Computational Chemistry

An Introduction to Numerical Methods

A. C. Norris
Department of Chemistry
Portsmouth Polytechnic

British Library Cataloguing in Publication Data:

Norris, A. C.
Computational chemistry.
1. Numerical analysis
2. Chemistry–Numerical analysis
I. Title
511'.7'02'4541 QA297 80-41691

ISBN 0 471 27949 8 (Cloth)
ISBN 0 471 27950 1 (Paper)

Typeset by Unicus Graphics Ltd, Horsham, and printed in the United States of America

Preface

One of the most important tasks facing the chemist is the need to extract useful information from numerical data. The requirement is increasing continuously as instrumental and computational methods produce not only more and better data, but the means to subject them to more detailed analysis. The general availability of cheap and powerful computers therefore makes it highly desirable that the chemist has some basic knowledge of the theory and practice of numerical computation.

The purpose of this book is to give a practical introduction to numerical methods at a level suitable for the undergraduate chemist. It therefore stresses chemical applications and attempts to show the student how to formulate a chemical problem in mathematical terms and select a numerical method to solve it. The book should also prove useful as an introduction to more powerful sources of programs such as the Numerical Algorithms Group (NAG) library.

The text has three introductory chapters which cover problem formulation and the treatment of computational and experimental errors. Chapters 4 to 8 discuss in turn, non-linear equations, simultaneous linear equations, numerical integration, the numerical solution of ordinary differential equations, and interpolation and approximation. The emphasis on selected, mainly experimental, applications has led to the omission of eigenvalue problems*.

The treatment of the chapters assumes a basic knowledge of algebra and calculus but no previous knowledge of numerical methods. Each chapter contains several worked examples and a wide range of problems varying in length and complexity. The longer exercises are structured into, frequently independent, subproblems and the many references given in the text and problems are designed to encourage students to search the literature and extend the exercises for themselves. Naturally, the problems have a numerical bias but numerical methods are no substitute for poor analysis and due consideration is given to problem formulation and the manipulation of equations. Detailed solutions and answers to the exercises are intended to strengthen understanding and promote a critical approach leading to further investigation.

No attempt is made to teach computer programming and ideally the computational chemistry would be taught alongside or immediately following a course on

* See the article by Bauer, H. and Roth, K., The numeric solution of eigenvalue problems, *J. Chem. Educ.*, 57, 423 (1980) and references contained therein.

the elements of FORTRAN. However, this arrangement may be impractical, or students may not take to the discipline of programming. The numerical methods chapters cater for these eventualities by providing two types of exercise. The first uses existing programs so that users with virtually no computing knowledge can enter data and interpret the results numerically and chemically. The second type assumes an elementary knowledge of FORTRAN and requires short programs to be written around available subprograms. These latter exercises present an additional challenge and the opportunity to develop programming skills but keep the emphasis on chemical problem-solving by providing clear directions and hints on writing the programs. The available software is described in the appropriate chapters and complete listings are given in the appendices.

Many of the exercises were developed whilst I held a BP Fellowship at Keble College, Oxford in 1976 and it is a pleasure to record my gratitude to the British Petroleum Company and the Warden and Fellows of Keble College for the facilities made available to me. My thanks are also due to Miss Patricia Lee and Mrs Angela Sharkey for their invaluable comments on the mathematical presentation. The task of removing errors has been greatly helped by the vigilance of my students and the expert typing of Mrs Anita Lemmon. Special thanks are due to my colleague Mr Stephen Broadbent for his ever-helpful comments and assistance in writing and testing the software. Naturally any errors that remain in the text or programs are my own responsibility and I shall be grateful to readers who bring them to my attention.

Portsmouth
September 1980 A. C. NORRIS

Notes of the Use of the Text

Units

SI Units are used throughout the book but the atm, mmHg, and occasionally the min, are retained for data originally published in these units.

Small Type Sections

Each of the chapters on numerical methods contains exercises that either require data to be entered into existing programs, or FORTRAN programs to be written around existing subroutines. To facilitate the use of these exercises, the Data Entry and Subprogram Specification sections of these chapters are accordingly set in small type to distinguish them from the main text.

Availability of Software

All the programs listed in the Appendices, together with test data, are available either on magnetic or paper tape. Also available are tapes, listings, and results for the 25 programming exercises described in sections 7 of Chapters 4–8. There is a small handling charge and lecturers interested in using these materials in their courses should write in the first instance to the author, Dr A. C. Norris, Department of Chemistry, Portsmouth Polytechnic, Portsmouth, Hampshire, England.

Contents

CHAPTER 1

Problem Formulation and Solution

1.1 INTRODUCTION

The chemist who wishes to solve a numerical problem must first decide its nature and its complexity since these factors will determine the approach and equipment needed to solve it. It must be stressed at the outset that little progress can be made in solving any but the simplest problems without at least a basic knowledge of algebra and calculus. Fortunately, this knowledge is not difficult to acquire and, with a little experience, it is possible to formulate the problem and determine whether pencil and paper, calculator or computer should be used for the solution. When deciding upon the equipment, it is best to use the simplest possible. This is not to suggest that the solution of say ten, simultaneous, linear equations in ten unknowns should be attempted by hand. The rule suggests rather that a little thought and scribbling can sometimes produce an answer more quickly, more cheaply, and even more accurately than the corresponding computer solution.

Given that the problem is sufficiently involved or repetitive to warrant a computer, the potential user is then faced with a number of additional questions:

(i) How do I formulate the problem for computer solution?
(ii) What methods are available to solve the problem in a reasonable time?
(iii) How do I obtain the best solution permitted by the data?
(iv) How do I know that this result has been obtained?
(v) Is it meaningful to correct an approximate result?
(vi) How can such a correction be applied?

The purpose of this book is to help the reader to answer these questions. The first step is to recognize that the questions should be answered at the design stage before any attempt is made to write a computer program. Certainly the computer will do exactly what the program tells it to, but it is up to the programmer to express his scientific knowledge in mathematical terms which can then be translated into an effective program. Also, the computer is not the modern equivalent of the philosopher's stone and electronic wizardry cannot turn poor data into good results. Furthermore, computers introduce error at almost every stage of a calculation and unless the user knows something about how susceptible his problem is to this error, and how it can be controlled, then sooner or later he will find himself accepting answers which are quite untrue.

In the following chapters we hope to show that the above questions are not nearly as daunting as they might seem. The remainder of this chapter is devoted to general advice on formulating and solving numerical problems and writing efficient computer programs. Exercises to demonstrate the points discussed are presented at the end of the chapter.

1.2 FORMULATING THE PROBLEM AND DESIGNING A SOLUTION

Many scientists become fascinated by computer programming and expend a great deal of energy devising programs which make efficient use of computer resources. Although this effort is often justified, the successful computer solution is likely to depend far less on program efficiency than on the correct formulation of the problem and the choice of an appropriate method to solve it. Not surprisingly, these latter tasks are determined largely by the nature of the problem and there are no simple rules which can be applied to solve the problem uniquely. Success, as always, hinges on an appropriate blend of theory and experience. Some useful guidelines can be given, however, and in this section we illustrate these with examples to indicate the value of a little thought before rushing to the computer.

1.2.1 Mathematical Expression and Problem Type

A mathematical and hence a computational method can only operate on formulae, sets of equations, inequalities, etc., and so it is necessary to present the scientific problem in mathematical terms. The translation process is achieved by adopting a suitable notation which preserves the chemical significance but reveals clearly the nature of the mathematical problem. Only when the problem type has been identified is it possible to consider a method of solution.

To illustrate these points consider the determination of the proton concentration in an aqueous solution of a weakly ionized acid HA,

$$HA + H_2O \rightleftharpoons H_3O^+ + A^-. \tag{1.1}$$

If the system is assumed to behave ideally, the acid ionization constant K_a is defined by

$$K_a = [H_3O^+]\,[A^-]/[HA]$$

and rearrangement of this expression yields the required equation for the proton concentration:

$$[H_3O^+] = K_a\,[HA]/[A^-]. \tag{1.2}$$

The chemist would probably interpret equation (1.2) as 'The proton concentration is the ratio of un-ionized acid and anion concentrations multiplied by the ionization constant'; a statement which implies that the problem is easily solved by substituting numerical values for the terms on the right-hand side. That this is too simple a view is readily seen by expressing the problem mathematically. If the total acid concentration is C_a and the proton concentration x, then

$$[H_3O^+] = [A^-] = x \qquad [HA] = C_a - x$$

and equation (1.2) becomes

$$x = K_a(C_a - x)/x \tag{1.3}$$

which rearranges to give

$$x^2 + K_a x - K_a C_a = 0 \tag{1.4}$$

showing that without further assumptions x, i.e. $[H_3O^+]$, must be evaluated as the root of a quadratic equation. A further difficulty is the need for some criterion to select the correct root.

Now this example is far too trivial for the complications not to be realized at an early stage. It does demonstrate, however, the need to reformulate the problem and so to replace the chemical questions (e.g. what is the proton concentration?) by mathematical ones (e.g. what is the appropriate root of the quadratic equation?). Only when the mathematical objectives are clearly defined can attention be given to the organization of the solution.

1.2.2 Well- and Poorly-formulated Problems

Broadly, a well-formulated problem is one which produces the correct solution with minimum error. Any steps that can be taken to ensure these properties are clearly worth considering. We can illustrate the first part of the definition by an example from chemical kinetics. The consecutive first-order decomposition

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

with rate constants k_1 and k_2 produces the relationship (see Exercise 4.7-4)

$$k_2 = g(k_2) = k_1 \exp[(k_2 - k_1)t_{\max}] \tag{1.5}$$

where $t_{\max}$ is the time at which the concentration of B achieves its maximum value. If the values of k_1 and $t_{\max}$ are known then, in principle, an initial estimate of k_2 can be inserted into the right-hand side of this non-linear equation and refined by successive approximation. However, equation (1.5) has two roots for k_2 and the theory[1] of iterative processes shows that the iteration can converge only to the root at which the derivative

$$|\mathrm{d}g(k_2)/\mathrm{d}k_2| \equiv |g'(k_2)| < 1.$$

Differentiating equation (1.5) demonstrates that this convergence condition is satisfied when $k_2 < 1/t_{\max}$. If we know from additional evidence that $k_2 > 1/t_{\max}$ then equation (1.5) is clearly poorly-formulated for the calculation of k_2. The solution is to take logarithms of equation (1.5), rearrange the result to give

$$k_2 = k_1 + [\ln(k_2/k_1)]/t_{\max} \tag{1.6}$$

and the problem is now well-formulated since a check on the convergence condition shows that this equation will produce the required root.

The effect of problem formulation on the accuracy of a solution is demonstrated by the numerical treatment of the ordinary differential equation

$$\mathrm{d}y/\mathrm{d}x \equiv y' = y - x$$

with the initial condition $y = 1$ when $x = 0$ $[y(0) = 1]$. The analytical, i.e. exact, solution is

$$y = x + 1$$

but if the problem is solved numerically and a small error is introduced which alters the initial condition to $y(0) = 1 + \epsilon$ then the solution is

$$y = \epsilon\,\mathrm{e}^{x} + x + 1$$

and the exponential term eventually swamps the correct 'linear' solution. The difficulty is caused by the poor choice of initial condition which produces an exponentially increasing component as the solution progresses from $x = 0$ to some value x_n. If we can reformulate the problem with the initial condition specified at some high value of x and proceed to $x = 0$ then the difficulty vanishes since the exponential contribution is now decreasing. The 'condition' of a problem is frequently improved by reformulation.

1.2.3 Transformation

The concept of a transform is familiar from the use of logarithms which turn a multiplication or division into a simpler addition or subtraction. If a parameter of interest occurs in an equation as a power then logarithmic transformation is frequently used to produce a linear relationship from which the parameter is obtained more readily. Numerous examples arise with kinetic and equilibrium processes which involve an energy barrier. Thus, the saturated vapour pressure (P) of a pure liquid is expressed as a function of temperature (T) by the equation

$$P = A\,\mathrm{e}^{-\Delta H/RT}$$

where ΔH is the enthalpy of vaporization and A and R are constants. The determination of ΔH is greatly simplified by taking logarithms to produce the more familiar expression

$$\ln P = -\,\Delta H/RT + \ln A.$$

ΔH is then estimated either from a linear plot of $\ln P$ as a function of $1/T$ or by a linear least-squares analysis with suitable weighting (see Chapter 8).

Logarithmic and other simple transforms[2,3] are often sufficient to extract the required information from the data but, in some cases, more complicated transforms may be needed. For example, the Fourier integral transform[4] is now used for the routine reduction of spectroscopic data[5] and the computer is playing an essential role in the development of instrumental methods.

1.2.4 Analytical Simplification

The discussion in Section 1.2.1 indicates that when chemical equations are formulated mathematically, the resulting expressions may need further rearrangement before they can be treated numerically. At the same time it may be possible to facilitate the numerical solution by simplifying the equations algebraically. A common instance is the use of integration by parts to produce recurrence relationships for numerical definite integrals. Thus, the gamma function $\Gamma(n)$ which finds application in areas such as unimolecular reaction theory[6] and the solution of differential equations[7] is defined by[8]

$$\Gamma(n) = \int_0^\infty x^{n-1}\,e^{-x}\,dx \qquad n > 0.$$

Integration by parts for $\Gamma(n+1)$ yields

$$\Gamma(n+1) = \int_0^\infty x^n\,e^{-x}\,dx = [-x^n\,e^{-x}]_0^\infty + n\int_0^\infty x^{n-1}\,e^{-x}\,dx,$$

that is,

$$\Gamma(n+1) = n\Gamma(n)$$

so that a series of integrals with non-integer n, $\Gamma(n+r)$, $r = 0, 1, 2, \ldots$, can be evaluated from this recurrence relationship and a single integration of $\Gamma(n)$. The decrease in the number of calculations makes this type of simplification not only faster but frequently more accurate than repeated numerical treatment.

1.2.5 Analytical Approximation

Analytical approximation refers to changes made to an equation to give a (usually slightly) different but not equivalent relationship. The object of course is to simplify the solution at the expense of an acceptable increase in (approximation) error. Common examples are the assumptions that $1 \pm z \simeq 1$ when $z \ll 1$ and the truncation of the exponential series.

An approximation frequently results from some assumption about the chemistry and it should in any case never be made without considering its physical consequences. For example, if the acid referred to in equation (1.1) is very weak then the proton concentration x might be considered negligible compared with the acid concentration C_a. In this case

$$C_a - x \simeq C_a \qquad (x/C_a \ll 1)$$

and equation (1.3) can be rewritten as

$$x \simeq (K_a C_a)^{1/2}$$

allowing x to be evaluated as a simple square root rather than one of the roots of the quadratic equation (1.4).

Similarly, the linearization of an exponential term may be possible under limiting conditions. Thus, the Debye equation for the heat capacity at constant volume of a monatomic solid is

$$C_V = (9R/x_m^3)\int_0^{x_m} [e^x x^4/(e^x - 1)^2]\, dx$$

where R is the gas constant and $x_m = \theta_D/T$; θ_D being the Debye temperature and T the temperature at which C_V is to be evaluated. Generally, the integral is found numerically but for sufficiently high temperatures

$$x \ll 1 \qquad (T \gg \theta_D)$$

and

$$e^x \simeq 1 + x$$

so that

$$e^x x^4/(e^x - 1)^2 \simeq e^x x^2 \simeq x^2.$$

The integral is then readily evaluated to give

$$C_V \simeq 3R,$$

the classical limiting value of the quantum prediction.

Naturally, neither of the above examples needs a computer to obtain the answer but, when a computer is necessary, care should be taken to ensure that it does not introduce any unsolicited approximations. For example, the integral of the Arrhenius function

$$\int_0^{T^*} e^{-E/RT}\, dT$$

involving the activation energy E is important in the non-isothermal decomposition of solids[9]. On many mainframe computers, the smallest positive number just greater than zero which the machine can store is about e^{-176}. Hence if $E/RT > 176$, the exponential will be set to zero and the relative error may be large though the absolute value of the integral is very small. Most computers give no indication of this underflow condition and it is advisable to know the computer's limitations.

1.2.6 Data Input and Output

The experimental conditions will determine the number and type of data available for analysis and these factors together with the format in which the results are to be presented may materially affect the formulation and solution of the problem. For example, the fugacity coefficient γ of a gas at a pressure P^* is obtained by a numerical integration of the relationship

$$\ln \gamma = \int_0^{P^*} [(C - 1)/P]\, dP$$