Announcements

- Next week: midterm exam
  - starts at 14:20
  - about 80-90 minutes
  - [Optional] bring a A4 cheat sheet (one side only)

Solving linear systems III

Mei-Chen Yeh

Review

- ***Direct*** methods for solving linear systems
  - Gaussian elimination with *partial pivoting*
  - The PA = LU factorization

Today

- ***Iterative*** methods for solving linear systems
  - Jacobi method
  - Gauss-Seidel method
  - Successive over-relaxation (SOR)

Solving A$\underline{x}$ = $\underline{b}$

- A is given, real, n x n, and $\underline{b}$ is given, real vector.
- Two types of approaches
  - Direct methods: yield exact solution in absence of roundoff error

    Example: Gaussian elimination and its variants
  - **Iterative** methods: iterate in a similar fashion to what we do for solving nonlinear equations

Iterative methods for solving A$\underline{x}$ = $\underline{b}$

- Starting from initial guess $\underline{x}_0$, generate iterates $\underline{x}_1$, $\underline{x}_2$, …, $\underline{x}_k$, hopefully converging to solution $\underline{x}$.
- But why not simply using a direct method (e.g., LU decomposition)?

Why using iterative methods?

- Can be faster if the input matrix is large
  - One step of an iterative method requires only a fraction of the floating operations of a full LU factorization.
- A good approximation to the solution is already known.
- The input matrix is *sparse*.

Direct vs. iterative linear solvers

- Direct solvers
  - Computation is numerically stable in many relevant cases. ☺
  - Can solve economically for several right-hand sides. ☺
  - But: fill-in limits usefulness (memory). ☹
- Iterative solvers
  - Only a rough approximation to $\underline{x}$ is required. ☺
  - A good $\underline{x}_0$ approximating $\underline{x}$ is generally known (warm start). ☺
  - Quality often depends on “right” choice of parameters. ☹

Typical scenarios

- Direct solvers
  - Many linear systems with the same matrix A
  - Applications that require very accurate solutions
- Iterative solvers
  - Many linear systems with “slightly changing” matrices
  - Matrix-free applications
  - Very large problems

Iterative methods

- Jacobi method
- Gauss-Seidel method
- Successive over-relaxation (SOR)

Jacobi me

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- A form of fixed-point iteration
- Example: Solve the linear system

  $3u + v = 5$

  $u + 2v = 5$
- The answer is $u = 1$, $v = 2$.
- First, we have

$$u = \frac{5 - v}{3}$$

$$v = \frac{5 - u}{2}.$$

- Start from an initial guess (0, 0)

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/2}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix}$$

$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-5/6}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix}$$

$$\vdots$$

continue

Jacobi method

- Use the example again, but now the equations are given in the *reverse* order.

  $u + 2v = 5$

  $3u + v = 5$

- The answer is $u = 1$, $v = 2$
- We have

$$u = 5 - 2v$$
$$v = 5 - 3u$$

- Start from an initial guess (0, 0)

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 5 - 2v_0 \\ 5 - 3u_0 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 5 - 2v_1 \\ 5 - 3u_1 \end{bmatrix} = \begin{bmatrix} -5 \\ -10 \end{bmatrix}$$

$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 5 - 2(-10) \\ 5 - 3(-5) \end{bmatrix} = \begin{bmatrix} 25 \\ 20 \end{bmatrix}$$

$$\vdots$$

The difference?

- $3u + v = 5$
- $u + 2v = 5$

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

The Jacobi method converges. ☺

vs.

- $u + 2v = 5$
- $3u + v = 5$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

The Jacobi method diverges. ☹

Strictly diagonally dominant

- **Definition**. The n x n matrix A = (a_{ij}) is **strictly diagonally dominant** if, for each $1 \leq i \leq n$, $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$.

- Which matrix is strictly diagonally dominant?

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -5 & 2 \\ 1 & 6 & 8 \end{bmatrix} \qquad B = \begin{bmatrix} 3 & 2 & 6 \\ 1 & 8 & 1 \\ 9 & 2 & -2 \end{bmatrix}$$

✓

Theorem. If the n x n matrix A is strictly diagonally dominant, then

1. A is a nonsingular matrix, and
2. for every vector $\underline{b}$ and every starting guess, the Jacobi method applied to A$\underline{x}$ = $\underline{b}$ converges to the solution.

The previous example

- $3u + v = 5$
- $u + 2v = 5$

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

strictly diagonally dominant

The Jacobi method converges. ☺

vs.

- $u + 2v = 5$
- $3u + v = 5$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

NOT strictly diagonally dominant

The Jacobi method diverges. ☹

A form of fixed-point iteration $\underline{x} = g(\underline{x})$

- Start with $A\underline{x} = \underline{b}$

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

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

  $\underline{x} = \underbrace{D^{-1}(\underline{b} - (L + U)\underline{x})}_{g(\underline{x})}$

U

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -5 & 2 \\ 1 & 6 & 8 \end{bmatrix}$$

L D

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 8 \end{bmatrix} \quad L = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 6 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

- Jacobi Method

$x_0 =$ initial vector

$x_{k+1} = D^{-1}(b - (L + U)x_k)$ for $k = 0, 1, 2, \ldots$

Example: Jacobi method

- Jacobi method

$$x_0 = \text{initial vector}$$
$$x_{k+1} = D^{-1}(b - (L+U)x_k) \text{ for } k = 0,1,2,\ldots$$

- The previous example
  - $3u + v = 5$
  - $u + 2v = 5$

$$D = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix}}_{D^{-1}} \left(\underbrace{\begin{bmatrix} 5 \\ 5 \end{bmatrix}}_{b} - \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{L+U} \underbrace{\begin{bmatrix} u_k \\ v_k \end{bmatrix}}_{x_k} \right) = \begin{bmatrix} (5 - v_k)/3 \\ (5 - u_k)/2 \end{bmatrix}$$

check

Iterative methods

- Jacobi method
- Gauss-Seidel method
- Successive over-relaxation (SOR)

Gauss-Seidel method

- Uses the most recently updated values of the knowns, even if the updating occurs in the current step!
- The previous exa

$3u + v = 5$

$u + 2v = 5$

$$u = \frac{5 - v}{3}$$

$$v = \frac{5 - u}{2}.$$

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{3} \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_2}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-10/9}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{35}{18} \end{bmatrix}$$

$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} \frac{5-v_2}{3} \\ \frac{5-u_3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-35/18}{3} \\ \frac{5-55/54}{2} \end{bmatrix} = \begin{bmatrix} \frac{55}{54} \\ \frac{215}{108} \end{bmatrix}$$

$$\vdots$$

A form of fixed-point iteration $\underline{x} = g(\underline{x})$

- Start with $A\underline{x} = \underline{b}$

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

  $(L+D)\underline{x}_{k+1} = \underline{b} - U\underline{x}_k$

  $D\underline{x}_{k+1} = \underline{b} - U\underline{x}_k - L\underline{x}_{k+1}$

  $$\underline{x}_{k+1} = \underbrace{D^{-1}(\underline{b} - U\underline{x}_k - L\underline{x}_{k+1})}_{g(\underline{x})}$$

U

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -5 & 2 \\ 1 & 6 & 8 \end{bmatrix}$$

L D

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 8 \end{bmatrix} \quad L = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 6 & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

- **Gauss-Seidel Method**

  $x_0 =$ initial vector

  $x_{k+1} = D^{-1}(b - Ux_k - Lx_{k+1})$ for $k = 0,1,2,\ldots$

- Gauss-Seidel Method

$$x_0 = \text{initial vector}$$
$$x_{k+1} = D^{-1}(b - Ux_k - Lx_{k+1}) \text{ for } k = 0,1,2,\ldots$$

- Example $\begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}$

$$\begin{bmatrix} u_{k+1} \\ v_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/5 \end{bmatrix}\left(\begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ -1 & 2 & 0 \end{bmatrix}\begin{bmatrix} u_{k+1} \\ v_{k+1} \\ w_{k+1} \end{bmatrix}\right)$$

$$= \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/5 \end{bmatrix}\left(\begin{bmatrix} 4 - v_k + w_k \\ 1 - w_k - 2u_{k+1} \\ 1 + u_{k+1} - 2v_{k+1} \end{bmatrix}\right)$$

$$3u + v - w = 4 \ \Rightarrow u = \frac{1}{3}(4 - v + w) \qquad -u + 2v + 5w = 1 \ \Rightarrow w = \frac{1}{5}(1 + u - 2v)$$

$$2u + 4v + w = 1 \ \Rightarrow v = \frac{1}{4}(1 - w - 2u)$$

Jacobi vs. Gauss-Seidel

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/2}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix}$$

$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-5/6}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix}$$

$$\underline{x}_{k+1} = D^{-1}(\underline{b} - (L+U)\underline{x}_k)$$

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{3} \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_2}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-10/9}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{35}{18} \end{bmatrix}$$

$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} \frac{5-v_2}{3} \\ \frac{5-u_3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-35/18}{3} \\ \frac{5-55/54}{2} \end{bmatrix} = \begin{bmatrix} \frac{55}{54} \\ \frac{215}{108} \end{bmatrix}$$

$$\underline{x}_{k+1} = D^{-1}(\underline{b} - U\underline{x}_k - L\underline{x}_{k+1})$$

Iterative methods

- Jacobi method
- Gauss-Seidel method
- Successive over-relaxation (SOR)

Successive Over-Relaxation (SOR)

- Uses a weighted average of the Gauss-Seidel and the current guess

$$\mathbf{x}_{k+1} \leftarrow \omega \mathbf{x}_{k+1} + (1 - \omega)\mathbf{x}_k$$

- The relaxation parameter: ω
  - $\omega \cdot$ Gauss-Seidel
  - $(1- \omega) \cdot$ current guess
- Over-relaxation: $\omega > 1$
- Also can be written as a form of fixed-point iteration

Gauss-Seidel vs. SOR

$$\begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}$$

Gauss-Seidel

$$u_{k+1} = \frac{4 - v_k + w_k}{3}$$
$$v_{k+1} = \frac{1 - 2u_{k+1} - w_k}{4}$$
$$w_{k+1} = \frac{1 + u_{k+1} - 2v_{k+1}}{5}$$

SOR

$$u_{k+1} = (1 - \omega)u_k + \omega \frac{4 - v_k + w_k}{3}$$
$$v_{k+1} = (1 - \omega)v_k + \omega \frac{1 - 2u_{k+1} - w_k}{4}$$
$$w_{k+1} = (1 - \omega)w_k + \omega \frac{1 + u_{k+1} - 2v_{k+1}}{5}$$

Successive Over-Relaxation (SOR)

- Also can be written as a form of fixed-point iteration

$x_0 =$ initial vector
$x_{k+1} = (\omega L + D)^{-1}[(1-\omega)Dx_k - \omega U x_k] + \omega(D + \omega L)^{-1}b$ for $k = 0,1,2,\ldots$

程式練習

And, please upload your program on moodle2.

- Please use Jacobi, Gauss-Seidel or SOR to solve the system of 10 equations in 10 unknowns:

$$\begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ \\ \vdots \\ \\ x_{10} \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ \vdots \\ 1 \\ 2 \end{pmatrix}$$