7.3 The Jacobi and Gauss-Seidel Iterative Methods

The Jacobi Method

Two assumptions made on Jacobi Method:

1. The system given by
   \[\begin{align*}
   a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
   a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
   &\vdots \\
   a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
   \end{align*}\]
   Has a unique solution.

2. The coefficient matrix \( A \) has no zeros on its main diagonal, namely, \( a_{11}, a_{22}, \ldots, a_{nn} \) are nonzeros.

Main idea of Jacobi
To begin, solve the 1st equation for \( x_1 \), the 2nd equation for \( x_2 \) and so on to obtain the rewritten equations:

\[\begin{align*}
   x_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n) \\
   x_2 &= \frac{1}{a_{22}} (b_2 - a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n) \\
   &\vdots \\
   x_n &= \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})
\end{align*}\]

Then make an initial guess of the solution \( \mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \ldots, x_n^{(0)}) \). Substitute these values into the right hand side of the rewritten equations to obtain the first approximation, \( \left( x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots, x_n^{(1)} \right) \).

This accomplishes one iteration.

In the same way, the second approximation \( \left( x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \ldots, x_n^{(2)} \right) \) is computed by substituting the first approximation’s \( x \)-values into the right hand side of the rewritten equations.

By repeated iterations, we form a sequence of approximations \( \mathbf{x}^{(k)} = \left( x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots, x_n^{(k)} \right)^t, \quad k = 1,2,3, \ldots \)
**The Jacobi Method.** For each \( k \geq 1 \), generate the components \( x_i^{(k)} \) of \( \mathbf{x}^{(k)} \) from \( \mathbf{x}^{(k-1)} \) by

\[
x_i^{(k)} = \frac{1}{a_{ii}} \left[ \sum_{j=1, j \neq i}^{n} (-a_{ij}x_j^{(k-1)}) + b_i \right], \quad \text{for } i = 1, 2, \ldots, n
\]

**Example.** Apply the Jacobi method to solve

\[
\begin{align*}
5x_1 - 2x_2 + 3x_3 &= -1 \\
-3x_1 + 9x_2 + x_3 &= 2 \\
2x_1 - x_2 - 7x_3 &= 3
\end{align*}
\]

Continue iterations until two successive approximations are identical when rounded to three significant digits.

**Solution**

To begin, rewrite the system

\[
\begin{align*}
x_1 &= \frac{-1}{5} + \frac{2}{5}x_2 - \frac{3}{5}x_3 \\
x_2 &= \frac{2}{9} + \frac{3}{9}x_1 - \frac{1}{9}x_3 \\
x_3 &= \frac{3}{7} + \frac{2}{7}x_1 - \frac{1}{7}x_2
\end{align*}
\]

Choose the initial guess \( x_1 = 0, x_2 = 0, x_3 = 0 \)

The first approximation is

\[
\begin{align*}
x_1^{(1)} &= \frac{-1}{5} + \frac{2}{5}(0) - \frac{3}{5}(0) = -0.200 \\
x_2^{(1)} &= \frac{2}{9} + \frac{3}{9}(0) - \frac{1}{9}(0) = 0.222 \\
x_3^{(1)} &= \frac{3}{7} + \frac{2}{7}(0) - \frac{1}{7}(0) = -0.429
\end{align*}
\]
Continue iteration, we obtain

<table>
<thead>
<tr>
<th>n</th>
<th>$k = 0$</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>$k = 4$</th>
<th>$k = 5$</th>
<th>$k = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1^{(k)}$</td>
<td>0.000</td>
<td>-0.200</td>
<td>0.146</td>
<td>0.192</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2^{(k)}$</td>
<td>0.000</td>
<td>0.222</td>
<td>0.203</td>
<td>0.328</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2^{(k)}$</td>
<td>0.000</td>
<td>-0.429</td>
<td>-0.517</td>
<td>-0.416</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When to stop: 1. $\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\|} < \varepsilon$; or 2. $\|x^{(k)} - x^{(k-1)}\| < \varepsilon$. Here $\varepsilon$ is a given small number.

**The Jacobi Method in Matrix Form**

Consider to solve an $n \times n$ size system of linear equations $Ax = b$ with $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$ and $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$ for $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$.

We split $A$ into

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -a_{n-1,n} \end{bmatrix} = D - L - U$$

$Ax = b$ is transformed into $(D - L - U)x = b$

$$Dx = (L + U)x + b$$

Assume $D^{-1}$ exists and $D^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{nn}} \end{bmatrix}$
Then
\[ x = D^{-1}(L + U)x + D^{-1}b \]

The matrix form of Jacobi iterative method is
\[ x^{(k)} = D^{-1}(L + U)x^{(k-1)} + D^{-1}b \quad k = 1, 2, 3, ... \]

Define \( T_j = D^{-1}(L + U) \) and \( c = D^{-1}b \), Jacobi iteration method can also be written as
\[ x^{(k)} = T_jx^{(k-1)} + c \quad k = 1, 2, 3, ... \]

**Numerical Algorithm of Jacobi Method**

Input: \( A = [a_{ij}] \), \( b, XO = x^{(0)} \), tolerance \( TOL \), maximum number of iterations \( N \).

Step 1 Set \( k = 1 \)

Step 2 while \( (k \leq N) \) do Steps 3-6

Step 3 For for \( i = 1, 2, ..., n \)
\[ x_i = \frac{1}{a_{ii}} \left( \Sigma_{j=1}^{n} \left( -a_{ij}XO_j \right) + b_i \right) \]

Step 4 If \( ||x - XO|| < TOL \), then OUTPUT \( (x_1, x_2, x_3, ..., x_n) \); STOP.

Step 5 Set \( k = k + 1 \).

Step 6 For for \( i = 1, 2, ..., n \)
Set \( XO_i = x_i \).

Step 7 OUTPUT \( (x_1, x_2, x_3, ..., x_n) \); STOP.

Another stopping criterion in Step 4: \[ \frac{||x^{(k)} - x^{(k-1)}||}{||x^{(k)}||} \]
The Gauss-Seidel Method

**Main idea of Gauss-Seidel**

With the Jacobi method, only the values of $x_i^{(k)}$ obtained in the $k$th iteration are used to compute $x_i^{(k+1)}$. With the Gauss-Seidel method, we use the new values $x_i^{(k+1)}$ as soon as they are known. For example, once we have computed $x_1^{(k+1)}$ from the first equation, its value is then used in the second equation to obtain the new $x_2^{(k+1)}$, and so on.

**Example.** Derive iteration equations for the Jacobi method and Gauss-Seidel method to solve

\[
\begin{align*}
5x_1 - 2x_2 + 3x_3 &= -1 \\
-3x_1 + 9x_2 + x_3 &= 2 \\
2x_1 - x_2 - 7x_3 &= 3
\end{align*}
\]

Choose the initial guess $x_1 = 0, x_2 = 0, x_3 = 0$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k = 0$</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>$k = 4$</th>
<th>$k = 5$</th>
<th>$k = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1^{(k)}$</td>
<td>0.000</td>
<td>-0.200</td>
<td>0.167</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2^{(k)}$</td>
<td>0.000</td>
<td>0.156</td>
<td>0.334</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2^{(k)}$</td>
<td>0.000</td>
<td>-0.508</td>
<td>-0.429</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**The Gauss-Seidel Method.** For each $k \geq 1$, generate the components $x_i^{(k)}$ of $x^{(k)}$ from $x^{(k-1)}$ by

\[
x_i^{(k)} = \frac{1}{a_{ii}} \left[ - \sum_{j=1}^{i-1} (a_{ij} x_j^{(k)}) - \sum_{j=i+1}^{n} (a_{ij} x_j^{(k-1)}) + b_i \right], \quad \text{for } i = 1, 2, \ldots, n
\]

Namely,

\[
\begin{align*}
a_{11} x_1^{(k)} &= -a_{12} x_2^{(k-1)} - \cdots - a_{1n} x_n^{(k-1)} + b_1 \\
a_{22} x_2^{(k)} &= -a_{21} x_1^{(k)} - a_{23} x_3^{(k-1)} - \cdots - a_{2n} x_n^{(k-1)} + b_2 \\
&\vdots \\
a_{nn} x_n^{(k)} &= -a_{n1} x_1^{(k)} - a_{n2} x_2^{(k)} - \cdots + b_n
\end{align*}
\]
Matrix form of Gauss-Seidel method.

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

Define \(T_g = (D - L)^{-1}U\) and \(c_g = (D - L)^{-1}b\), Gauss-Seidel method can be written as

\[x^{(k)} = T_gx^{(k-1)} + c_g \quad k = 1,2,3,\ldots\]

**Numerical Algorithm of Gauss-Seidel Method**

Input: \(A = [a_{ij}], b, x^O = x^{(0)}\), tolerance \(TOL\), maximum number of iterations \(N\).

Step 1 Set \(k = 1\)

Step 2 while \((k \leq N)\) do Steps 3-6

Step 3 For for \(i = 1,2,\ldots,n\)

\[x_i = \frac{1}{a_{ii}} \left[ - \sum_{j=1}^{i-1}(a_{ij}x_j) - \sum_{j=i+1}^{n}(a_{ij}x^O_j) + b_i \right],\]

Step 4 If \(||x - x^O|| < TOL\), then OUTPUT \((x_1, x_2, x_3, \ldots x_n)\);

STOP.

Step 5 Set \(k = k + 1\).

Step 6 For for \(i = 1,2,\ldots,n\)

Set \(x^O_i = x_i\).

Step 7 OUTPUT \((x_1, x_2, x_3, \ldots x_n)\);

STOP.
Convergence theorems of the iteration methods

Let the iteration method be written as

\[ x^{(k)} = T x^{(k-1)} + c \quad \text{for each } k = 1, 2, 3, \ldots \]

**Lemma 7.18** If the spectral radius satisfies \( \rho(T) < 1 \), then \((I - T)^{-1}\) exists, and

\[
(I - T)^{-1} = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j
\]

**Theorem 7.19** For any \( x^{(0)} \in \mathbb{R}^n \), the sequence \( \{x^{(k)}\}_{k=0}^{\infty} \) defined by

\[ x^{(k)} = T x^{(k-1)} + c \quad \text{for each } k \geq 1 \]

converges to the unique solution of \( x = T x + c \) if and only if \( \rho(T) < 1 \).

**Proof** (only show \( \rho(T) < 1 \) is sufficient condition)

\[ x^{(k)} = T x^{(k-1)} + c = T(T x^{(k-2)} + c) + c = \cdots = T^k x^{(0)} + (T^{k-1} + \cdots + T + I) c \]

Since \( \rho(T) < 1 \), \( \lim_{k \to \infty} T^k x^{(0)} = 0 \)

\[
\lim_{k \to \infty} x^{(k)} = 0 + \lim_{k \to \infty} \left( \sum_{j=0}^{k-1} T^j \right) c = (I - T)^{-1} c
\]

**Corollary 7.20** If \( ||T|| < 1 \) for any natural matrix norm and \( c \) is a given vector, then the sequence \( \{x^{(k)}\}_{k=0}^{\infty} \) defined by

\[ x^{(k)} = T x^{(k-1)} + c \]

converges, for any \( x^{(0)} \in \mathbb{R}^n \), to a vector \( x \in \mathbb{R}^n \), with \( x = T x + c \), and the following error bound hold:

(i) \[ ||x - x^{(k)}|| \leq ||T||^k ||x^{(0)} - x|| \]

(ii) \[ ||x - x^{(k)}|| \leq \frac{||T||^k}{1 - ||T||} ||x^{(1)} - x^{(0)}|| \]
**Theorem 7.21** If $A$ is strictly diagonally dominant, then for any choice of $x^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\{x^{(k)}\}_{k=0}^{\infty}$ that converges to the unique solution of $Ax = b$.

**Rate of Convergence**

**Corollary 7.20 (i)** implies $||x - x^{(k)}|| \approx \rho(T)^k ||x^{(0)} - x||$

**Theorem 7.22 (Stein-Rosenberg)** If $a_{ij} \leq 0$, for each $i \neq j$ and $a_{ii} \geq 0$, for each $i = 1,2,\ldots,n$, then one and only one of following statements holds:

(i) $0 \leq \rho(T_g) < \rho(T_j) < 1$;
(ii) $1 < \rho(T_j) < \rho(T_g)$;
(iii) $\rho(T_j) = \rho(T_g) = 0$;
(iv) $\rho(T_j) = \rho(T_g) = 1$. 
**Simple iteration**

Assume there is an initial guess \( x^{(0)} \) for the solution to \( Ax = b \). If one could compute the error \( e^{(0)} = A^{-1}b - x^{(0)} \), then one could find the solution \( x = x^{(0)} + e^{(0)} \). However, \( A^{-1}b \) is expensive to compute.

Instead, let’s compute the residual \( r^{(0)} = b - Ax^{(0)} \). Note that \( r^{(0)} = Ae^{(0)} \).

If \( M \) is a matrix with the property that \( M^{-1}A \) approximates the identity, yet \( r^{(0)} = Mz^{(0)} \) is easy to solve for \( z^{(0)} \), one could approximate the error by \( M^{-1}r^{(0)} \).

Thus, one solve \( Mz^{(0)} = r^{(0)} \) for \( z^{(0)} \), which approximates \( e^{(0)} \), and then replace \( x^{(0)} \) by \( x^{(1)} \equiv x^{(0)} + z^{(0)} \). Repeat this process with \( x^{(1)} \).

**Simple iteration algorithm**

Given an initial guess \( x^{(0)} \), compute \( r^{(0)} = b - Ax^{(0)} \), and solve \( r^{(0)} = Mz^{(0)} \) for \( z^{(0)} \).

for \( k = 1, 2, ... \)

- Set \( x^{(k)} \equiv x^{(k-1)} + z^{(k-1)} \).
- Compute \( r^{(k)} = b - Ax^{(k)} \).
- Solve \( Mz^{(k)} = r^{(k)} \) for \( z^{(k)} \).

**Remark:**

1. For \( M \equiv D \), the method is Jacobi iteration

\[
x^{(k)} = D^{-1}(L + U)x^{(k-1)} + D^{-1}b = D^{-1}(D - D + L + U)x^{(k-1)} + D^{-1}b = D^{-1}(D - A)x^{(k-1)} + D^{-1}b = x^{(k-1)} + D^{-1}(b - Ax^{(k-1)}) \equiv x^{(k-1)} + z^{(k-1)},
\]

2. For \( M \equiv D - L \), the method is GS.

\[
x^{(k)} = (D - L)^{-1}Ux^{(k-1)} + (D - L)^{-1}b = (D - L)^{-1}(U - D + L + D - L)x^{(k-1)} + (D - L)^{-1}b = (D - L)^{-1}(D - L - A)x^{(k-1)} + (D - L)^{-1}b = x^{(k-1)} - (D - L)^{-1}(b - Ax^{(k-1)}) \equiv x^{(k-1)} + z^{(k-1)}.
\]