Special Matrices and Gauss-Seidel

Chapter 11

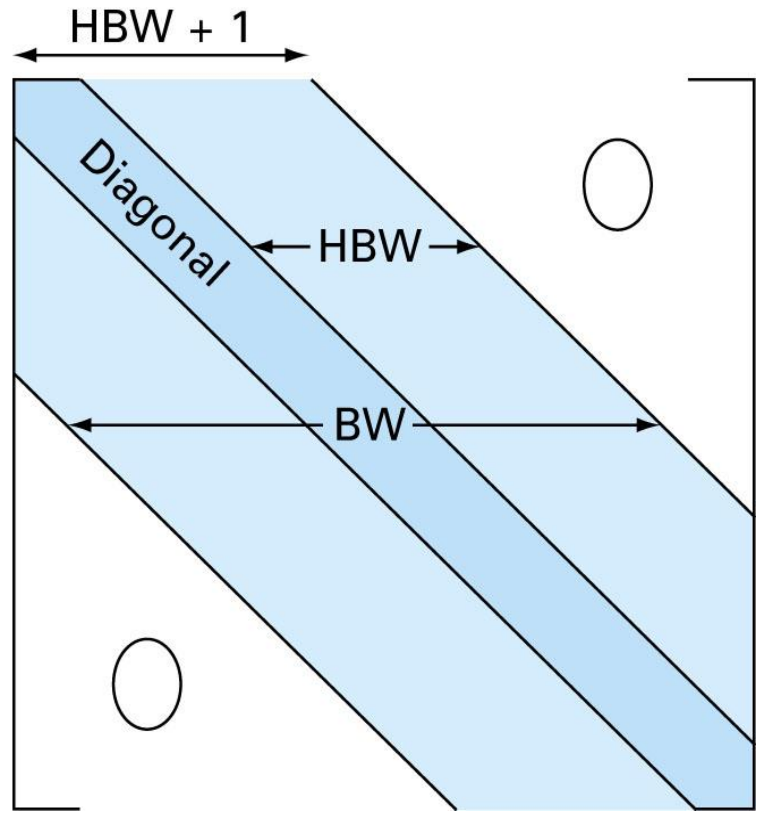
Credit: Prof. Lale Yurttas, Chemical Eng., Texas A&M University

- Certain matrices have particular structures that can be exploited to develop efficient solution schemes (e.g. *banded*, *symmetric*)

- A *banded matrix* is a square matrix that has all elements equal to zero, with the exception of a **band** centered on the main diagonal.

- Standard Gauss Elimination is inefficient in solving banded equations because unnecessary space and time would be expended on the storage and manipulation of **zeros**.

- There is no need to store or process the zeros (off of the band)



Solving Tridiagonal Systems (Thomas Algorithm)

A tridiagonal system has a bandwidth of 3

$$\begin{bmatrix} f_1 & g_1 & & \\ e_2 & f_2 & g_2 & \\ & e_3 & f_3 & g_3 \\ & & e_4 & f_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{Bmatrix}$$

$$\boldsymbol{A} = \boldsymbol{L} * \boldsymbol{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \boldsymbol{e}'_2 & 1 & 0 & 0 \\ 0 & \boldsymbol{e}'_3 & 1 & 0 \\ 0 & 0 & \boldsymbol{e}'_4 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{g}_1 & & \\ & \boldsymbol{f}_2' & \boldsymbol{g}_2 & \\ & & \boldsymbol{f}_3' & \boldsymbol{g}_3 \\ & & & \boldsymbol{f}_4' \end{bmatrix}$$

DECOMPOSITION

DO $k = 2, n$

$$e_k = e_k / f_{k-1}$$

$$f_k = f_k - e_k \, g_{k-1}$$

END DO

Time Complexity?

O(n)

vs. **O(n³)**

Tridiagonal Systems (cont.)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ e'_2 & 1 & 0 & 0 \\ 0 & e'_3 & 1 & 0 \\ 0 & 0 & e'_4 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} f_1 & g_1 & & \\ & f'_2 & g_2 & \\ & & f'_3 & g_3 \\ & & & f'_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\{d\}} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ e'_2 & 1 & 0 & 0 \\ 0 & e'_3 & 1 & 0 \\ 0 & 0 & e'_4 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} \qquad \begin{bmatrix} f_1 & g_1 & & \\ & f'_2 & g_2 & \\ & & f'_3 & g_3 \\ & & & f'_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

Forward Substitution

$d_1 = r_1$

DO $k = 2, n$

$d_k = r_k - e_k\, d_{k-1}$

END DO

Back Substitution

$x_n = d_n / f_n$

DO $k = n\text{-}1, 1, \text{-}1$

$x_k = (d_k - g_k \cdot x_{k+1}) / f_k$

END DO

Cholesky Decomposition

(for *Symmetric Positive Definite*[†] Matrices)

[†] A *positive definite matrix* is one for which the product $\{X\}^T[A]\{X\}$ is greater than zero for all nonzero vectors X

$$[A]=[A]^T=\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ a_{31} & a_{32} & a_{33} & a_{43} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$[A]=[A]^T=L*L^T=\begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ l_{31} & l_{32} & l_{33} & \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}\begin{bmatrix} l_{11} & l_{21} & l_{31} & l_{41} \\ & l_{22} & l_{32} & l_{42} \\ & & l_{33} & l_{43} \\ & & & l_{44} \end{bmatrix}$$

L^T means **Transpose** of L

$$l_{ki}=\frac{a_{ki}-\sum_{j=1}^{i-1} l_{ij}l_{kj}}{l_{ii}} \quad \text{for } k=1,2,\ldots,n \quad i=1,2,\ldots,k-1 \qquad l_{kk}=\sqrt{a_{kk}-\sum_{j=1}^{k-1} l_{kj}^2}$$

Time Complexity:

$O(n^3)$ but requires half the number of operations as standard Gaussian elimination.

Jacobi Iterative Method

Iterative methods provide an **alternative** to the *elimination methods*.

$$\boldsymbol{Ax} = \boldsymbol{b} \qquad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \boldsymbol{a}_{13} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} & \boldsymbol{a}_{23} \\ \boldsymbol{a}_{31} & \boldsymbol{a}_{32} & \boldsymbol{a}_{33} \end{bmatrix} \qquad \boldsymbol{D} = \begin{bmatrix} \boldsymbol{a}_{11} & 0 & 0 \\ 0 & \boldsymbol{a}_{22} & 0 \\ 0 & 0 & \boldsymbol{a}_{33} \end{bmatrix}$$

$$[\boldsymbol{D} + (\boldsymbol{A} - \boldsymbol{D})]\boldsymbol{x} = \boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{Dx} = \boldsymbol{b} - (\boldsymbol{A} - \boldsymbol{D})\boldsymbol{x} \quad \Rightarrow \quad \boldsymbol{x} = \boldsymbol{D}^{-1}[\boldsymbol{b} - (\boldsymbol{A} - \boldsymbol{D})\boldsymbol{x}]$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{k+1} = \begin{bmatrix} 1/a_{11} & 0 & 0 \\ 0 & 1/a_{22} & 0 \\ 0 & 0 & 1/a_{33} \end{bmatrix} * \left(\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} - \begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k \right)$$

$$x_1^{k+1} = \frac{b_1 - a_{12}x_2^k - a_{13}x_3^k}{a_{11}} \qquad x_2^{k+1} = \frac{b_2 - a_{21}x_1^k - a_{23}x_3^k}{a_{22}} \qquad x_3^{k+1} = \frac{b_3 - a_{31}x_1^k - a_{32}x_2^k}{a_{33}}$$

Choose an initial guess (i.e. all zeros) and Iterate until the equality is satisfied.
No guarantee for convergence! Each iteration takes O(n^2) time!

Gauss-Seidel

- The *Gauss-Seidel* method is a commonly used *iterative method*.

- It is same as **Jacobi technique** except with one important difference:
  A newly computed *x* value (say x_k) is substituted in the subsequent equations (equations $k+1$, $k+2$, …, n) **in the same iteration**.

Example: Consider the *3x3* system below:

$$x_1^{new} = \frac{b_1 - a_{12}x_2^{old} - a_{13}x_3^{old}}{a_{11}}$$

$$x_2^{new} = \frac{b_2 - a_{21}x_1^{new} - a_{23}x_3^{old}}{a_{22}}$$

$$x_3^{new} = \frac{b_3 - a_{31}x_1^{new} - a_{32}x_2^{new}}{a_{33}}$$

$$\{X\}_{old} \leftarrow \{X\}_{new}$$

- First, choose initial guesses for the x's.
- A simple way to obtain initial guesses is to assume that they are all **zero**.
- Compute **new** $\mathbf{x_1}$ using the previous iteration values.
- **New** x_1 is substituted in the equations to calculate x_2 and x_3
- The process is repeated for $\mathbf{x_2}$, $\mathbf{x_3}$, …

Convergence Criterion for Gauss-Seidel Method

- Iterations are repeated until the convergence criterion is satisfied:

$$\left|\varepsilon_{a,i}\right| = \left|\frac{x_i^j - x_i^{j-1}}{x_i^j}\right| 100\% < \varepsilon_s$$

For all *i*, where *j* and *j-1* are the *current* and *previous* iterations.

- As any other iterative method, the **Gauss-Seidel** method has problems:
  - It may not converge or it converges very slowly.

- If the coefficient matrix A is **Diagonally Dominant** Gauss-Seidel is guaranteed to converge.

Diagonally Dominant ➔

For each equation i:

$$\left|a_{ii}\right| > \sum_{\substack{j=1 \\ j\neq i}}^{n} \left|a_{i,j}\right|$$

- Note that this is not a necessary condition, i.e. the system *may* still have a chance to converge even if A is not diagonally dominant.

Time Complexity: Each iteration takes $O(n^2)$