UNIT 5  NUMERICAL METHODS TO SOLVE HEAT CONDUCTION PROBLEMS

Structure

5.1 Introduction
   Objectives

5.2 Analytical Solution

5.3 Graphical Method
   5.3.1 Determination of Heat Transfer Rate by Graphical Method
   5.3.2 The Conduction Shape Factor

5.4 Finite Difference Scheme
   5.4.1 Central, Forward and Backward Difference Schemes with Uniform Grid
   5.4.2 Forward Differences
   5.4.3 Backward Differences
   5.4.4 Condition for using Forward, Backward and Central Difference Expressions
   5.4.5 Difference Expressions of Higher Accuracy

5.5 Numerical Errors
   5.5.1 Round-off Error
   5.5.2 Truncation Error
   5.5.3 Discretisation Error

5.6 Accuracy of a Solution : Optimum Step Size

5.7 Numerical Methods for Conduction Heat Transfer
   5.7.1 Numerical Methods for a One Dimensional Steady Time Problems
   5.7.2 Numerical Methods for Two Dimensional Steady State Problems

5.8 Transient One Dimensional Problems

5.9 Summary

5.10 Key Words

5.11 Answers to SAQs

1.1 INTRODUCTION

Analytical method is applicable to simple geometries and boundary conditions only. In case of complicated geometries, boundary conditions, and temperature dependent thermal properties, analytical method can not be used. Further, most of the practical problems encountered are associated with complicated geometries, boundary conditions as well as variable thermal properties. In such situation, numerical methods are extensively used to find the heat transfer rate and temperature distribution. One of the most extensively used numerical methods for conduction heat transfer problems is the finite difference method. In the present unit analytical solution for simple geometry is first presented. Graphical method for certain category of problems are also described. Finite difference method is then elaborately described.

Objectives

After studying this unit, you should be able to

- know the importance of analytical, graphical and numerical methods in solving various heat conduction problems,
Conduction

- appreciate importance of numerical method for solving conduction heat transfer problems,
- formulate finite difference equations for a conduction heat transfer problem, and
- use various methods to solve the finite difference equation.

5.2 ANALYTICAL SOLUTION

Analytical method is applicable to simple geometries and boundary conditions only. The components of the heat flow per unit area (heat flux) $q''$ in the $x$ and $y$ directions are obtained from Fourier’s law

$$q''_x = -k \frac{dT}{dx} \quad \ldots \ (5.1)$$
$$q''_y = -k \frac{dT}{dy} \quad \ldots \ (5.2)$$

The heat flux depends on the temperature gradient and is, therefore, a vector, while temperature is a scalar.

The heat flux $q''$ at a given point $(x, y)$ is the resultant of the components $q''_x$ and $q''_y$ at that point and its directed perpendicular to the isotherm (Figure 5.1).

![Figure 5.1: Heat Flow in Two Dimensions](image)

If the temperature distribution in a system is known, the rate of heat flow can easily be calculated. Therefore, heat flow analyses usually concentrate on determining the temperature field.

Let us consider a simple case of a thin rectangular plate (Figure 5.2), free of heat sources and insulated at the top and bottom surfaces. For steady state two-dimensional heat conduction in the absence of any heat source, and uniform thermal conductivity, Laplace equation applies

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad \ldots \ (5.3)$$

Since $\frac{\partial T}{\partial z} \approx 0$, the temperature is a function of $x$ and $y$ only. If $k$ is uniform, the temperature distribution must satisfy Eq. (5.3), a linear and homogeneous partial differential equation that can be integrated by assuming a product solution for $T(x, y)$ of the form

$$T = X(x) \ Y(y) \quad \ldots \ (5.4)$$

where $X = X(x)$, is a function of $x$ only, and $Y = Y(y)$, a function of $y$ alone. Differentiating Eq. (5.4) twice, first with respect to $x$ and then with respect to $y$,
Numerical Methods to Solve Heat Conduction Problems

\[ \frac{\partial^2 T}{\partial x^2} = Y \frac{\partial^2 T}{\partial y^2} \quad \ldots (5.5) \]

\[ \frac{\partial^2 T}{\partial y^2} = \frac{\partial^2 T}{\partial x^2} \quad \ldots (5.6) \]

\[ \frac{\partial}{\partial x} = \frac{\partial}{\partial x} \quad \ldots (5.7) \]

\[ \frac{\partial}{\partial y} = \frac{\partial}{\partial y} \quad \ldots (5.8) \]

The variables are now separated. The LHS is a function of \( x \) only, while the RHS is a function of \( y \) alone. Since neither side can change as \( x \) and \( y \) vary, both must be equal to a constant, say \( \lambda^2 \). We have, therefore, two ordinary differential equations

\[ \frac{d^2 X}{dx^2} + \lambda^2 X = 0 \quad \ldots (5.9) \]

\[ \frac{d^2 Y}{dy^2} - \lambda^2 Y = 0 \quad \ldots (5.10) \]

The general solution of Eq. (5.9) is

\[ X = C_1 \cos \lambda x + C_2 \sin \lambda x \quad \ldots (5.11) \]

and the general solution of Eq. (5.10) is

\[ Y = C_3 e^{-\lambda y} + C_4 e^{\lambda y} \quad \ldots (5.12) \]

Therefore, from Eq. (5.4),

\[ T = X \cdot Y = (C_1 \cos \lambda x + C_2 \sin \lambda x)(C_3 e^{-\lambda y} + C_4 e^{\lambda y}) \quad \ldots (5.13) \]

where \( C_1, C_2, C_3 \) and \( C_4 \) are constants calculated from the boundary conditions. As shown in Figure 5.2, boundary conditions to be satisfied are

\[ T = 0 \text{ at } x = 0 \quad \ldots (5.14) \]

\[ T = 0 \text{ at } x = L \quad \ldots (5.15) \]

Figure 5.2: Rectangular Adiabatic Plate with Sinusoidal Temperature Distribution on One Edge
Conduction

\[ T = 0 \text{ at } y = 0 \] \hspace{1cm} \ldots (5.16)

\[ T = T_m \sin \left( \frac{\lambda x}{L} \right) \text{ at } y = b \] \hspace{1cm} \ldots (5.17)

Substituting these conditions in Eq. (5.13), from the 3rd condition (at \( y = 0 \))

\[ (C_1 \cos \lambda x + C_2 \sin \lambda x) (C_3 + C_4) = 0 \] \hspace{1cm} \ldots (5.18)

From the first condition (at \( x = 0 \))

\[ C_1 (C_3 e^{-\lambda y} + C_4 e^{\lambda y}) = 0 \] \hspace{1cm} \ldots (5.19)

From the second condition

\[ (C_1 \cos \lambda L + C_2 \sin \lambda L) (C_3 e^{-\lambda y} + C_4 e^{\lambda y}) = 0 \] \hspace{1cm} \ldots (5.20)

Eq. (5.18) gives \( C_3 = -C_4 \) and Eq. (5.19) gives \( C_1 = 0 \). Using these results in Eq. (5.20)

\[ 2C_2 C_3 \sin \lambda L \sin \lambda y = 0 \] \hspace{1cm} \ldots (5.21)

To satisfy this condition, \( \sin \lambda L = 0 \) or, \( \lambda = \frac{n \pi}{L} \), where \( n = 1, 2, 3, \ldots \). There exists therefore, a different solution for each integer \( n \), and each solution has a separate integration constant \( C_n \). Summing these solutions, we get from Eq. (5.13),

\[ T = \sum_{n=1}^{\infty} C_n \sin \left( \frac{n \pi x}{L} \right) \sinh \left( \frac{n \pi y}{L} \right) \] \hspace{1cm} \ldots (5.22)

The last boundary condition needs that at \( y = b \)

\[ T_m \sin \left( \frac{n \pi x}{L} \right) = C_n \sin \left( \frac{\pi x}{L} \right) \sinh \left( \frac{\pi b}{L} \right) \] \hspace{1cm} \ldots (5.23)

Hence,

\[ C_n = \frac{T_m}{\sinh \left( \frac{\pi b}{L} \right)} \] \hspace{1cm} \ldots (5.24)

The solution therefore becomes (From Eq. (5.22))

\[ T (x, y) = T_m \frac{\sin \left( \frac{\pi y}{L} \right) \sin \left( \frac{\pi x}{L} \right)}{\sinh \left( \frac{\pi b}{L} \right)} \] \hspace{1cm} \ldots (5.25)

Corresponding temperature field is shown in Figure 5.3.

**Figure 5.3**: Isotherms and Heat Flow Lines for the Plate in Figure 5.2

The solid lines are isotherms and the dashed lines are heat flow lines, which are orthogonal.
SAQ 1
(a) Write down the Laplace equation for steady state two dimensional heat conduction in the absence of any heat source, and uniform thermal conductivity.
(b) What is the significance of isotherms?
(c) Why are the isotherms and adiabatics orthogonal?

5.2.1 Extension to 3-D Problems
The separation of variable method can be extended to three-dimensional problems, by assuming that $T = X Y Z$. Substituting this expression for $T$ in equation
\[
\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0
\] 
Separating the variables, integrating the total differential equations and using the given boundary conditions, the solution for the temperature distribution can be obtained.

5.3 GRAPHICAL METHOD
Graphical method can rapidly provide an approximate estimate of the temperature distribution and heat flow in geometrically complex 2-D systems. However, this method is limited to isothermal and adiabatic boundary conditions only.

The objective of the graphical solution is to systematically construct a network of isotherms and adiabatics. The flux lines are analogous to the streamlines in a potential flow, i.e. tangent to the direction of heat flow at any point. Therefore, no heat can flow across the constant flux lines. The isotherms are analogous to constant potential lines and heat flows perpendicular to them. Thus, isotherms and adiabatics intersect at right angles.

Consider a square, two-dimensional channel whose inner and outer surfaces are maintained at temperature $T_1$ and $T_2$, respectively. A cross section of the channel is shown in Figure 5.4.

Following procedure is adapted to construct the flux plot:
Step 1

Conduction

Identify all relevant lines of symmetry. Such lines are determined by thermal as well as geometrical conditions. As for example, for the square channel in Figure 5.4(a), such lines include the designated vertical and horizontal and diagonal lines. For this system it is therefore possible to consider only one-eighth of the configuration as shown in Figure 5.4(b).

Step 2

Line of symmetry are adiabatic in the sense that there can be no heat transfer in a direction perpendicular to the lines. They are there for heat flow lines and should be treated as such. Since there is no heat flow in a direction perpendicular to a heat flow line, it can be termed as adiabat.

Step 3

After all the known lines of constant temperature are identified, an attempt should be made to sketch lines of constant temperature within the system. Note that isotherms should always be perpendicular to adiabats.

Step 4

Heat flow lines should be drawn with an eye towards creating a network of curvilinear squares. This is done by having heat flow lines and isotherms intersect at right angles and by requiring that all sides of each square be of approximately of same length. It is often impossible to satisfy this second requirement exactly, and it is more realistic to strive for equivalence between the sums of the opposite sides of each square (Figure 5.4(c)). Assuming $x$-coordinate to the direction of heat flow and the $y$-coordinate to the direction normal to this flow, the requirement may be expressed as

$$\Delta x \equiv \frac{ab + cd}{2} \approx \Delta y \equiv \frac{ac + bd}{2} \ldots (5.27)$$

Several iterations must often be made to draw a network of curvilinear squares. This trial and error process involves adjusting the isotherms and adiabats until satisfactory curvilinear squares are obtained for most of the network. Once the flux plot has been obtained, it may be used to infer the temperature distribution in the medium.

SAQ 2

(a) Explain the steps to be considered for construction of flux plot in graphical method.

(b) Explain the graphical method of solving a two-dimensional heat conduction problem.

5.3.1 Determination of Heat Transfer Rate by Graphical Method

Let, $q_i = \text{Energy flow rate through a lane (region between two adiabats).}$

If the flux plot is properly constructed, the value of $q_i$ will be approximately the same for all the lanes and the total heat transfer may be expressed as :

$$q = \sum_{i=1}^{M} q_i = M q_i \ldots (5.28)$$

where $M$ is the number of lanes associated with the plot. From the curvilinear plot of Figure 5.4(c) and application of Fourier’s law of heat conduction, $q_i$ may be expressed as
Numerical Methods to Solve Heat Conduction Problems

where $\Delta T_j$ is the temperature difference between successive isotherms, $A_i$ is the conduction heat transfer area for the lane, and $l$ is the length of the channel normal to the page. As the temperature increment is approximately same for all the adjoining isotherms, the overall temperature difference between boundaries, $\Delta T_{1-2}$, may be expressed as

$$\Delta T_{1-2} = \sum_{j=1}^{N} \Delta T_j = N \Delta T_j$$

(5.30)

where $N$ is the total number of temperature increments. Combining Eqs. (5.28)-(5.30) and recognizing that $\Delta x \approx \Delta y$, for curvilinear squares, we obtain

$$q \approx \frac{ML}{N} k \Delta T_{1-2}$$

(5.31)

The manner in which a flux plot may be used to obtain the heat transfer rate for a two-dimensional system is evident from Eq. (5.31). The ratio of $\left(\frac{M}{N}\right)$ may be obtained from the plot.

Recall that the specification of $N$ is based on Step 3 of the foregoing procedure, and the value, which is an integer, may be a consequence of following Step 4. Note that $M$ is not necessarily an integer, since a fraction may be needed to arrive at a satisfactory network of curvilinear squares. For the network of Figure 5.4(b), $N = 6$ and $M = 5$. Of course, as the network, or mesh of curvilinear squares is made finer, $N$ and $M$ increase and the estimate of $\left(\frac{M}{N}\right)$ becomes more accurate.

### 5.3.2 The Conduction Shape Factor

Eq. (5.31), may be used to define shape factor $S$, of a two-dimensional system. Heat transfer rate with shape factor may be expressed as

$$q = S k \Delta T_{1-2}$$

(5.32)

where for a flux plot

$$S = \frac{ML}{N}$$

(5.33)

From Eq. (5.32), it follows that a two-dimensional conduction resistance may be expressed as

$$R_{1,2-\text{cond}} = \frac{1}{S k}$$

(5.34)

Shape factors have been obtained for numerous two-dimensional systems and results are summarised in Figure 5.5.

<table>
<thead>
<tr>
<th>System</th>
<th>Schematic</th>
<th>Restrictions</th>
<th>Shape Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Isothermal sphere buried in a semi-infinite medium</td>
<td>$z &gt; \frac{D}{2}$</td>
<td>$\frac{2\pi D}{1 - \frac{D}{4z}}$</td>
</tr>
<tr>
<td>Case 2</td>
<td>Horizontal isothermal cylinder of length $L$ buried in a semi-infinite medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L &gt; D$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L &gt; D$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$z &gt; 3D$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\cosh^{-1} \left( \frac{2z}{D} \right)$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3</th>
<th>Vertical cylinder in a semi-infinite medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L &gt; D$</td>
<td></td>
</tr>
<tr>
<td>$\frac{2\pi L}{\ln \left( \frac{4L}{D} \right)}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 4</th>
<th>Conduction between two cylinders of length $L$ in infinite medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L &gt; D_1, D_2$</td>
<td></td>
</tr>
<tr>
<td>$L &gt; w$</td>
<td></td>
</tr>
<tr>
<td>$\cosh^{-1} \left( \frac{4w^3 - D_2^2 - D_1^2}{2D_1 D_2} \right)$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 5</th>
<th>Horizontal circular cylinder of length $L$, midway between parallel planes of equal length and infinite width</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L &gt; D_1, D_2$</td>
<td></td>
</tr>
<tr>
<td>$D_1, D_2 &gt; w$</td>
<td></td>
</tr>
<tr>
<td>$z &gt; D$</td>
<td></td>
</tr>
<tr>
<td>$\frac{2\pi L}{\ln \left( \frac{8L}{\pi D} \right)}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 6</th>
<th>Circular cylinder of length $L$ oriented in square solid of equal length</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w &gt; D$</td>
<td></td>
</tr>
<tr>
<td>$L &gt; w$</td>
<td></td>
</tr>
<tr>
<td>$\frac{2\pi L}{\ln \left( 1.08w \right)}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 7</th>
<th>Eccentric circular cylinder of length $L$ in a cylinder of equal length</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D &gt; d$</td>
<td></td>
</tr>
<tr>
<td>$L &gt; D$</td>
<td></td>
</tr>
<tr>
<td>$\cosh^{-1} \left( \frac{D^2 + a^2 - 4x^2}{2D d} \right)$</td>
<td></td>
</tr>
</tbody>
</table>
**Case 8**
Conduction through the edge of adjoining walls

$$D > D_e$$

0.54 \( D \)

**Case 9**
Conduction through corner of three walls with a temperature difference \( \Delta T_{1-2} \) across the walls

$$L < \text{length and width of wall}$$

0.15 \( L \)

**Case 10**
Disk of diameter \( D \) and \( T_1 \) on a semi-infinite medium of thermal conductivity \( k \) and \( T_2 \)

None

2 \( D \)

**Case 11**
Square channel of length \( L \)

\( \frac{W}{w} < 1.4 \)

\( \frac{2\pi L}{0.785 \ln \left( \frac{W}{w} \right)} \)

\( \frac{W}{w} > 1.4 \)

\( \frac{2\pi L}{0.930 \ln \left( \frac{W}{w} \right) - 0.050} \)

---

Figure 5.5 : Conduction Shape Factor for Selected Two-Dimensional Systems

**SAQ 3**

(a) What do you mean by conduction shape factor? How is it estimated?

(b) How does the conduction shape factor help in solving conduction heat transfer problems graphically?
5.4 Finite Difference Scheme

Basic approach in solving a problem with the finite difference method is to discretise the derivatives appearing in the governing differential equation at a finite number of uniformly or non-uniformly spaced grid points that fill the computational domain. The governing differential equation is then transformed into a system of difference equations. This means that if there are 100 grid points, where variables are unknown, there will be 100 equations to solve per variable. The necessity of using a computer arises because of the huge number of arithmetic operations that are required to be carried out in a reasonable time for solving a large number of equations. The simplification inherent in the use of algebraic equations rather than differential equations is what makes numerical methods so powerful and widely applicable.

5.4.1 Central, Forward and Backward Difference Schemes with Uniform Grid

Consider a smooth function \( y = f(x) \) as in Figure 5.6.

The Taylor series for the said function at \( x_i + h \) expanded about \( x_i \) is

\[
y (x_i + h) = y_i + y'_i h + \frac{y''_i h^2}{2!} + \frac{y'''_i h^3}{3!} + \ldots
\]

where \( h = \Delta x \) and \( y_i, y_{i+1}, y_{i-1} \) are the ordinates corresponding to \( x_i, x_i + h, x_i - h \), respectively. The function at \( (x_i - h) \) is similarly given by

\[
y (x_i - h) = y_i - y'_i h + \frac{y''_i h^2}{2!} + \frac{y'''_i h^3}{3!} - \ldots
\]

Subtracting Eq. (5.36) from Eq. (5.35) we get,

\[
y (x_i + h) - y (x_i - h) = 2y'_i h + \frac{2y''_i h^3}{3!} + \ldots
\]

or

\[
2y'_i h = y (x_i + h) - y (x_i - h) - \frac{y'''_i h^3}{3} + \ldots
\]

Therefore,

\[
y'_i = \frac{y (x_i + h) - y (x_i - h)}{2h} - \frac{y''_i h^2}{6} + \text{(higher order terms)}
\]

or

\[
y'_i = \frac{y_{i+1} - y_{i-1}}{2h} + \text{(error of order } h^2)\]

Figure 5.6 : Uniformly Spaced Grid Points on a Continuous and Differentiable Function \( y = f(x) \)
The notation \( o(h^2) \) means that in arriving at Eq. (5.41), terms of the order of \( h^2 \) and higher have been neglected. \( o(h^2) \) is called the truncation error. The truncation error is the difference between the exact mathematical expression and its numerical approximation.

Eq. (5.41) is called the central-difference approximation of \( y'(i.e., \frac{dy}{dx}) \) at \( x_i \) with an error of order \( h^2 \). In Figure 5.6, the approximation is depicted by the slope of the dashed line. The actual derivative is shown by the solid line drawn tangent to the curve at \( x_i \). The difference can be viewed as due to the truncation error resulting from using a truncated Taylor series.

Now, adding Eqs. (5.35) and (5.36), we get

\[
y_i^2 + 2y_i = 2y_i + y_i^* h^2 + \frac{&h^4}{12} + \ldots
\]

or

\[
y_i^* h^2 = y (x_i + h) + y (x_i - h) - 2y_i - \frac{&h^4}{12} + \text{higher-order terms}
\]

\[
y_i^* = \frac{y_i - 2y_i + y_i - 1}{h^2} + 0 (h^2)
\]

Eq. (5.44) is the central-difference approximation of the second derivative of the function with respect to \( x \) (i.e., \( \frac{d^2y}{dx^2} \)) evaluated at \( x_i \) with an error of order \( h^2 \). Alternatively, Eq. (5.44) may be expressed as

\[
y_i^* = \frac{y_{i+1} - y_i}{h} - \frac{y_i - y_{i-1}}{h}
\]

or

\[
y_i^* = \frac{y_{i+1/2} - y_{i-1/2}}{h}
\]

where \( y_{i+1/2} \) and \( y_{i-1/2} \) represent the slopes of the tangents to the curve at \( x_i + \frac{h}{2} \) and \( x_i - \frac{h}{2} \), respectively.

The central-difference expressions reveal that the first and second derivatives of the function involve values of the function on both sides of the \( x \)-value at which the derivative of the function is to be evaluated.

### 5.4.2 Forward Difference

From Taylor series expansions, it is also easy to obtain expressions for the derivatives which are entirely in terms of values of the function at \( x_i \) and points to the right of \( x_i \). These are called forward-difference expressions.

Starting from the Taylor series expansion as given in Eq. (5.35), we get

\[
y_i' h = y (x_i + h) - y_i - \frac{y_i^* h^2}{2!} - \frac{y_i^* h^3}{3!} - \ldots
\]
Conduction

or

\[ y'_i h = \frac{y(x_i + h) - y_i}{h} - \frac{y''_i h^2}{2!} - \frac{y'''_i h^3}{3!} - \cdots \]  \hspace{1cm} \ldots (5.48)

Dropping terms of the order of \( h \) and higher,

\[ y'_i = \frac{y(x_i + h) - y_i}{h} + 0 (h) \]  \hspace{1cm} \ldots (5.49)

Similarly,

\[ y(x_i + 2h) = y_i + y'_i (2h) + \frac{y''_i (2h)^2}{2!} - \frac{y'''_i (2h)^3}{3!} + \cdots \]

\[ = y_i + y'_i (2h) + 2y''_i h^2 + 0 (h^3) \]  \hspace{1cm} \ldots (5.50)

Also,

\[ y(x_i + h) = y_i + y'_i h + \frac{y''_i h^2}{2!} + 0 (h^3) \]  \hspace{1cm} \ldots (5.51)

Multiplying Eq. (5.51) by 2 and subtracting from Eq. (5.50) gives

\[ y''_i = \frac{y(x_i + 2h) - 2y(x_i + h) + y_i}{h^2} + 0 (h) \]  \hspace{1cm} \ldots (5.52)

\[ y'_i = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} + 0 (h) \]  \hspace{1cm} \ldots (5.53)

5.4.3 Backward Difference

Following the approach given in Section 5.4.2, one can easily obtain derivative expressions which are entirely in terms of the values of the function at \( x_i \). These are known as backward-difference expressions, which are given below for \( y'_i \) and \( y''_i \).

\[ y'_i = \frac{y_i - y_{i-1}}{h} + 0 (h) \]  \hspace{1cm} \ldots (5.54)

\[ y''_i = \frac{y_i - 2y_{i-1} + y_{i-2}}{h^2} + 0 (h) \]  \hspace{1cm} \ldots (5.55)

5.4.4 Conditions for using Forward, Backward, and Central-Difference Expressions

- Forward-difference expressions are used when data to the left of a point, at which a derivative is desired, are not available.
- Backward-difference expressions are used when data to the right of the desired point are not available.
- Central-difference expressions are used when data on both sides of the desired point are available and are more accurate than either forward- or backward-difference expressions.

SAQ 4

(a) Explain the forward difference, central difference and backward difference schemes.
(b) Under what conditions will you use these schemes?

5.4.5 Difference Expressions of Higher Accuracy

By retaining a greater number of terms in the Taylor series, it is possible to obtain forward-, backward-, and central difference expressions for a higher accuracy. The following expressions show central-difference expressions for \( y'_i \) and \( y''_i \) with an error of \( 0 (h^4) \) and forward- and backward-difference expressions for the same with an error of
0 \left( h^2 \right). \) It is apparent that for a greater accuracy more number of neighbouring points are involved. For example, Eq. (5.54) is a two-point forward-difference scheme for \( y'_i \), while Eq. (5.55) is a three-point forward-difference scheme. Central difference with an error of 0 \left( h^2 \right):

\[
y'_i = -\frac{y_{i+2} + 2y_{i+1} - 8y_{i+1} - y_{i-2}}{12h}
\]

\[
y''_i = \frac{-y_{i+2} + 6y_{i+1} - 8y_{i+1} + y_{i-2}}{12h^2}
\]  \ldots (5.56)

Forward difference with an error of 0 \left( h^2 \right):

\[
y'_i = \frac{-y_{i+2} + 2y_{i+1} + 3y_{i}}{2h}
\]

\[
y''_i = \frac{-y_{i+3} + 4y_{i+2} - 5y_{i+1} + 2y_{i}}{h^2}
\]  \ldots (5.57)

Backward difference with an error of 0 \left( h^2 \right):

\[
y'_i = \frac{3y_{i} - 4y_{i-1} + y_{i-2}}{2h}
\]

\[
y''_i = \frac{2y_{i} - 5y_{i-1} + 4y_{i-2} - y_{i-3}}{h^2}
\]  \ldots (5.58)

Now, the question is: When does one use a higher-order difference scheme? There is no set answer to this. It depends on the accuracy requirement of a problem, and the analyst will have to use his own judgment.

### 5.5 NUMERICAL ERRORS

Three most important errors that commonly occur in numerical solutions are the

(a) round-off error,

(b) truncation error, and

(c) discretization error.

#### 5.5.1 Round-Off Error

The round-off error is introduced because of the inability of the computer to handle a large number of significant digits. Typically, in single-precision, the number of significant digits retained ranges from 7 to 16, although it may vary from one computer system to another. The round-off error arises due to the fact that a finite number of significant digits or decimal places are retained and all real numbers are rounded off by the computer. The last retained digit is rounded off if the first discarded digit is equal to or greater than 5. Otherwise, it is unchanged. For example, if five significant digits are to be kept in place, 5.37527 is rounded off to 5.3753, and 5.37524 to 5.3752.

#### 5.5.2 Truncation Error

The truncation error is due to the replacement of an exact mathematical expression by a numerical approximation. This error has been discussed earlier in this chapter with respect to finite-difference approximations. Basically, it is the difference between an exact expression and the corresponding truncated form (for example, truncated Taylor series) used in the numerical solution.

#### 5.5.3 Discretization Error

...
The discretization error is the error in the overall solution that results from the truncation error assuming the round-off error to be negligible. Therefore,

\[(\text{Discretization error}) = (\text{exact solution}) – (\text{numerical solution with no round-off solution})\]

**SAQ 5**

What are different types of numerical errors? Explain with suitable examples.

### 5.6 ACCURACY OF A SOLUTION: OPTIMUM STEP SIZE

The accuracy of a numerical solution is determined by its total error, which is the sum of round-off error and truncation error. Hence, \( (\text{total error}) = (\text{round-off error}) + (\text{truncation error}) \). However, it is obvious that the round-off error increases as the total number of arithmetic operations increases. Again, the total number of arithmetic operations increases if the step size decreases (that is, when the number of grid points increases). Therefore, the round-off error is inversely proportional to the step size. On the other hand, the truncation error decreases as the step size decreases (or as the number of grid points increases). Because of the aforementioned opposing effects, an optimum step size is expected, which will produce minimum total error in the overall solution.

#### 5.6.1 Method of Choosing Optimum Step Size – Grid Independence Test

A numerical analyst has to be extremely careful as regards the accuracy of a solution. To get the most accurate solution (that is, the solution with the least total error), one has to perform a grid independence test. The test is carried out by experimenting with various grid sizes and watching how the solution changes with respect to the changes in the grid size. Finally, a stage will come when changing the grid spacing will not affect the solution. In other words, the solution will become independent of grid spacing. The largest value of grid spacing for which the solution is essentially independent of the step size is chosen so that both the computational time and effort and the round-off error are minimised.

### 5.7 NUMERICAL METHODS FOR CONDUCTION HEAT TRANSFER

Many difficult problems arise in conduction, for example, variable thermal conductivity, distributed energy sources, radiation boundary condition, for which analytical solution are not available. Approximate solutions for these are obtained by numerical methods. The basic approach is to arrive at the relevant governing differential equation based on the physics of the particular problem. They are then converted into the required finite-difference forms. To begin with, the numerical solution procedure for the problem of a simple one-dimensional steady-state heat conduction in a cooling fin described. It is to be noted that a simple, closed-form straightforward analytical solution for this problem is available. The idea is to show the use of the numerical method and to compare the numerical solution with its analytical counterpart.

#### 5.7.1 Numerical Methods for a One-Dimensional Steady-State Problem

Consider one-dimensional, steady-state heat conduction in an isolated rectangular fin as shown in Figure 5.7. The base temperature is maintained at \( T = T_0 \) and the tip of the fin is insulated. The fin is exposed to a convective environment (neglecting radiation heat transfer from the fin) which is at \( T_\infty \) \( (T_\infty < T_0) \). The average heat transfer coefficient of the fin to the surroundings is \( h \). The length of the fin is \( L \) and the coordinate axis begins at the base of the fin. The one-dimensionality arises from the fact that the thickness of the
Numerical Methods to Solve Heat Conduction Problems

A fin is much small as compared to its length, and the width can be considered either too long or the sides of the fin to be insulated.

\[ T = T_0 \]

Figure 5.7: Physical Domain of a Rectangular Fin

Governing Differential Equation

The energy equation for the fin in the steady state (assuming constant \( k \)) is

\[ \frac{d^2T}{dx^2} - \frac{hP}{kA} (T - T_\infty) = 0 \]  \hspace{1cm} \ldots (5.62)

where \( P \) and \( A \) are the perimeter and the cross-sectional area of the fin, respectively.

Boundary Conditions

Since Eq. (5.62) is a linear, second-order ordinary differential equation, two boundary conditions are needed to completely describe this problem. Boundary conditions are as follows:

- BC-1: at \( x = 0 \), \( T = T_0 \) \hspace{1cm} \ldots (5.63)
- BC-2: at \( x = L \), \( \frac{dT}{dx} = 0 \) \hspace{1cm} \ldots (5.64)

Non-Dimensionalisation

Non-dimensionalising Eqs. (5.62)-(5.64) and using the dimensionless variables

\[ \theta = \frac{T - T_\infty}{T_0 - T_\infty}, \quad X = \frac{x}{L} \]  \hspace{1cm} \ldots (5.65)

We obtain

\[ \frac{d^2\theta}{dx^2} - (mL)^2 \theta = 0 \]  \hspace{1cm} \ldots (5.66)

Where

\[ m^2 = \frac{hP}{kA} \]  \hspace{1cm} \ldots (5.67)

and

\[ \theta(0) = 1 \]  \hspace{1cm} \ldots (5.68)
\[ \theta'(1) = 0 \]  \hspace{1cm} \ldots (5.69)

where

\[ \theta' = \frac{d\theta}{dX} \]  \hspace{1cm} \ldots (5.70)

Discretisation

Eq. (5.66) is discretised at any interior grid point \( i \) (see Figure 5.8) using central difference for \( \frac{d^2\theta}{dx^2} \) as follows:

\[ \left( \frac{d^2\theta}{dx^2} \right)_i - [(mL)^2 \theta]_i = 0 \]  \hspace{1cm} \ldots (5.71)
Figure 5.8: Computational Domain of the Fin with Equally Spaced Grid Points

or \[
\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{(\Delta X)^2} - (mL)^2 \theta_i = 0
\] \hspace{1cm} \ldots (5.72)

or \[
\theta_{i+1} - D \theta_i + \theta_{i+1} = 0, \quad i = 2, \ldots, M
\] \hspace{1cm} \ldots (5.73)

where \( D = 2 + (mL)^2 (\Delta X)^2 \).

Handling of the Boundary Condition

At \( x = L \), i.e., at \( i = M \), Eq. (5.73) reduces to

\[
\theta_{M-1} - D \theta_M + \theta_{M+1} = 0
\] \hspace{1cm} \ldots (5.74)

A careful look at Eq. (5.74) reveals that \( \theta_{M+1} \) represents a fictitious temperature \( \theta \) at point \( M+1 \), which lies outside the computational domain. There is a remedy to tackle this issue.

Remedy: Image Point Technique

It is assumed that the \( \theta \) versus \( X \) curve extends beyond \( X = 1 \) so that at \( X = 1 \), the condition \( \frac{d\theta}{dX} = 0 \) is satisfied. In other words, the \( \theta \) versus \( X \) curve can be imagined to look as in Figure 5.9. The dotted line represents the mirror-image extension of the solid line, indicating that a minimum exists at \( X = 1 \). Figure 5.9(a) shows a mirror-image extension of the fin.

Therefore, the boundary condition at \( X = 1 \) can be approximately satisfied by taking
\[ \theta_{M+1} = \theta_{M-1} \quad \ldots (5.75) \]

Eq. (5.75) also follows from the central-difference approximation of \( \frac{d\theta}{dX} \) at \( i = M \).

Substituting Eq. (5.75) into Eq. (5.74), we get
\[ 2\theta_{M-1} - D \theta_{M} = 0 \quad \ldots (5.76) \]

Therefore, we can write that
\[ \theta_i = 1 \text{ for } i = 1 \text{ (known)} \quad \ldots (5.77) \]
\[ \theta_{i+1} - D \theta_i + \theta_{i+1} = 0 \text{ for } i = 2, \ldots, m - 1 \quad \ldots (5.78) \]
\[ 2\theta_{M-1} - D \theta_{M} = 0 \text{ for } i = M \quad \ldots (5.79) \]

Hence, we have a set of \( M - 1 \) linear simultaneous algebraic equations and \( M - 1 \) unknowns, which can be easily solved by standard numerical methods. For the case in which \( M = 5 \), we have \( N = M - 1 = 4 \) equations to solve. The four equations can be written in the matrix form as
\[
\begin{bmatrix}
D & -1 \\
-1 & D & -1 \\
-1 & D & -2 \\
-1 & D & D
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0 \\
0 \\
0
\end{bmatrix} \quad \ldots (5.80)
\]

It should be noted that \( \theta_1 \) corresponds to the temperature at grid point 2 in Figure 5.8, and so on for \( \theta_2, \theta_3, \theta_4 \). An alternative to the image-point scheme is to use a second-order backward difference for \( \frac{d\theta}{dX} \) at \( i = M \).

**Methods of Solution**

In Eq. (5.80), the coefficient matrix has three diagonals - the main diagonal, sub-diagonal and super-diagonal, and hence the name tridiagonal matrix (TDM). The set of equations in Eq. (5.80) is called tridiagonal system of equations. See Figure 5.10 for a pictorial representation of TDM.

![Figure 5.10: Pictorial Representation of a Tridiagonal Coefficient Matrix, an Unknown Vector and a Known Right Hand Vector](image-url)

The set of equations in Equation 5.80 can be solved by any of the three methods:
- Gaussian elimination,
Conduction

- Thomas algorithm (or tridiagonal matrix algorithm or simply TDMA), and
- Gauss-Seidel iterative method.

### Gaussian Elimination (GE)

This method reduces a given set of $N$ equations to an equivalent triangular set, so that one of the equations has only one unknown. This unknown is determined and the remaining unknowns are obtained by the process of back substitution. The basic approach is shown in a step-by-step form as given. The set of equations to be solved are written in a matrix form in Eq. (5.81)

$$
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3
\end{bmatrix}
= 
\begin{bmatrix}
  c_1 \\
  c_2 \\
  c_3
\end{bmatrix}
\quad \quad \quad \ldots \quad (5.81)
$$

$a_{11}$ is called the pivot, below which the terms are to be made zero.

**Step I**

$$
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{22} & a_{23} & a_{23} \\
  a_{32} & a_{33}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \quad (1) \\
  x_3 
\end{bmatrix}
= 
\begin{bmatrix}
  c_1 \\
  c_2 \quad (1) \\
  c_3 
\end{bmatrix}
\quad \quad \quad \ldots \quad (5.82)
$$

The superscript represents the step number. $a_{22}^{(1)}$ is now the pivot for the next operation.

**Step II**

$$
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{22} & a_{23} & a_{23} \\
  a_{32} & a_{33} & a_{33}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \quad (2) \\
  x_3 \quad (2)
\end{bmatrix}
= 
\begin{bmatrix}
  c_1 \\
  c_2 \quad (2) \\
  c_3 
\end{bmatrix}
\quad \quad \quad \ldots \quad (5.83)
$$

### Solution Accuracy

The round-off error may significantly affect the accuracy if a large number of equations is involved. In addition, the round-off error is cumulative because the errors are carried on from one step to the other during the elimination process. Consequently, GE is generally used if the number of equations is typically less than 20 when the coefficient matrix is dense. For a sparse coefficient matrix, however, a large number of equations can be solved. The TDM system is a good example of a sparse coefficient matrix. If Gaussian elimination is applied to this system, only one of the $a’$ s is eliminated from the column containing the pivot element in each step, since the remaining elements below the diagonal are zero. Therefore, only one elimination process is employed at each step. The number of operations needed for solving a tridiagonal system is of order $N$, that is, $O(N)$ as compared to $O(N^3)$ for a system with a dense coefficient matrix. Therefore, a much smaller number of operations and consequently much lower round-off errors arise in the solution of the systems. Obviously, the computer time is much less for solution by TDMA. Thus, large tridiagonal systems are generally solved by this method.

**Thomas Algorithm or TDMA**

The set of equations in Eq. (5.78) can be readily solved by the Gaussian elimination method with a maximum of three variables per equation. The solution can be expressed very concisely. Eq. (5.78) is actually a special form of the system (using $N = M - 1$).

$$
b_1 T_1 + c_1 T_2 = d_1 \\
a_2 T_1 + b_2 T_2 + c_2 T_3 = d_2
$$
Numerical Methods to Solve Heat Conduction Problems

First, let us demonstrate the validity of a recursion solution of the form (Carnahan et al. 1969)

\[ T_i = \gamma_i - \frac{c_i}{\beta_i} T_{i+1} \]  

where the constants \( \beta_i \) and \( \gamma_i \) are to be determined. The substitution of Eq. (5.85) into Eq. (5.84) gives

\[ a_i \left( \frac{\gamma_i - c_i - 1}{\beta_i} T_i \right) + b_i T_i + c_i T_{i+1} = d_i \]  

Rearranging Eq. (5.86), we obtain

\[ T_i = d_i - a_i \frac{\gamma_i - c_i - 1}{\beta_i} T_i - \frac{c_i T_{i+1}}{\beta_i} \]  

Eq. (5.87) verifies the form of Eq. (5.84), subject to the following recursion relations:

\[ \beta_i = \frac{b_i - a_i c_i - 1}{\beta_{i-1}} \]  

\[ \gamma_i = \frac{d_i - a_i \gamma_i - 1}{\beta_i} \]  

Also, from the first equation of Eq. (5.84),

\[ T_i = \frac{d_i}{b_i} - \frac{c_i}{b_i} T_2 \]  

From which we get

\[ \beta_i = \frac{b_i}{\gamma_i} \]  

\[ \gamma_i = \frac{d_i}{\beta_i} \]  

Finally, the substitution of the recursion solution into the last equation of Eq. (5.84) yields

\[ T_N = \frac{d_N - a_N T_{N-1}}{b_N} = \frac{d_N - a_N \left( \frac{\gamma_N - 1}{\beta_{N-1}} T_N \right)}{b_N} \]  

from which
In a nutshell, the complete algorithm for the solution of the tridiagonal system is

\[ T_N = \gamma_N \quad \ldots (5.94) \]

\[ T_i = \gamma_i - \frac{c_i T_{i+1}}{\beta_i}, \quad i = N-1, N-2, \ldots, 1 \quad \ldots (5.95) \]

where, \( \beta \)'s and \( \gamma \)'s are determined from the recursion formulae

\[ \beta_i = b_i, \quad \gamma_1 = \frac{d_1}{\beta_1} \quad \ldots (5.96) \]

\[ \beta_i = b_i - \frac{a_i \gamma_{i-1}}{\beta_{i-1}}, \quad i = 2, 3, \ldots, N \quad \ldots (5.97) \]

\[ \gamma_i = \frac{d_i - a_i \gamma_{i-1}}{\beta_i}, \quad i = 2, 3, \ldots, N \quad \ldots (5.98) \]

The above algorithm is also known as the Thomas algorithm.

Finally, it is to be noted that Eq. (5.78) might also be solved by the Gauss-Seidel iteration scheme discussed next.

The Gauss-Seidel Iterative Method (GS)

For a large number of equations (typically of the order of several hundred) iterative methods such as Jacobi, Gauss-Seidel, which initiate the computations with a guessed solution and iterate to the desired solution of the systems of equations within a specified convergence criterion, using improved guesses in the second and third iterations till the final one, are often more efficient. In the GS method only the values of the latest iteration are stored, and each iterative computation of the unknown employs the most recent values of the other unknowns. In this method, unlike in direct methods, such as Gaussian elimination, the round-off error does not accumulate. The round-off error after each iteration simply produces a less accurate input for the next iteration. Therefore, the resulting round-off error in the numerical solution is only what arises in the computation for the final iteration. However, the solution is not exact but is obtained to an arbitrary, specified, convergence criterion.

Convergence Criteria for the GS Method

Typical convergence criteria used are

1. \[ \left| x_i^{(p+1)} - x_i^{(p)} \right| \leq \varepsilon \quad \text{for} \quad i = 1, 2, \ldots, N \]

2. \[ \left| \frac{x_i^{(p+1)} - x_i^{(p)}}{x_i^{(p)}} \right| \leq \varepsilon \quad \text{for} \quad i = 1, 2, \ldots, N \]

where \( \varepsilon \) is a very small number, e.g. 0.01, 0.001, 0.00001. Criterion 2 is applicable if an estimate of the magnitude of the unknowns \( x_i \) is not available and none of the unknowns is expected to be zero.

Conditions for Convergence in the GS Method: Scarborough Criterion
Convergence is guaranteed for linear systems if
\[ |a_{ii}| \geq \sum_{j=1, j \neq i}^{N} |a_{ij}| \text{ for all } i, \]
and if
\[ |a_{ii}| > \sum_{j=1, j \neq i}^{N} |a_{ij}| \text{ for at least one } i, \]
that is, when the system is diagonally dominant. This is also known as the Scarborough criterion. This is a sufficient condition, which means that convergence may still be possible even if the above condition is not satisfied. Fortunately, it turns out that in fluid flow and heat transfer problems; finite-difference formulation indeed leads to a diagonally dominant coefficient matrix, which is the reason why for large systems the Gauss-Seidel method is so widely used.

**Application of the GS Iterative Method**

In order to demonstrate the iteration process, the following system of three linear equations is solved by the GS iterative method using a pocket calculator:

\[
\begin{align*}
10x_1 + x_2 + 2x_3 &= 44 \\
2x_1 + 10x_2 + x_3 &= 51 \\
x_1 + 2x_2 + 10x_3 &= 61
\end{align*}
\]  
\[\ldots (5.99)\]

Clearly, the coefficient matrix in Eq. (5.99) is diagonally dominant because
\[
\begin{align*}
|10| \geq |1| + |2| \\
|10| \geq |2| + |1| \\
|10| \geq |1| + |2|
\end{align*}
\]

Therefore, the Scarborough criterion is satisfied and hence, one is certain to get a converged solution using the GS iterative method. As a first guess, let us take

\[
[x_1, x_2, x_3]^0 = [0, 0, 0]
\]

We take \( \varepsilon = 0.02 \)

Then,
\[
x_1^{(1)} = \frac{1}{10} [44 - x_2^{(0)} - 2x_3^{(0)}] = \frac{1}{10} [44 - (0) - 2 (0)] = 4.40
\]
\[
x_2^{(1)} = \frac{1}{10} [51 - 2x_1^{(0)} - x_3^{(0)}] = \frac{1}{10} [51 - 2 (4.40) - 0] = 4.22
\]
\[
x_3^{(1)} = \frac{1}{10} [61 - x_1^{(0)} - 2x_2^{(0)}] = \frac{1}{10} [61 - 4.40 - 2 (4.22)] = 4.81
\]

Now, check for convergence after the first iteration:
\[
\begin{align*}
0 - 4.40 &= 4.40 \varepsilon \\
0 - 4.22 &= 4.22 \varepsilon \\
0 - 4.81 &= 4.81 \varepsilon
\end{align*}
\]

We find that there is no convergence. One more iteration gives

\[
[x_1^{(2)}, x_2^{(2)}, x_3^{(2)}] = [3.01, 4.01, 4.99]
\]

Again check the convergence after the second iteration:
Conduction

| 3.01−4.04 | = 1.39 > ε |
| 4.01−4.22 | = 0.21 > ε |
| 4.81−4.99 | = 0.18 > ε |

Still there is no convergence. One more iteration gives

\[ [x_1, x_2, x_3]^3 = [3.00, 4.00, 5.00] \]

We again check for convergence after the third iteration:

| 3.01−3.00 | = 0.01 < ε |
| 4.01−4.00 | = 0.01 < ε |
| 4.99−5.00 | = 0.01 < ε |

We see that convergence is reached. Hence, further computation stops. Therefore, it required three iterations to obtain a converged solution. Incidentally, the converged solution is also the exact solution of this set of equations. The reason is that the number of unknowns is very small in this case.

Relaxation : Over-Relaxation and Under-Relaxation

One of the problems with the GS method is that it is relatively slow to converge to the solution. The rate of convergence can often be improved by the relaxation method which is explained next.

Let us consider the example that has been solved by the GS method.

\[ x_i^{(0)} = x_i^{(0)} + \frac{1}{10} \left(44 - x_2^{(0)} - 2x_3^{(0)}\right) - x_i^{(0)} \] (5.100)

Now the change produced by the current iteration can be increased if we multiply it by a factor \( \alpha (\alpha > 1) \). However, \( \alpha \) also has an upper limit. For \( \alpha > 2 \), the change is so great that instead of convergence, divergence occurs, that is, the solution never converges. Therefore, Eq. (5.100) can now be written as

\[ x_i^{(0)} = x_i^{(0)} + \alpha \left[ \frac{1}{10} \left(44 - x_2^{(0)} - 2x_3^{(0)}\right) - x_i^{(0)} \right] \]

or

\[ x_i^{(0)} = \frac{\alpha}{10} \left(44 - x_2^{(0)} - 2x_3^{(0)}\right) + x_i^{(0)} \left(1 - \alpha\right) \] . . . (5.101)

From Eq. (5.101), it is readily seen that for

\[ \alpha = 0, \ x_i^{(1)} = x_i^{(0)} \] (no progress)

\[ \alpha = 1, \ x_i^{(1)} = x_i^{(1)}_{\text{GS}} \] (basic GS iteration)

0 < \( \alpha < 1 \), under-relaxation \( \rightarrow \) interpolation between \( x_i^{(0)} \) and \( x_i^{(1)}_{\text{GS}} \)

1 < \( \alpha < 2 \) \( \rightarrow \) over-relaxation \( \rightarrow \) extrapolation beyond \( x_i^{(1)}_{\text{GS}} \)

In a compact form, the relaxation method may be written as
Numerical Methods to Solve Heat Conduction Problems

Optimum Relaxation Factor $\alpha_{\text{opt}}$

The question is: What value of the over-relaxation factor should be used? There is no set rule to determine this. One has to do numerical experimentation to find out the relaxation factor which gives the highest rate of convergence. This is called the optimum relaxation factor $\alpha_{\text{opt}}$ which lies between 1 and 2 and varies from one problem to another. For the simple case of Laplace’s equation $(\nabla^2 T = 0)$ in a square with Dirichlet boundary conditions (that is, known temperature on the boundaries), Young (1954) and Frankel (1950) show that $\alpha_{\text{opt}}$ equals the smaller root of

$$t^2 \alpha^2 - 16\alpha + 16 = 0 \quad \ldots \ (5.103)$$

with $t = 2 \cos \left( \frac{\pi}{n} \right)$, where $n$ is the total number of increments into which the side of the square is divided. In other words, $n$ is the number of grid spacing. The number of iterations required for a given convergence criterion falls very rapidly when the parameter is in the immediate vicinity of $\alpha_{\text{opt}}$, and it is generally better to overestimate $\alpha_{\text{opt}}$ than to underestimate it (Carnahan, et al. 1969).

Solution of Eq. (5.80) by all Three methods

We shall now solve Eq. (5.80) having a tridiagonal coefficient matrix by the Gaussian elimination, TDMA, and Gauss-Seidel iterative method and choose the proper method.

Recall Eq. (5.80) is given below:

$$
\begin{bmatrix}
D & -1 & & & \\
-1 & D & -1 & & \\
& -1 & D & -1 & \\
& & -2 & D & \\
& & & & D
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
$$

where $D = 2 + (mL)^2 \ (\Delta X)^2$. Let us consider a fin with $mL = 2$. Also, $\Delta X = \frac{1}{4} = 0.25$. Therefore, $D = 2 + (2)^2 \ (0.25)^2 = 2.25$. Hence, Eq. (5.80) now becomes

$$
\begin{bmatrix}
2.25 & -1 & & & \\
-1 & 2.25 & -1 & & \\
& -1 & 2.25 & -1 & \\
& & -2 & 2.25 & \\
& & & & D
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\quad \ldots \ (5.104)
$$

Solution by Gaussian Elimination

Step I

The pivot is 2.25. So, first eliminate all terms below the pivot.
Conduction

\[
\begin{bmatrix}
R1 + R2 \\
R2 + R3 \\
R3 \times 1.176 + R4
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0.444 \\
0.246 \\
0.290
\end{bmatrix}
\]

**Step II**

Now, the pivot is 1.81. So eliminate all terms below the pivot.

\[
\begin{bmatrix}
2.25 & -1 \\
1.81 & -1 \\
1.7 & -1 \\
2.25 & -1
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0.444 \\
0.246 \\
0.290
\end{bmatrix}
\]

**Step III**

Now, the pivot is 1.7. So eliminate all terms below the pivot.

\[
\begin{bmatrix}
2.25 & -1 \\
1.81 & -1 \\
1.7 & -1 \\
1.07
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0.444 \\
0.246 \\
0.290
\end{bmatrix}
\]

**Step IV**

The last step as now the triangular coefficient matrix is produced. \( R1, R2, R3, R4 \) are used to denote the first, second, third, and fourth rows, respectively. The unknowns are obtained by back substitution.

**Back Substitution**

\[
\theta_4 = \frac{0.290}{1.07} = 0.271
\]

\[
1.7\theta_3 - \theta_4 = 0.246 \Rightarrow \theta_3 = 0.304
\]

\[
1.81\theta_2 - \theta_3 = 0.444 \Rightarrow \theta_2 = 0.413
\]

\[
2.25\theta_1 - \theta_2 = 1 \Rightarrow \theta_1 = 0.628
\]

Therefore, the unknown temperatures are

\[
\theta_1 = 0.628, \theta_2 = 0.413, \theta_3 = 0.304, \theta_4 = 0.271
\]

The total number of arithmetic operations (multiplications and divisions) to obtain the solution is 13 (9 for elimination and 4 for back substitution).

**Solution by TDMA**

Recall the tridiagonal matrix algorithm given by Eqs. (5.94) and (5.97). With respect to Eq. (5.104),

\[
d_1 = 1, d_2 = 0, d_3 = 0, d_4 = 0
\]

\[
b_1 = 2.25, b_2 = 2.25, b_3 = 2.25, b_4 = 2.25
\]

\[
a_2 = -1, a_3 = -1, a_4 = -2
\]

\[
c_1 = -1, c_2 = -1, c_3 = -1
\]

From the TDMA,

\[
\theta_4 = \gamma_4,
\]

\[
\beta_1 = b_1 = 2.25
\]
Numerical Methods to Solve Heat Conduction Problems

\( \gamma_1 = \frac{d_1}{\beta_1} = \frac{1}{2.25} = 0.444 \)

\( \beta_2 = b_2 - \left( \frac{a_2 c_1}{\beta_2} \right) = 2.25 - \left[ \frac{(1)(-1)}{2.25} \right] = 1.805 \)

\( \beta_3 = b_3 - \left( \frac{a_3 c_2}{\beta_2} \right) = 2.25 - \left[ \frac{(-1)(-1)}{1.805} \right] = 1.695 \)

\( \beta_4 = b_3 - \left( \frac{a_4 c_3}{\beta_3} \right) = 2.25 - \left[ \frac{(-2)(-1)}{1.69} \right] = 1.07 \)

\( \gamma_2 = \frac{(d_2 - a_2 \gamma_1)}{\beta_2} = \frac{[0 - (-1)(0.444)]}{1.805} = 0.246 \)

\( \gamma_3 = \frac{(d_3 - a_3 \gamma_2)}{\beta_3} = \frac{[0 - (-1)(0.246)]}{1.695} = 0.145 \)

\( \gamma_4 = \frac{(d_4 - a_4 \gamma_3)}{\beta_4} = \frac{[0 - (-2)(0.145)]}{1.07} = 0.271 \)

\( \theta_4 = \gamma_4 = 0.271 \)

\( \theta_3 = \gamma_3 - \frac{c_3 \theta_4}{\beta_3} = 0.145 - \frac{(-1)(0.271)}{1.695} = 0.304 \)

\( \theta_2 = \gamma_2 - \frac{c_2 \theta_3}{\beta_2} = 0.246 - \frac{(-1)(0.304)}{1.805} = 0.414 \)

\( \theta_1 = \gamma_1 - \frac{c_1 \theta_2}{\beta_1} = 0.444 - \frac{(-1)(0.414)}{2.25} = 0.628 \)

Therefore, \( \theta_1 = 0.628, \theta_2 = 0.414, \theta_3 = 0.304, \theta_4 = 0.271 \)

The total number of arithmetic operations required to obtain the solution is 10.

**Solution by the Gauss-Seidel Iteration**

Let \( [\theta_1, \theta_2, \theta_3, \theta_4]^0 = [1, 1, 1, 1] \) and \( \varepsilon = 0.001 \).

Iteration 1: \( [\theta_1, \theta_2, \theta_3, \theta_4]^1 = [0.888, 0.839, 0.817, 0.726] \)

Iteration 2: \( [\theta_1, \theta_2, \theta_3, \theta_4]^2 = [0.817, 0.726, 0.645, 0.573] \)

Iteration 3: \( [\theta_1, \theta_2, \theta_3, \theta_4]^3 = [0.767, 0.627, 0.6, 0.533] \)

Iteration 4: \( [\theta_1, \theta_2, \theta_3, \theta_4]^4 = [0.723, 0.588, 0.498, 0.442] \)

Iteration 5: \( [\theta_1, \theta_2, \theta_3, \theta_4]^5 = [0.705, 0.535, 0.434, 0.385] \)

Iteration 6: \( [\theta_1, \theta_2, \theta_3, \theta_4]^6 = [0.682, 0.496, 0.391, 0.347] \)

Iteration 7: \( [\theta_1, \theta_2, \theta_3, \theta_4]^7 = [0.664, 0.468, 0.362, 0.321] \)

Iteration 8: \( [\theta_1, \theta_2, \theta_3, \theta_4]^8 = [0.652, 0.45, 0.342, 0.304] \)

Iteration 9: \( [\theta_1, \theta_2, \theta_3, \theta_4]^9 = [0.664, 0.438, 0.329, 0.292] \)

Iteration 10: \( [\theta_1, \theta_2, \theta_3, \theta_4]^{10} = [0.639, 0.43, 0.32, 0.284] \)
Conduction

Iteration 11 : \([\theta_1, \theta_2, \theta_3, \theta_4]^{11} = [0.635, 0.424, 0.314, 0.279]\)

Iteration 12 : \([\theta_1, \theta_2, \theta_3, \theta_4]^{12} = [0.632, 0.42, 0.31, 0.275]\)

Iteration 13 : \([\theta_1, \theta_2, \theta_3, \theta_4]^{13} = [0.631, 0.418, 0.308, 0.273]\)

Iteration 14 : \([\theta_1, \theta_2, \theta_3, \theta_4]^{14} = [0.63, 0.416, 0.306, 0.272]\)

Iteration 15 : \([\theta_1, \theta_2, \theta_3, \theta_4]^{15} = [0.629, 0.415, 0.305, 0.271]\)

Convergence is reached on the 15th iteration. Hence, further computation stops.

Total number of iterations required = 15

Total number of arithmetic operations in each iteration = 5

Therefore, total number of arithmetic operations = 75

The Number of Arithmetic Operations for each Method : A Comparison

As can be seen from Table 5.1, it is obvious that TDMA is the fastest method and the Gauss-Seidel iteration is the worst method for solving a tridiagonal system of equations. Therefore, the choice falls on the TDMA for the solution of Eq. (5.80).

Table 5.1 : A Comparative Study of the Number of Arithmetic Operations Needed to Solve Eq. (5.80) by Three Methods

<table>
<thead>
<tr>
<th>Name of the Method Operations</th>
<th>Number of Arithmetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian elimination</td>
<td>13</td>
</tr>
<tr>
<td>Tridiagonal matrix algorithm (TDMA)</td>
<td>10</td>
</tr>
<tr>
<td>Gauss-Seidel iteration</td>
<td>75</td>
</tr>
</tbody>
</table>

As initial guess = \([1, 1, 1, 1]\) \(\varepsilon = 0.001\)

Hence, for solving a tridiagonal system of linear equations, Thomas algorithm (TDMA) is preferred.

Checking for Accuracy

The accuracy of a numerical solution is usually checked in one of the three ways:

- Comparison with the analytical solution: For most practical problems analytical solutions do not exist. But, this is a good way of checking the accuracy of a new numerical method.

- Comparison with the limiting case analytical solution: This is possible when the analytical solution for some limiting value of a parameter governing the solution is available.

- Comparison with experimental results: This is most desirable for complex problems, such as turbulence, combustion, non-Newtonian fluid flow, and heat transfer, which require many assumptions for the purpose of modeling.

Comparison of the Present Numerical Result with the Corresponding Analytical Solution

For the present problem of heat conduction in a fin, an analytical solution is available. Therefore, a comparison of the numerical results with the exact solution \((for mL = 2)\) will enable us to obtain an estimate of the numerical error. Table 5.2 gives a comparison of the numerical and analytical solutions.
Table 5.2: A Comparison of the Numerical and Analytical Solutions of the Fin Problem

<table>
<thead>
<tr>
<th>Location $X$</th>
<th>Temperature $\theta$</th>
<th>Absolute Percent Error with Respect to the Exact Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0^a$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0.628</td>
<td>0.625</td>
</tr>
<tr>
<td>0.48</td>
<td>0.414</td>
<td>0.410</td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.304</td>
<td>0.2995</td>
</tr>
<tr>
<td>1.50</td>
<td>0.271</td>
<td>0.266</td>
</tr>
</tbody>
</table>

Dirichlet boundary condition and hence not computed Table 5.2 clearly reveals that $\Delta X = 0.25$ is not good enough and the grid spacing needs to be finer. In other words, a higher number of grid points are necessary to obtain a more accurate solution. However, one has to be also careful in increasing the number of grid points as this will result in a higher round-off error. Therefore, a grid independence test, which gives an optimum $\Delta X$, is called for. A point to note is that even with a relatively coarse grid the accuracy is quite good. This means that with a slight decrease in the grid spacings, the numerical solution will be even closer to its analytical counterpart. Another interesting feature of Table 5.2 is the gradually increasing error for increasing $X$. This is possible because of the fact that at the left boundary ($X = 0$) the Dirichlet condition is imposed and, therefore, for both numerical and exact solutions the same temperature is used for the calculation of temperature at $X = 0.25$. Hence, the temperature of the grid point closest to the left boundary (i.e., at $X = 0.25$) computed by the numerical method is most accurate and the error accumulates as the distance of a grid point with respect to the left boundary increases.

Convective Boundary Condition

If the tip of the fin was convective instead of insulated, the discretisation equation at $i = M$ would have to be modified. The dimensionless boundary condition at the fin tip in the changed scenario would be written in the mathematical form as

$$\frac{d\theta}{dX} + \frac{h_e L}{k} \theta = 0 \quad \ldots (5.105)$$

where $h_e$ is the convection heat transfer coefficient from the tip of the fin to the surroundings.

Using the image-point technique as discussed earlier and the central-difference scheme for discretisation of $\frac{d\theta}{dX}$ at $i = M$, Eq. (5.105) is expressed as

$$\frac{\theta_{M+1} - \theta_{M-1}}{2\Delta X} + \frac{h_e L}{k} \theta_M = 0 \quad \ldots (5.106)$$

or

$$\theta_{M+1} = \theta_{M-1} - \frac{2h_e L \Delta X}{k} \theta_M \quad \ldots (5.107)$$

Substituting the expression for $\theta_{M+1}$ from Eq. (5.107) into Eq. (5.74), we obtain

$$2\theta_{M-1} - \theta_M \left(D + \frac{2h_e L \Delta X}{k}\right) = 0 \quad \ldots (5.108)$$
Therefore, only the last equation is changed. The method of solution remains the same as before. To check the accuracy of Eq. (5.107), substituting \( h_e = 0 \) (corresponding to the insulation condition), we obtain

\[
2\theta_{M-1} - D \theta_M = 0
\]

Which is the same as that obtained for the insulated tip.

### 5.7.2 Numerical Methods for Two-Dimensional Steady-State Problems

Consider the case of steady heat conduction in a long square slab \((2L \times 2L)\) in which heat is generated at a uniform rate of \( q'' W m^{-3} \). The problem can be assumed to be a two-dimensional one as the dimension of the slab is much longer in the direction normal to the cross-sectional plane. Therefore, the end efforts can be neglected. All four sides are maintained at \( T = T_\infty \), temperature of the surrounding fluid, assuming a large heat transfer coefficient.

![Physical Domain of the Slab with Square Cross Section (2L x 2L)](image)

**Consideration of Symmetry**

A close look at the physics of the problem reveals that the problem is geometrically and thermally symmetric. Therefore, from the temperature distribution in any quarter of the physical domain, by mirror-imaging one can get the solution for the entire region. Figure 5.12 shows the computational domain (top right-hand quarter). The use of symmetry enables the numerical analyst to obtain the solution much faster as the number of grid points is greatly reduced.

![Computational Domain (Top Right Hand Quarter) Considering Symmetry](image)
Numerical Methods to Solve Heat Conduction Problems

Governing Differential Equation

The governing non-dimensional energy equation (assuming constant $k$) is

$$\frac{\partial^2 \theta}{\partial X^2} + \frac{\partial^2 \theta}{\partial Y^2} + 1 = 0 \quad \ldots (5.110)$$

where

$$\theta = \frac{T - T_w}{q^* L^2}, \quad X = \frac{x}{L}, \quad Y = \frac{y}{L}$$

Boundary Conditions

The non-dimensional boundary conditions are as follows:

At $X = 0$, \( \frac{\partial \theta}{\partial X} = 0 \) \quad \ldots (5.111(a))

At $X = 1$, \( \theta = 0 \) \quad \ldots (5.111(b))

At $Y = 0$, \( \frac{\partial \theta}{\partial Y} = 0 \) \quad \ldots (5.111(c))

Discretisation

The computational domain including the notations for the interior grid points is shown in Figure 5.13.

Figure 5.13 : Interior Grid Points in the Computational Domain

Eq. (5.110) is discretised using the central difference for \( \frac{\partial^2 \theta}{\partial X^2} \) and \( \frac{\partial^2 \theta}{\partial Y^2} \) at the interior grid point \((i,j)\) as follows:

$$\theta_{i-1,j} - 2\theta_{i,j} + \theta_{i+1,j} + \theta_{i-1,j} - 2\theta_{i,j} + \theta_{i,j+1} + 1 = 0 \quad \ldots (5.112)$$

Taking $\Delta X = \Delta Y$, Eq. (5.112) reduces to

$$- \theta_{i-1,j} + \theta_{i,j-1} + 4\theta_{i,j} - \theta_{i,j-1} - \theta_{i,j+1} = (\Delta X)^2 \quad \ldots (5.113)$$

Boundary Condition along $X = 0$

Using the image-point technique,

$$\theta_{i-1,j} = \theta_{i+1,j} \quad \ldots (5.114)$$

and setting \( i = 0 \), Eq. (5.112) becomes

$$- \theta_{2,j} + \theta_{1,j-1} + 4\theta_{1,j} - \theta_{1,j-1} - \theta_{1,j+1} = (\Delta X)^2 \quad \ldots (5.115)$$
Conduction

Boundary Condition along \( Y = 0 \)

Using the image-point technique

\[
\theta_{i,j-1} = \theta_{i,j+1}
\]

and setting \( j = 1 \), Eq. (5.112) becomes

\[
-\theta_{i+1,1} - 2\theta_{i,2} + 4\theta_{i,1} - \theta_{i-1,1} = (\Delta X)^2
\]  \( \ldots (5.117) \)

Handling of Corner Points

Corner points need special attention because they belong to both horizontal and vertical surfaces. Therefore, the boundary conditions at both the surfaces apply there. However, if one or both of the surfaces have the Dirichlet condition (specified temperature), then there is no problem because the corner point can be assumed to have a specified temperature. But, if both surfaces have Neumann (insulation) and/or Robbins (convective) conditions, then the corner point needs to be handled separately since both conditions exist there. With respect to the present problem, out of the four corner points, only the bottom-left corner point is exposed to Neumann conditions in \( X \) and \( Y \) directions. Other three have either one or both surfaces exposed to the Dirichlet condition. Figure 5.14 shows the image points for the bottom-left-hand corner represented by the grid point \((1,1)\).

![Figure 5.14: Image Points for the Bottom Left Hand Corner Point (1,1)](image)

Using the image-point technique,

\[
\theta_{0,1} = \theta_{2,1}
\]  \( \ldots (5.118) \)

\[
\theta_{1,0} = \theta_{1,2}
\]  \( \ldots (5.119) \)

Now,

\[
\frac{\partial^2 \theta}{\partial X^2}_{1,1} = \frac{\theta_{0,1} - 2\theta_{1,1} - \theta_{2,1}}{(\Delta X)^2}
\]  \( \ldots (5.120) \)

\[
\frac{\partial^2 \theta}{\partial Y^2}_{1,1} = \frac{\theta_{1,0} - 2\theta_{1,1} + \theta_{1,2}}{(\Delta Y)^2}
\]  \( \ldots (5.121) \)

Substituting Eq. (5.118) into Eq. (5.120), and Eq. (5.119) into Eq. (5.121). We get

\[
\frac{\partial^2 \theta}{\partial X^2}_{1,1} = \frac{2\theta_{0,1} - 2\theta_{1,1}}{(\Delta X)^2}
\]  \( \ldots (5.122) \)

\[
\frac{\partial^2 \theta}{\partial Y^2}_{1,1} = \frac{2\theta_{1,0} - 2\theta_{1,1}}{(\Delta Y)^2}
\]  \( \ldots (5.123) \)

Setting \( i = 1, j = 1 \) and substituting Eqs. (5.122) and (5.123) into Eq. (5.112), we obtain for \( \Delta X = \Delta Y \),

\[
2\theta_{2,1} - 4\theta_{1,1} + 2\theta_{1,2} + (\Delta X)^2 = 0
\]  \( \ldots (5.124) \)
Methods of Solution

Let us consider an example in which \( \Delta X = \frac{1}{4} \). The grid points that are unlabelled (Figure 5.15) are all at temperature \( \theta = 0 \) as imposed by the boundary condition. Thus, there are 16 unknown temperature to find (Figure 5.16).

\[
\begin{bmatrix}
4 & -2 & 0 & 0 & -2 \\
-1 & 4 & -1 & 0 & 0 & -2 \\
0 & -1 & 4 & -1 & 0 & 0 & -2 \\
0 & 0 & -1 & 4 & 0 & 0 & 0 & -2 \\
-1 & 0 & 0 & 0 & 4 & -2 & 0 & -1 \\
-1 & 0 & 0 & 0 & 1 & 4 & -1 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & -1 \\
-1 & 0 & 0 & -1 & 1 & 4 & -1 & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & -1 \\
-1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0_1 \\
0_2 \\
0_3 \\
0_4 \\
0_5 \\
0_6 \\
0_7 \\
0_8 \\
0_9 \\
0_{10} \\
0_{11} \\
0_{12} \\
0_{13} \\
0_{14} \\
0_{15} \\
0_{16}
\end{bmatrix} = \begin{bmatrix} 1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \\
1/16 \end{bmatrix}
\]

Figure 5.15: Labelling of Grid Points using Double Subscript Notation

Figure 5.16: Labelling of the Same Grid Points using Single Subscript Notation

Since there are 16 unknowns, there will be 16 equations to solve. Using the matrix representation for these equations, Eq. (5.124(a)) is obtained from Eqs. (5.113)-(5.117) and Eq. (5.124)
Equation (5.124(a))

A close look at Eq. (5.124(a)) reveals the following:

- The division of $X$ and $Y$ into relatively coarse subdivisions leads to many equations ($4 \times 4 = 16$). In a practical situation, the number of equations may be hundred or more.

- The coefficient matrix is banded, which means that the non-zero components only appear in a band on either side of the main diagonal. There are 9 diagonals (Figure 5.17). So, the bandwidth is large as compared to TDM. The advantage of having a banded matrix (this is true also for TDM) is that special sub-routines can be written to solve the problem in less computer time than if the matrix was filled with non-zero components.

![Figure 5.17: Pictorial Representation of the Banded Coefficient Matrix Showing Nine Diagonals](image)

- The zero matrix components outside the band need not be stored in the computer. This is of great significance in large problems where the computer memory size becomes a limiting factor.

Choice of the Proper Method

Eq. (5.124(a)) can be solved in two ways:

(a) By Gaussian elimination, and

(b) By Gauss-Seidel iteration.

Let us weight the pros and cons of both methods before we make our final choice. It is interesting to note that the banded coefficient matrix in Eq. (5.124(a)) has 124 components within the band rather than the 256 spaces that would have been required to store the entire matrix. One could even reduce the bandwidth by recognizing the physical and geometrical symmetry across one of the diagonals of the square as shown in Figure 5.17. This means that $\theta_{i, j} = \theta_{j, i}$ or $\theta_2 = \theta_5$, $\theta_3 = \theta_9$, $\theta_4 = \theta_{13}$, and so on. This would reduce the number of equations from 16 to 10.

Furthermore, it may be noted that many of the components within the band itself are zero. In this case, 60 of 124 band components are zero. These components must still be stored, however, if Gaussian elimination is to be used, because during the elimination process they will, in general, change to non-zero values. If computer storage is critical, one might prefer to use a method that does not require storing these zero component in the band. The Gauss-Seidel iteration method is one way of doing this. In addition to this, the round-off error is minimum in the Gauss-Seidel method. Therefore, in view of the aforesaid two overwhelming merits, in spite of the clear-cut advantage of Gaussian elimination because of its non-iterative nature, the GS method is chosen to solve Eq. (5.124(a)).
Check for Accuracy

For the present problem, the accuracy of the numerical results can be checked by comparing it with the corresponding analytical solution available. Subsequently, a grid independence test must be done to obtain the desired results. The analytical (or exact) solution in the dimensionless form is given below:

\[ \theta (X, Y) = \frac{1}{2} \left[ 1 - X^2 \right] - 2 \sum_{n=0}^{\infty} \frac{(-1)^n \cosh \lambda_n Y}{\lambda_n^3 \cosh \lambda_n} \quad \ldots \ (5.125) \]

where \( \lambda_n = \frac{(2n + 1) \pi}{2} \), where \( n = 0, 1, 2, \ldots \)

5.8 TRANSIENT ONE-DIMENSIONAL PROBLEMS

Consider a hot infinite plate (Figure 5.18) of finite thickness \( 2L \). The plate is suddenly exposed to a cool fluid at \( T_\infty \). The initial temperature of the plate is \( T_0 \). The heat transfer coefficient is large. We wish to find the temperature of the plate as a function of space and time using a numerical method.

The problem can be modeled as a one-dimensional, unsteady-state problem because \( \frac{\partial T}{\partial y} = \frac{\partial T}{\partial z} = 0 \) as the plate is infinitely long in \( y \)- and \( z \)-directions.

Consideration of Symmetry

Since the problem is a thermally and geometrically symmetric one, only one half of the plate can be taken as the computational domain with the insulation boundary condition at \( x = L \) (Figure 5.19).
**Conduction**

**Governing Differential Equation**

For constant thermophysical properties $k$, $\rho$, $c$ the non-dimensional energy equation for the plate is

$$\frac{\partial \theta}{\partial \tau} = \frac{\partial^2 \theta}{\partial X^2}$$

where

$$\theta = \frac{T - T_\infty}{T_i - T_\infty}, \quad X = \frac{x}{L}, \quad \tau = \frac{at}{L^2}$$

**Initial and Boundary Conditions**

The initial and boundary conditions are as follows:

**IC**

At $\tau = 0$, $\theta = 1$ for all $x$  \( \ldots (5.127(a)) \)

For $\tau > 0$, BC 1

At $X = 0$, $\theta = 0$  \( \ldots (5.127(b)) \)

BC 2

At $X = 1$, $\frac{\partial \theta}{\partial X} = 0$  \( \ldots (5.127(c)) \)

**Discretisation**

For any interior grid point, the finite difference formulation give,

$$\frac{\partial \theta_i}{\partial \tau} = \frac{\theta_{i-1} + \theta_{i+1} - 2\theta_i}{(\Delta x)^2}$$  \( \ldots (5.128) \)

For $i = 1, \ldots, M$

The equation for $X = 1$ is obtained by using the image-point technique, i.e. by substituting $\theta_{M+1} = \theta_{M-1}$ in Eq. (5.128) for $i = M$. Therefore,

$$\frac{\partial \theta_M}{\partial \tau} = \frac{2\theta_{M-1} - 2\theta_M}{(\Delta X)^2}$$  \( \ldots (5.129) \)

For the sake of demonstration, let us take four equal subdivisions in the $X$-direction (Figure 5.20). Therefore, $\Delta X = \frac{1}{4}$.

**Figure 5.20**: Equally Spaced Grid points in the $x$-direction of the Computational Domain

At $i = 0$, $\theta = 0$ (known)

At $i = 1, 2, 3, \ldots$
From Eq. (5.128), we obtain
\[
\frac{d\theta_1}{d\tau} = \frac{1}{(\Delta X)^2} \left( \theta_0 - 2\theta_1 + \theta_2 \right) \quad \ldots (5.130)
\]
\[
\frac{d\theta_2}{d\tau} = \frac{1}{(\Delta X)^2} \left( \theta_1 - 2\theta_2 + \theta_3 \right) \quad \ldots (5.131)
\]
\[
\frac{d\theta_3}{d\tau} = \frac{1}{(\Delta X)^2} \left( \theta_2 - 2\theta_3 + \theta_4 \right) \quad \ldots (5.132)
\]

At \( i = 4 \), from Eq. (5.129), we obtain
\[
\frac{d\theta_4}{d\tau} = \frac{1}{(\Delta X)^2} \left( 2\theta_3 - 2\theta_4 \right) \quad \ldots (5.133)
\]

Thus, we have four simultaneous ordinary differential equations [Eqs. 5.130-5.131] to solve. This system of ordinary differential equations may be classified as initial value problems. This is because these equations are to be solved for the unknowns as a function of time, beginning with an initial value for each of the unknowns. In this case the initial values are obtained from the initial temperature distribution in the plate, which is given by
\[
\theta_1(0) = \theta_2(0) = \theta_3(0) = \theta_4(0) = 1 \quad \ldots (5.134)
\]

Methods of Solution

There are three methods by which this initial-value problem can be solved. These are the

(a) Euler (or explicit),

(b) Crank-Nicholson, and

(c) Pure implicit methods.

The Euler Method (also known as the Explicit Method)

Since the given problem is an initial-value one, we will know the solution \( \theta^0 \) and will seek \( \theta^{p+1} \) at some later point in time \( \tau^{p+1} = \tau^p + \Delta \tau \). In the Euler method of solution, the solution at a future time \( \tau^{p+1} \) is obtained by computing the derivative at the present time \( \tau^p \) and then by moving ahead in time in the following way:

\[
\theta^{p+1} = \theta^p + \frac{d\theta}{d\tau} \bigg|_{\tau^p} \Delta \tau \quad \ldots (5.135)
\]

The Euler scheme is pictorially represented in Figure 5.21 for the grid points 1, 2, 3, 4. Eq. (5.135) can be written as

\[
\theta_1^{p+1} = \theta_1^p + \frac{d\theta_1}{d\tau} \bigg|_{\tau^p} \Delta \tau \quad \ldots (5.136(a))
\]

\[
\theta_2^{p+1} = \theta_2^p + \frac{d\theta_2}{d\tau} \bigg|_{\tau^p} \Delta \tau \quad \ldots (5.136(b))
\]

\[
\theta_3^{p+1} = \theta_3^p + \frac{d\theta_3}{d\tau} \bigg|_{\tau^p} \Delta \tau \quad \ldots (5.136(c))
\]

\[
\theta_4^{p+1} = \theta_4^p + \frac{d\theta_4}{d\tau} \bigg|_{\tau^p} \Delta \tau \quad \ldots (5.136(d))
\]
Conduction

Figure 5.21: Pictorial Representation of the Euler Method

Substituting Eq. (5.136(a)) into Eq. (5.130), Eq. (5.136(b)) into Eq. (5.131), Eq. (5.136(c)) into Eq. (5.132), Eq. (5.136(d)) into Eq. (5.133), we obtain

\[
\frac{\theta_p^{t+1} - \theta_p^t}{\Delta t} = \frac{1}{(\Delta X)^2} (-2\theta_p^t + \theta_p^{t+1}) \quad \ldots (5.137)
\]

\[
\frac{\theta_q^{t+1} - \theta_q^t}{\Delta t} = \frac{1}{(\Delta X)^2} (\theta_q^t - 2\theta_q^t + \theta_q^{t+1}) \quad \ldots (5.138)
\]

\[
\frac{\theta_r^{t+1} - \theta_r^t}{\Delta t} = \frac{1}{(\Delta X)^2} (\theta_r^t - 2\theta_r^t + \theta_r^{t+1}) \quad \ldots (5.139)
\]

\[
\frac{\theta_s^{t+1} - \theta_s^t}{\Delta t} = \frac{1}{(\Delta X)^2} (2\theta_s^t - 2\theta_s^t) \quad \ldots (5.140)
\]

Eqs. (5.137)-(5.140) are then rearranged to give

\[
\theta_p^{t+1} = \left(1 - \frac{2\Delta t}{(\Delta X)^2}\right) \theta_p^t + \frac{\Delta t}{(\Delta X)^2} \theta_p^{t+1} \quad \ldots (5.141)
\]

\[
\theta_q^{t+1} = \frac{\Delta t}{(\Delta X)^2} \theta_q^t + \left(1 - \frac{2\Delta t}{(\Delta X)^2}\right) \theta_q^t + \frac{\Delta t}{(\Delta X)^2} \theta_q^{t+1} \quad \ldots (5.142)
\]

\[
\theta_r^{t+1} = \frac{\Delta t}{(\Delta X)^2} \theta_r^t + \left(1 - \frac{2\Delta t}{(\Delta X)^2}\right) \theta_r^t + \frac{\Delta t}{(\Delta X)^2} \theta_r^{t+1} \quad \ldots (5.143)
\]

\[
\theta_s^{t+1} = \frac{2\Delta t}{(\Delta X)^2} \theta_s^t + \left(1 - \frac{2\Delta t}{(\Delta X)^2}\right) \theta_s^t \quad \ldots (5.144)
\]

Eqs. (5.141)-(5.144) can be written in the following matrix form:

\[
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}^{t+1} =
\begin{bmatrix}
1 - 2r & r & 0 & 0 \\
4 & 1 - 2r & r & 0 \\
0 & 1 - 2r & r & 0 \\
0 & 0 & 2r & 1 - 2r
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4
\end{bmatrix}^t
\quad \ldots (5.145)
\]

where \( r = \frac{\Delta t}{(\Delta X)^2} \).
The TDM on the right-hand side of Eq. (5.145) is known (and constant) once the size of the time step $\Delta \tau$ is chosen. The known value of $\theta$ at $\tau^p$ (that is $\theta^p$) are multiplied by this TDM to obtain the new values of $\theta$ at $\tau^{p+1}$. This matrix multiplication is quite easy to carry out on the computer since only the non-zero term will contribute to the calculation. Thus, $\theta^{p+1}$ values are obtained explicitly in terms of $\theta^p$ values and hence the name, explicit method. Note that the solution of simultaneous algebraic equation is not necessary in this scheme, which makes it a very attractive method. Once $\theta^{p+1}$ are obtained, they are stored in $\theta^p$, and the computation is repeated for the next time step. This procedure continues until the result at the desired time is obtained or till the steady state is reached. However, a major drawback of this method is that for $r > 0.5$, that is when $1 - 2r$ is negative, the solution become unstable. Therefore, a stability limit of $r \leq 0.5$, is imposed, which result in considerable restriction on the time step $\Delta \tau$ for a particular value of $\Delta X$.

The Crank-Nicholson Method

In the Euler method the value of the derivative at the beginning of the time interval was used to progress in time. A more accurate method would be to use the arithmetic mean value of the derivatives at the beginning and at the end of the time interval, i.e. use the time derivative at $p + \frac{1}{2}$, a time which is midway between $p$ and $p + 1$. Therefore,

$$\theta^{p+1} = \theta^p + \frac{1}{2} \left[ \frac{d\theta}{d\tau} \right]^p + \frac{d\theta}{d\tau} \left[ \frac{\theta^{p+1}}{2} \right] \Delta \tau \quad \ldots (5.146)$$

Substituting Eqs. (5.130)-(5.133) in Eq. (5.146), we obtain

$$\frac{\theta^{p+1}_1 - \theta^p}{\Delta \tau} = \frac{1}{2 (\Delta x)^2} \left[ (-2 \theta^p_1 + \theta^p_2) + (-2 \theta^{p+1}_1 + \theta^{p+1}_2) \right] \quad \ldots (5.147)$$

$$\frac{\theta^{p+1}_2 - \theta^p_2}{\Delta \tau} = \frac{1}{2 (\Delta x)^2} \left[ (\theta^p_1 - 2 \theta^p_2 + \theta^p_3) + (\theta^{p+1}_1 - 2 \theta^{p+1}_2 + \theta^{p+1}_3) \right] \quad \ldots (5.148)$$

$$\frac{\theta^{p+1}_3 - \theta^p_3}{\Delta \tau} = \frac{1}{2 (\Delta x)^2} \left[ (\theta^p_2 - 2 \theta^p_3 + \theta^p_4) + (\theta^{p+1}_2 - 2 \theta^{p+1}_3 + \theta^{p+1}_4) \right] \quad \ldots (5.149)$$

$$\frac{\theta^{p+1}_4 - \theta^p_4}{\Delta \tau} = \frac{1}{2 (\Delta x)^2} \left[ (2 \theta^p_3 - 2 \theta^p_4) + (2 \theta^{p+1}_3 - 2 \theta^{p+1}_4) \right] \quad \ldots (5.150)$$

Eqs. (5.147)-(5.150) are rearranged, which results in a set of four simultaneous algebraic equations in $\theta^{p+1}_1, \theta^{p+1}_2, \theta^{p+1}_3, \theta^{p+1}_4$, represented in the matrix form as

$$\begin{bmatrix} 1 + r & -r/2 & 0 & 0 \\ -r/2 & 1 + r & -r/2 & 0 \\ 0 & -r/2 & 1 + r & -r/2 \\ 0 & 0 & -r & 1 + r \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \theta^p_1 \\ \theta^p_2 \\ \theta^p_3 \\ \theta^p_4 \end{bmatrix} \quad \ldots (5.151)$$
Similar to the case in the Euler method, the right-hand side can be computed directly because all the components are known. This results in a column matrix as before. The difference arises in the fact that this does not give an explicit results for the unknowns on the left-hand side; rather an implicit TDM system of algebraic equations results. This system of algebraic equations must then be solved in each time step. The Crank-Nicholson method, although a stable method, gives erroneous results in the early time if the time step is too large. However, the error damps out as time progresses towards the steady state.

The Pure Implicit Method

In contrast with the Euler or the Crank-Nicholson method, in the pure implicit scheme, the time derivative at the new time is used to move ahead in time. Thus,

$$\theta^p + 1 = \theta^p + \frac{d\theta}{d\tau} \bigg|_{\tau = \Delta \tau} \quad \ldots \text{(5.152)}$$

From Eqs. (5.130)-(5.133) and Eq. (5.152), we obtain

$$\frac{\theta^p + 1 - \theta^p}{\Delta \tau} = \frac{1}{(\Delta X)^2} \left[(-2\theta^p + 1 + \theta^p + 1)\right] \quad \ldots \text{(5.153)}$$

$$\frac{\theta^p + 1 - \theta^p}{\Delta \tau} = \frac{1}{(\Delta X)^2} \left[(\theta^p + 1 + 2\theta^p + 1 + \theta^p + 1)\right] \quad \ldots \text{(5.154)}$$

$$\frac{\theta^p + 1 - \theta^p}{\Delta \tau} = \frac{1}{(\Delta X)^2} \left[(\theta^p + 1 - 2\theta^p + 1 + \theta^p + 1)\right] \quad \ldots \text{(5.155)}$$

$$\frac{\theta^p + 1 - \theta^p}{\Delta \tau} = \frac{1}{(\Delta X)^2} \left[(2\theta^p + 1 - 2\theta^p + 1)\right] \quad \ldots \text{(5.156)}$$

$$\begin{bmatrix} 1+2r & -r \\ -r & 1+2r & -r \\ -r & 1+2r & -r \\ -2r & 1+2r & -r \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \theta_1^{p+1} \\ \theta_2^{p+1} \\ \theta_3^{p+1} \\ \theta_4^{p+1} \end{bmatrix} \quad \ldots \text{(5.157)}$$

Eq. (5.157) is an implicit set of equations to solve for the new temperatures at each time step. The pure implicit scheme is an unconditionally stable scheme, that is, there is no restriction on the time step, which is in sharp contrast with the Euler and the Crank-Nicholson method. However, the Euler and the pure implicit methods have the same order of accuracy, while the Crank-Nicholson method is more accurate than either of the two for the same time step. The accuracy and stability of each of the three methods are detailed in the sections to follow.

Accuracy of the Euler, Crank-Nicholson, and Pure Implicit Methods

In the Euler method, at any grid point $i$, $\frac{d\theta}{d\tau}$ is evaluated at $\tau^p$, that is,

$$\frac{d\theta}{d\tau} \bigg|_{\tau = \tau^p} = \frac{\theta^p + 1 - \theta^p}{\Delta \tau} \quad \ldots \text{(5.158)}$$

Eq. (5.158) is forward difference in time. Therefore, the order of accuracy in time is $0(\Delta \tau)$. In the pure implicit method, $\frac{d\theta}{d\tau}$ is evaluated at $\tau^{p+1}$, that is,

$$\frac{d\theta}{d\tau} \bigg|_{\tau = \tau^{p+1}} = \frac{\theta^{p+1} - \theta^p}{\Delta \tau} \quad \ldots \text{(5.159)}$$

Eq. (5.159) actually arises from Eq. (5.160) shown below
Although the RHS of Eq. (5.159) looks the same as that of Eq. (5.158), the former is actually backward difference in time, which is obvious from Eq. (5.160). Therefore, the order of accuracy in time is $0 (\Delta \tau)$. In the Crank-Nicholson method, $\frac{d\theta}{d\tau}$ is evaluated at $p + \frac{1}{2}$, that is,

$$
\frac{d\theta}{d\tau} \bigg|_{p+1} = \frac{\theta^{p+1} - \theta^p}{\Delta \tau} \quad \ldots (5.161)
$$

Again, although the RHS of Equation 5.161 looks the same as that of Eqs. (5.158) and (5.159), the former is actually central difference in time, which is obvious from Eq. (5.160). Therefore, the order of accuracy in time is $0 (\Delta \tau)^2$. This explains why the Crank-Nicholson scheme is one order more accurate in time as compared to the Euler or pure implicit scheme.

In all the three methods, the space derivatives are discretised using the central-difference scheme. Therefore, the order of accuracy in space in the Euler, pure implicit, and Crank-Nicholson methods is $0 (\Delta X)^2$.

The Euler method, the Crank-Nicholson method, and the pure implicit method are also called FTCS (forward-time, central space), CTCS (central-time, central space), and BTCS (backward-time, central space), respectively.

To summarize, the order of accuracy of each method can be written as follows:

- **Euler or explicit**: $0 [(\Delta X)^2, (\Delta \tau)]$ FTCS
- **Crank-Nicholson**: $0 [(\Delta X)^2, (\Delta \tau)^2]$ CTCS
- **Pure Implicit**: $0 [(\Delta X)^2, (\Delta \tau)]$ BTCS

**Stability : Numerically Induced Oscillations**

From the preceding discussion, it is apparent that all the three schemes will give better results if time steps are made smaller. In practice, however, one would usually like to take as large a time step as one can to reduce the computational effort and time. In addition to decreasing the accuracy of the solution, large time steps can introduce some unwanted, numerically induced oscillations into the solution, making it physically unrealistic. Such solutions are not acceptable and the method that produces such a solution is called unstable method. This brings us to the formal definition of a stable numerical scheme, which is the one for which errors from any source (round-off, truncation, mistakes, etc.) are not permitted to grow in the sequence of numerical procedures as the calculation proceeds from one step to the next.

**The Case of One Grid Point**

Consider the case of only one grid point, that is, the grid point on the insulation boundary of a plate (Figure 5.22). Therefore, $\Delta X = 1$, $r = \Delta \tau$. Hence, we have only one equation to solve, that is,

$$
\frac{d\theta_1}{d\tau} = -20_1 \quad \ldots (5.163)
$$
Figure 5.22: The Case of Only One Grid Point

Subjects to the initial conditions

\[ \theta_i (0) = 1 \]  \hspace{1cm} \ldots (5.164)

The analytical solution of Eq. (5.163) is

\[ \theta_i = e^{-2r} \]  \hspace{1cm} \ldots (5.165)

The following three equations are obtained corresponding to the three numerical schemes:

**Euler**

\[ \theta_i^{n+1} = (1 - 2r) \theta_i^n \]  \hspace{1cm} \ldots (5.166)

**Crank-Nicholson**

\[ (1 + r) \theta_i^{n+1} = (1 - r) \theta_i^n \]  \hspace{1cm} \ldots (5.167)

**Pure Implicit**

\[ (1 + 2r) \theta_i^{n+1} = \theta_i^n \]  \hspace{1cm} \ldots (5.168)

Each of these may be put in the following general form:

\[ \theta_i^{n+1} = \lambda \theta_i^n \]  \hspace{1cm} \ldots (5.169)

where \( \lambda \) is defined by

**Euler**

\[ \lambda = 1 - 2r \]  \hspace{1cm} \ldots (5.170)

**Crank-Nicholson**

\[ \lambda = \frac{1 - r}{1 + r} \]  \hspace{1cm} \ldots (5.171)

**Pure Implicit**

\[ \lambda = \frac{1}{1 + 2r} \]  \hspace{1cm} \ldots (5.172)

The value of \( \lambda \) determines the character of the solution. This is self-explanatory from Figure 5.23, which shows \( \lambda \) as a function of \( r \) (\( r = \Delta \tau \) for this special case) for each of the three numerical methods we have considered.

A close inspection of Figure 5.23 reveals that as \( r \to 0 \), that is, if the time step is made smaller and smaller, all three schemes become identical. As the time step is increased, in each case, the solutions begin to deviate from one another. The Euler method can have steady decay, stable oscillations, or unstable oscillations. The Crank-Nicholson method can have either steady decay or stable oscillations. The pure implicit method has only a steadily decaying type of solution. From the graph, it is also seen that the stability limit for the Euler method is 0.5 while that for the Crank-Nicholson method is 1.0. The Euler method becomes totally unstable at \( r = 1.0 \). While the Euler method is called conditionally stable, the Crank-Nicholson method is called unconditionally stable because the oscillations
ultimately damp out with time. The pure implicit method is truly unconditionally stable method.

Figure 5.23: Stability Curves for the Case of One GrPd Point

Figure 5.24 compares the three numerical solutions to the corresponding exact solution (drawn qualitatively) \( \theta_1 = e^{-2\tau} \) for \( r = 1.2 \) which exceeds the stability limit for both the Euler method and the Crank-Nicholson method.

The figures reveals that oscillations in the Euler solution grow without bound, and oscillations are seen in the Crank-Nicholson solution but gradually damp out for large time. The pure implicit solution does not show any oscillations.

The Case of More than One Grid Point

The example of one grid point may be extended to the more general case in which there are more than one grid points, that is, more than one equation. The matrix representation for any of three numerical schemes can be written as

\[
A\theta^{p+1} = B\theta^p
\]

where \( A \) and \( B \) are matrices that depend on the particular method. Eq. (5.102) can be written as

\[
\theta^{p+1} = A^{-1} B\theta^p
\]  

Note that in the right-hand side of Eq. (5.174) \( A^{-1} B \) is a square matrix. We also know that associated with every square matrix (let us call this matrix \( S \)) are a special set of vectors, called eigenvectors, and a related set of scalars, called eigenvalues. Formally, the vector \( x \) is an eigenvector of \( S \) if and only if \( x \) is a non-zero vector and \( \lambda \) is a scalar (which may be zero), such that

\[
Sx = \lambda x
\]
Conduction

The scalar \( \lambda \) is an eigenvalue of \( S \) if and only if there exists a non-zero vector \( x \) such that Eq. (5.175) holds. The eigenvalues \( \lambda \) of the matrix \( A^{-1} B \) play a similar role to the \( \lambda \) in the case of one grid point. If there are \( N \) simultaneous equations being handled, there will be \( N \) eigenvalues of \( A^{-1} B \). These values will determine the character of the solution. Now,

\[
A^{-1} B \theta^P \lambda \theta^P = 0 \quad \ldots (5.176)
\]

or

\[
(A^{-1} B - \lambda I) \theta^P = 0 \quad \ldots (5.177)
\]

where \( I \) is an \( N \times N \) unite matrix. To get a non-trivial solution of Eq. (5.177),

\[
det (A^{-1} B - \lambda I) = 0 \quad \ldots (5.178)
\]

We may now multiply both side of Eq. (5.178) by \( \det (A) \) to get

\[
det (A) \det (A^{-1} B - \lambda I) = 0 \quad \ldots (5.179)
\]

or

\[
det (AA^{-1} B - A \lambda I) = 0 \quad \ldots (5.180)
\]

or

\[
det (B - \lambda A) = 0 \quad \ldots (5.181)
\]

There will be three general classes of solutions which will arise in this problem.

**Case I**

If all eigenvalues are between 0 and 1, there will be no oscillations. The solution will gradually approach a steady-state value.

**Case II**

If one of the eigenvalues falls between 0 and –1, numerically induced oscillation will appear.

**Case III**

If one of the eigenvalues is less than –1, the oscillations will be unstable.

**An example of the Two-Grid-Point Case for the Euler Method**

As an example, let us consider the two grid point case for Euler method. The matrices \( A \) and \( B \) are then given by

\[
A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

\[
B = \begin{bmatrix} 1 - 2r & r \\ 2r & 1 - 2r \end{bmatrix}
\]

Then,

\[
B - \lambda A = \begin{bmatrix} 1 - 2r \lambda & r \\ 2r & 1 - 2r - \lambda \end{bmatrix}
\]

The determinant of the above matrix is given by

\[
det (B - \lambda A) = (1 - 2r - \lambda)^2 - 2r^2 = 0 \quad \ldots (5.182)
\]

Solving Eq. (5.182), we get

\[
\lambda_1 = 1 - r (2 + \sqrt{2})
\]

\[
\lambda_2 = 1 - r (2 - \sqrt{2})
\]
The value of \( \lambda_1 \) will determine the character of the solution since it is this value that is most likely to be negative (because of the larger coefficient or \( r \)).

The \( \lambda \) versus \( r \) plots (Figure 5.25) show the same general trend of Figure 5.23, but the curves have shifted to the left so that the critical values of \( r \) are smaller than those for the one-grid-point case. The upper limit for stable oscillations of the Euler method is now \( \frac{2}{2 + \sqrt{2}} = 0.586 \) [since \( \lambda_{\text{crit}} = -1 - r (2 + \sqrt{2}) \)] as compared to 1.0 in the one grid point case.

![Figure 5.25: Stability Curves for the case of Two Grid Points](image)

**SAQ 6**

(a) Explain discretization and stability with respect to the numerical method of solving a two-dimensional heat conduction problem.

(b) Discuss Gauss-Siedel iteration technique to determine temperature at a nodal point in a two-dimensional solid.

(c) What is relaxation method? What do you mean by residuals?

(d) Briefly describe the Thomas algorithm.

**Exercise 1**

Consider the following one dimensional steady state conduction problem

\[
\begin{align*}
\frac{d^2 T(x)}{dx^2} + \frac{1}{k} g &= 0 & \text{in} & \quad 0 < x < L \\
\frac{dT(x)}{dx} &= 0 & \text{at} & \quad x = 0 \\
k \frac{dT(x)}{dx} + hT(x) &= 0 & \text{at} & \quad x = L
\end{align*}
\]

Write the finite difference formulation of this heat transfer problem by dividing the region \( 0 < x < L \) into four equal parts.
**Exercise 2**

In a parallel plate fuel element for a gas cooled reactor, the heat generation in the fuel element has approximately a cosine distribution. The simplest steady state model for the temperature distribution in the fuel element may be taken as

\[
\frac{d^2 T}{dx^2} + \frac{1}{k} g_0 \cos \frac{\pi x}{2L} = 0 \quad \text{in} \quad 0 < x < L
\]

\[
\frac{dT}{dx} = 0 \quad \text{at} \quad x = 0
\]

\[
T = 0 \quad \text{at} \quad x = L
\]

where \( L \) is the half thickness of the fuel element. By dividing the region into four equal parts, calculate the temperature distribution with finite difference for \( k = 12 \text{ W/m} \cdot \text{°C} \), \( L = 5 \times 10^{-3} \text{ m} \), and \( g = 6 \times 10^8 \text{ W/m}^2 \). Compare the numerical results with the exact solution.

**Exercise 3**

Consider a straight fin of rectangular cross section having thermal conductivity \( k = 40 \text{ W/m} \cdot \text{°C} \), length \( L = 30 \text{ cm} \), thickness \( t = 0.5 \text{ cm} \), and a large width perpendicular to the plane of Figure 5.26. The base is at \( T_0 = 130 \text{°C} \), and the fin tip is regarded insulated. The fin dissipates heat by convection with a heat transfer coefficient \( h = 400 \text{ W/m}^2 \cdot \text{°C} \) into a fluid at \( T_\infty = 30 \text{°C} \).

![Figure 5.26](image)

(a) By using a one-dimensional mesh of size \( \Delta x = 0.3 \text{ cm} \), calculate by finite difference the temperature distribution along the fin.

(b) Estimate the heat transfer rate through the fin per 10 – cm width perpendicular to the plane of the paper.

(c) Compare the numerical results with those obtained from the analytical solution of the one-dimensional equation.

**Exercise 4**

By writing an energy balance on a differential volume element, derive the finite difference form of the heat conduction equation

\[
\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{g}{k} = 0
\]

for the nodal point \( A \) in each of the accompanying Figure 5.27 for boundary conditions indicated.
Exercise 5

Write the finite difference formulation for two dimensional, steady state heat conduction with no heat generation for a square region of side $L$ by using mesh size $\Delta x = \Delta y = \frac{L}{3}$ for the boundary conditions shown in Figure 5.28. Also, express the resulting finite difference equations in matrix form for the nine unknown node temperatures $T_m, m = 1 - 9$. 

![Figure 5.28](image-url)
Conduction

**Exercise 6**

A large and very thick brick wall \((\alpha = 5 \times 10^{-7} \text{ m}^2/\text{s})\) which is initially at a uniform temperature \(T_i = 125^\circ\text{C}\) is suddenly exposed to cooling by maintaining its surface at \(x = 0\) at \(T_0 = 25^\circ\text{C}\). To calculate the temperature transients at depths small in comparison to the thickness, the wall can be regarded as a semi infinite medium confined to the region \(x \geq 0\). By using a suitable method and mesh size \(\Delta x = 0.3\) cm, calculate the temperature at \(x = 1.2\) cm from the surface \(t = 1\) and \(5\) min after the exposure.

**Exercise 7**

A marble slab \((k = 2 \text{ W/m . }^\circ\text{C}, \alpha = 1 \times 10^{-6} \text{ m}^2/\text{s})\) that is \(L = 4\) cm thick is initially at a uniform temperature \(T_i = 200^\circ\text{C}\). Suddenly one of its surfaces is lowered to \(0^\circ\text{C}\) and is maintained at that temperature, while the other surface is kept insulated. Develop an explicit finite difference scheme for the determination of the temperature distribution in the slab as a function of position and time as well the heat flux at the boundary surface.

**5.9 SUMMARY**

In the present unit, different solution methods of conduction heat transfer problems are discussed. Analytical method is restricted to simple geometry and boundary conditions. Graphical method can be applied to some complex problems but its workability is restricted. One of the best methods and most frequently used numerical scheme is the finite difference method for solution of conduction heat transfer problems. In this method the entire domain is divided into a grid and differential equations are rewritten as difference equation. Depending upon the nature of the problem, various iterative methods can be used. Accuracy and grid independence test are important for numerical problems which are discussed in details. Solution for 1-D and 2-D conduction problems are done by analytical as well as finite difference methods. You can also understand how finite difference method can be applied to solve transient conduction problems. Computer knowledge and computation skill are essential in solving the problems.

**5.10 KEY WORDS**

- **Error**: Difference between the exact (true) value and actual value.
- **Computational Domain**: Space considered for computation.
- **Grid Independent**: Final parameter under consideration is independent of grid size.

**5.11 ANSWERS TO SAQs**

Please refer the relevant preceding text in this unit for answers to SAQs.
REFERENCES

