of the values of several variables onto the single value which is that of the function. Moreover, the functions themselves involve parameters which give to the model its scale, rates of change, units of measurement, and other aspects which distinguish an individual occurrence of a phenomenon from all similar events or systems.

In talking about nonlinear models, we are referring to the kind of equations and inequalities from which a mathematical model is made. Specifically, we refer to the manner in which the parameters enter into the equations or inequalities and/or the difficulties this form of appearance imposes on the scientist wishing to estimate the parameters. We shall expand on this topic below.

Early mathematical models of reality were frequently centered on the physical world. For example, Kepler’s Laws of Planetary motion state:

1. The orbit of a planet is an ellipse with the sun at one focus.

2. The radius vector from the sun to the planet sweeps over equal areas in equal time intervals.

3. The squares of the periodic times are proportional to the cubes of the major axes.

Sommerfeld (1964, pages 38-44) gives the classical mechanics derivation of these "laws," but for our purposes it shall suffice to re-phrase the last law as an equation:

$$T^2 = K\ a^3 \qquad (1\text{-}1\text{-}1)$$

where T is the period of the orbit (time for one complete trip around the sun), a is a measure of the size of the ellipse of the orbit, and K is a constant which relates T and a.

Since a appears to the power 3 and T to the power 2, the relationship between T and a is nonlinear, that is, T cannot be expressed as directly proportional to a to the power 1. The parameter K does, however, appear to the power 1 in the model, and the estimation of K could be carried out with calculations

requiring the solution of linear equations.

Early scientists sought to find "natural laws" which were universally applicable. The parameters and equations of such laws appear from the present perspective to have been regarded as exact. The consideration of errors or fluctuations in observations, which would in the course of calculations alter the values of parameters, is generally attributed to Gauss (1809). Furthermore, Gauss astutely chose to presume the errors in his observations to be distributed according to the so-called Normal distribution, a name statisticians now prefer to replace with "Gaussian distribution" in respect for its inventor and in the hope that a term which is misleading to the untrained reader may gradually drop from usage. The Gaussian distribution of errors, when superposed on a mathematical model, allows us to find the set of parameters which maximizes the probability that the actual observations would be seen. In the case where the model is a linear function of the parameters, that is, each parameter occurs to the power 1 and does not multiply or divide any other parameter, such maximum likelihood parameter estimates can be found by solving a set of linear equations, as shown in Neter and Wasserman (1974, pages 48-49).

Many models of reality use functional forms to relate variables which are linear in the parameters. The most widely used of these is the multiple linear regression model, relating variable y with the varables x(i), i = 1, 2, ..., n-1.

$$y(t) = \sum_{i=1}^{n-1} B(i)\ x(t,i) + B(n) + error(t) \qquad (1\text{-}1\text{-}2)$$

where y(t), x(t,i) and error(t) are the respective values of variable y, variable x(i) and the error at observation t. Parameter B(n), which in some notations multiplies an artificial variable whose observations are all 1, is called the constant. Maximizing the probability of observing specific data for a set of observations t=1,2,...,m under the assumption that the errors are Gaussian distributed with zero mean and variance sigma^2 (read "sigma squared") requires us to find the maximum of the likelihood function for the Gaussian distribution

(1-1-3) $$f(\underline{B}) = \prod_{t=1}^{m} \exp(- error(t)^2) / (2\ sigma^2)$$

where

(1-1-4) $$\underline{B} = (B(1), B(2), \ldots , B(n))^T$$

For convenience, we have left out the normalizing constant in Equation (1-1-3), which does not change the result. The maximization of the function in Equation (1-1-3) is obtained by noting that the logarithm of f(B) takes on its maximum where f(B) does. Thus the probability of observing y and x(i), i=1,2,...,n at their data positions is maximized by minimizing the sum of squared errors -- hence the concept of least squares.

(1-1-5) $$\text{LS: minimize } S(\underline{B}) = \sum_{t=1}^{m} error(t)^2 \quad \text{w.r.t. } \underline{B}$$

Minimizing the sum of squared errors for the linear regression model involves only functions which are linear in the parameters B. This can be accomplished via the solution of a set of simultaneous linear equations, a task which we shall consider as "easy" from our current perspective, despite the variety of obstacles and pitfalls to obtaining good parameter estimates (see Nash, 1979, pages 37-39,53-58, for a discussion of some of these difficulties).

Our present concern will be with models which do not

permit estimation via the solution of a set of simultaneous linear equations. Thus we shall generally not be interested in models which can, by suitable transformations, be estimated using linear equations. For example, the model

(1-1-6) y(t) = B(1) + exp(-B(2) + x(t)) + error(t)

which presents the parameter B(2) within a transcendental function can be transformed to the model

(1-1-7) y(t) = B(1) + B'(2) exp(x(t)) + error(t)

where

(1-1-8) B'(2) = exp(- B(2))

which requires only linear equation solving to estimate B(1) and B'(2). B(2) is then recovered as

(1-1-9) B(2) = - log(B'(2))

On the other hand, the linear regression model (1-1-2) can be rendered highly nonlinear in the parameters by the imposition of the constraint

(1-1-10) B(2) B(5) = B(3) B(4)

For a problem of this type having n = 6 parameters and m = 26 observations, see Nash (1979, pages 184-186).

Gauss, despite the very limited computational tools available, recognized the difficulties of nonlinear parameter estimation and described a variant of the Newton-Raphson iteration to attempt a solution. This method, called the Gauss-Newton method, is at the foundation of a number of the more common techniques used today for the nonlinear estimation problem. Its failings are primarily in situations where the errors (or residuals) at the minimum sum of squares are still large. Gauss himself was quite clear on this point -- he only envisaged using the method for problems where the errors were relatively small compared to the quantities measured.

To conclude this section, we formalize our definition of "nonlinear parameter." In order to do this, we first state that a function Z(B) which is linear in the parameters B(1), B(2),...,B(n) to be one where each term in the function has at most one parameter, and each parameter appears only to the first degree, that is, to the first power. This means that Z(B) can be written as a linear combination

$$(1\text{-}1\text{-}11) \qquad Z(\underline{B}) = z_0 + \sum_{i=1}^{n} z_i\, B(i)$$

where the values of the numbers

$$(1\text{-}1\text{-}12) \qquad z_i \;,\; i = 1, 2, \ldots, n$$

define the function. A linear equation results by setting Z(B) to zero. In their forthcoming book on nonlinear least squares problems, Bates and Watts (1986) consider the term "nonlinear" to imply that the gradient of the modelling function is not constant. We will consider that the parameters B are nonlinear if the equations which must be solved or functions which must be minimized in order to estimate their values cannot be reduced to a set of simultaneous linear equations. This is closely related to the Bates/Watts definition, but will encompass estimation problems wider in scope than nonlinear least squares.

In some examples, we may have a choice of estimation techniques. A set of parameters which is linear under one choice may nevertheless be nonlinear under another. It is the linearity of the equations used to calculate the estimates which will govern our definition here.

Some problems have a number of parameters which can be considered linear if other parameters are fixed. For example, if we wish to model some phenomenon by a simple exponential growth model

$$(1\text{-}1\text{-}13) \qquad Z(B(1), B(2), Y(i,2)) = B(1)\exp(\, B(2) * Y(i,2)\,) \quad \text{for } i = 1,2, \ldots, m$$

then B(1) can be estimated by solving a linear equation if B(2) is fixed. That is, we could simply average over the observations to give

$$(1\text{-}1\text{-}14) \qquad \sum_{i=1}^{m} Y(i,1) = B(1) \sum_{i=1}^{m} \exp(B(2) * Y(i,2))$$

(Lawton and Sylvestre, 1971). Such problems have interested a number of researchers whose work has attempted to improve the efficiency of algorithms to estimate parameters by using the special structure (see Section 14-3).

1-2. Purposes

This book is an attempt to provide a software collection for the estimation of nonlinear parameters in mathematical models which arise in diverse scientific disciplines. We shall not restrict our attention to any particular area of application. Indeed, there are applications of nonlinear models in practically every area of study -- wherever people attempt to codify their understanding of the operation of a system or the evolution of a phenomenon by means of a mathematical model, the requirement that parameters be estimated may arise. In the situation that these parameters are nonlinear in the sense described in Section 1-1, the ideas to be discussed in this book may be useful.

In what way will our work be useful to others? Clearly, we cannot discuss all aspects of parameter estimation in a volume of this size. Indeed, we have no intention of creating an encyclopaedic masterwork on the subject. Our goal is more modest: to provide researchers in many disciplines with an approach and with the software tools to solve nonlinear parameter estimation problems using small computers.

First, the approach that we propose to nonlinear parameter estimation is via the minimization of loss functions. That is, it is presumed the researcher has some measure of "goodness" of his model and, given two sets of parameters, can state which is "better" from the point of view of this measure or loss function. We fully expect users to want to alter such loss functions to explore different aspects of their problem:

- statisticians may wish to impose different distributional assumptions on the unknown errors in their models, thereby altering the types of estimators and consequent estimation problems;
- chemical engineers may want to hypothesize alternative chemical reaction paths in the model of a reactor to produce an industrial chemical;
- biologists may want to choose different environmental constraints for a population of animals or plants under study.

Within our approach are certain simplifications and precautions. For example,

- we suggest the use of certain checking programs to verify the operation of program code written by users;
- we urge but do not insist, that users scale their parameters and functions to avoid numbers of widely different magnitude in calculations;
- we suggest that crude estimation methods be used to gain insight into the effects of changes in the parameters on the loss functions used, as well as to impose constraints and develop feasible sets of parameters. Some of these methods have fallen into disuse on large mainframe computers because they are inefficient in finding estimates of nonlinear parameters if used as general tools. However, in the context of

small computers, they are useful for explorations of the problem.

"Small computer" is a term which is in need of definition. We shall use it to refer to a personal computer or segment of a larger machine which allows interactive problem solving with easy access to program and data files. In this book we present all programs in BASIC, since most microcomputers either come equipped with a BASIC interpreter or one can be bought for them. Furthermore, many minicomputers and mainframes have BASIC sub-systems with similar facilities. We prefer to discuss programs operating in an interpretive environment for the type of parameter estimation tasks in this book. This is because the ability to quickly modify programs, print variables or array elements and, in some systems, change values and continue execution of a program is highly desirable in this particular context. For "production" tasks, or where the problem is so large that an interpreted program is simply too slow and inefficient, compiled versions of the software are more appropriate. In such cases, however, other programming languages and a somewhat different approach may be more suitable. (Depending on the response to the software presented here, we may undertake versions in other programming languages. Turbo Pascal is a likely environment for early consideration.)

The second goal of this book is to present the software tools to actually carry out the estimation tasks. Primarily, these tools are presented as BASIC "subprograms" which minimize nonlinear functions -- either general or specialized to sums of squares -- and which may or may not require derivative information to be provided by the user.

The third goal is to present the rudiments of the software "glue" to integrate the tools above into an easily used system for solving estimation problems. The individual estimation programs are useful and the core of our approach, but they do not lend themselves to easy use by themselves. Extra programs or subprograms are included to aid the user in related tasks such as

- to check program code for the loss function or its derivatives
- to incorporate bounds constraints on parameters
- to plot two-dimensional data
- to determine the computational environment
- to perform postestimation analysis to determine the sensitivity of the parameters.

We do not provide tools for the entry and edit of data, since there are many good, inexpensive tools for this purpose on most small computers.

The fourth goal is to document both the approach and software in a unified and consistent way. Since BASIC uses global variable and array names, this part of our work is extremely important -- should a user employ a variable name already storing a key control value, our software will not work as anticipated and the results may be totally wrong. In this regard, the choice of interpreted BASIC can lose a great deal of the benefit gained in the ease of development of function and derivative subprograms. However, we feel this benefit outweighs the disadvantage of potential variable name confusion. For convenience, we have chosen to work in the Microsoft flavor of BASIC. Although this allows us to use relatively long variable names, we shall restrict ourselves to a single letter or a single letter plus a single digit in conformance with the ISO Minimal BASIC standard (ISO 6373-1984). By so doing, we create a few headaches for ourselves in preparing and testing the programs here, but allow many more users to adapt the software to their own particular

computing environments. Appendix D presents details of the choices we have made.

Our four goals are thus summarized as

1. approach
2. software tools
3. system design and prototype programs
4. documentation of methods and programs

We anticipate that most readers of this book will use it in one of three ways, which represent differing levels of use:

1. as an introduction to nonlinear estimation for non-specialists
2. to learn/use a particular tool
3. to build their own nonlinear parameter estimation system or to integrate our methods with another (statistical or database) package.

Having presented our intentions, it is also pertinent to list some of the aspects of nonlinear parameter estimation we will NOT address. We do not intend to make more than passing reference to specialist techniques in particular disciplines. There are, in fact, many special models for which highly efficient and well-established estimation methods have been derived. However, these do not, we feel, fit well into the development of a general parameter estimation package for small computers. The difficulty is not one of the length or complexity of program code to include a number of such methods -- it is the amount of effort to properly explain and document such a compendium of tools which may have only a limited range of applicability and small target audience.Nor do we intend to discuss at any length the foundations of the statistical theory upon which estimation methods are based. This is an important subject area, but we shall focus our effort on solving problems after they have been posed, rather than developing the formulation from first principles.

1-3. Classic Problems

While Chapter 2 introduces more examples of nonlinear parameter estimation problems, we wish to point out at this very early stage that these tasks are well-established throughout science and technology. Some of the particular cases which have developed a large literature are those of finding models to describe:

- the relationship between plant yield and density of planting;
- the change over time of the size of plants, animals or economic activity -- growth curves;
- the rate of production or disappearance of a chemical or radio-isotope in a chemical or nuclear reaction.

Examples of data for such situations are:

- the data for growth of White Imperial Spanish Onions at Purnong Landing given by Ratkowsky (1983, page 58) and presented in Table 1-3-1 and plotted in Figure 1-3-1;
- the weed infestation data discussed in Section 2-1 and presented in Table 2-1-1 (next chapter) and plotted in Figure 2-1-1;
- data for the decomposition of methyl iodide in alkaline solution given by Moelwyn-Hughes (1961, page 1253) and presented in Table 1-3-2 and plotted in Figure 1-3-2. This problem is discussed in more detail in Section 16-5.

Many of the examples in this book arise in these "classic" tasks, but we shall introduce problems from other situations. Since one of our major goals is to provide software, we shall also present easily formulated test problems to allow methods to be compared. Readers may note that our plotted curves often have a very simple structure. Most, but not all, have been created with the program PLOT.BAS which is described and listed in Chapter 15. While more sophisticated plots could have been used for presentation in this book, we prefer to show the reader the utility of simple graphs which can be created within an inexpensive and commonly available computing environment.

Table 1-3-1. Yield versus density for White Imperial Spanish Onions at Purnong Landing

X = density	Y = yield	X = density	Y = yield
23.48	223.02	70.06	116.31
26.22	234.24	70.45	120.71
27.79	221.68	73.98	134.16
32.88	221.94	73.98	114.48
33.27	197.45	78.67	91.17
36.79	189.64	95.90	101.27
37.58	211.20	96.68	97.33
37.58	191.36	96.68	101.37
41.49	156.62	101.38	97.20
42.66	168.12	103.72	87.12
44.23	197.89	104.51	81.71
44.23	154.14	105.68	76.44
51.67	153.26	108.03	87.10
55.58	142.79	117.82	84.54
55.58	126.17	127.21	69.09
57.93	167.95	134.26	64.40
58.71	144.54	137.39	66.81
59.50	151.30	151.87	63.01
60.67	130.52	163.61	55.45
62.63	125.30	166.35	62.54
67.71	114.05	184.75	54.68

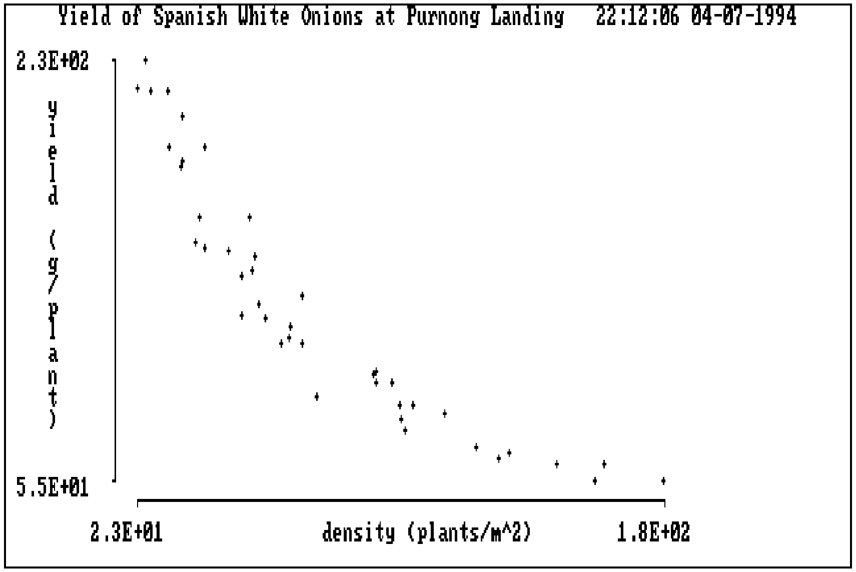

Figure 1-3-1. Plot of the "ONION" data set in Table 1-3-1.

Table 1-3-2. The fractional decomposition of methyl iodide in aqueous alkaline solution at temperature 342.8 K

t=time in minutes	observed percent completion
0	0
3.5	16.8
8.5	36.0
13.5	50.7
19.5	64.3
26.5	75.2
37.5	85.1
infinity	100.0

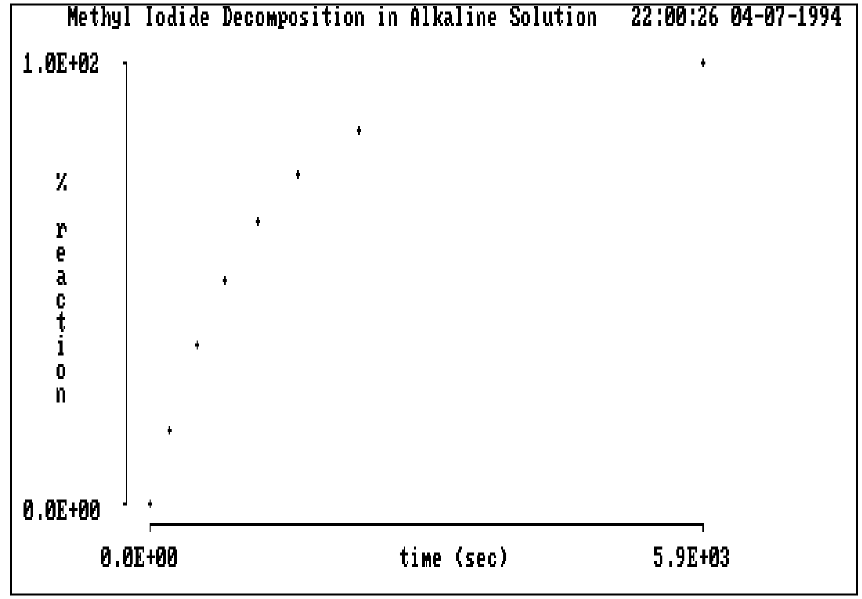

Figure 1-3-2. Plot of the percent completion of decomposition of methyl iodide with time.

1-4. Landmarks in Nonlinear Estimation Methods

This section presents a brief historical overview of the topic of nonlinear estimation. We make no claim to being exhaustive, and wish only to provide a perspective for the material we will present in later chapters.

Isaac Newton provided scientists and mathematicians with much food for thought. His developments in the calculus were not only theoretical, and from the late 17th century to the beginning of the 19th century, Newton's method (also referred to as the Newton-Raphson method) was the only approach to the solution of nonlinear equations, and hence of parameter estimation problems. However, it cannot be said that there are many examples of such problems being solved. It is only when Gauss and others, taking advantage of the developments in instrumentation which accompanied the industrial revolution, began to calculate orbital parameters for planets and other astronomical objects that we see attempts made to develop the computational techniques. Gauss (1809), by recognizing that the calculations for the Newton method is simplified if the residuals (model minus observation) were small, developed the Gauss-Newton method for nonlinear least squares problems.

In large measure, it can be stated that most function minimization methods which use the gradient of a function still owe much to Newton, and most nonlinear least squares methods still find their origins in the Gauss-Newton method. The developments which have occurred have mainly improved the Newton and Gauss methods in

- increasing their ability to obtain correct solutions when presented with poor guesses for the parameter estimates, since initial values of the parameters must be provided so that the iteration can proceed;

- reducing the computational effort required within the Newton and Gauss-Newton methods either in the calculation of derivatives or in the solution of the iteration equations;

- reducing the storage requirements of the methods, that is, the amount of information needed to operate them.

Particular developments of note in the progress of nonlinear estimation techniques follow.

Cauchy (1848) suggested the steepest descents method, that is, an iteration wherein the gradient is used as a

search direction, the loss function is reduced along the gradient direction, and the process is repeated from a new (lower) point. In practice, this is NOT a good method in general, but incorporating elements of it can be useful for ensuring progress of a method. Thus, many of the approaches we shall suggest will make use of occasional explicit or implicit steepest descent steps.

Levenberg (1944) suggested that the iteration equations for the Gauss-Newton method could be stabilized by modifying the iteration equations to include a steepest-descent component. While Levenberg's work was largely ignored by practitioners, its rediscovery by Marquardt (1963) and the distribution of a computer program implementing the ideas were perhaps the biggest practical contribution to nonlinear parameter estimation to date.

A somewhat different approach was used by Hartley (1961), who modified the Gauss-Newton method by treating the solution of the iteration equations as a search vector (see Chapter 11).

From the point of view of function minimization, as opposed to nonlinear least squares, the developments of digital computers prompted workers to try direct search methods for function minimization. Methods were proposed by, among others, Rosenbrock (1960), Hooke and Jeeves (1961) and Box (1957). Of these methods, the Hooke and Jeeves method survives almost in its original form and is presented in Chapter 5, while the "simplex" method of Box was improved by Nelder and Mead (1965) to a form similar to that which appears in Chaper 6. Other methods, such as that of Rosenbrock, appear to have fallen into disuse.

Gradient methods have also been improved to provide efficient techniques which avoid the computational effort of Newton's method and the slow convergence of Steepest Descents. Davidon (1959) showed how to build up the Newton solution from a sequence of gradient steps. Thus the work at each iteration of his "variable metric" method was comparable to that of Steepest Descents but convergence was more like that of Newton’s method. Fletcher and Powell (1965) published what has become one of the most popular algorithms of this type. In Chapter 10 we use a variant by Fletcher (1970).

Variable metric (or quasi-Newton) methods still require a lot of data storage to operate, and for functions in a large number of parameters we would still be required to use steepest descents if Fletcher and Reeves (1964) had not noticed that ideas from iterative approaches to the solution of linear algebraic equations could be applied to nonlinear minimization. The resulting conjugate gradients method, modified and adapted, appears in Chapter 8. A further refinement along somewhat similar lines resulted in the truncated-Newton method (S.G.Nash, 1982; Dembo et al., 1982). See Chapter 9.

The nonlinear least squares problem has received special attention from time to time. Of note is the program NL2SOL of Dennis, Gay and Welsch (1981), which uses the secant method. While the length and complexity of this code mean that we have not considered including it here, it should be kept in mind by those working in FORTRAN environments on larger computers.

An important area of research has been the attempt to understand the statistical meaning and consequences of nonlinearity in models and loss functions. Bard (1974) remains a source of information and inspiration in this area. On the topic of nonlinear least squares, Donald Watts and his associates have done much to advance understanding of the subject (for example, in the forthcoming book by Bates and Watts, 1986). In a very different fashion, work on generalized linear models, as illustrated in the book by

McCullagh and Nelder (1983), will have an important place in nonlinear modeling.

This brief survey ignores a great deal of research work which is important in the field. For example, no mention has been made of the efforts spent in incorporating constraints in nonlinear function minimization methods.

1-5. Statistical Approaches

Underlying all scientific and technological developments are the many years of tedious observation, data analysis and modeling and model testing needed to establish the "laws" of nature. With all observations, an element of error creeps into data. For the most part we will ignore (unexplained) systematic errors in observations, for example, due to equipment or the observational environment. Note, however, that systematic errors may be uncovered by careful analysis of the problem and data (Havriliak and Watts, 1985). We will also generally assume that observations with obviously wrong data have been eliminated from the problems we shall study. This task -- the detection and removal of outliers -- is one which cannot be taken lightly. However, we believe that it is outside our present purview, despite being a topic of great importance. (For a review, see Beckman and Cook, 1983.)

In the data left after systematic errors and (supposedly) accidental outliers have been removed, there will then remain some level of "error" from an ideal form which is due to the uncontrollable fluctuations in the system under study or the observational equipment. Ultimately, because of the Heisenberg uncertainty principle (Schiff, 1955, page 7), there is a limit to the control of such observational errors, and we are forced to take a statistical view of the fluctuations.

A simplified description of the statistical approach to estimation is as follows:

1. specify the mathematical form of the model of the system under study;
2. specify the nature of the fluctuations about this model (error distribution);
3. derive the relationship between the observed data and the unknown parameters of the model;
4. use this relationship to determine the most probable parameter values or those which minimize the deviation between predicted and observed behavior of the system.

Note that the first two of these four steps require that we assume a great deal about the system, namely, the form of the functional relationship describing the system of interest as well as the distribution of errors made in observing it. Clearly, one needs to have quite strong reasons for making such assumptions. Such reasons may be based on well-established theories, or the necessity of obtaining results, but the assumptions should ultimately be tested by comparison with the consequences of other possibilities. In some cases, it may be possible to show that the resultant model is relatively insensitive to the particular functional form of the model or error distribution suggested. In other situations, we may have to make a difficult decision between alternative models with different interpretations if they are accepted as reliable descriptions of the properties and behavior of the system under study. More observations and experiments are then required to resolve the uncertainties.

Given a form for the model function and for the error distribution, it is straightforward to design estimators for the parameters of the model. It is not our intention to explain statistical estimation theory, but the following example of maximum likelihood estimators illustrates one application of the methods we present later.

Let us suppose that the model function at the i-th observation yields a residual function (that is, deviation between observed and predicted behavior)

(1-5-1) $r(\underline{Y}_i^T, \underline{B})$ or $r(i,\underline{B})$

where Y_i^T represents the i-th row of the matrix of observed data, Y.

The error distribution function describes the relative probability that a given value of a residual is observed. If the true model parameters $\underline{B}^*$ are known, then the error distribution function describes the relative probability that an observation $\underline{Y}^T$ is recorded. Alternatively, we may consider the observed data to be given facts, so that the error distribution allows the relative probability (or likelihood) of different parameter sets to be compared.

With respect to the residual $r(i,\underline{B})$, we will write the probability density function as

(1-5-2) $P(r(i, \underline{B}))$

That is, the integral of this function over all possible values of r is one. The joint probability density that observations i and j are sampled i.e., that they are independently drawn at random in a sample of size 2, is

(1-5-3) $P(r(i, \underline{B})) * P(r(j, \underline{B}))$

and the joint probability density of an entire sample of observations is the product of the density functions for each real observation. The most probable set of parameters $\underline{B}$ is the set which maximizes this probability density. If the function can be written down, then methods for function maximization can be applied. (Usually we minimize the negative of the product of probability densities, or, frequently, the negative of the logarithm of this product to convert the functional form to a summation.)

This maximum likelihood approach to parameter estimation is an important class of estimators. However, its dependence on a specific probability density function for the error distribution should be kept clearly in mind. We maximize the likelihood of occurrence of a set of parameters under the assumed error distribution and under specific assumptions regarding the validity of the observations.

Other authors have explored these statistical aspects of nonlinear parameter estimation in more detail than we shall here. Bard (1974, Chapter 4) surveys a variety of statistical estimation techniques. Ratkowsky (1983), while restricting attention to errors which in one problem are independent of each other and all drawn from the same ("identical") Gaussian distribution, covers a number of important classes of nonlinear models in some detail. McCullagh and Nelder (1983) approach the task from the view of generalized linear models. While their approach and motivations are different from our own, their success in solving difficult modeling problems deserves the wide respect it has earned. The programs we present could be used within the McCullagh/Nelder formulation, which requires a sound understanding of relatively advanced statistics. The approaches we present are frequently less elegant routes to solutions of parameter estimation problems. They also admit constraints in a way which we believe to be easier for users to apply, but provide generally less statistical information about the parameters and hence models estimated. Research is continuing in both areas and one may hope to see a comprehensive and constructive synthesis of the two approaches in the not-too-distant future.

1-6. Software Approaches

Methods for nonlinear estimation have been a popular subject for research, as the bibliography will substantiate. Software for particular methods has also been announced from time to time. (For a review to 1973, see Chambers, 1973.) We divide the forms of this software into three classes:

- software packages
- parts of a program library
- individual programs or subroutines

Packages allow the user the greatest facility in solving parameter estimation problems. That is, a package provides for data entry, edit and transformation, the development of the model function and error distribution and the estimation of parameters. The price paid for convenience is usually the effort to learn the operation of the package, the cost of acquiring or using it and the general lack of flexibility.

Packages which allow nonlinear parameters to be estimated are:

- GLIM, a generalized linear modeling package (McCullagh and Nelder, 1983, page 239);
- GENSTAT, a statistical computing language (McCullagh and Nelder, 1983, page 239); and
- SAS, via the NLIN function (SAS User’s Guide, 1982).

Libraries of computer programs have been an important source of tools for solving nonlinear estimation problems. Four of the major libraries which are widely available are:

- the NAG library, which includes several nonlinear optimization methods, including at least one for nonlinear least squares (Because the library changes slightly at each release, we refer the reader to the latest documentation for precise information.);

- the IMSL library, which similarly offers several routines for nonlinear optimization and least squares, for example, the nonlinear least squares routine RNSSQ in the STAT/PC-LIBRARY (IMSL, 1984; Nash, 1986a; Nash, 1986b);
- the Harwell library, which is especially rich in nonlinear optimization routines;
- BMDP, which offers routines PAR and P3R to solve nonlinear least squares problems (BMDP, 1981, page 289ff.).

A number of individual programs have been publicized over the years. Many of these are linked to particular subject areas. For example, on pharmacokinetics, programs have been reported or discussed by Duggleby (1984a,1984b), Sedman and Wagner (1976), Pfeffer (1973), Pedersen (1977), Brown and Manno (1978), Peck and Barrett (1979) and Landriani, Guardabasso and Rocchetti (1983). For physics, James and Roos (1975) propose a polyalgorithm (several methods in one program) for the fitting and analysis of parameters. Using simple examples from chemistry, Caceci and Cacheris (1984) presented a Nelder-Mead polytope program in PASCAL. (This listing may appear heavily weighted towards chemical kinetics, either general or pharmacological. We believe that programs have been developed in other fields, but that they may not have been published. For example, we have had enquiries for our own work from engineers who were working on defense projects who did not wish to discuss the nature of the problems they were solving.)

One of the more widely discussed individual programs for nonlinear estimation is that of Marquardt (1964,1966). A simplified version of Marquardt's method has been presented as a step-and-description algorithm by Nash (1979). Bard (1974, Appendix G) lists several other nonlinear estimation

programs, but omits the more recent program NL2SOL of Dennis, Gay and Welsch (1979).

To conclude this section, we would stress that nonlinear parameter estimation cannot be fully automated because of the important decisions which must be made concerning model and error distribution forms, the rejection of outliers and the analysis of results. There is here the necessity for a partnership between user, software and machine if nonlinear estimation problems are to be efficiently solved. No one piece of software is likely to satisfy all requirements. Our own preference is to have at hand a number of software building blocks which can be quickly put together to approach a given problem. This approach is reflected in the programs presented later in the book.

1-7. Characterization of Problems

In this section we shall specifically define the classes of problems which will be of interest to us. We will use the vector

$$\underline{B} = (B(1), B(2),..., B(n))^T = (B_1, B_2, ..., B_n)^T \qquad (1\text{-}7\text{-}1)$$

to represent the set of parameters to be estimated.

In estimating the parameters in nonlinear models, one of the most important general problems is that of finding the minimum of a function of n parameters $\underline{B}$ which is itself the sum of the squares of m functions -- the nonlinear least squares problem (NLLS).

<u>Problem NLLS: Nonlinear least squares.</u>

Find the set of parameters $\underline{B}$ which minimizes the loss function

$$(1\text{-}7\text{-}2) \qquad f(\underline{B}) = \sum_{i=1}^{m} [r(i,\underline{B})]^2.$$

The functions r(i,B) are called residuals.

By taking the first derivatives of f(B) and setting them to zero in the classical method for finding the stationary points of a function (Protter and Morrey, 1964, Chapter 20), that is, the maxima, minima or saddle points, we obtain the nonlinear equations problem (NLE).

Problem NLE: Nonlinear equations or rootfinding.

Find the set of parameters B which causes the set of equations

(1-7-3) r(i,B) = 0 i = 1,2,...,m

Frequently m = n, so that the number of equations equals the number of parameters.

It is also possible to consider parameter estimation problems in which the loss function f(B) does not appear in the form of a sum of squares, in which case we may wish to consider the general unconstrained function minimization problem (FMIN).

Problem FMIN: Unconstrained function minimization.

Find the set of parameters B which minimizes f(B). This may be a local minimum i.e.

(1-7-4) f(B) < f(B + delta)

for all vectors delta such that length(delta) < deltasize.

Alternatively, we may pose the much more difficult problem of finding the global minimum of the function, that is, the B where it takes on its lowest value for all possible sets of parameters.

Problems FMIN and NLLS can have constraints imposed on the allowed parameters B. Such constraints may be in the form of m3 equations, or equality constraints,

(1-7-5) c(i,B) = 0 i = 1,2,...,m3

or appear as inequalities

(1-7-6) c(i,B) ≥ 0 i = m3+1, m3+2,...,m4.

Of the inequalities, one class is so common that our methods will generally be set up to take account of them. This is the class of bounds, that is, constraints of the form

(1-7-7) 2 ≤ B(3) ≤ 4

We rewrite (1-7-7) as two constraints to obtain the general form (1-7-6)

(1-7-8) B(3) ≥ 2
-B(3) ≥ -4

The general nonlinear function minimization problem can now be written down as follows:

Problem NLP: General function minimization or nonlinear programming.

Given data Y, minimize f(B, Y) with respect to B under the constraints

(1-7-5) c(i,B) = 0 i = 1,2,...,m3

and

(1-7-6) c(i,B) ≥ 0 i = m3+1, m3+2,...,m4.

It is straightforward to write down this problem, and not too difficult to actually specify all the functional forms needed for a number of practical problems. However, finding a solution is not trivial, and at this point we have no guarantee that a solution even exists. Moreover, the specific expression of a particular problem is open to a number of choices. For example, a single equality constraint,

B(1) + B(2) * B(3) = 4

is equivalent to the two inequalities

B(1) + B(2) * B(3) - 4 $\geq$ 0

and

- B(1) - B(2) * B(3) + 4 $\geq$ 0

A more serious problem arises when the constraints are such that there is NO solution, for example, if

B(1) + B(2) * B(3) $\geq$ 5

and

B(1) + B(2) * B(3) $\leq$ 3

are supposed to be satisfied simultaneously. In this case the pair of constraints is infeasible.

In general, each equality constraint should allow one parameter to be calculated from the rest, thereby reducing the dimensionality of the problem which has to be solved. In practice, it may be difficult to determine if each of the constraints imposes an independent relationship on the set of parameters B. However, for linear constraints, this is a well-understood procedure.

A categorization of constraints concerns their number. When there are very many constraints, particularly linear ones, it is common to approach the function minimization problem by exploring the possible feasible points. That is, we do not change the parameters B primarily to reduce the function f(B), but to move between points which are feasible with respect to the set of constraints. The general name given to this process is mathematical programming, which has unfortunate consequences for the development of indexes to the mathematical and computing literature.

Linear programming (LP) involves the minimization of

a linear function with respect to linear constraints.

Integer programming (IP) problems have parameters B which are integers. The problems are usually of the LP type. Mixed-Integer Programming problems involve both integer and floating-point parameters, and are among the more difficult to solve efficiently.

Quadratic programming (QP) problems have a quadratic function to be minimized subject to linear constraints. They are extremely common, arising in many statistical and image processing applications.

Nonlinear programming (NLP) is the general function minimization problem defined above, though the term is usually reserved for cases where there are a large number of constraints or parameters.

In this book, we will concentrate on problems having only a few explicit constraints. We shall approach such problems from the point of view of the loss function which we shall attempt to minimize. Constraints will be included by either altering the function so that the solution of an unconstrained problem coincides with that of the constrained one, or by forcing our search into the feasible region. Generally we expect that there will be only a few constraints in addition to bounds on the parameters.

As indicated, the various problems above are quite closely related:

1. Problem NLE is a problem NLLS with the loss function required to be zero at the minimum;

2. Classical calculus solution of problem FMIN sets the partial derivatives of the loss function to zero, a problem of type NLE. Note that additional conditions must be imposed on the solution to avoid local maxima and saddle points (Gill, Murray and Wright, 1981, pp 63-65);

3. Problem NLLS is clearly a special case of problem FMIN.

Just as the problems are closely linked, so are the methods for their solution. A discussion of these relationships is given in Section 3.2 and in some of the examples throughout the book.

Gill, Murray and Wright (1981) give problem FMIN without constraints the label UCP, while with constraints, as NCP. Conn (1985) calls it problem NLP as we have done here.

1-8. Outline of Intentions

In Section 1-2 we have presented the general purposes of this book and software. Now we will discuss what types of material in relation to nonlinear parameter estimation methods we hope to present in the following chapters.

1. We will deal primarily with methods for unconstrained problems. Bounds on parameters will be an important exception to this generalization. We will handle other constraints by methods mentioned briefly in Section 3-2 and detailed in examples.

2. The methods we present in this book are intended to be simple but effective. We will sacrifice machine cycles for user convenience and will not be overly bothered if convergence of an iteration is several or even many times slower than the "best" method for a problem. We will, however, wish the methods we employ to allow users to make steady progress toward solving real-world problems, and to do this without a great deal of difficulty.

3. We desire our methods to be useful on a variety of personal computers. This should include machines such as the Apple II family, the IBM-PC family, various CP/M machines and portables such as the Radio Shack TRS-80 Model 100. In this incarnation of our methods, we have used BASIC as the programming language. To allow for file input/output, we

have used the Microsoft dialect of BASIC (Microsoft, 1983) for those parts of the code which require file I/O functions. However, for the most part our code conforms to the International Standard ISO 6373-1984 for Minimal BASIC (Appendix D).

4. Our methods should be capable of handling most nonlinear parameter estimation problems which present themselves to users. There will surely be problems too extreme, too large, or too specialized for our methods, but by and large, we expect that the package of techniques presented herein to be comprehensive of most user problems.

5. Algorithm presentation is not a priority in this work, though algorithms are of course imbedded in the software. Other works, including our own, (Nash, 1979) present the details of how many of the algorithms work. Here we shall be satisfied with a general perspective on parameter estimation and function minimization methods.

6. Despite our intention NOT to present algorithms as such, we shall try to point out the important details of coding of parameter estimation programs, and of the loss functions and their derivatives, along with the background reasons for these details, which make a great deal of difference to the performance and success of programs and their users.

7. We hope that our work will provide a useful and inexpensive learning tool for this field.