Finite Difference Method for the Solution of Laplace Equation

Ambar K. Mitra
Department of Aerospace Engineering
Iowa State University

Introduction

Laplace Equation is a second order partial differential equation (PDE) that appears in many areas of science an engineering, such as electricity, fluid flow, and steady heat conduction. Solution of this equation, in a domain, requires the specification of certain conditions that the unknown function must satisfy at the boundary of the domain. When the function itself is specified on a part of the boundary, we call that part the Dirichlet boundary; when the normal derivative of the function is specified on a part of the boundary, we call that part the Neumann boundary. In a problem, the entire boundary can be Dirichlet or a part of the boundary can be Dirichlet and the rest Neumann. A problem with Neumann condition specified on the entire boundary does not have a unique solution. In some problems, a linear combination of the function and its normal derivative is specified; such situations are called Robin boundary. We will not deal with the Robin problem, but it is fairly straightforward to extend the method described here to these problems. A typical Laplace problem is schematically shown in Figure-1. In domain *D*,

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$$

and on the boundary

$$\phi = f \quad on \quad S_D \quad and \quad \frac{\partial\phi}{\partial n} = g \quad on \quad S_N$$

where n is the normal to the boundary, S_D is the Dirichlet boundary, and S_N is the Neumann boundary.

In this paper, the finite-difference-method (FDM) for the solution of the Laplace equation is discussed. In this method, the PDE is converted into a set of linear, simultaneous equations. When the simultaneous equations are written in matrix notation, the majority of the elements of the matrix are zero. Such matrices are called ”sparse matrix”. However, for any meaningful problem, the number of simultaneous equations becomes very large, say of the order of a few thousand. There are special purpose routines that deal with very large, sparse matrices. Furthermore, one needs skillful ways of storing such large matrices, otherwise, several Gigabits will be used up just for the storing.

An alternative way of solving very large system of simultaneous equations is iterative. The advantage of iterative solution is that the storing of large matrices is unnecessary. In this paper, we will demonstrate the use of one such iterative technique. We will also explore one way of accelerating the convergence of the iterative method.

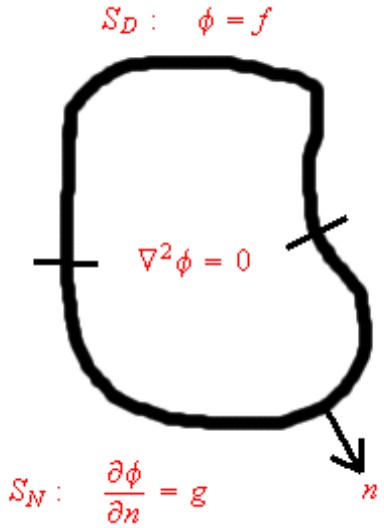

Figure-1: Laplace problem.

Finite Difference (FD)

Consider three points on the x-axis separated by a distance h, as shown in Figure-2a. For convenience, we have labeled these points as $i-1, i,$ and $i+1$. Let the value of a function $\phi(x,y)$ at these three points be $\phi_{i-1}, \phi_i,$ and ϕ_{i+1}. Now we can write two Taylor expansions for ϕ_{i-1} and ϕ_{i+1} as follows.

$$\phi_{i-1} = \phi_i - \frac{\partial\phi}{\partial x}|_i h + \frac{\partial^2\phi}{\partial x^2}|_i \frac{h^2}{2!} - \frac{\partial^3\phi}{\partial x^3}|_i \frac{h^3}{3!} + \frac{\partial^4\phi}{\partial x^4}|_i \frac{h^4}{4!} + O(h^5) \tag{1}$$

$$\phi_{i+1} = \phi_i + \frac{\partial\phi}{\partial x}|_i h + \frac{\partial^2\phi}{\partial x^2}|_i \frac{h^2}{2!} + \frac{\partial^3\phi}{\partial x^3}|_i \frac{h^3}{3!} + \frac{\partial^4\phi}{\partial x^4}|_i \frac{h^4}{4!} + O(h^5) \tag{2}$$

where $|_i$ means that the derivative is computed at the point i. Adding Eqns(1,2), we get

$$\phi_{i-1} + \phi_{i+1} = 2\phi_i + \frac{\partial^2\phi}{\partial x^2}|_i h^2 + \frac{\partial^4\phi}{\partial x^4}|_i \frac{h^4}{12} + O(h^5)$$

After some re-arrangement, we write

$$\frac{\partial^2\phi}{\partial x^2}|_i = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2} + O(h^2) \tag{3}$$

The right-hand-side of Eqn.(3) is the 2nd order accurate FD approximation of $\frac{\partial^2\phi}{\partial x^2}|_i$. The expression is second order accurate because the error is of the order of h^2.

Subtracting Eqn.(1) from Eqn.(2), we get

$$\phi_{i+1} - \phi_{i-1} = 2\frac{\partial\phi}{\partial x}|_i h + \frac{\partial^3\phi}{\partial x^3}|_i \frac{h^3}{3} + O(h^5)$$

or

$$\frac{\partial \phi}{\partial x}|_i = \frac{\phi_{i+1} - \phi_{i-1}}{2h} + O(h^2) \tag{4}$$

Eqn.(4) is the 2nd order accurate FD approximation of $\frac{\partial \phi}{\partial x}|_i$.

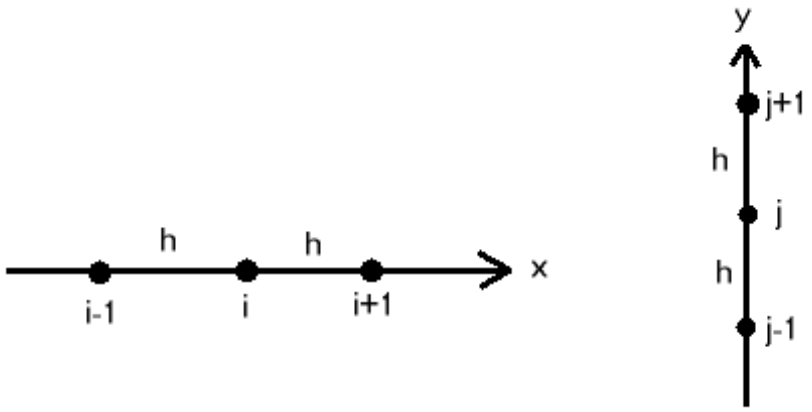

Figure-2a,2b : Finite Differencing along x and y

In a similar manner, we can take three points $j-1, j$, and $j+1$ along the y-axis, as shown in Figure-2b. For this arrangement of points, we can write

$$\frac{\partial^2 \phi}{\partial y^2}|_j = \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{h^2} + O(h^2) \tag{5}$$

and

$$\frac{\partial \phi}{\partial y}|_j = \frac{\phi_{j+1} - \phi_{j-1}}{2h} + O(h^2) \tag{6}$$

By superposing the arrangements of Figure-2a,2b , we arrive at the 5-point stencil of Figure-3 and ϕ acquires two subscripts, one for the i or x direction and the other for the j or y direction. By combining the Eqns.(3,5), we write

$$\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) |_{i,j} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2} + \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{h^2} \tag{7}$$

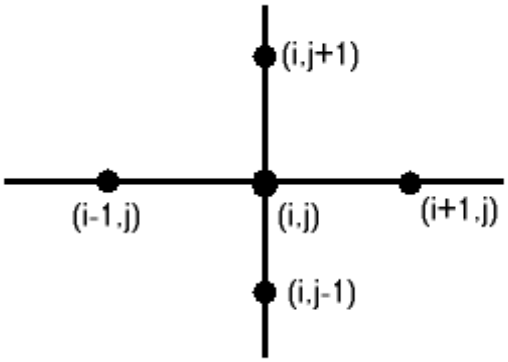

Figure-3 : 5-point stencil for Laplace equation.

By substituting Eqn.(7) in the Laplace equation, we find

$$\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j} + \phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1} = 0$$

or

$$\phi_{i,j} = \frac{1}{4}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) \tag{8}$$

Eqn.(8) is a wonderful result that leads to:

Corollary *: If ϕ satisfies Laplace equation, then ϕ, at any point in the domain D, is the average of the values of ϕ at the four surrounding points in the 5-point stencil of Figure-3.*

This corollary is the basis of the iterative method.

We need to make a small modification in Eqn.(8) when we wish to solve the Poisson equation

$$\nabla^2\phi = F(x,y)$$

where $F(x,y)$ is a known function. In this situation, Eqn.(8) is modified to

$$\phi_{i,j} = \frac{1}{4}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) - \frac{h^2}{4}F_{i,j} \tag{9}$$

Dirichlet Problem

Consider the simple problem of Figure-4 posed in a box with only four interior points. ϕ is given on the East, West, North, and South walls. Thus, $\phi(1,2)$, $\phi(1,3)$, $\phi(2,4)$, $\phi(3,4)$, $\phi(4,3)$, $\phi(4,2)$, $\phi(3,1)$, and $\phi(2,1)$ are known. We have to calculate the values of $\phi(2,2)$, $\phi(3,2)$, $\phi(2,3)$, and $\phi(3,3)$. We begin the iterative process by assuming

$$\phi^{(0)}(2,2) = \phi^{(0)}(3,2) = \phi^{(0)}(2,3) = \phi^{(0)}(3,3) = 0 \tag{10}$$

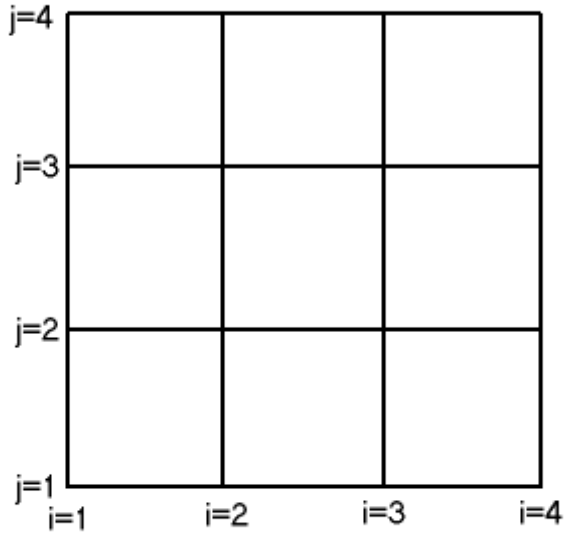

Figure-4: Dirichlet problem on a 4 by 4 box.

The superscript (0) is an iteration counter. Starting at bottom left, we write

$$\begin{aligned}
\phi^{(1)}(2,2) &= \frac{1}{4}[\phi(1,2) + \phi^{(0)}(3,2) + \phi(2,1) + \phi^{(0)}(2,3)] \\
\phi^{(1)}(3,2) &= \frac{1}{4}[\phi^{(1)}(2,2) + \phi(4,2) + \phi(3,1) + \phi^{(0)}(3,3)] \\
\phi^{(1)}(2,3) &= \frac{1}{4}[\phi(1,3) + \phi^{(0)}(3,3) + \phi^{(1)}(2,2) + \phi(2,4)] \\
\phi^{(1)}(3,3) &= \frac{1}{4}[\phi^{(1)}(2,3) + \phi(4,3) + \phi^{(1)}(3,2) + \phi(3,4)]
\end{aligned} \tag{11}$$

In the first line of Eqn.(11), we compute the first iterated value of $\phi^{(1)}(2,2)$. Note that, as soon as this first iterate becomes available, it is used in the calculation of the first iterate of $\phi^{(1)}(3,2)$, in the second line of Eqn.(11). This is very easily accomplished by storing $\phi^{(0)}(2,2)$ and $\phi^{(1)}(2,2)$ in the same memory location, thereby overwriting $\phi^{(0)}(2,2)$ with $\phi^{(1)}(2,2)$. Therefore, for any location (i,j), $\phi^{(iteration+1)}(i,j)$ is overwritten on $\phi^{(iteration)}(i,j)$.

However, there is one little calculation that you need to do before you overwrite. First, store $\phi^{(iteration+1)}(i,j)$ in *temp*. Then compute the relative error at (i,j) as

$$\epsilon(i,j) = \left| \frac{temp - \phi^{(iteration)}(i,j)}{temp} \right| \tag{12}$$

Then overwrite $\phi^{(iteration)}(i,j)$ by *temp*. In this manner, we will keep track of the relative errors at all the (i,j) locations. Then at the end of the iteration loop ($iteration + 1$), we determine a representative relative error as

$$\epsilon_{\max} = \max \textit{ of all } \quad \epsilon(i,j) \tag{13}$$

At this point, you need to think about something critical. Do you really need to store all the $\epsilon(i,j)$ in order to calculate $\epsilon_{\max}$? Or can you keep updating the value of $\epsilon_{\max}$ as you travel from point to point within an iteration loop. For a large problem, storing all the $\epsilon(i,j)$ can become a

very bad idea from the point of view of memory usage.

When $\epsilon_{\max}$ becomes available, compare this with the user-specified tolerance 10^{-n} for n digit accuracy. If $\epsilon_{\max}$ is larger than the specified tolerance, you need to go through another iteration loop. Otherwise, the iteration is stopped and the "converged" solution for ϕ is outputted.

It is a good idea to keep track of the iteration count, because it is a measure of the cost. Secondly, you must specify the "*maximum iteration allowed*". Otherwise, the program may get caught in a never-ending iteration loop due to some error in the specification of data.

Now, let us consider the large problem of Figure-5 and fix our ideas regarding various do-loops that we will use in the program. The grid for this problem is defined by $1 \leq i \leq N$ and $1 \leq j \leq M$. The east, west, north, and south walls are Dirichlet boundaries. The boundary conditions $\{\phi(1,j); j = 2, M-1\}$, $\{\phi(N,j); j = 2, M-1\}$, $\{\phi(i,1); i = 2, N-1\}$, and $\{\phi(i,M); i = 2, N-1\}$ are inserted as data.

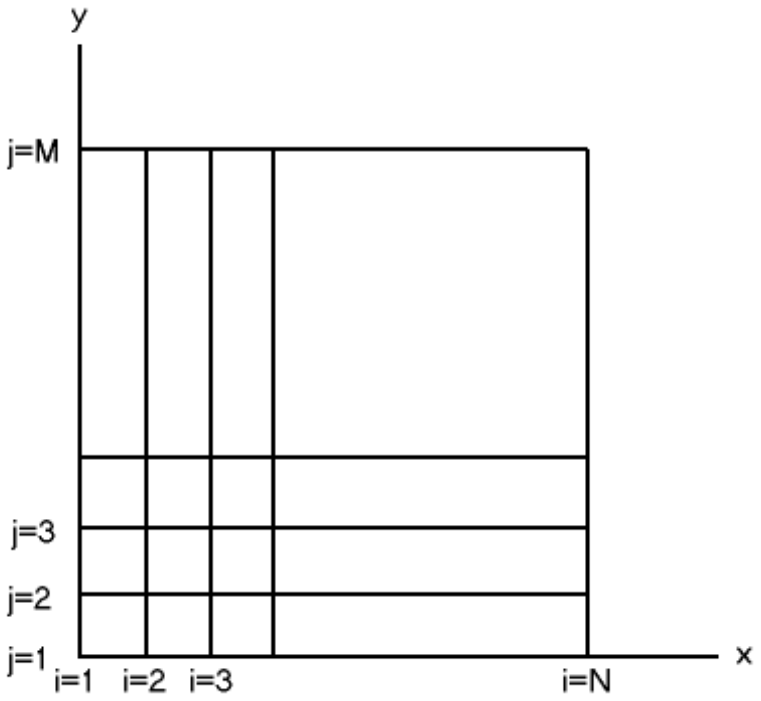

Figure-5: Dirichlet problem on a generalized grid.

The algorithm for the iterative solution is

$$\begin{aligned} &For \quad i = 2, N-1 \\ &For \quad j = 2, M-1 \\ &\phi(i,j) = \frac{1}{4}[\phi(i+1,j) + \phi(i-1,j) + \phi(i,j-1) + \phi(i,j+1)] \\ &\qquad End \quad Do \\ &End \quad Do \end{aligned} \tag{14}$$

You can easily see that the corner values $\phi(1,1), \phi(N,1), \phi(1,M), \phi(N,N)$ do not appear in the iteration loops of Eqn.(14). These corner values are computed from

$$\phi(1,1) = \frac{1}{2}[\phi(1,2) + \phi(2,1) \tag{15}$$

$$\phi(N,1) = \frac{1}{2}[\phi(N-1,1) + \phi(N,2)$$

$$\phi(1,M) = \frac{1}{2}[\phi(1,M-1) + \phi(2,M)$$

$$\phi(N,M) = \frac{1}{2}[\phi(N,M-1) + \phi(N-1,M)$$

Eqn.(15) is a statement of continuity of ϕ as a corner is approached along the two walls meeting at the corner.

Summary

- $\{\phi(1,j); j = 2, M-1\}, \{\phi(N,j); j = 2, M-1\}, \{\phi(i,1); i = 2, N-1\}, \{\phi(i,M); i = 2, N-1\}$ are known from boundary conditions.
- Corner values are obtained from Eqn.(15).
- All other values are iteratively computed from Eqn.(14).

Neumann Problem

Consider the Neumann problem posed on the grid of Figure-6. The Dirichlet condition is specified on north, east, and south walls. The west wall has Neumann condition specified as:

$$\frac{\partial \phi}{\partial n} = -\frac{\partial \phi}{\partial x} = g(y) \tag{16}$$

As the partial derivatives in the Laplace equation are approximated by 2nd order FD scheme as in Eqn.(7), the partial derivative of Eqn.(16) needs to be approximated by a second order scheme also. We accomplish this by taking a fictitious grid point $(0,j)$ outside the domain of the problem, as shown in Figure-6.

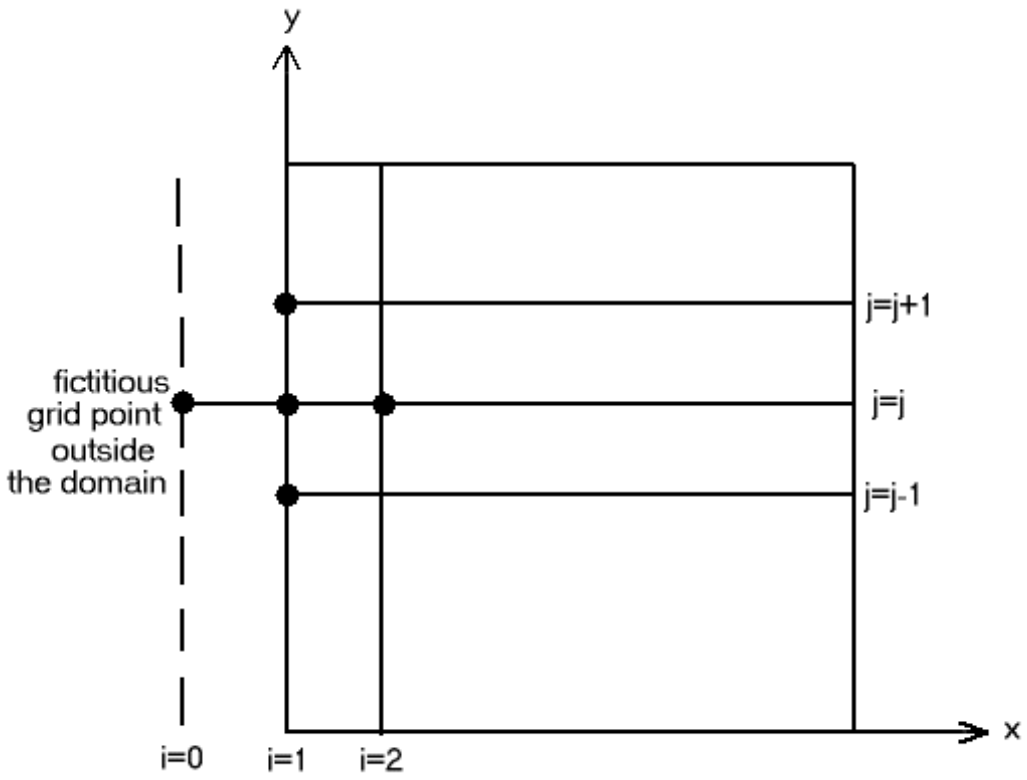

Figure-6 : Modeling the Neumann condition.

By utilizing the second order FD scheme for a first order derivative from Eqn.(4), we write Eqn.(16) as

$$\frac{\partial \phi}{\partial x}|_{(1,j)} = \frac{\phi(2,j) - \phi(0,j)}{2h} = -g(1,j) \tag{17}$$

However, we cannot keep the fictitious $\phi(0,j)$ within our formulation. Therefore, we write the Laplace equation (Eqn.(8)) at the point $(1,j)$ as

$$\phi(1,j) = \frac{1}{4}[\phi(2,j) + \phi(0,j) + \phi(1,j+1) + \phi(1,j-1)] \tag{18}$$

We re-write Eqn.(17) as

$$\phi(0,j) = \phi(2,j) + 2h\,g(1,j) \tag{19}$$

By substituting Eqn.(19) into Eqn.(18), we get

$$\phi(1,j) = \frac{1}{4}[2\phi(2,j) + 2h\,g(1,j) + \phi(1,j+1) + \phi(1,j-1)] \tag{20}$$

We use Eqn.(20) for $2 \le j \le M-1$, where $g(1,j)$ is a specified function. As Dirichlet condition is specified on north, west, and south walls, the values $[\phi(i,M), 2 \le i \le N-1], [\phi(N,j), 2 \le j \le M-1], [\phi(i,1), 2 \le i \le N-1]$ are known.

Hence, the simple algorithm for the iterative solution is

$$
\begin{aligned}
&For \quad i = 1, N-1 \\
&For \quad j = 2, M-1 \\
&\quad If \quad i = 1 \quad Then \\
&\phi(1,j) = \frac{1}{4}[2\phi(2,j) + 2h\,g(1,j) + \phi(1,j+1) + \phi(1,j-1)] \\
&\quad Else \\
&\phi(i,j) = \frac{1}{4}[\phi(i+1,j) + \phi(i-1,j) + \phi(i,j-1) + \phi(i,j+1)] \\
&\quad End \quad Do \\
&End \quad Do
\end{aligned} \tag{21}
$$

The values of ϕ at the corner points are computed by using Eqn.(15).

Summary

- $[\phi(i,M), 2 \le i \le N-1], [\phi(N,j), 2 \le j \le M-1], [\phi(i,1), 2 \le i \le N-1]$ are known as boundary conditions.
- Corner values are obtained from Eqn.(15).
- All other values are iteratively computed from Eqn.(21).

Exercise-1

On a $1 \le i \le N$ and $1 \le j \le M$ grid, write the iterative scheme when the Neumann condition is specified on the north AND the east walls as

$$
\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial y} = f(x); \quad on \quad North
$$

$$
\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} = g(y); \quad on \quad East
$$

and Dirichlet condition on the south and west walls are

$$
\phi = p(x); \quad on \quad South
$$

$$
\phi = q(y); \quad on \quad West
$$

Exercise-2

Solve the Neumann problem of Exercise-1 by taking $p(x) = 20$, $q(y) = 200$, $f(x) = 0$, and $g(y) = 4$ in a (1×1) box with (i) $N = 5$ and $M = 5$, (ii) $N = 17$ and $M = 17$, and (iii) $N = 65$ and $M = 65$.

Sample Calculation for Exercise-2

On a $5 \otimes 5$ grid, the iteration converges after *twentyone* cycles, when the tolerance is set to 0.01. The values of ϕ at the center of the domain, i.e. at location $(3,3)$, at the end of each iteration are given below for debugging purposes: 20.63, 37.81, 50.44, 60.21, 68.11, 74.68, 80.23, 84.96, 89.02, 92.52, 95.54, 98.14, 100.4, 102.3, 104.0, 105.5, 106.7, 107.8, 108.8, 109.6, 110.3. (These numbers are obtained on a Compaq Compiler.)

The values of ϕ on the $5 \otimes 5$ grid at the end of the first cycle of iteration are given below.

100.0	57.97	25.31	9.980	7.848
200.0	65.94	21.64	7.305	5.715
200.0	63.75	20.63	7.578	6.250
200.0	55.00	18.75	9.688	7.844
110.0	20.00	20.00	20.00	10.00

Accelerating the Convergence

Here we describe a method for accelerating the convergence of an iterative scheme. This method is known as ”successive over/under relaxation”. We show this method for the iterative scheme

$$\phi(i,j) = \frac{1}{4}[\phi(i+1,j) + \phi(i-1,j) + \phi(i,j-1) + \phi(i,j+1)]$$

We modify this equations as follows

$$\widetilde{\phi(i,j)} = \frac{1}{4}[\phi(i+1,j) + \phi(i-1,j) + \phi(i,j-1) + \phi(i,j+1)] \tag{22}$$

$$\phi(i,j)^{(New)} = (1-\omega)\phi(i,j)^{(Old)} + \omega\widetilde{\phi(i,j)} \tag{23}$$

Successive over/under relaxation is a two step process. In the first step of Eqn.(22), we calculate an *intermediate update* for $\phi(i,j)$. Then the *true update* of $\phi(i,j)$ is computed in Eqn.(23). This *true update* is a weighted combination of the *intermediate update* and the old value of $\phi(i,j)$.

- When the relaxation factor ω is less than one, the method is *under relaxed*.
- When the relaxation factor ω is more than one, the method is *over relaxed*.

A few general observations for over relaxed iterative scheme are: (i) for moderately high values of ω, the convergence is accelerated, (ii) for high values of ω, the convergence is oscillatory, and (iii) for very high values of ω, the oscillations becomes divergently oscillatory and the iterative scheme becomes unstable. Under relaxed schemes seldom diverge, but the rate of convergence may become painfully slow. For some simple geometries, the optimal convergence parameter ω for the fastest rate of convergence can be calculated. However, for general problems the optimal ω can only be estimated through numerical experiments.

For an $N \otimes M$ grid spanning a rectangular domain, the optimum ω is given by

$$\omega = \frac{4}{2 + \sqrt{4 - \left(\cos\left[\frac{\pi}{N-1}\right] + \cos\left[\frac{\pi}{M-1}\right]\right)^2}} \tag{24}$$

Exercise-3

Through numerical experiments, compute the optimal ω for the Neumann problem of Exercise-2, Case (iii).

Example

A dirichlet problem is posed on a square domain as follows:

$$\phi_{North} = 100$$

$$\phi_{East} = 50$$

$$\phi_{South} = 0$$

$$\phi_{West} = 75$$

When the problem is solved on a $5 \otimes 5$ grid, by using Gauss-Seidel Iterative scheme, nine iterations are necessary to reach a solution with 99% accuracy. The progress of the iterative scheme is shown in the Table below.

87.5	100.0	100.0	100.0	75.0
75.00	49.61 63.38 71.18 74.90 76.74 77.66 78.11 78.34 78.46	39.16 60.51 68.67 72.44 74.28 75.20 75.66 75.89 76.00	51.71 61.91 65.93 67.81 68.73 69.18 69.41 69.53 69.59	50.00
75.00	23.44 39.36 49.21 55.93 59.52 61.34 62.25 62.71 62.94	7.031 26.95 41.60 48.93 52.59 54.42 55.33 55.79 56.02	17.68 37.12 45.04 48.78 50.62 51.54 52.00 52.23 52.34	50.00
75.00	18.75 25.78 31.49 35.94 39.25 41.03 41.94 42.20 42.63	4.687 11.62 19.57 26.05 29.61 31.43 32.34 32.80 33.03	13.67 19.82 26.67 30.27 32.10 33.01 33.47 33.70 33.81	50.00
37.50	0.000	0.000	0.000	25.00

In an optimally over-relaxed iterative scheme with $\omega = 1.1716,$ calculated from Eqn.(24), the 99% accurate solution is obtained after six iterations. The progress of the over-relaxed scheme is shown below.

87.5	100.0	100.0	100.0	75.0
75.00	59.58 69.03 76.43 78.01 78.47 78.55	49.73 70.94 74.46 75.75 76.05 76.10	65.08 67.44 69.06 69.53 69.63 69.64	50.00
75.00	28.40 45.85 55.46 61.65 62.86 63.09	10.20 37.23 51.92 55.18 56.03 56.20	22.48 47.43 50.84 52.09 52.39 52.44	50.00
75.00	21.97 28.40 34.93 39.61 42.56 42.72	6.434 15.04 25.23 31.86 32.94 33.18	16.53 22.80 32.02 33.37 33.82 33.91	50.00
37.50	0.000	0.000	0.000	25.00