Chapter 3

Finite Difference Methods

3.1 Introduction

The finite difference methods have dominated computational science since its inception and were *the* method of choice in the 1960s and 1970s. While other methods, such as the finite element method and boundary element have enjoyed recent popularity, finite difference methods are still utilized for a wide array of computational engineering and science problems.

Briefly, the finite difference method can be characterized as follows. **Finite Difference Method**

- Utilizes uniformly spaced grids.
- At each node, each derivative is approximated by an algebraic expression which references the adjacent nodes.
- A system of algebraic equations is obtained by evaluating the previous step for each node.
- The system is solved for the dependent variable.

3.2 Taylor Series

Before proceeding to discuss the mathematical formalities of the finite difference method, it would probably be a good idea to review some facts concerning Taylor's series. From elementary calculus, we remember that the Taylor series formula is used to expand a function, $u(x)$ as a power series,

$$u(x) = u(a) + \frac{(x-a)}{1!}u'(a) + \frac{(x-a)^2}{2!}u''(a) + \ldots = \sum_{n=0}^{\infty} \frac{(x-a)^n}{n!}u^{(n)}(a) \tag{3.1}$$

For the numerical treatment of field equations, we often represent the Taylor series in somewhat different forms. Usually, we replace x and a in the previous formula by $x+h$ and x respectively. Here, h is denoted as an increment in x and is assumed positive. The Taylor series now becomes

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{(h)^2}{2!}u''(x) + \ldots = \sum_{n=0}^{\infty} \frac{(h)^n}{n!}u^{(n)}(x) \tag{3.2}$$

For the treatment of many problems, it is convenient to take only the first two terms of the right hand side of the previous equation,

$$u(x+h) = u(x) + hu'(x) + O(h^2) \tag{3.3}$$

where the expression $O(h^2)$, represents the error of the approximation is proportional to h^2. From this relationship, we can easily define what is known as the first order, forward difference approximant to $\frac{du}{dx}$,

$$\frac{du}{dx} = \frac{u(x+h) - u(x)}{h} + O(h) \tag{3.4}$$

Likewise, we can define the first order, backward difference approximant to $\frac{du}{dx}$ (just substitute $-h$ for h in the expansion),

$$\frac{du}{dx} = \frac{u(x) - u(x-h)}{h} + O(h) \tag{3.5}$$

If instead, we take terms up to and including h^3 on the right hand side of the two previous equations, then subtract them, we can obtain,

$$\frac{du}{dx} = \frac{u(x+h) - u(x-h)}{2h} + O(h^3). \tag{3.6}$$

This is the second order, central difference approximation to $\frac{du}{dx}$. Taking up to forth order yields,

$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + O(h^4) \tag{3.7}$$

the terms h and h^3 having canceled. Now we can obtain the second order, central difference approximation to $\frac{d^2u}{dx^2}$

$$\frac{d^2u}{dx^2} = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2). \tag{3.8}$$

Now that were warming up, let's see what excitement awaits us by considering the Taylor series for a function of two independent variables, $u(x,t)$,

$$u(x+h,t) = u(x,t) + \frac{h}{1!}\frac{\partial u(x,t)}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 u(x,t)}{\partial x^2} + \ldots \tag{3.9}$$

likewise,

$$u(x-h,t) = u(x,t) - \frac{h}{1!}\frac{\partial u(x,t)}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 u(x,t)}{\partial x^2} + \dots \tag{3.10}$$

thus, we can generate first and second (and higher) order approximations for partial derivatives, for example:

$$\frac{\partial u}{\partial x} = \frac{u(x+h,t) - u(x,t)}{h} + O(h) \tag{3.11}$$

$$\frac{\partial u}{\partial x} = \frac{u(x,t) - u(x-h,t)}{h} + O(h) \tag{3.12}$$

$$\frac{\partial u}{\partial x} = \frac{u(x+h,t) - u(x-h,t)}{2h} + O(h^2) \tag{3.13}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h} + O(h^2). \tag{3.14}$$

It is easy to show that there are equivalent expressions for time,

$$\frac{\partial u}{\partial t} = \frac{u(x,t+l) - u(x,t)}{l} + O(l) \tag{3.15}$$

as well as for other spacial variables and mixed partial derivatives,

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} &= \frac{u(x+h,y+k) - u(x+h,y-k) - u(x-h,y+k) + u(x-h,y-k)}{4hk} + \\ &\quad O(\frac{(h+k)^4}{hk}) \end{aligned} \tag{3.16}$$

Alright, now that we know everything about how to derive difference formulations from Taylors series, let's talk about how to use such results in approximating a field problem.

3.3 Finite difference formulation for Elliptic Equations

As a simple illustration, let's consider Laplace's equation for a planar region, Ω with Dirichlet boundary data on the boundary of the region, $\partial\Omega$. To make this even simpler still, consider the region to be bounded by the lines $x = 0$, $x = 1$, $y = 0$, and $y = 1$. We now discretize the bounded region with a square mesh as show in Figure 3.3. We note that we can use non-square elements, but for now, squares will suffice. We might ask ourselves the pertinant question, "how many squares do we use to discretize the solution domain?" While we intuitively know that our solution will probably get better as the discretization level increases, how fast it will get better and how much it will cost are other questions to think about. We impose a mesh of grid points and define h to be the distance between mesh points. The **interior** grid points are given by $(x_i, y_j) = (ih, jh)$, where $i, j = 1, \dots, N$. Now we look up

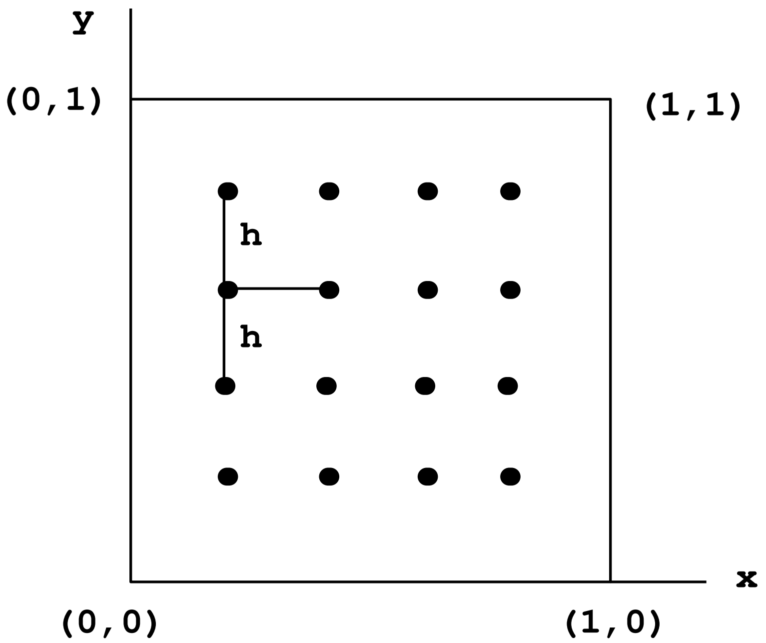

Figure 3.1: Discretization for Laplace's Equation.

above for the Taylors series expansion which represents second order approximations for x and y and substitute them in:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{3.17}$$

$$\nabla^2 u = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{(\Delta x)^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{(\Delta y)^2} = 0. \tag{3.18}$$

As we mentioned before, we chose a uniform grid such that $\Delta x = \Delta y = h$. This yields the simplification,

$$\nabla^2 u_{i,j} = \frac{1}{h^2}[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}] = 0 \tag{3.19}$$

Which is the finite difference approximation of Laplace's equation. If we had carried a higher order term through the calculations, we would have found out that the local truncation error is

$$\frac{1}{12}h^2(\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4})_{i,j} \tag{3.20}$$

which is usually written as $O(h^2)$, such that the error is proportional to the space constant squared. Note that five points are involved in the approximation of the Laplacian (points to the left, right, above and below of a central point, (x_i, y_j)) such that this is called the *five point* formula.

Since for the Dirichlet problem we know the values of the mesh points (nodes) on the region of the boundary, we are left applying the approximation to N^2 internal mesh points.

This yields a linear system of equations of N^2 equations of the function $u_{i,j}$. We usually write the system as:

$$A\mathbf{u} = \mathbf{b} \tag{3.21}$$

where A is an $N^2 \times N^2$ matrix of block tridiagonal form:

$$A = \begin{pmatrix} T_N & I_N & & & \\ I_N & T_N & \ddots & & \\ & & \ddots & & \\ & & \ddots & & I_N \\ & & & I_N & T_N \end{pmatrix} \tag{3.22}$$

Here, I_N is the $N \times N$ identity matrix and T_N is the $N \times N$ tridiagonal matrix,

$$T_N = \begin{pmatrix} -4 & 1 & & & \\ 1 & -4 & \ddots & & \\ & & \ddots & & \\ & & \ddots & & 1 \\ & & & 1 & -4 \end{pmatrix} \tag{3.23}$$

The vector $\mathbf{u}$ consists of the unknown values at the nodes and is of dimension $N^2 \times 1$, while $\mathbf{b}$ is the $N^2 \times 1$ vector consisting of boundary data (with $4N - 4$ non-zero elements).

Thus the solution to our problem can be found by solving for the vector u,

$$\mathbf{u} = A^{-1}\mathbf{b} \tag{3.24}$$

since A is nonsingular, we will obtain a unique solution.

3.3.1 Laplace's Equation – A Numerical Example

To illustrate the method, consider the steady-state heat conduction problem defined over our unit square. We want to find the finite difference approximation to Laplace's equation subject to the Dirichlet boundary conditions,

$$u(x,0) = u_0 = 300; \qquad u(1,y) = u(x,1) = u(0,y) = 0 \tag{3.25}$$

If we discretize our square into a uniform mesh with $h = 0.25$ as shown in Figure 3.3.1. The nodes are numbered from $i,j = 0,1,\ldots,N+1 = 4$, where nodes $i,j = 1,2,3$ represent the interior nodes which we are solving for. We can apply the finite difference memthod using equation 3.19. Writing out a few of the nodes, we obtain for $u_{1,1}$

$$u_{2,1} + u_{0,1} + u_{1,2} + u_{1,0} - 4u_{1,1} = 0 \tag{3.26}$$

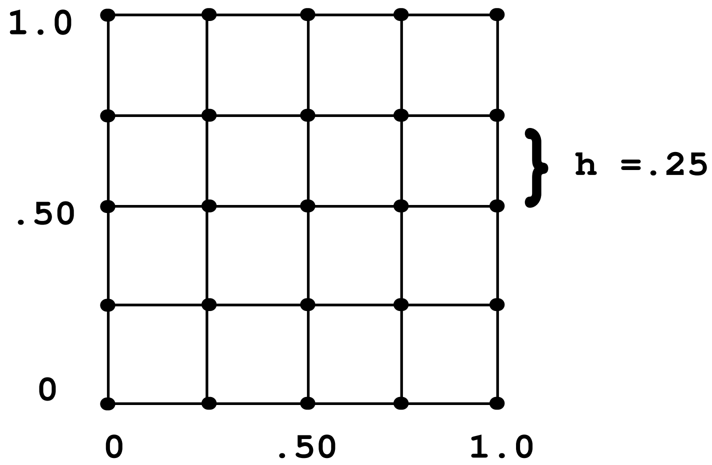

Figure 3.2: Mesh used for the numerical solution of the steady-state heat equation.

and for $u_{1,2}$

$$u_{2,2} + u_{0,2} + u_{1,3} + u_{1,1} - 4u_{1,2} = 0 \tag{3.27}$$

We continue the process for all nine interior nodes numbered from the left corner, proceeding along the x-axis, $u_{0,0}, u_{1,0}, u_{2,0}, u_{3,0}, u_{0,1}, u_{1,1}, u_{2,1}, u_{3,1} \ldots u_{3,3}$ and obtain the following linear system of equations (notice that we have multiplied through equations 3.26 and 3.27 by a negative 1):

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{31} \\ u_{12} \\ u_{22} \\ u_{32} \\ u_{13} \\ u_{23} \\ u_{33} \end{pmatrix} = \begin{pmatrix} 300 \\ 300 \\ 300 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.28}$$

Upon solving the linear system, we obtain a vector **u** of the temperatures. It is interesting to look at the distributions of temperature as a function of the number of elements (and thus step-size, h). The solution of the system shown in figure 3.3.1 and equation 3.28, we obtain the temperature distribution in 3.3.1. You'll note that the solution does not look very smooth. This is because we a fair amount of error due to inadequately discretizing the solution domain. As we increase the number of elements (and thus decrease the step-size), we are able to reduce the discretization area. Figures 3.3.1, 3.3.1, and 3.3.1 show the iso-tempurature contours for succesive double of the number of elements along the x and y

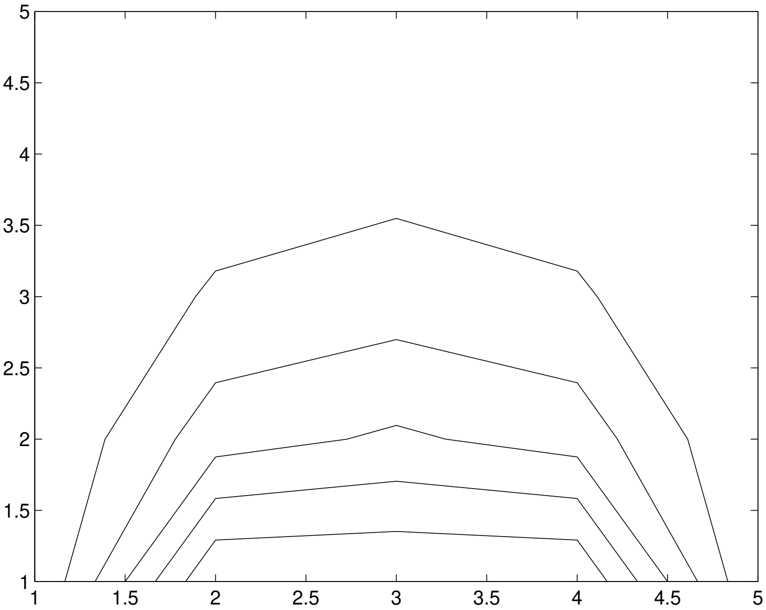

Figure 3.3: Temperature distribution, shown as iso-temperature contours for the solution of Laplace's equation from example 3.3.1. Here we have divided up the solution domain into 8 panels with $h = 0.25$.

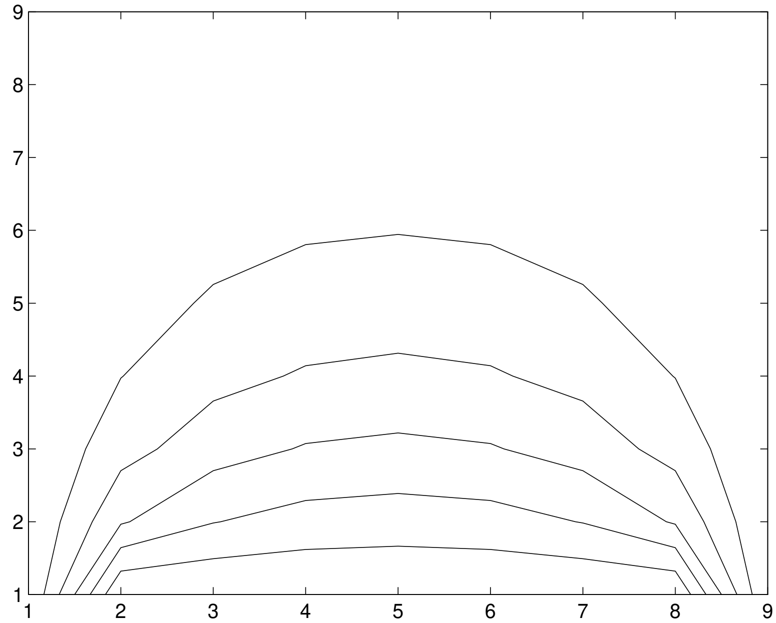

Figure 3.4: Iso-temperature contours for the solution of Laplace's equation from example 3.3.1. Here the solution domain is divided into 64 panels with $h = 0.125$.

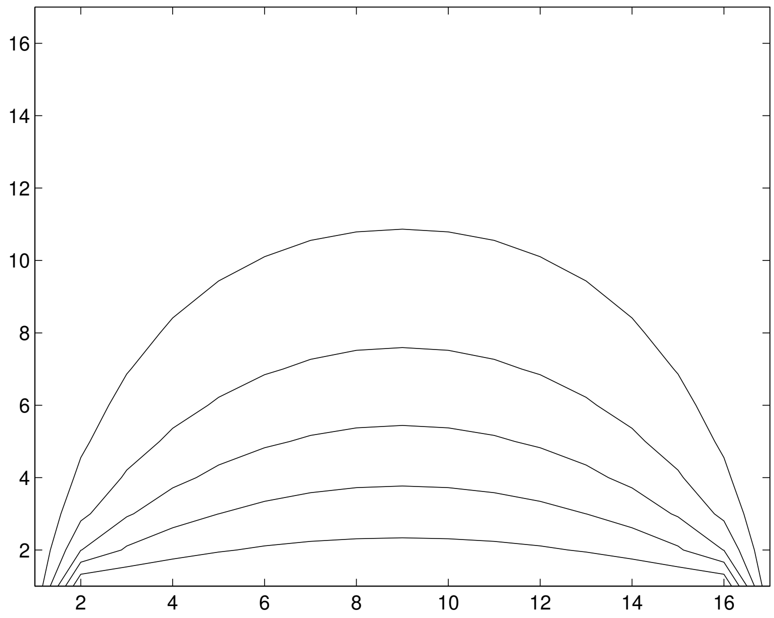

Figure 3.5: Iso-temperature contours for the solution of Laplace's equation from example 3.3.1. Here the solution domain is divided into 256 panels with $h = 0.0625$.

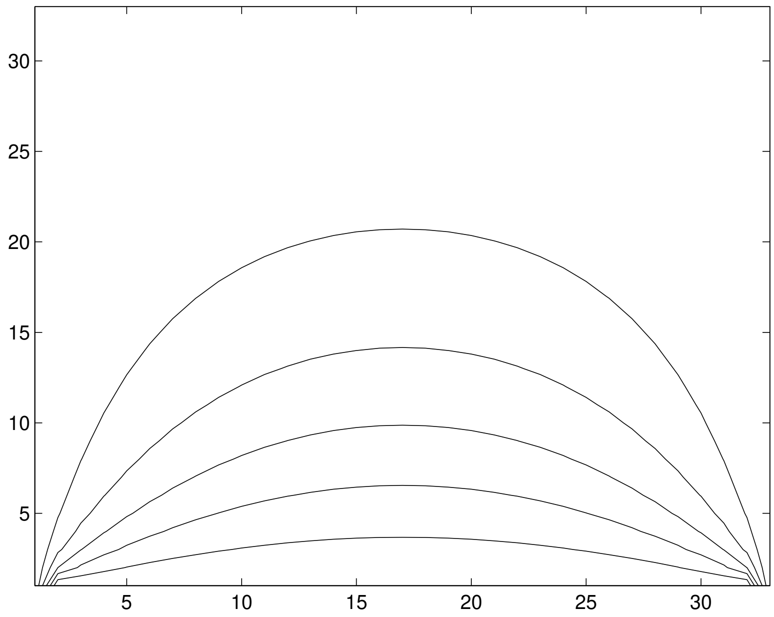

Figure 3.6: Iso-temperature contours for the solution of Laplace's equation from example 3.3.1. Here the solution domain is divided into 1024 panels with $h = 0.03125$.

axes. As you can see, the solutions become smoother as we increase the number of elements (although there is still a problem near the bottom corners). Note that we have significantly increased the computational costs to obtain a more accurate solution. We started with a 9×9 system of equations, and ended up using a 30×30 system of equations. Since the solution of a system of equations is $O(N^3$, we increased the number of operations by a factor of 37 and our storage costs by a factor of 100 (assuming double precision floats).

We have increased the solution accuracy by uniformly dividing the number of elements into smaller paritions. However, as you might imagine, this isn't necessarily the most computationally efficient way to decrease the discretization error. It would seemingly make sense to only increase the number of elements in the regions with the largest amount of error. But how do we do that if the finite different methodology utilizes uniform partitions? One answer is to use overlapping grids. We will address this method later on in the section on Adaptive Methods.

I note that this problem can be solved analytically using a Fourier series [21]:

$$u(x,y) = \frac{4u_0}{\pi} \sum_{n=0}^{\infty} \frac{1}{2n+1} \sin((2n+1)\pi x) sinh[(1-y)(2n+1)\pi] cosech(2n+1). \tag{3.29}$$

thus we can compare our finite difference solution with the solution generated by the Fourier series [21]:

Node	Fourier Solution	Finite Difference Solution	Absolute Error
(1,1)	129.608	128.571	-1.0
(2,1)	162.159	158.036	-4.1
(1,2)	54.608	56.250	1.6
(2,2)	75.000	75.000	0.0
(1,3)	20.391	21.429	1.0
(2,3)	28.624	29.464	0.8

Table 3.1: Comparison of F.D. and Fourier Solutions

3.3.2 Poisson's Equation

Before going on, let's take a minute to extend our method to including solving Poisson's equation. The Poisson equation has a right hand side which does not equal zero,

$$\nabla^2 u = f(x,y) \tag{3.30}$$