Iterative Techniques in Matrix Algebra

Jacobi & Gauss-Seidel Iterative Techniques I

Numerical Analysis (9th Edition)

R L Burden & J D Faires

Beamer Presentation Slides
prepared by
John Carroll
Dublin City University

Outline

1 Introducing Iterative Techniques for Linear Systems

Outline

1. Introducing Iterative Techniques for Linear Systems
2. The Jacobi Iterative Method

Outline

1. Introducing Iterative Techniques for Linear Systems

2. The Jacobi Iterative Method

3. Converting $A\mathbf{x} = \mathbf{b}$ into an Equivalent System

Outline

1. Introducing Iterative Techniques for Linear Systems
2. The Jacobi Iterative Method
3. Converting $A\mathbf{x} = \mathbf{b}$ into an Equivalent System
4. The Jacobi Iterative Algorithm

Outline

1. Introducing Iterative Techniques for Linear Systems
2. The Jacobi Iterative Method
3. Converting $A\mathbf{x} = \mathbf{b}$ into an Equivalent System
4. The Jacobi Iterative Algorithm

The Jacobi & Gauss-Seidel Methods

Intyroduction

- We will now describe the Jacobi and the Gauss-Seidel iterative methods, classic methods that date to the late eighteenth century.

The Jacobi & Gauss-Seidel Methods

Intyroduction

- We will now describe the Jacobi and the Gauss-Seidel iterative methods, classic methods that date to the late eighteenth century.
- Iterative techniques are seldom used for solving linear systems of small dimension since the time required for sufficient accuracy exceeds that required for direct techniques such as Gaussian elimination.

The Jacobi & Gauss-Seidel Methods

Intyroduction

- We will now describe the Jacobi and the Gauss-Seidel iterative methods, classic methods that date to the late eighteenth century.
- Iterative techniques are seldom used for solving linear systems of small dimension since the time required for sufficient accuracy exceeds that required for direct techniques such as Gaussian elimination.
- For large systems with a high percentage of 0 entries, however, these techniques are efficient in terms of both computer storage and computation.

The Jacobi & Gauss-Seidel Methods

Iterative Technique

An iterative technique to solve the $n \times n$ linear system

$$A\mathbf{x} = \mathbf{b}$$

The Jacobi & Gauss-Seidel Methods

Iterative Technique

An iterative technique to solve the $n \times n$ linear system

$$A\mathbf{x} = \mathbf{b}$$

starts with an initial approximation

$$\mathbf{x}^{(0)}$$

to the solution $\mathbf{x}$

The Jacobi & Gauss-Seidel Methods

Iterative Technique

An iterative technique to solve the $n \times n$ linear system

$$A\mathbf{x} = \mathbf{b}$$

starts with an initial approximation

$$\mathbf{x}^{(0)}$$

to the solution $\mathbf{x}$ and generates a sequence of vectors

$$\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$$

that converges to $\mathbf{x}$.

Outline

1 Introducing Iterative Techniques for Linear Systems

2 The Jacobi Iterative Method

3 Converting $A\mathbf{x} = \mathbf{b}$ into an Equivalent System

4 The Jacobi Iterative Algorithm

Jacobi's Method

The Jacobi iterative method is obtained by solving the ith equation in $A\mathbf{x} = \mathbf{b}$ for x_i to obtain (provided $a_{ii} \neq 0$)

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

Jacobi's Method

The Jacobi iterative method is obtained by solving the ith equation in $A\mathbf{x} = \mathbf{b}$ for x_i to obtain (provided $a_{ii} \neq 0$)

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

For each $k \geq 1$, generate the components $x_i^{(k)}$ of $\mathbf{x}^{(k)}$ from the components of $\mathbf{x}^{(k-1)}$ by

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

Jacobi's Method

Example

The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{array}{rrrrrrrrrl}
E_1: & 10x_1 & - & x_2 & + & 2x_3 & & & = & 6 \\
E_2: & -x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25 \\
E_3: & 2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
E_4: & & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}$$

has the unique solution $\mathbf{x} = (1, 2, -1, 1)^t$.

Jacobi's Method

Example

The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{aligned}
E_1: &\quad 10x_1 - x_2 + 2x_3 &= 6\\
E_2: &\quad -x_1 + 11x_2 - x_3 + 3x_4 &= 25\\
E_3: &\quad 2x_1 - x_2 + 10x_3 - x_4 &= -11,\\
E_4: &\quad 3x_2 - x_3 + 8x_4 &= 15
\end{aligned}$$

has the unique solution $\mathbf{x} = (1, 2, -1, 1)^t$. Use Jacobi's iterative technique to find approximations $\mathbf{x}^{(k)}$ to $\mathbf{x}$ starting with $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$ until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_\infty}{\|\mathbf{x}^{(k)}\|_\infty} < 10^{-3}$$

Jacobi's Method: Example

Solution (1/4)

We first solve equation E_i for x_i, for each $i = 1, 2, 3, 4$, to obtain

$$
\begin{aligned}
x_1 &= \phantom{-\frac{1}{5}x_1 +} \frac{1}{10}x_2 - \frac{1}{5}x_3 \phantom{+ \frac{1}{10}x_4} + \frac{3}{5} \\
x_2 &= \frac{1}{11}x_1 \phantom{+ \frac{1}{10}x_2} + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11} \\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 \phantom{+ \frac{1}{11}x_3} + \frac{1}{10}x_4 - \frac{11}{10} \\
x_4 &= \phantom{-\frac{1}{5}x_1} - \frac{3}{8}x_2 + \frac{1}{8}x_3 \phantom{+ \frac{1}{10}x_4} + \frac{15}{8}
\end{aligned}
$$

Jacobi's Method: Example

Solution (2/4)

From the initial approximation $\mathbf{x}^{(0)} = (0,0,0,0)^t$ we have $\mathbf{x}^{(1)}$ given by

$$
\begin{aligned}
x_1^{(1)} &= \phantom{-\frac{1}{5}x_1^{(0)}} \quad \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} \phantom{- \frac{3}{11}x_4^{(0)}} + \frac{3}{5} = 0.6000 \\
x_2^{(1)} &= \frac{1}{11}x_1^{(0)} \phantom{+ \frac{1}{10}x_2^{(0)}} + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11} = 2.2727 \\
x_3^{(1)} &= -\frac{1}{5}x_1^{(0)} + \frac{1}{10}x_2^{(0)} \phantom{+ \frac{1}{11}x_3^{(0)}} + \frac{1}{10}x_4^{(0)} - \frac{11}{10} = -1.1000 \\
x_4^{(1)} &= \phantom{-\frac{1}{5}x_1^{(0)}} - \frac{3}{8}x_2^{(0)} + \frac{1}{8}x_3^{(0)} \phantom{- \frac{3}{11}x_4^{(0)}} + \frac{15}{8} = 1.8750
\end{aligned}
$$

Jacobi's Method: Example

Solution (3/4)

Additional iterates, $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)})^t$, are generated in a similar manner and are summarized as follows:

k	0	1	2	3	4		10
$x_1^{(k)}$	0.0	0.6000	1.0473	0.9326	1.0152	$\cdots$	1.0001
$x_2^{(k)}$	0.0	2.2727	1.7159	2.053	1.9537	$\cdots$	1.9998
$x_3^{(k)}$	0.0	-1.1000	-0.8052	-1.0493	-0.9681	$\cdots$	-0.9998
$x_4^{(k)}$	0.0	1.8750	0.8852	1.1309	0.9739	$\cdots$	0.9998

Jacobi's Method: Example

Solution (4/4)

The process was stopped after 10 iterations because

$$\frac{\|\mathbf{x}^{(10)} - \mathbf{x}^{(9)}\|_\infty}{\|\mathbf{x}^{(10)}\|_\infty} = \frac{8.0 \times 10^{-4}}{1.9998} < 10^{-3}$$

In fact, $\|\mathbf{x}^{(10)} - \mathbf{x}\|_\infty = 0.0002$.

Outline

1. Introducing Iterative Techniques for Linear Systems
2. The Jacobi Iterative Method
3. Converting $A\mathbf{x} = \mathbf{b}$ into an Equivalent System
4. The Jacobi Iterative Algorithm

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation

- In general, iterative techniques for solving linear systems involve a process that converts the system $A\mathbf{x} = \mathbf{b}$ into an equivalent system of the form

$$\mathbf{x} = T\mathbf{x} + \mathbf{c}$$

for some fixed matrix T and vector $\mathbf{c}$.

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation

- In general, iterative techniques for solving linear systems involve a process that converts the system $A\mathbf{x} = \mathbf{b}$ into an equivalent system of the form

$$\mathbf{x} = T\mathbf{x} + \mathbf{c}$$

  for some fixed matrix T and vector $\mathbf{c}$.
- After the initial vector $\mathbf{x}^{(0)}$ is selected, the sequence of approximate solution vectors is generated by computing

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

  for each $k = 1, 2, 3, \ldots$ (reminiscent of the fixed-point iteration for solving nonlinear equations).

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

- The Jacobi method can be written in the form

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

by splitting A into its diagonal and off-diagonal parts.

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

- The Jacobi method can be written in the form

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

  by splitting A into its diagonal and off-diagonal parts.

- To see this, let D be the diagonal matrix whose diagonal entries are those of A, $-L$ be the strictly lower-triangular part of A, and $-U$ be the strictly upper-triangular part of A where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

We then write $A = D - L - U$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

We then write $A = D - L - U$ where

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix}$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

We then write $A = D - L - U$ where

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

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

We then write $A = D - L - U$ where

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

and

$$U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

The equation $A\mathbf{x} = \mathbf{b}$, or $(D - L - U)\mathbf{x} = \mathbf{b}$, is then transformed into

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b}$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

The equation $A\mathbf{x} = \mathbf{b}$, or $(D - L - U)\mathbf{x} = \mathbf{b}$, is then transformed into

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b}$$

and, if D^{-1} exists, that is, if $a_{ii} \neq 0$ for each i, then

$$\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

The equation $A\mathbf{x} = \mathbf{b}$, or $(D - L - U)\mathbf{x} = \mathbf{b}$, is then transformed into

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b}$$

and, if D^{-1} exists, that is, if $a_{ii} \neq 0$ for each i, then

$$\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}$$

This results in the matrix form of the Jacobi iterative technique:

$$\mathbf{x}^{(k)} = D^{-1}(L + U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b}, \quad k = 1, 2, \ldots$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

Introducing the notation $T_j = D^{-1}(L + U)$ and $\mathbf{c}_j = D^{-1}\mathbf{b}$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

Introducing the notation $T_j = D^{-1}(L+U)$ and $\mathbf{c}_j = D^{-1}\mathbf{b}$ gives the Jacobi technique the form

$$\mathbf{x}^{(k)} = T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

A More General Representation (Cont'd)

Introducing the notation $T_j = D^{-1}(L+U)$ and $\mathbf{c}_j = D^{-1}\mathbf{b}$ gives the Jacobi technique the form

$$\mathbf{x}^{(k)} = T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j$$

In practice, this form is only used for theoretical purposes while

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

is used in computation.

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

Example

Express the Jacobi iteration method for the linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{aligned}
E_1 &: \quad 10x_1 - x_2 + 2x_3 = 6 \\
E_2 &: \quad -x_1 + 11x_2 - x_3 + 3x_4 = 25 \\
E_3 &: \quad 2x_1 - x_2 + 10x_3 - x_4 = -11 \\
E_4 &: \quad 3x_2 - x_3 + 8x_4 = 15
\end{aligned}$$

in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$.

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

Solution (1/2)

We saw earlier that the Jacobi method for this system has the form

$$\begin{aligned}
x_1 &= \phantom{-\frac{1}{5}x_1 +{}} \frac{1}{10}x_2 - \frac{1}{5}x_3 \phantom{{}+ \frac{1}{10}x_4} + \frac{3}{5} \\
x_2 &= \frac{1}{11}x_1 \phantom{{}+ \frac{1}{10}x_2} + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11} \\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 \phantom{{}+ \frac{1}{11}x_3} + \frac{1}{10}x_4 - \frac{11}{10} \\
x_4 &= \phantom{-\frac{1}{5}x_1} - \frac{3}{8}x_2 + \frac{1}{8}x_3 \phantom{{}+ \frac{1}{10}x_4} + \frac{15}{8}
\end{aligned}$$

Jacobi's Method in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$

Solution (2/2)

Hence, we have

$$T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}$$

Outline

1. Introducing Iterative Techniques for Linear Systems
2. The Jacobi Iterative Method
3. Converting $A\mathbf{x} = \mathbf{b}$ into an Equivalent System
4. The Jacobi Iterative Algorithm

The Jacobi Iterative Algorithm (1/2)

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

The Jacobi Iterative Algorithm (1/2)

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

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 $\mathbf{b}$;
the entries XO_i, $1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$;
tolerance TOL;
maximum number of iterations N.

The Jacobi Iterative Algorithm (1/2)

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

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 $\mathbf{b}$;
the entries XO_i, $1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$;
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.

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + b_i \right]$$

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + b_i \right]$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + b_i \right]$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$

(*The procedure was successful*)

STOP

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + b_i \right]$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$

(*The procedure was successful*)

STOP

Step 5 Set $k = k + 1$

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + b_i \right]$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$

(*The procedure was successful*)

STOP

Step 5 Set $k = k + 1$

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$

The Jacobi Iterative Algorithm (2/2)

Step 1 Set $k = 1$

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

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + b_i \right]$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$

(*The procedure was successful*)

STOP

Step 5 Set $k = k + 1$

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$

Step 7 OUTPUT ('Maximum number of iterations exceeded')

(*The procedure was successful*)

STOP

Jacobi's Method

Comments on the Algorithm

- Step 3 of the algorithm requires that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$.

Jacobi's Method

Comments on the Algorithm

- Step 3 of the algorithm requires that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$. If one of the a_{ii} entries is 0 and the system is nonsingular, a reordering of the equations can be performed so that no $a_{ii} = 0$.

Jacobi's Method

Comments on the Algorithm

- Step 3 of the algorithm requires that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$. If one of the a_{ii} entries is 0 and the system is nonsingular, a reordering of the equations can be performed so that no $a_{ii} = 0$.
- To speed convergence, the equations should be arranged so that a_{ii} is as large as possible.

Jacobi's Method

Comments on the Algorithm

- Step 3 of the algorithm requires that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$. If one of the a_{ii} entries is 0 and the system is nonsingular, a reordering of the equations can be performed so that no $a_{ii} = 0$.
- To speed convergence, the equations should be arranged so that a_{ii} is as large as possible.
- Another possible stopping criterion in Step 4 is to iterate until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|}{\|\mathbf{x}^{(k)}\|}$$

is smaller than some prescribed tolerance.

Jacobi's Method

Comments on the Algorithm

- Step 3 of the algorithm requires that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$. If one of the a_{ii} entries is 0 and the system is nonsingular, a reordering of the equations can be performed so that no $a_{ii} = 0$.
- To speed convergence, the equations should be arranged so that a_{ii} is as large as possible.
- Another possible stopping criterion in Step 4 is to iterate until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|}{\|\mathbf{x}^{(k)}\|}$$

  is smaller than some prescribed tolerance.
- For this purpose, any convenient norm can be used, the usual being the l_∞ norm.

Questions?