2. A SECOND-ORDER THREE-POINT SCHEME

2.1 Deriving a Second-Order Three-Point Scheme

The equation under consideration in one dimension is the equation (1.1)

\[ y'' = (r(x)y)' + q(x)y + f(x), \]

\[ y(a) = \theta, \quad y(b) = \theta, \tag{1.1} \]

on the interval \( a \leq x \leq b \). This equation is rewritten as the first-order system of ODE's

\[ y' = r(x)y + z, \tag{2.1a} \]

\[ z' = q(x)y + f(x). \tag{2.1b} \]

The transformation to the system (2.1) has some advantages over the more common transformation

\[ \begin{pmatrix} y \\ y' \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ q(x) + r'(x) & r(x) \end{pmatrix} \begin{pmatrix} y \\ y' \end{pmatrix} + \begin{pmatrix} 0 \\ f(x) \end{pmatrix}. \tag{2.2} \]

For the system (2.1), \( r'(x) \) does not need to be evaluated, whereas to solve the system (2.2) it is necessary to evaluate \( r'(x) \). This function is not always explicitly available.
Also for some applications, \((r(x) y)'\) may be smooth while \(y'\) is not. Since (2.1) does not separate \(r(x) y\) and does not require evaluation of the potentially non-smooth function \(y'\), the system (2.1) can be used in such cases but the system (2.2) cannot. Here the transformation to (2.1) is done partly to facilitate the discussion of singularly perturbed problems in Chapter 4. The method described in this section will work equally well for any transformation of the equation (1.1) to a first-order system of equations.

Typically the system (2.1) is solved by discretizing both equations on a mesh \(\pi_n\). Difference schemes are applied separately to these two equations to find approximate values of \(y\) and \(z\) at the mesh points. (See Figure 1)

**Figure 1**

Standard Evaluation Points on a Mesh

\[
\begin{array}{ccc}
  z_{i-1} & z_i & z_{i+1} \\
  y_{i-1} & y_i & y_{i+1} \\
  \hline
  x_{i-1} & x_i & x_{i+1}
\end{array}
\]

As Keller shows in his paper [4], the system

\[B_1 y_\Pi = b_1\]  \hspace{1cm} (1.3)

results. This system is not tridiagonal in general.

Now let us introduce the concept of a staggered mesh. "Staggered mesh" means that equations (2.1a) and (2.1b) are not discretized at the same points on the mesh. The values of \(y\) are approximated at the meshpoints,

\[y_i \sim y(x_i) \hspace{1cm} i = 1, \ldots, n + 1,\]
but the values of $z$ are approximated at the points half-way between the mesh points,

$$z_{i+1/2} \sim z(x_{i+1/2}) \quad x_{i+1/2} = x_i + \frac{h_i}{2} \quad i = 1, \ldots, n$$

(See Figure 2 below).

**Figure 2**

Evaluation Points on a Staggered Mesh

On a staggered mesh, the discretization of the system of first-order ordinary differential equations (2.1) by one-step centered difference schemes can be represented by the banded matrix system

$$Bv_\pi = b$$  \hspace{1cm} (2.3)

where

$$v_\pi = \{y_1, z_{1+1/2}, y_2, z_{2+1/2}, \ldots, z_{n+1/2}, y_{n+1}\}^T$$

$$b = \{\theta, 0, l_2, 0, l_3, \ldots, l_n, 0, \vartheta\}^T$$

and the matrix $B$ has the form
\[
B = \begin{pmatrix}
1 \\
\frac{1}{s_1} & -1 & t_1 \\
\frac{1}{u_2} & v_2 & \frac{1}{w_2} \\
\frac{1}{u_3} & \frac{1}{w_3} & v_3 \\
& & \ddots \\
\frac{1}{u_n} & \frac{1}{w_n} & v_n \\
s_n & -1 & t_n \\
1
\end{pmatrix}
\]

The \( s_i \)'s and \( t_i \)'s are determined by the equation

\[
s_i \, y_i + t_i \, y_{i+1} - z_{i+1/2} = 0, \quad i = 1, \ldots, n,
\]

the \( u_i \)'s, \( v_i \)'s, and \( w_i \)'s by

\[
u_i \, y_{i-1} + \frac{1}{h_i} z_{i-1/2} + v_i \, y_i + \frac{1}{h_i} z_{i+1/2} + w_i \, y_{i+1} = l_i, \quad i = 2, \ldots, n
\]

and \( \tilde{h}_i = \frac{1}{2}(h_i + h_{i-1}) \). Because the discretization was performed on a staggered mesh, the system (2.3) can be reduced to a tridiagonal system

\[
A \, y = g. \quad (1.2)
\]

The \( i^{th} \) row of the tridiagonal system, \( i = 2, \ldots, n \), is created by adding from (2.3)

\[
\frac{1}{h_{2i}} \, \text{row}[2i] + \text{row}[2i-1] - \frac{1}{h_{2i}} \, \text{row}[2i-2]. \quad (2.4)
\]

This operation eliminates the auxiliary unknowns by elementary row operations (thus the operation is stable) and leaves an equation in only the principal unknowns \( y_{i-1}, y_i, y_{i+1} \). The boundary conditions, as represented in (2.3) form the 1\(^{st} \) and \( n + 1 \^{st} \) rows of the system (1.2). The tridiagonal system is a matrix representation of a three-point formula

\[
a_i \, y_{i-1} + b_i \, y_i + c_i \, y_{i+1} = g_i, \quad i = 2, \ldots, n.
\]
In order for the resulting three-point formula to have second-order accuracy on a staggered mesh with the development shown here, any difference scheme used must be one-step and centered, (and hence second-order accurate). The reasons for this will be discussed in the next section.

For example if the midpoint scheme is used, the discretization for equation (2.1a) on \([x_i, x_{i+1}]\), \(i = 1, \ldots, n\) is

\[
\frac{y_{i+1} - y_i}{\Delta x} = r(x_{i+1/2}) y(x_{i+1/2}) + z(x_{i+1/2})
\]

(2.5a)

where \(x_{i+1/2} = x_i + \frac{h_i}{2}\) as above. Since \(r(x)\) is a given function, \(r(x_{i+1/2})\) can be evaluated exactly. The value for \(y(x_{i+1/2})\) will be approximated by \(\frac{1}{2}(y_{i-1} + y_i)\), and the value for \(z(x_{i+1/2})\) will be approximated by \(z_{i+1/2}\). So (2.5a) becomes

\[
\frac{y_{i+1} - y_i}{\Delta x} = \frac{1}{2} r(x_{i+1/2}) (y_{i+1} + y_i) + z_{i+1/2}.
\]

(2.6a)

Equation (2.1b) can also be discretized using the midpoint scheme. But instead of being discretized on \([x_i, x_{i+1}]\), equation (2.1b) is discretized on the interval \([x_{i-1/2}, x_{i+1/2}]\), \(i = 2, \ldots, n\). This interval has length \(h_i\), and midpoint, called \(\tilde{x}_i\)

\[\tilde{x}_i = x_i + \frac{1}{4} (h_i - h_{i-1})\]

The midpoint scheme applied to (2.1b) therefore yields

\[
\frac{z_{i+1/2} - z_{i-1/2}}{\tilde{h}_i} = q(\tilde{x}_i) y(\tilde{x}_i) + f(\tilde{x}_i).
\]

(2.5b)

The known functions \(q(x)\) and \(f(x)\) are easily evaluated at \(\tilde{x}_i\). An approximation \(\bar{y}_i\) for \(y(\tilde{x}_i)\) is obtained by quadratic interpolation on \(y_{i-1}, y_i, y_{i+1}\). (For details on this calculation, see, for example [1, 2, 14].) This yields

\[
\bar{y}_i = \alpha_i y_{i-1} + \beta_i y_i + \gamma_i y_{i+1},
\]

where

\[
\alpha_i = \frac{(3h_i + h_{i-1})(h_{i+1} - h_i)}{16h_{i-1}(h_i + h_{i-1})},
\]

\[
\beta_i = \frac{(3h_i + h_{i-1})(3h_{i-1} + h_i)}{16h_{i-1}h_i},
\]

\[
\gamma_i = \frac{(3h_{i-1} + h_i)(h_i - h_{i-1})}{16h_i(h_i + h_{i-1})}.
\]
Note that quadratic interpolation has second-order accuracy. Equation (2.5b) then becomes,

\[
\frac{z_{i+1/2} - z_{i-1/2}}{h_i} = q(\bar{x}_i) (\alpha_i, y_{i-1} + \beta_i, y_i + \gamma_i, y_{i+1}) + f(\bar{x}_i). \tag{2.6b}
\]

We can use the difference formulas (2.6a) and (2.6b) directly to develop a three-point formula. Then the system (1.2) can be formed so that, ultimately, approximate values of \( y \) at the mesh points can be determined. We use equation (2.6a) as a definition of \( z_{i+1/2} \)

\[
z_{i+1/2} = \frac{y_{i+1} - y_i}{h_i} - \frac{1}{2} r(x_{i+1/2})(y_{i+1} + y_i). \tag{2.7}
\]

Substituting this definition in for \( z_{i+1/2} \), \( z_{i-1/2} \) in equation (2.6b) to eliminate all occurrences of \( z \) leaves an equation in only \( y \).

\[
\frac{h_{i-1}(y_{i+1} - y_i) - h_i(y_i - y_{i-1})}{1/2(h_i + h_{i-1})h_i h_{i-1}} = \frac{r(x_{i-1/2})(y_i + y_{i-1}) - r(x_{i+1/2})(y_{i+1} + y_i)}{h_i + h_{i-1}} q(\bar{x}_i) (\alpha_i, y_{i-1} + \beta_i, y_i + \gamma_i, y_{i+1}) + f(\bar{x}),
\]

\[i = 2, \ldots, n.\] \tag{2.8}

Equation (2.8) can be rearranged to give the three-point formula

\[
a_i y_{i-1} + b_i y_i + c_i y_{i+1} = g_i, \quad i = 2, \ldots, n,
\]

\[
a_i = \frac{2}{h_{i-1}(h_i - h_{i-1})} + \frac{r(x_{i-1/2})}{h_i + h_{i-1}} - q(\bar{x}_i) \alpha_i,
\]

\[
b_i = \frac{-2}{h_i h_{i-1}} + \frac{r(x_{i-1/2}) - r(x_{i+1/2})}{h_i + h_{i-1}} - q(\bar{x}_i) \gamma_i,
\]

\[
c_i = \frac{2}{h_i(h_i - h_{i-1})} + \frac{r(x_{i+1/2})}{h_i + h_{i-1}} - q(\bar{x}_i) \beta_i,
\]

\[
g_i = f(\bar{x}_i). \tag{2.9}
\]

For a uniform mesh, the interval \([x_{i-1/2}, x_{i+1/2}]\) has length \( h \) and midpoint \( x_i \). So then discretization of (2.1b) by the midpoint scheme gives

\[
\frac{z_{i+1/2} - z_{i-1/2}}{h} = q(x_i) y_i + f(x_i). \tag{2.10}
\]
Substituting (2.7) into (2.10) yields the "usual" three-point formula

\[ a_i \, y_{i-1} + b_i \, y_i + c_i \, y_{i+1} = g_i, \quad i = 2, \ldots, n, \]

where

\[ a_i = \frac{1}{h^2} + \frac{r(x_{i-1/2})}{2h}, \]
\[ b_i = -\frac{2}{h^2} - q(x_i) + \frac{r(x_{i-1/2}) - r(x_{i+1/2})}{2h}, \]
\[ c_i = \frac{1}{h^2} - \frac{r(x_{i+1/2})}{2h}, \]
\[ g_i = f(x_i). \tag{2.11} \]

Note that equation (2.11) is the same as equation (2.9) with \( \alpha_i = \gamma_i = 0 \), and \( \beta_i = 1 \).

The three-point formulas (2.9) and (2.11) were derived using the midpoint scheme. Other difference schemes can also be used. If the trapezoidal scheme were used on (2.1b), then instead of equation (2.5b), the discretization of (2.1b) would be

\[ \frac{z_{i+1/2} - z_{i-1/2}}{1/2(h_i + h_{i-1})} = \]
\[ \frac{1}{4} q(x_{i-1/2}) (y_i + y_{i-1}) + \frac{1}{2} f(x_{i-1/2}) + \]
\[ \frac{1}{4} q(x_{i+1/2}) (y_{i+1} + y_i) + \frac{1}{2} f(x_{i+1/2}). \tag{2.12} \]

Substituting (2.7) into (2.12) to eliminate \( z \) from the equation leads to the three-point formula where

\[ a_i = \frac{4}{h_{i-1}(h_i + h_{i-1})} + \frac{2r(x_{i-1/2})}{h_i + h_{i-1}} - \frac{1}{2} q(x_{i-1/2}), \]
\[ b_i = \frac{-4}{h_{i-1}(h_i + h_{i-1})} - \frac{4}{h_i(h_i + h_{i-1})} + \frac{2r(x_{i-1/2}) - 2r(x_{i+1/2})}{h_i + h_{i-1}} \]
\[ - \frac{1}{2} q(x_{i-1/2}) - \frac{1}{2} q(x_{i+1/2}), \]
\[ c_i = \frac{4}{h_i(h_i + h_{i-1})} - \frac{2r(x_{i+1/2})}{h_i + h_{i-1}} - \frac{1}{2} q(x_{i+1/2}), \]
\[ g_i = f(x_{i+1/2}) + f(x_{i-1/2}) \tag{2.13}, \]
\[ i = 2, \ldots, n. \]

Finite element and quadrature approaches can also be used. For example, an implementation of the Ritz-Galerkin method using roof functions as basis functions yields a scheme that is related to the trapezoidal scheme:

\[
z(x_{i+1/2}) - z(x_{i-1/2}) = \\
\int_{x_{i-1/2}}^{x_i} (q(x) y(x) + f(x)) \, dx + \int_{x_i}^{x_{i+1/2}} (q(x) y(x) + f(x)) \, dx. \tag{2.14}
\]

When the two integrals in (2.14) are replaced by simple midpoint quadrature formulas, the equation becomes

\[
z_{i+1/2} - z_{i-1/2} = \\
\frac{1}{4} h_{i-1} q(x_{i-1/2}) (y_i + y_{i-1}) + \frac{1}{2} h_{i-1} f(x_{i-1/2}) + \\
\frac{1}{4} h_i q(x_{i+1/2}) (y_i + y_{i+1}) + \frac{1}{2} h_i f(x_{i+1/2}). \tag{2.15}
\]

This yields the three-point formula where

\[
a_i = \frac{1}{h_{i-1}^2} + \frac{r(x_{i-1/2})}{2} - \frac{h_{i-1} q(x_{i-1/2})}{4},
\]

\[
b_i = \frac{-1}{h_{i-1}^2} - \frac{1}{h_i^2} + \frac{r(x_{i-1/2})}{2} - \frac{h_{i-1} q(x_{i-1/2})}{4} + \frac{h_i q(x_{i+1/2})}{4},
\]

\[
c_i = \frac{1}{h_i^2} + \frac{r(x_{i+1/2})}{2} - \frac{h_i q(x_{i-1/2})}{4},
\]

\[
g_i = \frac{1}{2} (f(x_{i+1/2}) + f(x_{i-1/2})),
\]

\[ i = 2, \ldots, n. \tag{2.16}
\]

Note that if the three-point formulae (2.13) or (2.16) are applied on a uniform mesh, they do not reduce to the usual three-point formula (2.11), but (2.13) and (2.16) are still expected to be second-order accurate.
2.2 Determining the Order of the Error

How is the order of the discretization error of these three-point formulae determined? First the local truncation error over the \(i^{th}\) sub-interval, \(\tau_{h_i}\), is found for equation (2.7). This is done using the Taylor expansions for \(y(x_{i+1})\) and \(y(x_i)\) about the point \(x_{i+1/2}\). (These expansions hold for all \(i = 1, \ldots, n\)).

\[
y(x_{i+1}) = y(x_{i+1/2}) + \frac{h_i}{2} y'(x_{i+1/2}) + \frac{h_i^2}{8} y''(x_{i+1/2}) + \frac{h_i^3}{48} y'''(x_{i+1/2}) + O(h_i^4),
\]

\[
y(x_i) = y(x_{i+1/2}) - \frac{h_i}{2} y'(x_{i+1/2}) + \frac{h_i^2}{8} y''(x_{i+1/2}) - \frac{h_i^3}{48} y'''(x_{i+1/2}) + O(h_i^4).
\]

Then it is easily determined that

\[
y(x_{i+1}) - y(x_i) = h_i y'(x_{i+1/2}) + \frac{h_i^3}{24} y'''(x_{i+1/2}) + O(h_i^4), \quad (2.17)
\]

\[
y(x_{i+1}) + y(x_i) = 2 y(x_{i+1/2}) - \frac{h_i^2}{4} y''(x_{i+1/2}) + O(h_i^4), \quad (2.18)
\]

\(i = 1, \ldots, n\).

Substituting the expansions (2.17) and (2.18) into equation (2.7) gives an expression for the local truncation error in equation (2.7).

\[
\tau_{h_i} = - z_{i+1/2} + y'(x_{i+1/2}) + \frac{h_i^3}{24} y'''(x_{i+1/2}) + O(h_i^4) - \frac{1}{2} r(x_{i+1/2}) \left( 2 y(x_{i+1/2}) - \frac{h_i^2}{4} y''(x_{i+1/2}) + O(h_i^4) \right), \quad (2.19)
\]

\(i = 1, \ldots, n\).

But \(y' = r y + z\) according to equation (2.1a), or

\[
y'(x_{i+1/2}) = r(x_{i+1/2}) y(x_{i+1/2}) + z_{i+1/2}. \quad (2.20)
\]

By replacing \(y'(x_{i+1/2})\) with its equivalent (2.20), the local truncation error over the \(i^{th}\) sub-interval for equation (2.7) can be simplified to

\[
\tau_{h_i} = - \frac{h_i^2}{8} y''(x_{i+1/2}) r(x_{i+1/2}) + \frac{h_i^2}{24} y'''(x_{i+1/2}) + O(h_i^4) \cdot y''(x_{i+1/2}) r(x_{i+1/2}) + O(h_i^4). \quad (2.21)
\]
If \( h \) is defined to be the maximum step size on the interval \([a,b]\),

\[
h = \max_{1 \leq i \leq n} (h_i),
\]

then, over the interval \([a,b]\), the maximum local truncation error of equation (2.7) obviously satisfies

\[
\tau_h = \max_{1 \leq i \leq n} |\tau_{h_i}| = O(h^2).
\]

The local truncation error for (2.6b) can be found using the Taylor expansions for \( z_{i+1/2}, z_{i-1/2} \) about the point \( \tilde{x}_i \). As before, let \( \tilde{h}_i = \frac{1}{2}(h_i + h_{i-1}) \), the length of the interval \([x_{i-1/2}, x_{i+1/2}]\), and let \( \bar{z}_i = z(\tilde{x}_i) \). (Note that \( \tilde{h} = \max_{2 \leq i \leq n} \tilde{h}_i = O(h) \).

\[
z(x_{i+1/2}) = \bar{z}_i + \frac{\tilde{h}_i}{2} \bar{z}'_i + \frac{1}{2} \left( \frac{\tilde{h}_i}{2} \right)^2 \bar{z}''_i + \frac{1}{6} \left( \frac{\tilde{h}_i}{2} \right)^3 \bar{z}'''_i + O(\tilde{h}_i^4), \quad (2.22)
\]

\[
z(x_{i-1/2}) = \bar{z}_i - \frac{\tilde{h}_i}{2} \bar{z}'_i + \frac{1}{2} \left( \frac{\tilde{h}_i}{2} \right)^2 \bar{z}''_i - \frac{1}{6} \left( \frac{\tilde{h}_i}{2} \right)^3 \bar{z}'''_i + O(\tilde{h}_i^4). \quad (2.23)
\]

Now substitute the exact expansions into (2.6b) and include the \( O(h^2) \) error for quadratic interpolation to get

\[
\tau_{h_i} = \bar{z}'_i + \frac{1}{24} \tilde{h}_i^2 \bar{z}'''_i + O(h_i^4) - \frac{1}{2} q(\tilde{x}_i) (\tilde{y}_i + O(h_i^2)) - f(\tilde{x}_i). \quad (2.24)
\]

But if this expression is simplified by using equation (2.1b) as a definition for \( \bar{z}_i \), the local truncation error for (2.6b) is found to be

\[
\tau_{h_i} = \frac{1}{24} \tilde{h}_i^2 \bar{z}'''_i + O(h_i^4) + q(\tilde{x}_i) O(h_i^2). \quad (2.25)
\]

Therefore, both equations (2.6a) and (2.6b) have second-order local truncation error.

The three-point formula (2.9) derived using these equations (2.6a) and (2.6b) will have a second-order discretization error on an arbitrary mesh even though the local truncation error of the formula is only first order. (The local truncation error is found by substituting Taylor expansions for \( y_{i+1}, y_i, y_{i-1} \), and \( \tilde{y}_i \) about the point \( x_{i+1/2} \) into equation (2.8).) The same will hold for any three-point formula developed as outlined
above using one-step, centered, second-order difference schemes on a staggered mesh: the discretization errors will be second-order. A more generalized look at the errors in the system will show why.

Recall that the discretization can be represented by the system

\[ B \mathbf{v}_x = b. \tag{2.3} \]

Since each of the difference equations represented in (2.3) has second-order local truncation errors, the system (2.3) as a whole will also have second-order local truncation errors. To show stability of (2.3) we first demonstrate that, after appropriate scaling, \( B \) resembles a multiple-shooting matrix

\[
M = \begin{pmatrix}
-Y(x_2, x_1) & I \\
& -Y(x_3, x_2) & I \\
& & & \ddots & \ddots \\
& & & -Y(x_n, x_{n-1}) & I \\
B_a & & & & B_b
\end{pmatrix}
\]  

where the \( Y(x_j, x_i) \) are fundamental solutions of the system (2.1) satisfying \( Y(x_i, x_i) = I \). Thus these are 2x2 matrices that solve the homogeneous form of (2.1)

\[
Y'(x, x_i) = \begin{pmatrix} r(x) & 1 \\ q(x) & 0 \end{pmatrix} Y(x, x_i).
\]

The matrices \( B_a \) and \( B_b \) represent the Dirichlet boundary conditions

\[
B_a = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad B_b = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.
\]

First note that, for the multiple shooting matrix \( M \), both dependent variables in the system must be evaluated at the mesh points. Presently in (2.3) \( y \) is evaluated at the mesh points but \( z \) is not; however the evaluation of \( z \) at the midpoints of the mesh in (2.3) can be viewed as an approximation to the values of \( z \) at the mesh points. By Taylor series we know that \( z(x_{i+1/2}) = z(x_i) + O(h) \). \( x_{i+1/2} \), an approximation to \( z(x_{i+1/2}) \), can therefore be considered a first-order approximation to \( z(x_i) \), and
consequently the system (2.3) can be seen to evaluate both \( y \) and \( z \) at the mesh points with first-order accuracy.

Next we see that the elemental structure of \( B \) is different from that of \( M \). \( M \) has \((2n+2)\) rows and columns whereas \( B \) has only \((2n+1)\). We can extend \( B \) to the "proper" size by adding the equation

\[
\frac{z_{n+3/2} - z_{n+1/2}}{\frac{1}{2}(h_{n+1} + h_n)} = q(x_{n+1}) y_{n+1} + f(x_{n+1})
\]

to the system. \( z_{n+3/2} \) will give a first-order approximation to \( z(b) \). If we also move the first row of the system (2.3) so that the boundary conditions are represented in the last two rows of the system, and then multiply on the left by

\[
D_1 = \begin{pmatrix}
1 & \frac{-w_2}{t_2} & \frac{-w_2}{t_3} & \cdots & \frac{-w_2}{t_n} & 1 \\
1 & \frac{-w_3}{t_3} & & & & \\
1 & & \frac{-w_3}{t_4} & & & \\
1 & & & \frac{-w_3}{t_5} & & \\
1 & & & & \frac{-w_3}{t_n} & \\
1 & & & & & \frac{-w_3}{t_{n+1}}
\end{pmatrix},
\]

then the correct zero structure of \( M \) is attained. The system that results from these transformations is

\[
\bar{B}\bar{v}_\pi = D_1\bar{b}
\]

(2.27)

where we denote

\[
\bar{v}_\pi = \{y_1, z_1, y_2, z_2, \ldots, y_n, z_n, y_{n+1}, z_{n+1}\}^T,
\]

with

\[
\bar{b} = \{0, l_2, 0, l_3, \ldots, l_n, 0, f(x_{n+1}), \theta, \vartheta\}^T,
\]
and

\[
\tilde{B} = \begin{pmatrix}
  s_1 & -1 & t_1 \\
  u_2 \frac{-1}{h_2} & v_2 - \frac{v_2}{t_2} & \frac{\frac{1}{h_2} + \frac{w_2}{t_2}}{h_2} \\
  \vdots & \vdots & \vdots \\
  s_n & -1 & t_n \\
  \frac{-1}{h_{n+1}} & q(x_{n+1}) & \frac{1}{h_{n+1}} \\
  1 & 0 & 0 \\
  0 & 1 & 0
\end{pmatrix}
\]

Note that multiplication by \(D_1\) is the same as solving for \(y_{i+1}\) in equation (2.6a) and then substituting into equation (2.6b). A look at the local truncation errors shows that \(\frac{w_i}{t_i} = O(h_i)\), so \(D_1 = I + O(h)\). Thus, for \(h\) small enough, the inverses of \(B\) and \(\tilde{B}\) can be bounded by similar constants.

From here we proceed as in Theorem 5.38 in [1]. To simplify the rest of the discussion we write the system (2.27) in block form using 2x2 matrices \(S_i\) and \(R_i\)

\[
\tilde{B} = \begin{pmatrix}
  S_1 & R_1 \\
  S_2 & R_2 \\
  \vdots & \vdots \\
  S_n & R_n \\
  B_a & B_b
\end{pmatrix}
\]

\(\tilde{B}\) can be equilibrated by multiplying by the matrix

\[
D_2 = diag \left( R_1^{-1}, R_2^{-1}, \ldots, R_n^{-1}, I \right)
\]

to form the matrix

\[
\hat{B} = D_2 \tilde{B} = \begin{pmatrix}
  R_1^{-1}S_1 & I \\
  R_2^{-1}S_2 & I \\
  \vdots & \vdots \\
  R_n^{-1}S_n & I \\
  B_a & B_b
\end{pmatrix}
\]

We now show that the block entries \(R_i^{-1}S_i\) approximate the fundamental solution matrices \(Y(x_{i+1}, x_i)\). As discussed earlier, when the scheme represented by (2.3)
is applied to the fundamental solutions, it can be considered a first-order difference scheme and so yields

\[ S_i Y(x_i, x_{i+1}) + R_i Y(x_{i+1}, x_i) = O(h_i) \]  

\[ i = 1, \ldots, n. \]

Since \( ||R_i^{-1}||_\infty = O(h_i) \), the multiplication of (2.27) by \( D_2 \), which corresponds to multiplying (2.28) by \( R_i^{-1} \), gives

\[ R_i^{-1} S_i + Y(x_{i+1}, x_i) = O(h_i^2). \]  

(2.29)

Seen in this light it becomes obvious that the matrix \( \hat{B} \) is row-wise a second-order approximation to the fundamental solution matrix, \( M \), and the errors in (2.29) can be represented in block diagonal form by

\[ E = \text{diag}(E_1, E_2, \ldots, E_n, 0) \]

where the last entry corresponds to the errors in the approximation of the Dirichlet boundary conditions. Note that

\[ ||E_i|| = O(h_i^2). \]

Thus we write \( \hat{B} = M + E \), or

\[ (M + E) \tilde{v}_n = D_2 D_1 \tilde{b}. \]  

(2.30)

Now to demonstrate stability of (2.3), we find a suitable bound on \( \hat{B}^{-1} \) (a bound on \( \tilde{B}^{-1} \)). First recall that the inverse of \( M \) is given by

\[ M^{-1} = \begin{pmatrix}
G(x_1, x_2) & \cdots & G(x_1, x_{n-1}) & Y(x_1, x_1)Q^{-1} \\
G(x_2, x_2) & \cdots & G(x_2, x_{n-1}) & Y(x_2, x_1)Q^{-1} \\
\vdots & \ddots & \vdots & \vdots \\
G(x_{n+1}, x_2) & \cdots & G(x_{n+1}, x_{n-1}) & Y(x_{n+1}, x_1)Q^{-1}
\end{pmatrix} \]

where \( Q \) is a nonsingular matrix representing the fundamental solutions at the boundary conditions

\[ Q = B_2 Y(x_1, x_1) + B_3 Y(x_{n+1}, x_1). \]
and \( G(x_i, x_j) \) is the Green's function of the problem (see for example [1]). Hence a bound on the norm of \( M^{-1} \) is given by \( \| M^{-1} \|_\infty \leq n \kappa \) where \( \kappa \), the conditioning constant of the problem, gives a measure of the sensitivity of the boundary-value problem to perturbations in the data. Then it is easy to see that
\[
\| M^{-1} D_2 \|_\infty \leq \kappa + O(h)
\]
and
\[
\| M^{-1} E \|_\infty \leq c\kappa h,
\]
for some constant \( c \). If \( c\kappa h \leq 1 \) then
\[
\| (I + M^{-1} E)^{-1} \|_\infty \leq 1 + O(h).
\]
Now
\[
\bar{B}^{-1} = (I + M^{-1} E)^{-1} M^{-1} D_2.
\]
So for any mesh with \( h \) sufficiently small,
\[
\| \bar{B}^{-1} \| = \kappa + O(h)
\]
and a similar bound holds for \( B^{-1} \).

Therefore, (2.3) represents a stable difference scheme with a stability constant \( K = \kappa + O(h) \). We already know that (2.3) has second-order local truncation errors when applied to the problem (1.1) on a staggered mesh. Thus, by stability and consistency arguments, the solutions obtained using the difference methods derived in Section 2.1 and represented by the system (2.3) are second-order approximations to the exact solutions. Since the system (1.2) is just another representation of the same method, it also yields solutions with second-order discretization errors.
2.3 Numerical Results

Numerical results were calculated for six test boundary-value problems with Dirichlet boundary conditions. The exact solution is known for each boundary-value problem.

1) \[ y'' = \frac{x}{4} y' - \frac{x e^x}{2} y - \frac{x}{2} e^{-3x} + 4e^{-2x} + \frac{1}{4} (1 + x) e^{-x}, \]

\[ y(0) = 2, \quad y(2) = e^{-4} + e^{-2}, \]

solution : \[ y = e^{-2x} + e^{-x}. \]

2) \[ x^2 y'' = -xy' + 4y, \]

\[ y(1) = 2, \quad y(2) = 4.25, \]

solution : \[ y = x^2 + \frac{1}{x^2}. \]

3) \[ x^2 y'' = xy' + 3y - 9x^2 - 8x - 3, \]

\[ y(1) = 7, \quad y(5) = 211, \]

solution : \[ y = x^3 + 3x^2 + 2x + 1. \]

4) \[ y'' = \lambda^2 y + (\lambda^2 + 4\pi^2) \cos^2(\pi x) - 2\pi^2, \]

\[ y(0) = 0, \quad y(1) = 0, \]

solution : \[ y = \frac{\exp(\lambda^2 x - \lambda^2) + \exp(-\lambda^2 x)}{1 + \exp(-\lambda^2)} - \cos^2(\pi x). \]

5) \[ y'' = -\pi \tan(\pi x) y', \]

\[ y(0) = 0, \quad y(0.25) = \sqrt{2}/2, \]

solution : \[ y = \sin(\pi x). \]

6) \[ y'' = -\frac{1}{x} y' + \left(\frac{8}{8 - x^2}\right)^2, \]

\[ y(1) = 0, \quad y(2) = 2 \ln(1.75), \]

solution : \[ y = 2 \ln\left(\frac{7}{8 - x^2}\right). \]

The solution to each problem was approximated 30 times on each of eight different types of meshes. Each time a different number of mesh points was used. A random
number generator was used to pick the number of mesh points between 9 and 300. The points on the meshes were calculated using formulas relating to transcendental functions. The first mesh type was created according to a sine curve. Points are taken at even sub-intervals along the sine curve over the interval $[0, \pi/2]$:

$$x_i = \sin \left( \frac{\pi i}{2(n+1)} \right), \quad i = 0, \ldots, n + 1$$

where $n-1$ is the number of mesh sub-intervals. (See Figure 3 below.)

**Figure 3**

Determining a Sine-Type Mesh for $n=5$

The mesh is then scaled and translated to fit the interval of the boundary-value problem.
under consideration. The resulting formula is

\[ x_i = a + \sin \left( \frac{\pi i}{2(n + 1)} \right) (b - a), \quad i = 0, \ldots, n + 1 \]

where \( a \) and \( b \) are the endpoints of the interval.

Meshes of other types were similarly created according to the \( \cos \) curve, the \( \ln \) curve, the \( \log_{10} \) curve, and linear combinations of the above. The meshes were created in this way to ensure non-uniformity in the meshes, and to allow calculated variations in the amount of non-uniformity. It also simplified the creation of a mesh with a random number of mesh points.

For each problem on each mesh, the approximate solution at each mesh point was compared to the exact solution at the mesh points. The pointwise error was recorded. From these local errors an average error and a maximum error were calculated. The average error is calculated to ensure that the maximum error is not unrepresentative of the local errors. The traditional "sub-divide the mesh" method for experimentally determining the order of accuracy of discretization schemes involves first the repeated sub-division of each mesh subinterval, starting from an arbitrary first mesh, and then a comparison of the resulting solutions. This method is inappropriate for our purposes, however, because the refined meshes are not arbitrary anymore. So, in the experiments reported in this thesis, the order of accuracy of the approximate solution is determined using the more statistical method described in Manteuffel and White [8]. A log-log plot is created of the maximum local error against the maximum step size for each problem on each mesh. Figure 4 shows the first-order errors obtained when the usual three-point formula (2.11) is applied to problems 1) to 6) on the nonuniform meshes. Figures 5 to 10 show the results for problems 1) to 6), respectively, when the three-point formula (2.9) derived from the midpoint scheme is used. The solid line appearing is the linear least-squares fit to the plotted points. The slope of this line gives the order of accuracy of the method used. Figure 11 shows the combined results for problems 1) to 6). The accuracy of the three-point formula (2.9) is determined to be second-order. Results are also given for the three-point trapezoidal formula (2.13). Figure 12 shows
that this method is also second-order accurate. These results are for problems 1) to 6) together.
Figure 4
Usual Three-Point Formula: Results for Problems 1 to 6

slope = 0.98
Figure 5
Three-point Midpoint Formula: Results for Problem 1

slope = 2.14
Figure 6
Three-point Midpoint Formula: Results for Problem 2

slope = 2.39
Figure 7

Three-point Midpoint Formula: Results for Problem 3

slope = 2.01
Three-point Midpoint Formula: Results for Problem 4

slope = 2.35
Figure 9

Three-point Midpoint Formula: Results for Problem 5

slope = 2.41
Figure 10
Three-point Midpoint Formula: Results for Problem 6

slope = 2.43
Figure 11
Three-point Midpoint Combined Results for Problems 1 to 6

slope = 2.34
Figure 12
Three-point Trapezoidal Combined Results for Problems 1 to 6

slope = 2.31