In general,

$$A_{ii} X_i^{(k)} = - \sum_{j=1}^{n} A_{ij} X_j^{(k)} - \sum_{j=1}^{n} A_{ij} X_j^{(k-1)} + b_i \quad (6.1)$$

or

$$X_i^{(k)} = \frac{1}{A_{ii}} \left[ - \sum_{j=1}^{n} A_{ij} X_j^{(k)} - \sum_{j=1}^{n} A_{ij} X_j^{(k-1)} + b_i \right] \quad (6.2) \quad i = 1, \ldots, n$$

Successive Overrelaxation (SOR)

Once $X_i^{(k)}$ has been obtained from Gauss-Seidel iteration, the convergence can be accelerated by computing a new $X_i^{(k)}$ by means

$$X_i^{(k)} = \omega X_i^{(w)} + (1-\omega) X_i^{(k-1)} \quad (6.3)$$

where $0 < \omega < 2$
Applying (6.3) to our example. It means combining (6.3) and (5.3).

\[
\begin{align*}
X_1^{(k)} &= \omega \left[ \frac{1}{3} X_2^{(k-1)} + \frac{1}{3} \right] + (1 - \omega) X_1^{(k-1)} \\
X_2^{(k)} &= \omega \left[ \frac{1}{3} X_1^{(k)} + \frac{1}{3} X_3^{(k-1)} + \frac{\theta}{3} \right] + (1 - \omega) X_2^{(k-1)} \\
X_3^{(k)} &= \omega \left[ \frac{1}{3} X_2^{(k)} - \frac{\theta}{3} \right] + (1 - \omega) X_3^{(k-1)}
\end{align*}
\]

In general,

\[
X_i^{(k)} = \frac{\omega}{a_{ii}} \left[ - \sum_{j=1}^{n} a_{ij} X_i^{(k)} - \sum_{j=1}^{n} a_{ij} X_j^{(k-1)} + b_i \right] + (1 - \omega) X_i^{(k-1)}
\]

Starting with \( X^{(0)} = (0, 0, 0)^T \) in our example and \( \omega = 1.1 \)

\[
X^{(1)} = (0.55, 3.135, -1.026)^T
\]

\[
X^{(2)} = (2, 3, -1) \quad \text{exact soln.}
\]
SOR method.

Gauss-Seidel:

\[ X_i^{(w)} = \frac{1}{a_{ii}} \left( -\sum_{j=1}^{i-1} a_{ij} X_j^{(w)} - \sum_{j=i+1}^{n} a_{ij} X_j^{(w-1)} + b_i \right) \]

SOR is obtained by introducing a relaxation factor \(\omega\). It helps to reduce the residual error of \(\vec{A} \hat{x} = \vec{b}\) when an approx. \(\hat{x}^*\) is obtained.

\[ \vec{r} = \vec{A} \hat{x}^* - \vec{b} \]

\[ X_i^{(w)} = \omega \left[ \frac{1}{a_{ii}} \left( -\sum_{j=1}^{i-1} a_{ij} X_j^{(w)} - \sum_{j=i+1}^{n} a_{ij} X_j^{(w-1)} + b_i \right) \right] + (1-\omega) X_i^{(w-1)} \]

\[ A_{ii} X_i^{(w)} = \omega \left[ -\sum_{j=1}^{i-1} a_{ij} X_j^{(w)} - \sum_{j=i+1}^{n} a_{ij} X_j^{(w-1)} + b_i \right] + (1-\omega) A_{ii} X_i^{(w-1)} \]

\[ A_{ii} X_i^{(w)} + \sum_{j=1}^{i-1} \omega a_{ij} X_j^{(w)} = -\sum_{j=i+1}^{n} \omega a_{ij} X_j^{(w-1)} + \omega b_i + (1-\omega) A_{ii} X_i^{(w-1)} \]

\[ (D - \omega L) \dot{X}^{(w)} = \omega U \dot{X}^{(w-1)} + (1-\omega) D \dot{X}^{(w-1)} + \omega \dot{b} \]

\[ \dot{X}^{(w)} = (D - \omega L)^{-1} \left[ \omega U + (1-\omega) D \right] \dot{X}^{(w-1)} + \omega (D - \omega L)^{-1} \dot{b}. \]
Matrix of SOR Method.

(9.1) Can be rewritten as

\[
A_{ii} X_i^{(k)} + \omega \sum_{j=1}^{i-1} A_{ij} x_j^{(k)} = (1-\omega) A_{ii} X_i^{(k-1)} - \omega \sum_{j=i+1}^{n} A_{ij} x_j^{(k-1)} + \omega b_i
\]

(10.1)

Using the definition given previously of the matrices \(D, L, \) and \(U,\) the equation (10.1) can be written as

\[
(D - \omega L) X^{(k)} = [(1-\omega) D + \omega U] X^{(k-1)} + \omega b.
\]

Therefore,

\[
\hat{X}^{(k)} = (D - \omega L)^{-1} [(1-\omega) D + \omega U] \hat{X}^{(k-1)} + \omega \hat{b}
\]

or

\[
\hat{X}^{(k)} = T_w \hat{X}^{(k-1)} + \hat{c}_w
\]

Where

\[
T_w = (D - \omega L)^{-1} [(1-\omega) D + \omega U] \quad \text{and}
\]

\[
\hat{c}_w = \omega (D - \omega L)^{-1} \hat{b}.
\]
Popular iterative algorithm used to approximate \( A \hat{x} = \hat{b} \), where \( A = D - L - U \).

Can be written as

\[
D \hat{x}^{(w)} = (L + U) \hat{x}^{(w-1)} + \hat{b}.
\]

**Jacobi Method:**

\[
Q = D
\]

\[
(D - L) \hat{x}^{(w)} = U \hat{x}^{(w-1)} + \hat{b}.
\]

**Gauss-Seidel Method:**

\[
(D - wL) \hat{x}^{(w)} = [wU + (1-w)D] \hat{x}^{(w-1)} + w \hat{b}
\]

or

\[
\frac{D - wL}{w} \hat{x}^{(w)} = \left[U + \frac{1-w}{w}D\right] \hat{x}^{(w-1)} + \hat{b}.
\]

They can also be written as fixed point problems:

\[
\hat{x} = T \hat{x} + \hat{c}
\]

**Jacobi:**

\[
\hat{x}^{(w)} = T_j \hat{x}^{(w-1)} + \hat{c}_j,
\]

\[
T_j \equiv D'(L + U), \quad \hat{c}_j = D^{-1} \hat{b}.
\]

**G-S:**

\[
\hat{x}^{(w)} = T_{gs} \hat{x}^{(w-1)} + \hat{c}_{gs},
\]

\[
T_{gs} \equiv (D - L)^{-1} U, \quad \hat{c}_{gs} = (D - L)^{-1} \hat{b}
\]

**SOR:**

\[
\hat{x}^{(w)} = T_w \hat{x}^{(w-1)} + \hat{c}_w,
\]

\[
T_w \equiv (D - wL)^{-1} [wU + (1-w)D],
\]

\[
\hat{c}_w \equiv \omega (D - wL)^{-1} \hat{b}.
\]
Before illustrating the advantages of the SOR method, we note that by using Eq. (7.13), Eq. (7.16) can be reformulated for calculation purposes to

\[ x_i^{(k)} = (1 - \omega)x_i^{(k-1)} + \omega \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \right). \]

To determine the matrix of the SOR method, we rewrite this as

\[ a_{ii}x_i^{(k)} + \omega \sum_{j=1}^{n} a_{ij}x_j^{(k)} = (1 - \omega)a_{ii}x_i^{(k-1)} - \omega \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} + \omega b_i, \]

so that in vector form, we have

\[ (D - \omega L)x^{(k)} = [(1 - \omega)D + \omega U]x^{(k-1)} + \omega b \]

or

\[ x^{(k)} = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]x^{(k-1)} + \omega(D - \omega L)^{-1}b. \] (7.17)

If we let \( T_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U] \) and \( c_\omega = \omega(D - \omega L)^{-1}b \), the SOR technique has the form

\[ x^{(k)} = T_\omega x^{(k-1)} + c_\omega. \] (7.18)

**EXAMPLE 3**

The linear system \( Ax = b \) given by

\[
\begin{align*}
4x_1 + 3x_2 &= 24, \\
3x_1 + 4x_2 - x_3 &= 30, \\
-x_2 + 4x_3 &= -24,
\end{align*}
\]

has the solution \((3, 4, -5)^t\). The Gauss-Seidel method and the SOR method with \( \omega = 1.25 \) will be used to solve this system, using \( x^{(0)} = (1, 1, 1)^t \) for both methods. For each \( k = 1, 2, \ldots, \), the equations for the Gauss-Seidel method are

\[
\begin{align*}
x_1^{(k)} &= -0.75x_2^{(k-1)} + 6, \\
x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5, \\
x_3^{(k)} &= 0.25x_2^{(k)} - 6,
\end{align*}
\]

and the equations for the SOR method with \( \omega = 1.25 \) are

\[
\begin{align*}
x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5, \\
x_2^{(k)} &= 0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375, \\
x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5.
\end{align*}
\]

The first seven iterates for each method are listed in Tables 7.3 and 7.4. For the iterates to be accurate to seven decimal places, the Gauss-Seidel method requires 34 iterations, as opposed to 14 iterations for the over-relaxation method with \( \omega = 1.25 \).
Table 7.3  Gauss-Seidel

<table>
<thead>
<tr>
<th>k</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_k</td>
<td>1</td>
<td>5.250000</td>
<td>3.1406250</td>
<td>3.0878906</td>
<td>3.0549316</td>
<td>3.0343323</td>
<td>3.0214577</td>
<td>3.0134110</td>
</tr>
<tr>
<td>y_k</td>
<td>1</td>
<td>3.812500</td>
<td>3.8828125</td>
<td>3.9267578</td>
<td>3.9542236</td>
<td>3.9713898</td>
<td>3.9821186</td>
<td>3.9888241</td>
</tr>
<tr>
<td>z_k</td>
<td>1</td>
<td>-5.046875</td>
<td>-5.0292969</td>
<td>-5.0183105</td>
<td>-5.0114441</td>
<td>-5.0071526</td>
<td>-5.0044703</td>
<td>-5.0027940</td>
</tr>
</tbody>
</table>

Table 7.4  SOR with $\omega = 1.25$

<table>
<thead>
<tr>
<th>k</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_k</td>
<td>1</td>
<td>6.312500</td>
<td>2.6223145</td>
<td>3.1333027</td>
<td>2.9570512</td>
<td>3.0037211</td>
<td>2.9963276</td>
<td>3.0000498</td>
</tr>
<tr>
<td>y_k</td>
<td>1</td>
<td>3.5195313</td>
<td>3.9585266</td>
<td>4.0102646</td>
<td>4.0074839</td>
<td>4.0029200</td>
<td>4.0009262</td>
<td>4.0002586</td>
</tr>
<tr>
<td>z_k</td>
<td>1</td>
<td>-6.6501465</td>
<td>-4.6004238</td>
<td>-5.0966863</td>
<td>-4.9734897</td>
<td>-5.0057135</td>
<td>-4.9982822</td>
<td>-5.0003486</td>
</tr>
</tbody>
</table>

The obvious question to ask is how the appropriate value of $\omega$ is chosen. Although no complete answer to this question is known for the general $n \times n$ linear system, the following results can be used in certain situations.

**Theorem 7.24  (Kahan)**

If $a_{ij} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$.

The proof of this theorem is considered in Exercise 13. The proof of the next two results can be found in [Or2, pp. 123–133]. These results will be used in Chapter 12.

**Theorem 7.25  (Ostrowski-Reich)**

If $A$ is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $x^{(0)}$.

**Theorem 7.26**

If $A$ is positive definite and tridiagonal, then $\rho(T_\omega) = [\rho(T_j)]^2 < 1$, and the optimal choice of $\omega$ for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}.$$

With this choice of $\omega$, we have $\rho(T_\omega) = \omega - 1$.

**EXAMPLE 4**

The matrix

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix},$$

given in Example 3, is positive definite and tridiagonal, so Theorem 7.26 applies. Since
Application of theorem 7.26 to our linear system.

\[
A = \begin{bmatrix}
2 & -1 & 0 \\
-1 & 3 & -1 \\
0 & 1 & 2
\end{bmatrix}
\]

A is P.D. since \( |A|=2 \), \( |2\times3|=5>0 \)
and \( |A|=12-4=8>0 \), and it's also tri-diagonal.

Then thin 7.26 can be applied to A.

\[
A = D - L - U = \begin{bmatrix}
2 & 0 & 0 \\
0 & 3 & 0 \\
0 & 0 & 2
\end{bmatrix} - \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} - \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
A\ddot{x} = \ddot{b} \Rightarrow D\ddot{x} = (L+U)\ddot{x} + \ddot{b} \Rightarrow \ddot{x} = D^{-1}(L+U)\ddot{x} + D^{-1}\ddot{b}
\]

or \( \ddot{x} = T_j\ddot{x} + C_j \), \( T_j = D^{-1}(L+U) \).

For our example,

\[
T_j = \begin{bmatrix}
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{3} & 0 \\
0 & 0 & \frac{1}{2}
\end{bmatrix}
\]

\[
D^{-1} = \begin{bmatrix}
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{3} & 0 \\
0 & 0 & \frac{1}{2}
\end{bmatrix}
\]

\[
D^{-1}(L+U) = \begin{bmatrix}
0 & \frac{1}{2} & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\Rightarrow \quad T_j - \lambda I = \begin{bmatrix}
-\lambda & \frac{1}{2} & 0 \\
\frac{1}{3} & -\lambda & \frac{1}{3} \\
0 & \frac{1}{2} & -\lambda
\end{bmatrix}
\]

\[
= -\lambda^3 + \frac{1}{6}\lambda = 0 \Rightarrow \lambda\left(\lambda^2 - \frac{1}{6}\right) = 0
\]

\[
\Rightarrow \lambda_1 = 0, \quad \lambda_3 = \pm \frac{\sqrt{3}}{\sqrt{6}} = \pm \frac{1}{\sqrt{2}}
\]

\[
\Rightarrow \rho(T_j) = \frac{\sqrt{3}}{3} \approx 0.58.
\]

\[
\Rightarrow \text{Optimal } \omega, \quad \omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho^2(T_j)}} \approx 1.1.
\]
We have

\[
T_j - \lambda I = \begin{bmatrix}
-\lambda & -0.75 & 0 \\
-0.75 & -\lambda & 0.25 \\
0 & 0.25 & -\lambda \\
\end{bmatrix},
\]

so

\[
\det(T_j - \lambda I) = -\lambda(\lambda^2 - 0.625).
\]

Thus,

\[
\rho(T_j) = \sqrt{0.625}
\]

and

\[
\omega = \frac{2}{1 + \sqrt{1 - (\rho(T_j))^2}} = \frac{2}{1 + \sqrt{1 - 0.625}} \approx 1.24.
\]

This explains the rapid convergence obtained in Example 1 when using \(\omega = 1.25\).

We close this section with Algorithm 7.3 for the SOR method.

### Algorithm 7.3

To solve \(Ax = b\) given the parameter \(\omega\) and an initial approximation \(x^{(0)}\):

**INPUT**  
the number of equations and unknowns \(n\); the entries \(a_{ij}, 1 \leq i, j \leq n\), of the matrix \(A\); the entries \(b_i, 1 \leq i \leq n\), of \(b\); the entries \(XO_i, 1 \leq i \leq n\), of \(XO = x^{(0)}\); the parameter \(\omega\); tolerance \(TOL\); maximum number of iterations \(N\).

**OUTPUT**  
the approximate solution \(x_1, \ldots, x_n\) or a message that the number of iterations was exceeded.

**Step 1**  
Set \(k = 1\).

**Step 2**  
While \((k \leq N)\) do Steps 3–6.

**Step 3**  
For \(i = 1, \ldots, n\)

set \(x_i = (1 - \omega)XO_i + \omega\left(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i\right) / a_{ii}.

**Step 4**  
If \(||x - XO|| < TOL\) then OUTPUT \((x_1, \ldots, x_n)\);  
(The procedure was successful.)

STOP.
Here problem: Sect 7.2

(5b) \[
\begin{align*}
10x_1 - x_2 &= 9 \\
-x_1 + 10x_2 - 2x_3 &= 7 \\
-2x_2 + 10x_3 &= 6 \\
\end{align*}
\]

\[
A = \begin{bmatrix} 10 & -1 & 0 \\
-1 & 10 & -2 \\
0 & -2 & 10 \end{bmatrix}
\]

A is positive definite. Prove it!
And also tridiagonal.

Using Theorem 7.26 there is an optimal \( \omega \) for SOR method given by

\[
\omega = \frac{2}{1 + \sqrt{1 - (\rho(T_j))^2}}
\]

where \( T_j \) is Jacobi's method matrix.

\[
T_j = \begin{bmatrix} 0 & \frac{4}{10} & 0 \\
\frac{4}{10} & 0 & \frac{2}{10} \\
0 & \frac{2}{10} & 0 \end{bmatrix}
\]

\[
|T_j - \lambda I| = 0 \quad \Leftrightarrow \quad \lambda = \frac{3}{10}, \frac{1}{100}, \frac{1}{25}
\]

\[\rho(T_j) \approx 0.2237\]

and \( \omega = \frac{2}{1 + \sqrt{1 - (0.2237)^2}} \approx 1.0128 \)
\[ A = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\theta_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \theta_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \theta_x & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \theta_x & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \theta_x & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \theta_x & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \theta_x & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_x & 0 \\
\end{pmatrix} \]

or

\[ A = \begin{pmatrix}
\theta_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \theta_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \theta_y & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \theta_y & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \theta_y & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \theta_y & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \theta_y & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_y & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_y \\
\end{pmatrix} \]

where

\[ C_k = \begin{pmatrix}
-1 & +\theta_x & 0 \\
\theta_x & -1 & \theta_x \\
0 & \theta_x & -1 \\
\end{pmatrix}_{3\times3} \]

\[ D_k = \begin{pmatrix}
\theta_y & 0 & 0 \\
0 & \theta_y & 0 \\
0 & 0 & \theta_y \\
\end{pmatrix}_{3\times3} \]

\[ \theta_x = \frac{\Delta y^2}{2(\omega x^2+\omega y^2)} \]

\[ \theta_y = \frac{\Delta x^2}{2(\omega x^2+\omega y^2)} \]

The 9 equations can be represented by

\[ A \ddot{u} = b. \]

\[ 0 = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{pmatrix}_{3\times3} \]
Matrix (4.1) is not strictly diagonally dominant.

In row 5 for example:

$$|a_{55}| = 1 = \sum_{j=1}^{5} |a_{5j}| = 2\theta x + 2\theta y = 1$$

It means that 9.2.3 does not apply.

However, that 9.2.2 does apply. In fact, Jacobi's method for Laplace's eq. is given by:

$$\frac{v_j^{(k+1)}}{2} = \Theta_x (v_j^{(w)} + v_{j+1}^{(w)}) + \Theta_y (v_{j+1}^{(w)} + v_{j+1}^{(w)})$$

$$j = 1, 2, \ldots, J - 1, \quad k = 1, 2, \ldots, K - 1$$

It can be proved that the eigenvalues of \( M_j \) are given by:

$$\mu_{mn} = 1 - 4\Theta_x \sin^2 \left( \frac{m\pi}{2J} \right) - 4\Theta_y \sin^2 \left( \frac{n\pi}{2K} \right)$$

$$m = 1, 2, \ldots, J - 1, \quad n = 1, 2, \ldots, K - 1$$

The largest eigenvalue can be obtained when \( m = n = 1 \).

\[ \rho(T_j) = 1 - 4\Theta_x \sin^2 \left( \frac{\pi}{2J} \right) - 4\Theta_y \sin^2 \left( \frac{\pi}{2K} \right) \]
If \( \theta_x = \theta_y = \frac{1}{4} \) (\( \Delta x = \Delta y \)), and \( J = K \)

then \[ p(15) = 1 - 2 \sin^2 \frac{\pi}{2J} \]

Obviously for \( J \) moderately large

\[ 0 < p(15) < 1 \]

and then 9.2.2 applies

Jacobi Iteration for the linear system defined by \( A \) (4.1)

(Approximation of Poisson’s eq.) converges for any initial guess \( \tilde{u}(0) \)
Example 9.2.8 (book).

Solution of Laplace's equation for iteration.

For $J = K = 3$, $\theta_x = \theta_y = \frac{1}{4}$

Gauss-Seidel:

\[
\hat{u}_{j\ell}^{(k+1)} = \frac{1}{4} \left[ \hat{u}_{j+1\ell}^{(k)} + \hat{u}_{j-1\ell}^{(k)} + \hat{u}_{j\ell+1}^{(k)} + \hat{u}_{j\ell-1}^{(k)} \right]
\]  

(13.1)

Relaxation:

\[
\hat{u}_{j\ell}^{(k+1)} = \omega \hat{u}_{j\ell}^{(k+1)} + (1-\omega) \hat{u}_{j\ell}^{(k)}
\]  

(13.2)

Combining (13.1) and (13.2)

\[
\hat{u}_{j\ell}^{(k+1)} = \frac{\omega}{4} \left[ \hat{u}_{j+1\ell}^{(k)} + \hat{u}_{j-1\ell}^{(k)} + \hat{u}_{j\ell+1}^{(k)} + \hat{u}_{j\ell-1}^{(k)} \right] + (1-\omega) \hat{u}_{j\ell}^{(k)}
\]

The spectral radius of $M_f$ is given by

\[
\rho(M_f) = 1 - 2 \sin^2 \left( \frac{\pi}{2J} \right)
\]

So in this case, $\rho(M_f) = 1 - 2 \sin^2 \left( \frac{\pi}{6} \right) = \frac{1}{2}$

Therefore,

\[
\omega_{opt} = \frac{2}{1 + \sqrt{1 - \frac{1}{4}}} \approx 1.07
\]