

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

Zhilin Li<sup>1</sup>

Zhonghua Qiao<sup>2</sup>

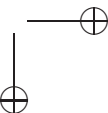
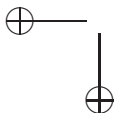
Tao Tang<sup>3</sup>

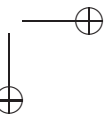
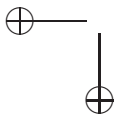
December 17, 2012

<sup>1</sup>Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA

<sup>2</sup>Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong

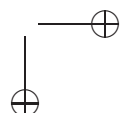
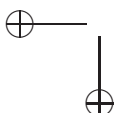
<sup>3</sup>Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong



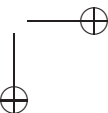
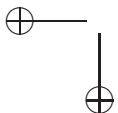


# Contents

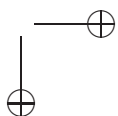
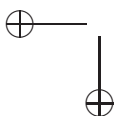
<b>Preface</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Boundary value problems of differential equations . . . . .	3
1.1.1 Main features of finite difference and finite element methods . . . . .	6
1.2 Further Reading . . . . .	7
<b>I Finite Difference Methods</b>	<b>9</b>
<b>2 Finite difference methods for one-dimensional problems</b>	<b>11</b>
2.1 A simple example of a finite difference method . . . . .	11
2.1.1 A Matlab code for the model problem . . . . .	13
2.1.2 Questions one may ask from this example . . . . .	15
2.2 Fundamentals of FD methods . . . . .	15
2.2.1 Forward, backward, and central FD formulas for $u'(x)$	16
2.2.2 Verification and grid refinement analysis . . . . .	18
2.3 Deriving FD formulas using the method of undetermined coeffi- cients . . . . .	20
2.3.1 FD formulas for second order derivatives . . . . .	21
2.3.2 FD formulas for higher order derivatives . . . . .	22
2.4 Consistency, stability, convergence, and error estimates of FD methods . . . . .	23
2.4.1 Global error . . . . .	23
2.4.2 Local truncation error . . . . .	23
2.4.3 The effect of round-off errors . . . . .	27
2.5 FD methods for 1-D self-adjoint elliptic equations . . . . .	28
2.6 FD methods for general 1D elliptic equations . . . . .	30
2.7 The ghost point method for boundary conditions involving deriva- tives . . . . .	30
2.7.1 A Matlab code of the ghost point method . . . . .	32
2.7.2 The Matlab driver program . . . . .	33
2.7.3 Dealing with mixed boundary conditions . . . . .	34



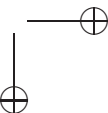
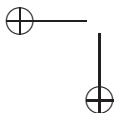
2.8	An example of a non-linear boundary value problem . . . . .	35
2.9	The grid refinement analysis technique . . . . .	37
2.10	Exercises . . . . .	40
<b>3</b>	<b>Finite difference methods for 2D elliptic PDE</b>	<b>43</b>
3.1	Boundary and compatibility conditions . . . . .	45
3.2	The central finite difference method for Poisson equations . . . . .	46
3.2.1	The matrix-vector form of the FD equations . . . . .	48
3.3	The maximum principle and error analysis . . . . .	50
3.3.1	The discrete maximum principle . . . . .	52
3.3.2	Error estimates of the FD method for Poisson equations . . . . .	53
3.4	Finite difference methods for general second-order elliptic PDE	54
3.5	Solving the resulting linear system of algebraic equations . . . . .	56
3.5.1	The Jacobi iterative method (solving the diagonals)	57
3.5.2	The Gauss-Seidel iterative method (solving the diagonals and using the most updated information)	58
3.5.3	The successive over-relaxation (SOR( $\omega$ )) method: an extrapolation technique . . . . .	59
3.5.4	Convergence of stationary iterative methods . . . . .	60
3.6	A fourth-order compact FD scheme for Poisson equations . . . . .	61
3.7	A finite difference method for Poisson equations in polar coordinates . . . . .	64
3.7.1	Treating polar boundary conditions . . . . .	65
3.7.2	Using the FFT to solve Poisson equations in polar coordinates . . . . .	66
3.8	Exercises . . . . .	66
<b>4</b>	<b>FD methods for linear parabolic PDE</b>	<b>69</b>
4.1	The Euler method . . . . .	71
4.1.1	Forward Euler method (FT-CT) . . . . .	72
4.1.2	The backward Euler method (BW-CT) . . . . .	73
4.2	The method of lines (MOL) . . . . .	74
4.3	The Crank-Nicolson scheme . . . . .	76
4.4	Stability analysis for time-dependent problems . . . . .	78
4.4.1	Review of the Fourier transform (FT) . . . . .	78
4.4.2	The discrete Fourier transform . . . . .	81
4.4.3	Definition of the stability of a FD scheme . . . . .	82
4.4.4	The von Neumann stability analysis for FD methods	83
4.4.5	Simplification of the von Neumann stability analysis for one-step time marching methods . . . . .	84
4.5	FD methods and analysis for 2D parabolic equations . . . . .	86
4.5.1	The backward Euler method (BW-CT) in 2D . . . . .	87
4.5.2	The Crank-Nicolson (C-N) scheme in 2D . . . . .	87
4.6	The alternating directional Implicit (ADI) method . . . . .	87
4.6.1	Implementation of the ADI algorithm . . . . .	88



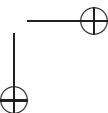
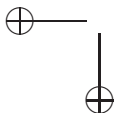
4.6.2	A pseudo-code of the ADI method in Matlab . . . . .	89
4.6.3	Consistency of the ADI method . . . . .	90
4.6.4	Stability analysis for the ADI method . . . . .	91
4.7	An implicit-explicit method for diffusion and advection equations	91
4.8	Solving elliptic PDE using numerical methods for parabolic PDE	92
4.9	Exercises . . . . .	92
<b>5</b>	<b>Finite difference methods for linear hyperbolic PDE</b>	<b>95</b>
5.1	FD methods . . . . .	96
5.1.1	Lax-Friedrichs method . . . . .	97
5.1.2	The upwind scheme . . . . .	98
5.1.3	The Leap-Frog scheme . . . . .	98
5.2	Modified PDE and numerical diffusion/dispersion . . . . .	100
5.3	The Lax-Wendroff scheme and other FD methods . . . . .	101
5.3.1	The Beam-Warming method . . . . .	103
5.3.2	The Crank-Nicholson scheme . . . . .	103
5.3.3	The method of lines (MOL) . . . . .	104
5.4	Numerical boundary conditions (NBC) . . . . .	104
5.5	Finite difference methods for second order linear hyperbolic PDE	105
5.5.1	A FD method (CT-CT) for the second order wave equation . . . . .	106
5.5.2	Transforming the second order wave equation to a first order system . . . . .	107
5.5.3	Initial and boundary conditions for the system . . . . .	107
5.6	Some commonly used FD methods for a linear system of hyperbolic PDE . . . . .	109
5.7	Finite difference methods for conservation laws . . . . .	110
5.8	Conservative FD methods for conservation laws . . . . .	111
5.9	Exercises . . . . .	113
<b>II</b>	<b>Finite Element Methods</b>	<b>115</b>
<b>6</b>	<b>Finite Element Methods for 1D Elliptic Problems</b>	<b>117</b>
6.1	The FE method for the 1D model . . . . .	117
6.2	Different mathematical formulations for the 1D model . . . . .	120
6.2.1	A physical example . . . . .	121
6.2.2	Mathematical equivalence . . . . .	122
6.3	Key components of the FE method for the 1D model . . . . .	124
6.3.1	Key components of any FE method . . . . .	124
6.3.2	The Ritz method . . . . .	128
6.3.3	Assembling the stiffness matrix element by element	129
6.4	Matlab programming of the FE method for the 1D model problem	133
6.4.1	Define the basis functions . . . . .	134
6.4.2	Define $f(x)$ . . . . .	135
6.4.3	The main FE routine . . . . .	135



	6.4.4	A test example . . . . .	136
6.5		Exercises . . . . .	138
<b>7</b>		<b>Theoretical Foundations of the Finite Element Method</b>	<b>141</b>
7.1		Functional spaces . . . . .	141
	7.1.1	Multi-dimensional spaces and multi-index notation . . . . .	142
7.2		Spaces for integral forms, $L^2(\Omega)$ and $L^p(\Omega)$ . . . . .	143
	7.2.1	The inner product in $L^2$ . . . . .	144
	7.2.2	The Cauchy-Schwartz inequality in $L^2(\Omega)$ . . . . .	145
	7.2.3	$L^p(\Omega)$ spaces . . . . .	145
7.3		Sobolev spaces and weak derivatives . . . . .	146
	7.3.1	Definition of weak derivatives . . . . .	146
	7.3.2	Definition of Sobolev spaces $H^m(\Omega)$ . . . . .	146
	7.3.3	Inner products in $H^m(\Omega)$ spaces . . . . .	147
	7.3.4	Relations between $C^m(\Omega)$ and $H^m(\Omega)$ — the Sobolev embedding theorem . . . . .	148
7.4		FE analysis for the 1D model problem . . . . .	148
	7.4.1	Conforming FE methods . . . . .	149
	7.4.2	FE analysis for a 1D Sturm-Liouville problem . . . . .	149
	7.4.3	The bilinear form . . . . .	150
	7.4.4	The FE method for the 1D Sturm-Liouville problem using piecewise linear basis functions in $H_0^1(x_l, x_r)$ . . . . .	150
	7.4.5	Local stiffness matrix and load vector using the hat basis functions . . . . .	152
7.5		Error analysis for the FE method . . . . .	153
	7.5.1	Interpolation functions and error estimates . . . . .	154
	7.5.2	Error estimates of the FE methods using the interpolation function . . . . .	156
	7.5.3	Error estimates in pointwise norm . . . . .	156
7.6		Exercises . . . . .	157
<b>8</b>		<b>Issues of the FE Method in one space dimensions</b>	<b>159</b>
8.1		Boundary Conditions . . . . .	159
	8.1.1	Mixed BC . . . . .	160
	8.1.2	Non-homogeneous Dirichlet BC . . . . .	161
8.2		The FE method for Sturm-Liouville problems . . . . .	162
	8.2.1	Numerical treatments of Dirichlet BC . . . . .	163
	8.2.2	Contributions from Neumann or mixed BC . . . . .	163
	8.2.3	Pseudo-code of the FE method for the Sturm-Liouville problem, using the piecewise linear basis functions. . . . .	164
8.3		High order elements . . . . .	165
	8.3.1	Piecewise quadratic basis functions . . . . .	165
	8.3.2	Assembling the stiffness matrix and the load vector . . . . .	167
	8.3.3	The cubic basis functions in $H^1(x_l, x_r)$ space . . . . .	170
8.4		A 1D Matlab FE package . . . . .	170
	8.4.1	Gaussian quadrature formulas . . . . .	170



	8.4.2	Shape functions . . . . .	173
	8.4.3	The main data structure . . . . .	176
	8.4.4	Outline of the algorithm . . . . .	177
	8.4.5	Assembling element by element . . . . .	177
	8.4.6	The treatment of BC . . . . .	180
	8.4.7	An example . . . . .	181
	8.5	The FE method for fourth order ODEs . . . . .	182
	8.5.1	The FE method . . . . .	184
	8.5.2	The shape functions . . . . .	186
	8.6	The Lax-Milgram Lemma and the existence of FE solutions . .	186
	8.6.1	General settings: Assumptions, and conditions . . .	188
	8.6.2	The Lax-Milgram Lemma . . . . .	188
	8.6.3	An example of the Lax-Milgram theorem . . . . .	190
	8.6.4	Abstract FE methods . . . . .	192
	8.7	Exercises . . . . .	194
<b>9</b>	<b>The Finite Element Method for 2D Problems</b>		<b>199</b>
	9.1	The second Green's theorem and integration by parts in 2D . .	199
	9.1.1	Boundary conditions . . . . .	201
	9.1.2	Weak form of the second order self-adjoint elliptic PDE . . . . .	201
	9.1.3	Verification of the conditions of the Lax-Milgram Lemma . . . . .	202
	9.2	Triangulation and basis functions . . . . .	203
	9.2.1	Triangulation and mesh parameters . . . . .	203
	9.2.2	FE space of piecewise linear functions over a mesh .	203
	9.2.3	Global basis functions . . . . .	207
	9.2.4	The interpolation function and error analysis . . . .	209
	9.2.5	Error estimates of the FE solution . . . . .	213
	9.3	Transforms, shape functions, and quadrature formulas . . . . .	214
	9.3.1	Quadrature formulas . . . . .	215
	9.4	Some implementation details . . . . .	215
	9.4.1	Description of a triangulation . . . . .	215
	9.4.2	Outline of the FE algorithm using the piecewise lin- ear basis functions . . . . .	217
	9.5	Simplification of the FE method for Poisson equations . . . . .	219
	9.5.1	Main pseudo-code of the simplified FE method . . .	221
	9.6	Some FE spaces in $H^1(\Omega)$ and $H^2(\Omega)$ . . . . .	223
	9.6.1	The piecewise quadratic functions space . . . . .	225
	9.6.2	Cubic basis functions in $H^1 \cap C^0$ . . . . .	228
	9.6.3	Basis functions in $H^2 \cap C^1$ . . . . .	229
	9.6.4	Finite element spaces on quadrilaterals . . . . .	231
	9.6.5	Other FE spaces . . . . .	233
	9.7	The FE method for parabolic problems . . . . .	233
	9.8	Exercises . . . . .	236
	9.9	Matlab PDE-Toolbox Lab Exercises . . . . .	237

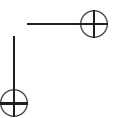
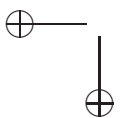


**Bibliography**

**241**

**Index**

**243**



# Preface

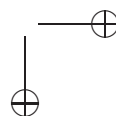
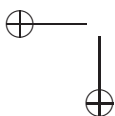
The purpose of this book is to provide an introduction to the numerical solution of partial differential equations using finite difference and finite element methods. The book is designed for beginning graduate students, upper level undergraduate students, students from interdisciplinary areas including 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.

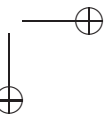
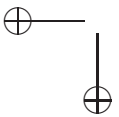
The emphasis is on the mathematical theory of finite difference and finite element methods and essential details in their implementation. Part I considers finite difference methods, and Part II is about finite element methods. In each part, we start with a comprehensive discussion of one-dimensional 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](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.

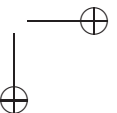
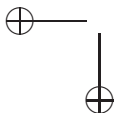
A *differential equation* involves the ordinary derivatives of an unknown function of one independent variable (say  $u(x)$ ), or the partial derivatives of an unknown function of more than one independent variable (say  $u(x, y)$ , or  $u(t, x)$ , or  $u(t, x, y, z)$  etc.). Differential equations have been used extensively to model many problems in fluid and solid mechanics, biology, material sciences, economics, ecology, sports and computer sciences.<sup>1</sup> 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:

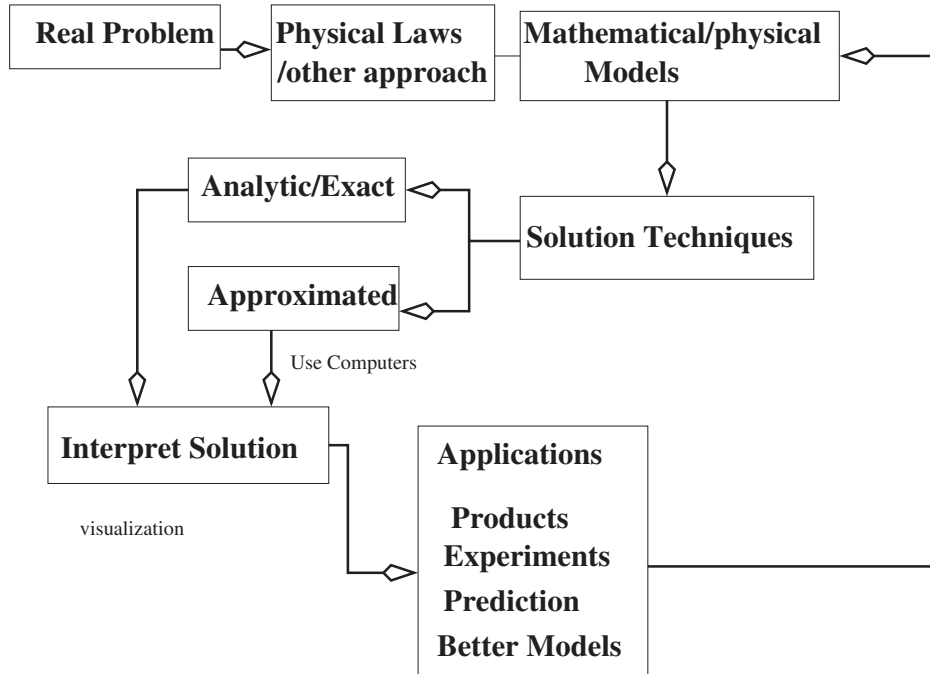
- Semi-analytic methods. Sometimes we can use series, integral equations, perturbation techniques, or asymptotic methods to obtain an approximate solution expressed in terms of simpler functions.
- Numerical solutions. Discrete numerical values may represent the solution to a certain accuracy. Nowadays, 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.

---

<sup>1</sup>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

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

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

$$u''(t) + a(t)u'(t) + b(t)u(t) = f(t), \quad (1.2)$$

$$u(0) = u_0, \quad \boxed{u'(0) = v_0}.$$

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  $\Omega$  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.,

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

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  $\lambda$  (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.

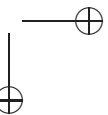
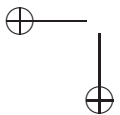
6. Systems of PDE. The incompressible Navier-Stokes model is an important nonlinear example:

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

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

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

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  $\Omega$ , 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

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

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

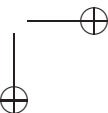
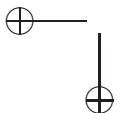
Many problems can be solved numerically by some finite difference or finite element method. We strongly believe that any numerical analyst should be familiar with methods from both categories, and a brief list of some of their important features follows.

#### 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].

