2.29 Numerical Fluid Mechanics
Spring 2015 – **Lecture 7**

REVIEW Lecture 6:

- **Direct Methods for solving linear algebraic equations**
  - LU decomposition/factorization
    - Separates time-consuming elimination for **A** from that for b / **B**

$$\overline{\overline{\mathbf{A}}} = \overline{\overline{\mathbf{L}}} \cdot \overline{\overline{\mathbf{U}}} \longrightarrow \begin{aligned} \overline{\overline{\mathbf{L}}}\vec{y} &= \vec{b} \\ \overline{\overline{\mathbf{U}}}\vec{x} &= \vec{y} \end{aligned}$$

    - Derivation, assuming no pivoting needed: $a_{ij} = \sum_{k=1}^{\min(i,j)} m_{ik} a_{kj}^{(k)}$
    - Number of Ops: Same as for Gauss Elimination
    - Pivoting: Use pivot element "pointer vector"
    - Variations: Doolittle and Crout decompositions, Matrix Inverse
  - Error Analysis for Linear Systems
    - Matrix norms
    - Condition Number for Perturbed RHS and LHS: $K(\overline{\overline{\mathbf{A}}}) = \left\| \overline{\overline{\mathbf{A}}}^{-1} \right\| \left\| \overline{\overline{\mathbf{A}}} \right\|$
  - Special Matrices: Intro

TODAY (Lecture 7): Systems of Linear Equations III

- **Direct Methods**
  - Gauss Elimination
  - LU decomposition/factorization
  - Error Analysis for Linear Systems
  - Special Matrices: LU Decompositions
    - Tri-diagonal systems: Thomas Algorithm
    - General Banded Matrices
      - Algorithm, Pivoting and Modes of storage
      - Sparse and Banded Matrices
    - Symmetric, positive-definite Matrices
      - Definitions and Properties, Choleski Decomposition
- **Iterative Methods**
  - Jacobi's method
  - Gauss-Seidel iteration
  - Convergence

Reading Assignment

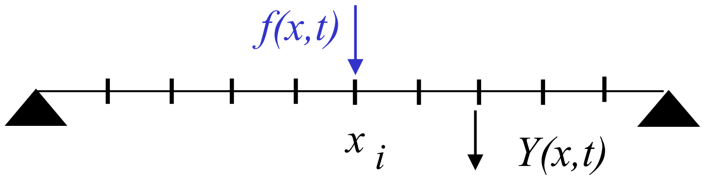
- Chapters 11 of "Chapra and Canale, Numerical Methods for Engineers, 2006/2010/2014."
  - Any chapter on "Solving linear systems of equations" in references on CFD references provided. For example: chapter 5 of "J. H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. Springer, NY, 3rd edition, 2002"

Special Matrices

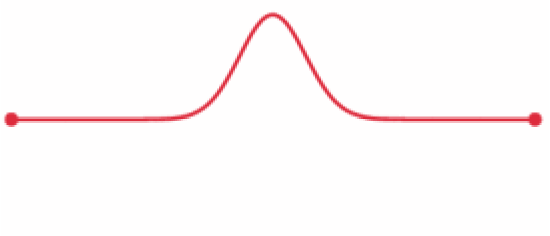
- **Certain Matrices have particular structures that can be exploited, i.e.**
  - Reduce number of ops and memory needs
- **Banded Matrices:**
  - Square banded matrix that has all elements equal to zero, excepted for a band around the main diagonal.
  - Frequent in engineering and differential equations:
    - Tri-diagonal Matrices
    - Wider bands for higher-order schemes
  - Gauss Elimination or LU decomposition inefficient because, if pivoting is not necessary, all elements outside of the band remain zero (but direct GE/LU would manipulate these zero elements anyway)
- **Symmetric Matrices**
- **Iterative Methods:**
  - **Employ initial guesses, than iterate to refine solution**
  - **Can be subject to round-off errors**

Special Matrices:
Tri-diagonal Systems Example

Forced Vibration of a String



Example of a travelling pluse:



Consider the case of a Harmonic excitation

$$f(x,t) = -f(x)\, cos(\omega t)$$

Applying Newton's law leads to the wave equation:
With separation of variables, one obtains the
equation for modal amplitudes, see eq. (1) below:

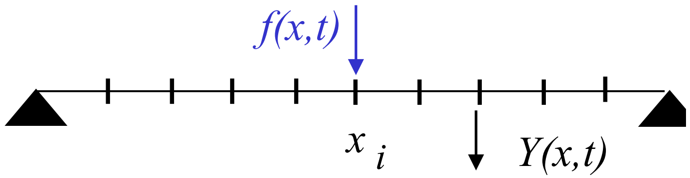
$$\begin{cases} Y_{tt} - c^2 Y_{xx} = f(x,t) \\ Y(x,t) = \tau(t)\, y(x) \end{cases}$$

Differential Equation for the amplitude: $$\frac{d^2y}{dx^2} + k^2 y = f(x) \qquad (1)$$

Boundary Conditions: $$y(0) = 0\ , \quad y(L) = 0$$

Special Matrices: **Tri-diagonal Systems**

Forced Vibration of a String



Harmonic excitation

$$f(x,t) = f(x)\cos(\omega t)$$

Differential Equation:

$$\frac{d^2y}{dx^2} + k^2 y = f(x) \qquad (1)$$

Boundary Conditions:

$$y(0) = 0\,, \quad y(L) = 0$$

Finite Difference

$$\left.\frac{d^2y}{dx^2}\right|_{x_i} \simeq \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + O(h^2)$$

Discrete Difference Equations

$$y_{i-1} + \left((kh)^2 - 2\right) y_i + y_{i+1} = f(x_i)h^2$$

Matrix Form:

$$\begin{bmatrix} (kh)^2-2 & 1 & \cdot & \cdot & \cdot\ \cdot & 0 \\ 1 & (kh)^2-2 & 1 & & & \cdot \\ \cdot & & \cdot & \cdot & & \cdot \\ \cdot & & 1 & (kh)^2-2 & 1 & \cdot \\ \cdot & & & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot\ 1 & (kh)^2-2 \end{bmatrix} \bar{y} = \begin{Bmatrix} f(x_1)h^2 \\ \cdot \\ \cdot \\ f(x_i)h^2 \\ \cdot \\ \cdot \\ f(x_n)h^2 \end{Bmatrix}$$

Tridiagonal Matrix

If $kh < 1$ or $kh > \sqrt{3}$ symmetric, negative or positive definite: No pivoting needed

Note: for 0< kh <1 Negative definite => Write: **A**'=-**A** and $\bar{y}' = -\bar{y}'$ to render matrix positive definite

Special Matrices: **Tri-diagonal Systems**

General Tri-diagonal Systems: Bandwidth of 3

$$\begin{bmatrix} a_1 & c_1 & \cdot & \cdot & \cdot & \cdot & 0 \\ b_2 & a_2 & c_2 & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & b_i & a_i & c_i & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & b_n & a_n \end{bmatrix} \bar{\mathbf{x}} = \begin{Bmatrix} f_1 \\ \cdot \\ \cdot \\ f_i \\ \cdot \\ \cdot \\ f_n \end{Bmatrix}$$

$$\bar{\bar{\mathbf{L}}} = \begin{bmatrix} 1 & & \cdot & \cdot & \cdot & \cdot & 0 \\ \beta_2 & 1 & & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & \beta_i & 1 & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \beta_n & 1 \end{bmatrix}$$

LU Decomposition

$$\bar{\bar{\mathbf{A}}} = \bar{\bar{\mathbf{L}}}\bar{\bar{\mathbf{U}}} \qquad \left\{ \begin{array}{lcl} \bar{\bar{\mathbf{L}}}\bar{\mathbf{y}} & = & \bar{\mathbf{f}} \\ \bar{\bar{\mathbf{U}}}\bar{\mathbf{x}} & = & \bar{\mathbf{y}} \end{array} \right.$$

$$\bar{\bar{\mathbf{U}}} = \begin{bmatrix} \alpha_1 & c_1 & \cdot & \cdot & \cdot & \cdot & 0 \\ & \alpha_2 & c_2 & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & & \alpha_i & c_i & & \cdot \\ \cdot & & & & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & & \alpha_n \end{bmatrix}$$

Three steps for LU scheme:

1. Decomposition (GE): $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$
2. Forward substitution $\bar{\bar{\mathbf{L}}}\bar{\mathbf{y}} = \bar{\mathbf{f}}$
3. Backward substitution $\bar{\bar{\mathbf{U}}}\bar{\mathbf{x}} = \bar{\mathbf{y}}$

Special Matrices: Tri-diagonal Systems Thomas Algorithm

By identification with the general LU decomposition, $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$

one obtains,

$$\overline{\overline{\mathbf{L}}} = \begin{bmatrix} 1 & & \cdot & \cdot & \cdot & \cdot & 0 \\ \beta_2 & 1 & & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & \beta_i & 1 & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \beta_n & 1 \end{bmatrix}$$

$$\overline{\overline{\mathbf{U}}} = \begin{bmatrix} \alpha_1 & c_1 & \cdot & \cdot & \cdot & \cdot & 0 \\ & \alpha_2 & c_2 & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & & \alpha_i & c_i & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & & \alpha_n \end{bmatrix}$$

1. Factorization/Decomposition $\quad \alpha_1 = a_1$

$$\beta_k = \frac{b_k}{\alpha_{k-1}}, \quad \alpha_k = a_k - \beta_k c_{k-1}, \quad k = 2, 3, \ldots n$$

2. Forward Substitution

$$y_1 = f_1, \quad y_i = f_i - \beta_i y_{i-1}, \quad i = 2, 3, \ldots n$$

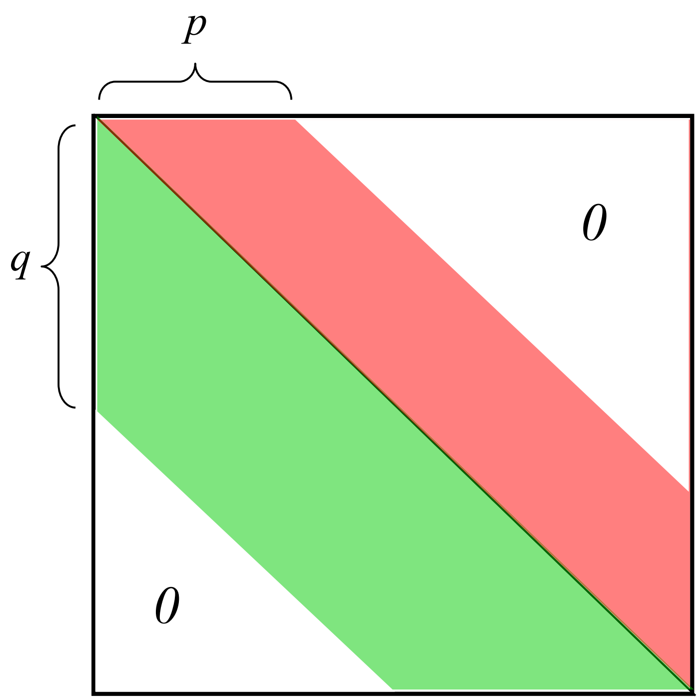
3. Back Substitution

$$x_n = \frac{y_n}{\alpha_n}, \quad x_i = \frac{y_i - c_i x_{i+1}}{\alpha_i}, \quad i = n-1, \ldots 1$$

Number of Operations: Thomas Algorithm

LU Factorization:	3*(n-1) operations
Forward substitution:	2*(n-1) operations
Back substitution:	3*(n-1)+1 operations
Total:	8*(n-1) ~ O(n) operations

Special Matrices:
General, Banded Matrix



General Banded Matrix (p ≠ q)

$$\left.\begin{array}{l} j > i + p \\ \\ i > j + q \end{array}\right\} a_{ij} = 0$$

Banded Symmetric Matrix (p = q = b)

$$\begin{aligned} a_{ij} &= a_{ji} \,, \quad |i - j| \leq b \\ a_{ij} &= a_{ji} = 0 \,, \quad |i - j| > b \end{aligned}$$

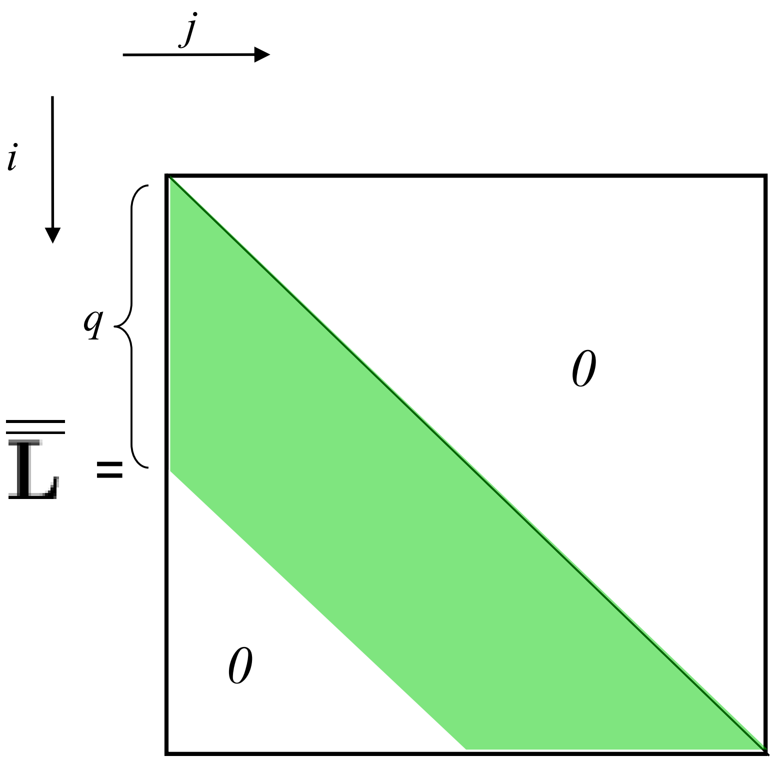
p super-diagonals
q sub-diagonals
$w = p + q + 1$ bandwidth

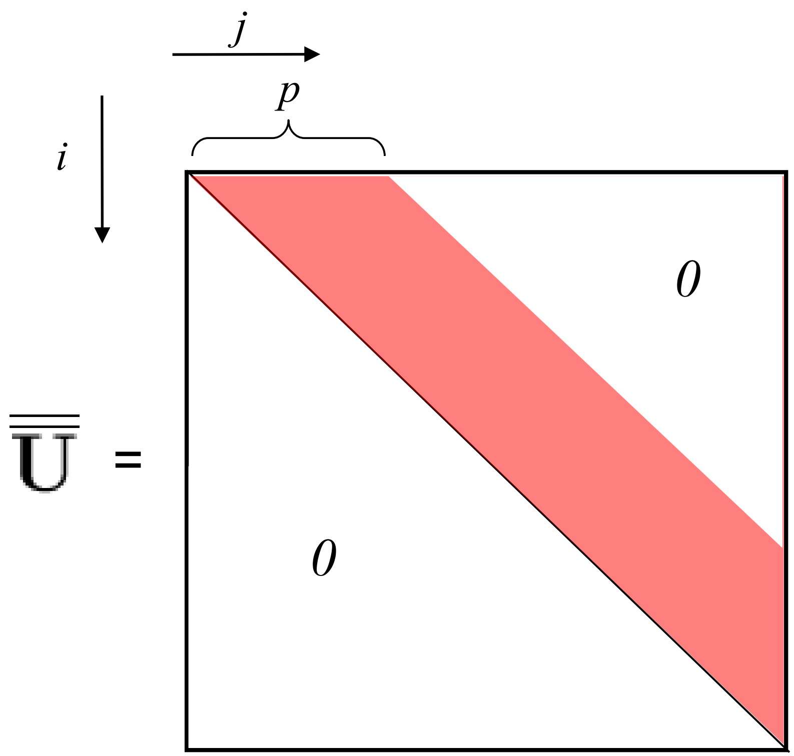
$w = 2\,b + 1$ is called the bandwidth
b is the half-bandwidth

Special Matrices:
General, Banded Matrix

LU Decomposition via Gaussian Elimination

If No Pivoting: the zeros are preserved





$$m_{ij} = \frac{a_{ij}^{(j)}}{a_{jj}^{(j)}} = 0 \quad \text{if} \quad j > i \quad \text{or} \quad i > j + q$$

(as gen. case) (banded)

$$u_{ij} = a_{ij}^{(i)} = a_{ij}^{(i-1)} - m_{i,i-1} a_{i-1,j}^{(i-1)}$$

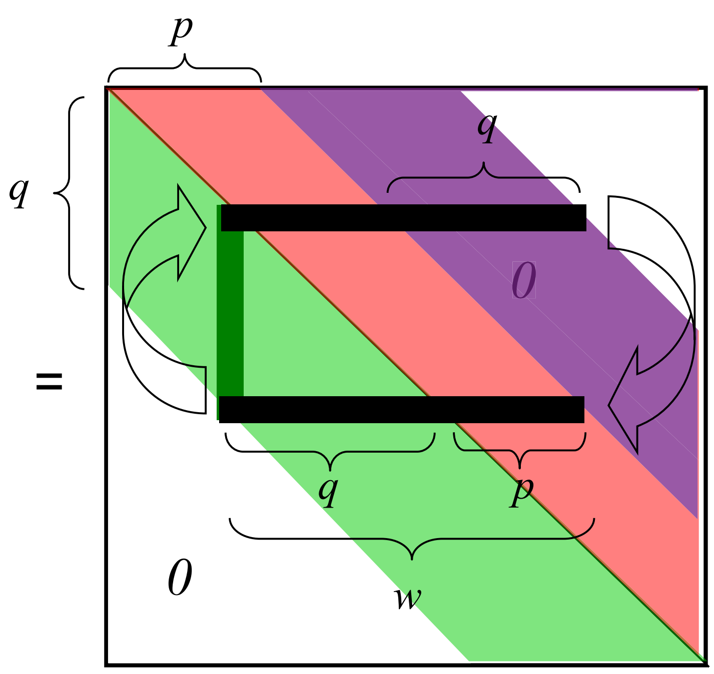
$$u_{ij} = 0 \quad \text{if} \quad i > j \quad \text{or} \quad j > i + p$$

(as gen. case) (banded)

Special Matrices:
General, Banded Matrix

LU Decomposition via Gaussian Elimination
With **Partial Pivoting** (by rows):

Consider pivoting the 2 rows as below:



Then, the bandwidth of L remains unchanged,

$$m_{ij} = 0 \quad \text{if} \quad j > i \quad \text{or} \quad i > j + q$$

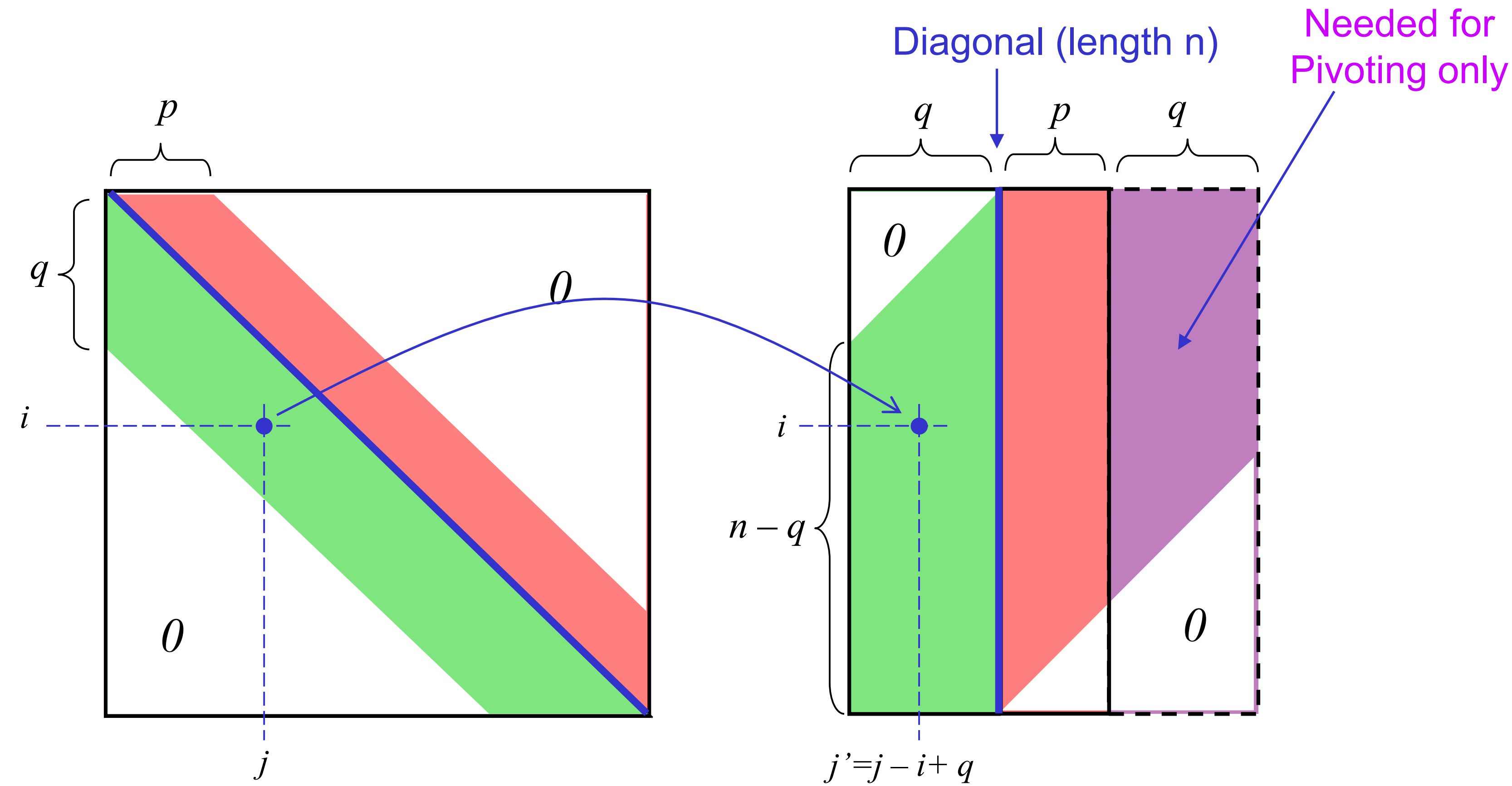
but the bandwidth of U becomes as that of A

$$u_{ij} = 0 \quad \text{if} \quad i > j \quad \text{or} \quad j > i + p + q$$

$w = p + 2\,q + 1$ bandwidth

Special Matrices:
General, Banded Matrix

Compact Storage

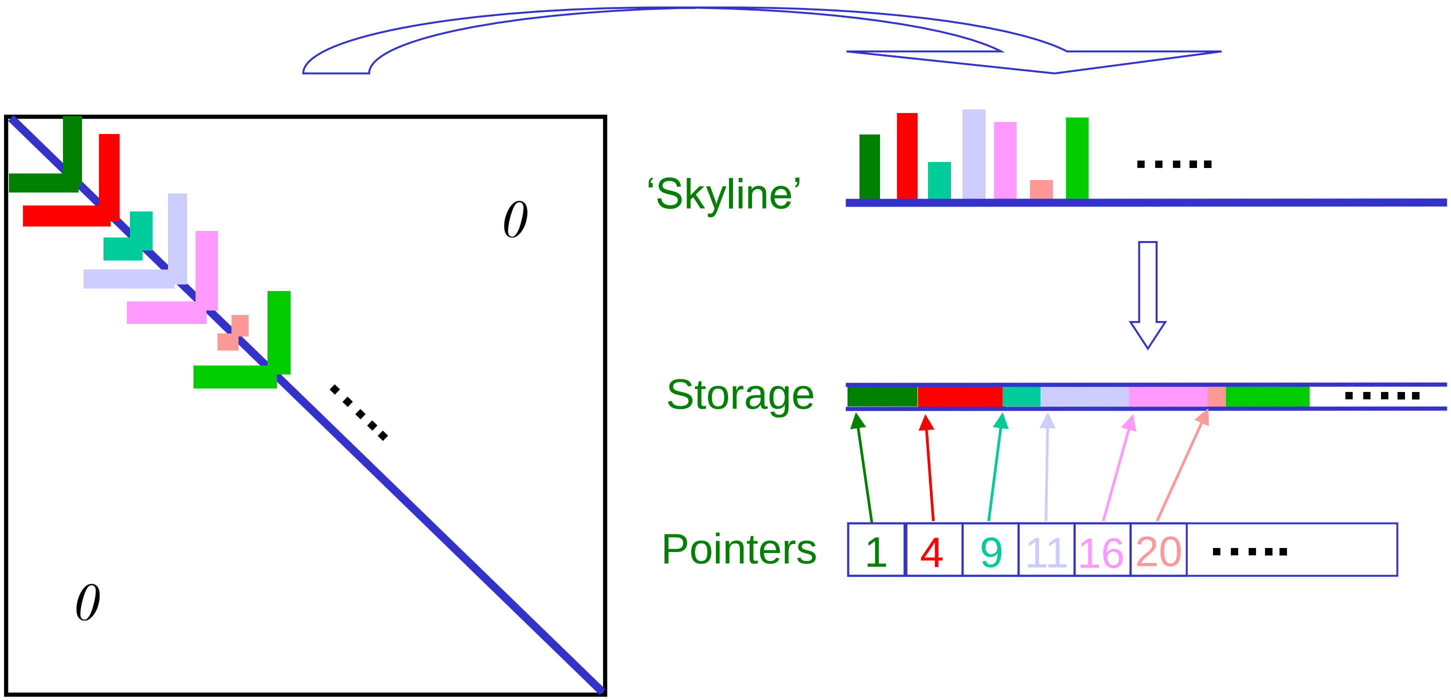


Matrix size: n^2

Matrix size: $n\,(p+2q+1)$

Special Matrices:
Sparse and Banded Matrix

‘Skyline’ Systems
(typically for symmetric matrices)



Skyline storage applicable when no pivoting is needed, e.g. for banded, symmetric, and positive definite matrices: FEM and FD methods. Skyline solvers are usually based on Cholesky factorization (which preserves the skyline)

Special Matrices:
Symmetric (Positive-Definite) Matrix

Symmetric Coefficient Matrices:

- If no pivoting, the matrix remains symmetric after Gauss Elimination/LU decompositions

Proof: Show that if $a_{ij}^{(k)} = a_{ji}^{(k)}$ then $a_{ij}^{(k+1)} = a_{ji}^{(k+1)}$ using:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$$

- Gauss Elimination symmetric (use only the upper triangular portion of **A**):

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}\ a_{kj}^{(k)}$$

$$m_{ik} = \frac{a_{ki}^{(k)}}{a_{kk}^{(k)}}, \qquad i = k+1, k+2, \ldots, n \qquad j = i, i+1, \ldots, n$$

- About half the total number of ops than full GE

Special Matrices:
Symmetric, Positive Definite Matrix

1. Sylvester Criterion:

A symmetric matrix is Positive Definite if and only if:

$\det(\mathbf{A}_k) > 0$ for $k=1,2,\ldots,n,$ where $\mathbf{A}_k$ is matrix of k first lines/columns

Symmetric Positive Definite matrices frequent in engineering

2. For a symmetric positive definite A, one thus has the following properties

a) The maximum elements of $\mathbf{A}$ are on the main diagonal

b) For a Symmetric, Positive Definite $\mathbf{A}$: No pivoting needed

c) The elimination is stable: $\left|a_{ii}^{(k+1)}\right| \le 2\left|a_{ii}^{(k)}\right|$. To show this, use $a_{kj}^2 \le a_{kk}\, a_{jj}$ in

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}\, a_{kj}^{(k)}$$

$$m_{ik} = \frac{a_{ki}^{(k)}}{a_{kk}^{(k)}}, \qquad i = k+1, k+2, \ldots, n \qquad j = i, i+1, \ldots, n$$

Special Matrices:
Symmetric, Positive Definite Matrix

The general GE

$$\begin{cases} a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}\, a_{kj}^{(k)} \\ m_{ik} = \dfrac{a_{ki}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, k+2, \ldots, n \quad j = i, i+1, \ldots, n \end{cases}$$

$$a_{ij} = \sum_{k=1}^{\min(i,j)} m_{ik} a_{kj}^{(k)}$$

becomes:

$$\overline{\overline{\mathbf{A}}} = \overline{\overline{\mathbf{L}}}\,\overline{\overline{\mathbf{U}}} = \overline{\overline{\mathbf{U}}}^{\dagger}\overline{\overline{\mathbf{U}}}$$

Choleski Factorization

$$\overline{\overline{\mathbf{U}}}^{\dagger} = [m_{ij}]$$

Complex Conjugate

where

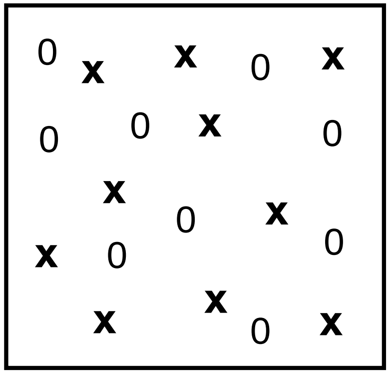
$$\left.\begin{aligned} m_{kk} &= \left(a_{kk} - \textstyle\sum_{\ell=1}^{k-1} m_{k\ell}\overline{m}_{k\ell}\right)^{1/2} \\ m_{ik} &= \frac{a_{ik} - \sum_{\ell=1}^{k-1} m_{i\ell}\overline{m}_{k\ell}}{m_{kk}},\ i = k+1, \ldots n \end{aligned}\right\} k = 1, \ldots n$$

No pivoting needed

$\dagger$ Complex Conjugate and Transpose

Linear Systems of Equations: **Iterative Methods**

Sparse (large) Full-bandwidth Systems (frequent in practice)



Iterative Methods are then efficient

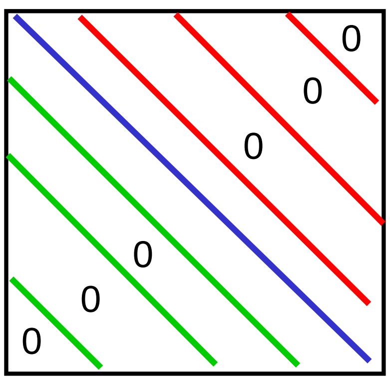
Analogous to iterative methods obtained for roots of equations, i.e. Open Methods: Fixed-point, Newton-Raphson, Secant

Example of Iteration equation

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{A}\,\mathbf{x} - \mathbf{b} = 0$$

$$\mathbf{x} = \mathbf{x} + \mathbf{A}\,\mathbf{x} - \mathbf{b} \quad \Rightarrow$$

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \mathbf{A}\,\mathbf{x}^{k} - \mathbf{b} = (\mathbf{A} + \mathbf{I})\,\mathbf{x}^{k} - \mathbf{b}$$



General Stationary Iteration Formula

$$\mathbf{x}^{k+1} = \mathbf{B}\,\mathbf{x}^{k} + \mathbf{c} \qquad k = 0, 1, 2, \ldots$$

Compatibility condition for $\mathbf{Ax}=\mathbf{b}$ to be the solution:

$$\left.\begin{array}{l} \text{Write} \quad \mathbf{c} = \mathbf{C}\,\mathbf{b} \\ \mathbf{A}^{-1}\mathbf{b} = \mathbf{B}\,\mathbf{A}^{-1}\,\mathbf{b} + \mathbf{C}\,\mathbf{b} \end{array}\right\} \Rightarrow \quad (\mathbf{I} - \mathbf{B})\,\mathbf{A}^{-1} = \mathbf{C} \quad \text{or} \quad \mathbf{B} = \mathbf{I} - \mathbf{C}\,\mathbf{A}$$

ps: $\mathbf{B}$ and $\mathbf{c}$ could be function of k (non-stationary)

Convergence

Convergence

$$\left\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}\right\| \to 0 \;\; \text{for} \;\; k \to \infty$$

Iteration – Matrix form

$$\bar{\mathbf{x}}^{(k+1)} = \bar{\bar{\mathbf{B}}}\bar{\mathbf{x}}^{(k)} + \bar{\mathbf{c}} \; , \;\; k = 0, \ldots$$

Convergence Analysis

$$\begin{aligned}
\bar{\mathbf{x}}^{(k+1)} &= \bar{\bar{\mathbf{B}}}\bar{\mathbf{x}}^{(k)} + \bar{\mathbf{c}} \\
\bar{\mathbf{x}} &= \bar{\bar{\mathbf{B}}}\bar{\mathbf{x}} + \bar{\mathbf{c}}
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad \bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}} &= \bar{\bar{\mathbf{B}}}\left(\bar{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}\right) \\
&= \bar{\bar{\mathbf{B}}} \cdot \bar{\bar{\mathbf{B}}}\left(\bar{\mathbf{x}}^{(k-1)} - \bar{\mathbf{x}}\right) \\
&\;\; \cdot \\
&= \bar{\bar{\mathbf{B}}}^{k+1}\left(\bar{\mathbf{x}}^{(0)} - \bar{\mathbf{x}}\right)
\end{aligned}$$

$$\left\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}\right\| \le \left\|\bar{\bar{\mathbf{B}}}^{k+1}\right\| \left\|\bar{\mathbf{x}}^{(0)} - \bar{\mathbf{x}}\right\| \le \left\|\bar{\bar{\mathbf{B}}}\right\|^{k+1} \left\|\bar{\mathbf{x}}^{(0)} - \bar{\mathbf{x}}\right\|$$

Sufficient Condition for Convergence: $\left\|\bar{\bar{\mathbf{B}}}\right\| < 1$

||B||<1 for a chosen matrix norm
Infinite norm often used in practice

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}|$$

"Maximum Column Sum"

$$\|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}|$$

"Maximum Row Sum"

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}$$

"The Frobenius norm" (also called Euclidean norm)", which for matrices differs from:

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^* A)}$$

"The l-2 norm" (also called spectral norm)

Convergence: Necessary and Sufficient Condition

Convergence

$$\left\|\overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}}\right\| \to 0 \ \text{ for } \ k \to \infty$$

Iteration – Matrix form

$$\overline{\mathbf{x}}^{(k+1)} = \overline{\overline{\mathbf{B}}}\overline{\mathbf{x}}^{(k)} + \overline{\mathbf{c}} \ , \ k = 0, \ldots$$

Convergence Analysis

$$\begin{aligned} \overline{\mathbf{x}}^{(k+1)} &= \overline{\overline{\mathbf{B}}}\overline{\mathbf{x}}^{(k)} + \overline{\mathbf{c}} \\ \overline{\mathbf{x}} &= \overline{\overline{\mathbf{B}}}\overline{\mathbf{x}} + \overline{\mathbf{c}} \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad \overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}} &= \overline{\overline{\mathbf{B}}}\left(\overline{\mathbf{x}}^{(k)} - \overline{\mathbf{x}}\right) \\ &= \overline{\overline{\mathbf{B}}} \cdot \overline{\overline{\mathbf{B}}}\left(\overline{\mathbf{x}}^{(k-1)} - \overline{\mathbf{x}}\right) \\ &\cdot \\ &= \overline{\overline{\mathbf{B}}}^{k+1}\left(\overline{\mathbf{x}}^{(0)} - \overline{\mathbf{x}}\right) \end{aligned}$$

$$\left\|\overline{\mathbf{x}}^{(k+1)} - \overline{\mathbf{x}}\right\| \le \left\|\overline{\overline{\mathbf{B}}}^{k+1}\right\| \left\|\overline{\mathbf{x}}^{(0)} - \overline{\mathbf{x}}\right\| \le \left\|\overline{\overline{\mathbf{B}}}\right\|^{k+1} \left\|\overline{\mathbf{x}}^{(0)} - \overline{\mathbf{x}}\right\|$$

Necessary and Sufficient Condition for Convergence:

Spectral radius of **B** is smaller than one: $\rho(\mathbf{B}) = \max_{i=1\ldots n} |\lambda_i| < 1, \quad \text{where } \lambda_i = \text{eigenvalue}(\mathbf{B}_{n\times n})$

(proof: use eigendecomposition of **B**)

(This ensures ||**B**||<1)