Numerical Solutions of Partial Differential Equations– An Introduction to Finite Difference and Finite Element Methods

Zhilin Li[1] Zhonghua Qiao[2] Tao Tang[3]

December 17, 2012

[1]Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA
[2]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong
[3]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong

Contents

Preface

The purpose of this book is to provide an introduction to the numerical solution of partial differential equations using finite difference and finite element methods. The book is designed for beginning graduate students, upper level undergraduate students, students from interdisciplinary areas including engineers, and others who need to obtain such numerical solutions. The prerequisite is a basic knowledge of calculus, linear algebra, and ordinary differential equations. Some knowledge of numerical analysis and partial differential equations would also be helpful.

The emphasis is on the mathematical theory of finite difference and finite element methods and essential details in their implementation. Part I considers finite difference methods, and Part II is about finite element methods. In each part, we start with a comprehensive discussion of one-dimensional problems before proceeding to consider two or higher dimensions. We also list some useful references for those who wish to know more in related areas.

This is a textbook based on materials that the authors have used in teaching graduate courses on the numerical solution of differential equations. Most sample computer programming is written in Matlab. Some advantages of Matlab are its simplicity, a wide range of library subroutines, double precision accuracy, and many existing and emerging tool-boxes.

A web-site http://www4.ncsu.edu/~zhilin/FD_FEM_Book has been set up, to post or link computer codes accompanying this textbook.

We would like to thank Dr. Roger Hoskin, Lilian Wang, and Hongru Chen for proofreading the book.

Chapter 1

Introduction

1.1 Boundary value problems of differential equations

We discuss *numerical* solutions of problems involving ordinary differential equations (ODE) or partial differential equations (PDE), especially linear second order ODE and PDE, and problems involving systems of first order differential equations.

A *differential equation* involves the ordinary derivatives of an unknown function of one independent variable (say $u(x)$), or the partial derivatives of an unknown function of more than one independent variable (say $u(x, y)$, or $u(t, x)$, or $u(t, x, y, z)$ etc.). Differential equations have been used extensively to model many problems in fluid and solid mechanics, biology, material sciences, economics, ecology, sports and computer sciences.[1] Examples include the Laplace equation for potentials, the Navier-Stokes equations in fluid dynamics, biharmonic equations for stresses in solid mechanics, and the Maxwell equations in electro-magnetics. For more examples and for mathematical theory of partial differential equations, we refer the reader to [10] and references therein.

However, although differential equations have such wide applications, too few can be solved exactly in terms of elementary functions such as polynomials, $\log x$, e^x, trigonometric functions ($\sin x$, $\cos x$, ...), etc. and their combinations. Even if a differential equation can be solved analytically, considerable effort and sound mathematical theory are often needed, and the closed form of the solution may even turn out to be too messy to be useful. If the analytic solution of the differential equation is unavailable or too difficult to obtain, or takes some complicated form that is unhelpful to use, we may try to find an approximate solution. There are two traditional approaches:

- Semi-analytic methods. Sometimes we can use series, integral equations, perturbation techniques, or asymptotic methods to obtain an approximate solution expressed in terms of simpler functions.
- Numerical solutions. Discrete numerical values may represent the solution to a certain accuracy. Nowadays, these number arrays (and associated tables or plots) are obtained using computers, to provide the effective solution of many problems that were impossible to obtain before.

[1] There are other models in practice, for example statistical models.

In this book, we mainly adopt the second approach and focus on numerical solutions using computers, especially the use of finite difference or finite element methods for differential equations. In Fig.1.1, we show a flow chart of the problem solving process.

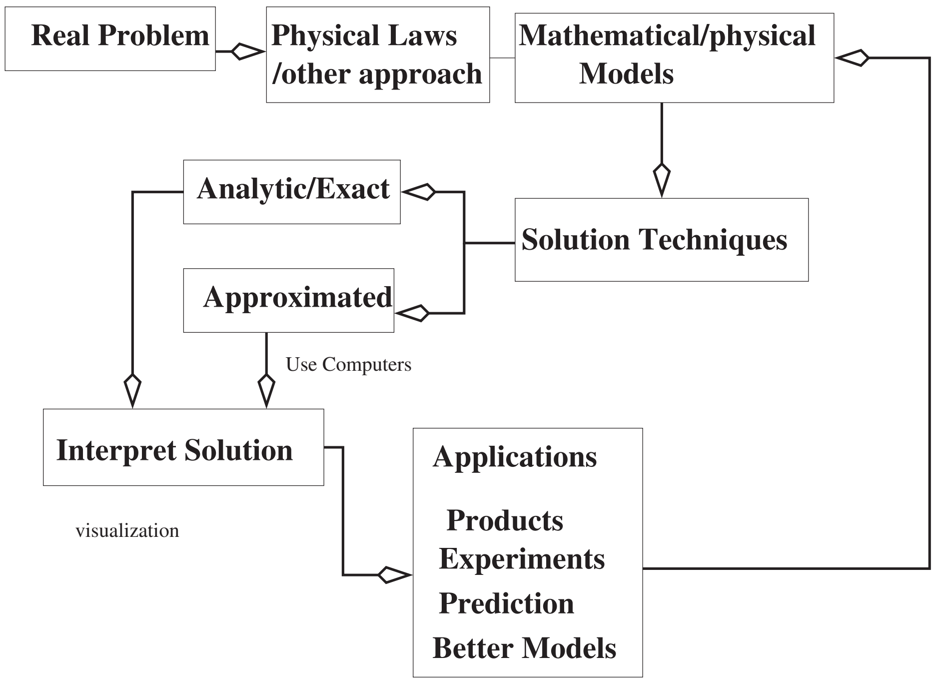

Figure 1.1. *A flow chart of a problem solving process.*

Some examples of ODE/PDE are as follows.

1. Initial value problems (IVP). The canonical first order system is

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0 \;; \tag{1.1}$$

and a single higher order differential equation may be rewritten as a first order system. For example, a second order ordinary differential equation

$$\begin{aligned} &u''(t) + a(t)u'(t) + b(t)u(t) = f(t), \\ &u(0) = u_0, \quad \boxed{u'(0) = v_0}. \end{aligned} \tag{1.2}$$

is converted into a first order system by setting $y_1(t) = u$ and $y_2(t) = u'(t)$.

An ODE IVP can often be solved using Runge-Kutta methods, with adaptive time steps. In Matlab, there is the ODE-SUITE which includes ode45, ode23, ode23s, ode15s, *etc.* For a stiff ODE system, either ode23s or ode15s is recommended.

2. Boundary value problems (BVP). An example of an ODE BVP is

$$\begin{aligned} & u''(x) + a(x)u'(x) + b(x)u(x) = f(x), \\ & u(0) = u_0, \quad u(1) = u_1; \end{aligned} \tag{1.3}$$

and a PDE BVP example is

$$\begin{aligned} & u_{xx} + u_{yy} = f(x,y), \quad (x,y) \in \Omega \\ & u(x,y) = u_0(x,y), \quad (x,y) \in \partial\Omega, \end{aligned} \tag{1.4}$$

in a domain Ω with boundary $\partial\Omega$. The above PDE is linear and classified as *elliptic*, and there are two other classifications for linear PDE, namely, *parabolic* and *hyperbolic*, as briefly discussed below.

3. Boundary and initial value problems, *e.g.*,

$$\begin{aligned} & u_t = a u_{xx} + f(x,t) \\ & u(0,t) = g_1(t), \quad u(1,t) = g_2(t), \qquad \text{BC} \\ & u(x,0) = u_0(x), \qquad \text{IC}. \end{aligned} \tag{1.5}$$

4. Eigenvalue problems, *e.g.*,

$$\begin{aligned} & u''(x) = \lambda u(x), \\ & u(0) = 0, \quad u(1) = 0. \end{aligned} \tag{1.6}$$

In this example, both the function $u(x)$ (the *eigenfunction*) and the scalar λ (the *eigenvalue*) are unknowns.

5. Diffusion and reaction equations, *e.g.*,

$$\frac{\partial u}{\partial t} = \nabla \cdot (\beta \nabla u) + \mathbf{a} \cdot \nabla u + f(u) \tag{1.7}$$

where $\mathbf{a}$ is a constant vector, $\nabla \cdot (\beta \nabla u)$ is a diffusion term, $\mathbf{a} \cdot \nabla u$ is called an advection term, and $f(u)$ a reaction term.

6. Systems of PDE. The incompressible Navier-Stokes model is an important nonlinear example:

$$\begin{aligned} & \rho\left(\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u}\right) = \nabla p + \mu \Delta \mathbf{u} + \mathbf{F}, \\ & \nabla \cdot \mathbf{u} = 0. \end{aligned} \tag{1.8}$$

In this book, we will consider *linear* PDE in either one dimension (1D) or two dimensions (2D). A 2D linear PDE has the general form

$$\begin{aligned} & a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy} \\ & \qquad + d(x,y)u_x + e(x,y)u_y + g(x,y)u(x,y) = f(x,y) \end{aligned} \tag{1.9}$$

where the coefficients are independent of $u(x,y)$ so the equation is linear in u and its partial derivatives. In the example above, the solution of the 2D linear PDE is sought in some bounded domain Ω, and the classification of the PDE form (1.9) is:

- Elliptic if $b^2 - ac < 0$ for all $(x, y) \in \Omega$,
- Parabolic if $b^2 - ac = 0$ for all $(x, y) \in \Omega$, and
- Hyperbolic if $b^2 - ac > 0$ for all $(x, y) \in \Omega$.

The appropriate solution method typically depends on the equation class. For the first order system

$$\frac{\partial \mathbf{u}}{\partial t} = A(\mathbf{x}) \frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \tag{1.10}$$

the classification is determined from the eigenvalues of the coefficient matrix $A(\mathbf{x})$.

Finite difference and finite element methods are suitable techniques to solve differential equations (ODE and PDE) numerically. There are other methods as well, for example, finite volume methods, collocation methods, and spectral methods, *etc.*

1.1.1 Main features of finite difference and finite element methods

Many problems can be solved numerically by some finite difference or finite element method. We strongly believe that any numerical analyst should be familiar with methods from both categories, and a brief list of some of their important features follows.

Finite difference (FD) methods:

- Often relatively simple to use, and quite easy to understand.
- Easy to implement for regular domains, *e.g.*, rectangular domains in Cartesian coordinates, and circular or annular domains in polar coordinates.
- Their discretization and approximate solutions are pointwise, and the fundamental mathematical tool is the Taylor expansion.
- There are many fast solvers and packages for regular domains, *e.g.*, the Poisson solvers Fishpack [1] and Clawpack [15].
- Difficult to implement for complicated geometries.
- Have strong regularity requirements (the existence of high order derivatives).

Finite element (FE) methods:

- Very successful for structural (elliptic type) problems.
- Suitable approach for problems with complicated boundaries.
- Sound theoretical foundation, at least for elliptic PDE, using Sobolev space theory.

- *Weaker* regularity requirements.
- Many commercial packages, *e.g.*, Ansys, Matlab PDE Tool-Box, Triangle, and PLTMG.
- Usually coupled with multigrid solvers.
- Mesh generation can be difficult, but there are now many packages that do this, *e.g.*, Matlab, Triangle, Pltmg, Fidap, and Ansys.

1.2 Further Reading

This text book provides an introduction to finite difference and finite methods. There are many other books for readers who wish to become expert in finite difference and finite element methods.

For FD methods, we recommend [12, 20, 16, 23, 24]. The text books [23, 24] are classical, while [12, 20, 16] are relatively new. With [16], the readers can find the accompanying Matlab codes from the author's website.

A classic book on FE methods is [8], while [14, 22] have been widely used as graduate text books. The series by Orden *et. al.* [6] not only presents the mathematical background of FE methods, but also gives some details on FE method programming in Fortran. Newer text books include [2, 3].