Computer Science
and Scientific Computing

NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

THIRD EDITION

William F. Ames

THIRD EDITION

Numerical Methods for Partial Differential Equations

This is a volume in
COMPUTER SCIENCE AND SCIENTIFIC COMPUTING
A Series of Monographs and Textbooks

Editor: WERNER RHEINBOLDT

A complete list of titles in this series is available from the publisher upon request.

THIRD EDITION

Numerical Methods for Partial Differential Equations

WILLIAM F. AMES

School of Mathematics
Georgia Institute of Technology
Atlanta, Georgia

ACADEMIC PRESS, INC.

Harcourt Brace Jovanovich, Publishers

Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper ♾

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24-28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data

Ames, William F. date.
Numerical methods for partial differential equations / W.F. Ames.
—3rd ed.
p. cm. — (Computer science and scientific computing)
Includes bibliographical references and indexes.
ISBN 0-12-056761-X (alk. paper)
1. Differential equations. Partial—Numerical Solutions.
I. Title. II. Series
QA374.A46 1992
515'.353—dc20 91-32811
CIP

Printed in the United States of America
92 93 94 95 EB 9 8 7 6 5 4 3 2 1

To Professor Garrett Birkhoff: Scholar, Advisor and Friend

Contents

This page intentionally left blank

Preface to the Third Edition

The fifteen years since the appearance of the second edition have been years of great accomplishments in scientific computation. Indeed, the number of journals dedicated to the subject of this book is now more than 20! To reflect this growth, this third edition is a drastic revision of the second one. Finite elements have been merged with the material on finite differences and they now constitute equal partners. Additional material has been added in the areas of boundary elements, spectral methods, the method of lines, and invariant methods. The self-contained nature of the previous editions has been maintained insofar as possible.

The references have been brought up to date and new problems reflect the additional material.

I am grateful to previous users for alerting me to misprints and errors of commission and omission. I am also thankful to Mrs. Annette Rohrs for her skillful preparation of the manuscript on the word processor. Professor Garrett Birkhoff made a number of thoughtful suggestions as I worked on the manuscript. For these and for his other advice I am most appreciative.

W. F. Ames
Georgia Institute of Technology

This page intentionally left blank

Preface to Second Edition

Since the publication of the first edition, research in and applications of numerical analysis have expanded rapidly. The past few years have witnessed the maturation of numerical fluid mechanics and finite element techniques. Numerical fluid mechanics is addressed in substance in this second edition. I have also added material in several other areas of promise, including hopscotch and other explicit–implicit methods, Monte Carlo techniques, lines, the fast Fourier transform, and fractional steps methods. A new sixth chapter introduces the general concepts of weighted residuals, with emphasis on orthogonal collocation and the Bubnov–Galerkin method. In turn, the latter procedure is used to introduce the finite element concepts.

The spirit of the first edition was to be as self-contained as possible, to present many applications illustrating the theory, and to supply a substantial number of recent references to supplement the text material. This spirit has been retained—there are 38 more problems and 138 additional references. Also, a substantial number of additional applications have been included and references to others appended.

I wish to extend my special thanks to Ms. Mildred Buckalew for the preparation of an outstanding manuscript on the typewriter.

Georgia Institute of Technology

This page intentionally left blank

Preface to First Edition

That part of numerical analysis which has been most changed by the ongoing revolution in numerical methods is probably the solution of partial differential equations. The equations from the technological world are often very complicated. Usually, they have variable coefficients, nonlinearities, irregular boundaries, and occur in coupled systems of differing types (say, parabolic and hyperbolic). The 'curse of dimensionality' is ever present – problems with two or three space variables, and time, are within our computational grasp.

Early development of calculational algorithms was based more upon the extension of methods for hand computation, empiricism, and intuition than on mathematical analyses. With increasing education and the subsequent development of the professional numerical analyst, the pattern is changing. New, useful methods are evolving which come closer to full utilization of the inherent powers of high-speed, large-memory computing machines. Many significant and powerful methods await discovery both for problems which are computable with existing techniques and those which are not.

Unfortunately, as in other portions of mathematics, the abstract and the applications have tended to diverge. A new field of pure mathematics has been generated and while it has produced some results of value to users, the complexities of real problems have yet to be significantly covered by the presently available theorems. Nevertheless, guidelines are now available for the person wishing to obtain the numerical solution to a practical problem.

The present volume constitutes an attempt to introduce to upper-level engineering and science undergraduate and graduate students the concepts of modern numerical analyses as they apply to partial differential equations. The book, while sprinkled liberally with practical problems and their solutions, also strives to point out the pitfalls – e.g., overstability, consistency requirements, and the danger of extrapolation to nonlinear problems methods which have proven useful on linear problems. The mathematics is by no means ignored, but its development to a keen-edge is not the major goal of this work.

The diligent student will find 248 problems of varying difficulty to test his mettle. Additionally, over 400 references provide a guide to the research and practical problems of today. With this text as a bridge, the applied student should find the professional numerical analysis journals more understandable.

I wish to extend special thanks to Mrs. Gary Strong and Mrs. Steven Dukeshier for the typing of a difficult manuscript and Mr. Jasbir Arora for preparation of the ink drawings. Lastly, the excellent cooperation and patience of Dr. Alan Jeffrey and my publishers have made the efforts of the past two years bearable.

This page intentionally left blank

1
Fundamentals

1-0 Introduction

Numerical calculation is commonplace today in fields where it was virtually unknown before 1950. The high-speed computing machine has made possible the solution of scientific and engineering problems of great complexity. This capability has, in turn, stimulated research in numerical analysis since effective utilization of such devices depends strongly upon the continual advance of research in relevant areas of mathematical analysis. One measure of the growth is the upsurge of books devoted to the subject in the years after 1953. A second measure is the development, during the same period, of many research journals and societies, whose primary concern is numerical analysis. Whole new disciplines, such as numerical fluid mechanics and computational mechanics, have arisen.

Finite difference approximations for derivatives were already in use by Euler [1]† in 1768. The simplest finite difference procedure for dealing with the problem $dx/dt = f(x, t)$, $x(0) = a$ is obtained by replacing $(dx/dt)_{n-1}$ with the crude approximation $(x_n - x_{n-1})/\Delta t$. This leads to the recurrence relation $x_0 = a$, $x_n = x_{n-1} + \Delta t f(x_{n-1}, t_{n-1})$ for $n > 0$. This procedure is known as Euler's method. Thus, we see that for one-dimensional systems the finite difference approach has been deeply ingrained in computational algorithms for quite some time.

For two-dimensional systems, the first computational application of finite difference methods was probably carried out by Runge [2] in 1908. He studied the numerical solution of the Poisson equation $u_{xx} + u_{yy} = \text{constant}$. At approximately the same time Richardson [3], in England, was carrying on similar research. His 1910 paper was the earliest work on the application of iterative methods to the solution of continuous equilibrium problems by finite differences. In 1918, Liebmann [4], in considering the finite difference approximation to Laplace's equation, suggested an improved method of iteration. Today the name of Liebmann is associated with any method of iteration by single steps in which a fixed calculation sequence is followed.

The study of errors in finite difference calculations is still an area of prime research interest. Early mathematical convergence proofs were carried out by LeRoux [5], Phillips, and Wiener [6], and Courant *et al.* [7]. Some consider the celebrated 1928 paper of Courant, Friedrichs, and Lewy as the birthdate of the modern theory of numerical methods for partial differential equations.

† Numbers in brackets refer to the references at the end of each chapter.

The algebraic solution of finite difference approximations is usually accomplished by some iteration procedure. Various schemes have been proposed to accelerate the convergence of the iteration. A summary of those that were available in 1950, and that are adaptable to automatic programming, is given by Frankel [8]. Other methods require considerable judgment on the part of the computer and are therefore better suited to hand computation. Higgins [9] gives an extensive bibliography of such techniques. In the latter category, the method of *relaxation* has received the most complete treatment. Relaxation was the most popular method in the decade of the thirties. Two books by Southwell [10, 11] describe the process and detail many examples. The *successive over-relaxation method*, extensively used in some areas, is an outgrowth of this highly successful hand computation procedure.

Let us now consider some of the early technical applications. The pioneering paper of Richardson [3] discussed the approximate solution by finite differences of differential equations describing stresses in a masonry dam. Equilibrium and eigenvalue problems were successfully handled. Binder [12] and Schmidt [13] applied finite difference methods to obtain solutions of the diffusion equation. The classical explicit recurrence relation

$$u_{i,j+1} = ru_{i-1,j} + (1 - 2r)u_{i,j} + ru_{i+1,j}, \qquad r = \Delta t/(\Delta x)^2,$$

for the diffusion equation $u_t = u_{xx}$ was given by Schmidt [13] in 1924.

For any given continuous system there are a multiplicity of discrete models, which are usually comparable in terms of their relative truncation errors. Early approximations were second order—that is, $O(h^2)$†—and these still play an important role today. Higher order procedures were promoted by Collatz [14, 15] and Fox [16]. The relative economy of computation and accuracy of second-order processes utilizing a small interval size, compared with higher order procedures using larger interval sizes, has been discussed in the papers of Southwell [17] and Fox [18].

It is quite possible to formulate a discrete model in an apparently natural way that, upon computation, is of little use. This is especially true in propagation problems—that is, problems described by parabolic and hyperbolic equations. An excellent historical example is provided by Richardson's pioneering paper [3], in which his suggested method for the diffusion equation, describing the cooling of a rod, was found to be completely unstable by O'Brien, Hyman, and Kaplan [19]. Nevertheless, the method can be used for short-time calculations, as discussed by Bell [76]. Another example concerns the transverse vibration of a beam. In 1936, Collatz [20] proposed a "natural"

† The notation $O(h^2)$ is read "(term of) order h^2" and can be interpreted to mean "when h is small enough the term behaves essentially like a constant times h^2."

finite difference procedure for the beam equation $u_{tt} + u_{xxxx} = 0$, but fifteen years later [21] the algorithm was found to be computationally unstable.

Nevertheless, the analyst usually strives to use methods dictated by the problem under consideration—these we call *natural methods*. Thus, a natural coordinate system may be toroidal (see Moon and Spencer [22]) instead of cartesian. Certain classes of equations have natural numerical methods that may be distinct from the finite difference methods. Typical of these are the *method of lines* for propagation problems and the *method of characteristics* for hyperbolic systems. Characteristics also provide a convenient way to classify partial differential equations.

1-1 Computer Program Packages

The user should be aware that there exist today a number of general purpose computer packages for the numerical solution of PDEs. This is especially true of *quasilinear* PDEs involving a single unknown function of *two* independent (real) variables. The following paragraphs will briefly identify a few such packages, giving references where more information about them can be found.

(A) *DISPL* was developed and written at *Argonne National Laboratory*. For details one good contact is Byrne [23] (see also Byrne [24]). DISPL is designed for nonlinear second order PDEs (parabolic, elliptic, hyperbolic (some cases), and parabolic–elliptic). Boundary conditions of a general nature and material interfaces are allowed. The spatial dimension can be either one or two and in cartesian, cylindrical, or spherical (one dimension only) geometry. The PDEs are reduced to ordinary DEs by Galerkin discretization of the spatial variables. The resulting ordinary DEs in the time-like variable are then solved by an ODE software package (such as GEAR). Software features include graphics capabilities, printed output, dump/restart facilities, and free format input. DISPL is intended to be an engineering and scientific tool and is not a finely tuned production code for a small set of problems.

DISPL makes no effort to control the spatial discretization errors. It has been used to successfully solve a variety of problems in chemical transport, heat and mass transfer, pipe flow, etc.

(B) *PDELIB* was developed and written at Los Alamos Scientific Laboratory. For details write Hyman [25]. PDELIB is a library of subroutines to support the numerical solution of evolution equations with a time-like variable and one or two space variables. The routines are grouped into a dozen independent modules according to their function—i.e., accepting initial data, approximating spatial derivatives, advancing the solution in time, etc. Each task is isolated in a distinct module. Within a module, the basic task is further refined into general purpose flexible lower level routines.

PDELIB can be understood and used at different levels. Within a small period of time a large class of problems can be solved by a novice. Moreover, it can provide a wide variety of outputs.

(C) *DSS/2* is a differential systems simulator developed at Lehigh University as a transportable Numerical Method of Lines (NMOL) code. See also LEANS. For details write Schiesser [26].

(D) *FORSIM* is designed for the automated solution of sets of implicitly coupled PDEs of the form

$$\frac{\partial u_i}{\partial t} = \phi_i\left(x, t, u_i, u_j, \ldots, (u_i)_x, \ldots, (u_i)_{xx}, (u_j)_{xx}, \ldots\right),$$

for $i = 1, \ldots, N$.

The user specifies the ϕ_i in a simple FORTRAN subroutine. Finite difference formulae of any order may be selected for the spatial discretization and the spatial grid need not be equidistant. The resulting system of time dependent ODEs is solved by the method of lines. For details write Carver [27] (see also Carver [28]).

(E) *SLDGL* is a program package for the self-adaptive solution of nonlinear systems of elliptic and parabolic PDEs in up to and including three space dimensions. Variable step size and variable order are permitted. The discretization error is estimated and used for the determination of the optimum grid and optimum orders. This is the most general of the codes described here (not for hyperbolic systems of course). For details write Schönauer [29] (see Schönauer *et al.* [30]). This package has seen extensive use in Europe.

(F) *FIDISOL* (finite difference solver) is a program package for nonlinear systems of two- or three-dimensional elliptic and parabolic systems in rectangular domains or in domains that can be transformed analytically to rectangular domains. This package is actually a redesign of parts of SLDGL, (E), primarily for the solution of large problems on vector computers. It has been tested on the CYBER 205, CRAY-1M, CRAY X-MP/22, and VP 200. The program vectorizes very well and uses the vector arithmetic efficiently. In addition to the numerical solution, a reliable error estimate is computed. For more details write Schönauer [29].

(G) *CAVE* is a program package [31] for conduction analysis via eigenvalues for three-dimensional geometries using the method of lines. In many problems much time is saved since only a few terms suffice.

(H) *IMSL Library* (Rice [32]). Many industrial and university computing services subscribe to the IMSL Software Library. Announcements of new software appear in *Directions*, a publication of IMSL. A brief description of some IMSL packages applicable to PDEs and associated problems is now given.

(1) *Linear Equation Packages.* There are three complementary linear equation packages of note.

LINPACK is a collection of programs concerned with *direct* methods for general (or full) symmetric, symmetric positive definite, triangular, and tridiagonal matrices. There are also programs for least squares problems along with the QR algorithm for eigensystems and the singular value decompositions of rectangular matrices. The programs are intended to be completely machine independent, fully portable, and run with good efficiency in most computing environments. The LINPACK User's Guide by Dongarra, *et al.* [33] is the basic reference.

ITPACK is a modular set of programs for iterative methods. The package is oriented toward the sparse matrices that arise in the solution of PDEs and other applications. While the programs apply to full matrices, that is rarely profitable. Four basic iteration methods and two convergence acceleration methods are in the package. There is Jacobi, SOR (with optimum relaxation parameter estimated), symmetric SOR, and reduced system (red–black ordering) iteration, each with semi-iteration and conjugate gradient acceleration. All parameters for these iterations are automatically estimated. The practical and theoretical background for ITPACK is found in Hageman and Young [34].

YALEPACK is a substantial collection of programs for sparse matrix computations.

(2) *Special PDE Packages.* In addition to those described in A to G two additional software packages bear mention. The first of these, the *ELLPACK* System, available from Rice [35], solves elliptic problems in two dimensions with general domains and in three dimensions with box-shaped domains. The system contains over 30 numerical methods modules, thereby providing a means of evaluating and comparing different methods for solving elliptic problems. ELLPACK has a special high level language making it easy to use. New algorithms can be added or deleted from the system with ease. See also Birkhoff and Lynch [36, Chapter 9].

Lastly, TWODEPEP is IMSL's general finite element system for two-dimensional elliptic, parabolic, and eigenvalue problems. The Galerkin finite elements available are triangles with quadratic, cubic, or quartic basic functions, with one edge curved when adjacent to a curved boundary, according to the isoparametric method. Nonlinear equations are solved by Newton's method with the resulting linear system solved directly by Gauss elimination. PDE/PROTRAN is also available. It uses triangular elements with piecewise polynomials of degree 2, 3, or 4 to solve quite general

steady-state, time dependent and eigenvalue problems in general two-dimensional regions. There is a simple user input. Additional information may be obtained from IMSL [37]. NASTRAN and STRUDL are two advanced finite element computer systems available from a variety of sources. Another, UNAFEM has been extensively used (see Burnett [38]).

1-2 Typical Problems

The majority of the problems of physics and engineering fall naturally into one of three *physical* categories: *equilibrium problems*, *eigenvalue problems*, and *propagation problems*.

Equilibrium problems are problems of steady state in which the equilibrium configuration ϕ in domain D is to be determined by solving the differential equation

$$L[\phi] = f \tag{1-1}$$

within D, subject to certain boundary conditions

$$B_i[\phi] = g_i \tag{1-2}$$

on the boundary of D. Very often, but not always, the integration domain D is closed and bounded. In Fig. 1-1 we illustrate the general equilibrium problem. In mathematical terminology such problems are known as *boundary value problems*. *Typical physical examples* include steady viscous flow, steady temperature distributions, equilibrium stresses in elastic structures, and steady voltage distributions. Despite the apparent diversity of the physics we shall shortly see that the governing equations for equilibrium problems are *elliptic*.†

Eigenvalue problems may be thought of as extensions of equilibrium problems wherein critical values of certain parameters are to be determined in addition to the corresponding steady-state configurations. Mathematically, the problem is to find one or more constants (λ), and the corresponding functions (ϕ), such that the differential equation

$$L[\phi] = \lambda M[\phi] \tag{1-3}$$

is satisfied within D and the boundary conditions

$$B_i[\phi] = \lambda E_i[\phi] \tag{1-4}$$

hold on the boundary of D. *Typical physical examples* include buckling and stability of structures, resonance in electric circuits and acoustics, natural

† The original mathematical formulation of an equilibrium problem will generate an elliptic equation or system. Later mathematical approximations may change the type. A typical example is the boundary layer approximation of the equations of fluid mechanics. Those elliptic equations are approximated by the parabolic equations of the boundary layer. Yet the problem is still one of equilibrium.

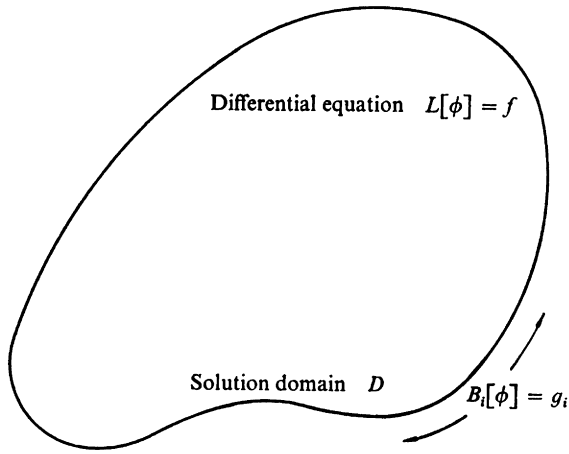

Figure 1-1 Representation of the general equilibrium problem

frequency problems in vibrations, and so on. The operators L and M are of elliptic type.

Propagation problems are initial value problems that have an unsteady state or transient nature. One wishes to predict the subsequent behavior of a system given the initial state. This is to be done by solving the differential equation

$$L[\phi] = f \tag{1-5}$$

within the domain D when the initial state is prescribed as

$$I_i[\phi] = h_i \tag{1-6}$$

and subject to prescribed conditions

$$B_i[\phi] = g_i \tag{1-7}$$

on the (open) boundaries. The integration domain D is open. In Fig. 1-2 we illustrate the general propagation problem. In mathematical parlance such problems are known as *initial boundary value problems.*† *Typical physical examples* include the propagation of pressure waves in a fluid, propagation of stresses and displacements in elastic systems, propagation of heat, and the development of self-excited vibrations. The physical diversity obscures the fact that the governing equations for propagation problems are *parabolic or hyperbolic.*

The distinction between equilibrium and propagation problems was well stated by Richardson [39] when he described the first as *jury* problems and the second as *marching* problems. In equilibrium problems the entire solution is passed on by a jury requiring satisfaction of all the boundary conditions and all

† Sometimes only the terminology initial value problem is utilized.

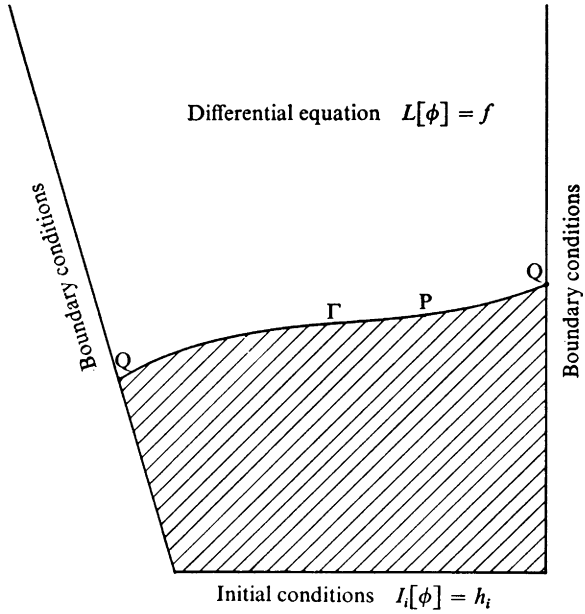

Figure 1-2 Representation of the general propagation problem

the internal requirements. In propagation problems the solution marches out from the initial state guided and modified in transit by the side boundary conditions.

1-3 Classification of Equations

The previous physical classification emphasized the distinctive features of basically two classes of problems. These distinctions strongly suggest that the governing equations are quite different in character. *From this we infer that the numerical methods for both problems must also have some basic differences.* Classification of the equations is best accomplished by developing the concept of *characteristics.*

Let the coefficients $a_1, a_2, \ldots, f_1, f_2$ be functions of x, y, u, and v and consider the simultaneous first-order quasilinear system†

$$\begin{aligned} a_1 u_x + b_1 u_y + c_1 v_x + d_1 v_y &= f_1 , \\ a_2 u_x + b_2 u_y + c_2 v_x + d_2 v_y &= f_2 . \ddagger \end{aligned} \tag{1-8}$$

This set of equations is sufficiently general to represent many of the problems encountered in engineering where the mathematical model is second order.

† A quasilinear system of equations is one in which the highest order derivatives occur linearly.
‡ We shall often use the notation u_x to represent $\partial u / \partial x$.