# 2.29 Numerical Fluid Mechanics
# Fall 2011 – Lecture 10

## REVIEW Lecture 9:

- Direct Methods for solving linear algebraic equations
  - Gauss Elimination
  - LU decomposition/factorization
  - Error Analysis for Linear Systems and Condition Numbers
  - Special Matrices: LU Decompositions
    - Tri-diagonal systems: Thomas Algorithm (Nb Ops: $8\,O(n)$)
    - General Banded Matrices
      - Algorithm, Pivoting and Modes of storage
      - Sparse and Banded Matrices

      $p$ super-diagonals
      $q$ sub-diagonals
      $w = p + q + 1$ bandwidth
    - Symmetric, positive-definite Matrices
      - Definitions and Properties, Choleski Decomposition

# 2.29 Numerical Fluid Mechanics
# Fall 2011 – Lecture 10

## REVIEW Lecture 9, Cont'd:

- Direct Methods
  - Gauss Elimination
  - LU decomposition/factorization
  - Error Analysis for Linear Systems and Condition Numbers
  - Special Matrices (Tri-diagonal, banded, sparse, positive-definite, etc)

- Iterative Methods:

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

  - Convergence
    - Nec. & Suf.: $\rho(\mathbf{B}) = \max_{i=1\ldots n} |\lambda_i| < 1, \quad \text{where } \lambda_i = \text{eigenvalue}(\mathbf{B}_{n\times n})$ (ensures ||**B**||<1)
  - Jacobi's method, Gauss-Seidel iteration

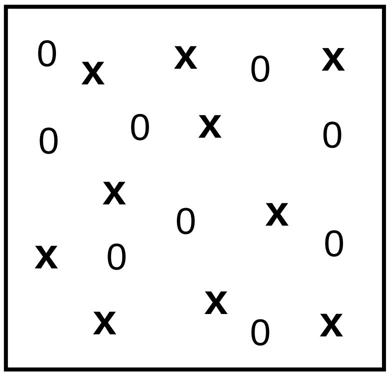
# TODAY (Lecture 10): Systems of Linear Equations IV

- **Direct Methods**
  - Gauss Elimination
  - LU decomposition/factorization
  - Error Analysis for Linear Systems
  - Special Matrices: LU Decompositions
- **Iterative Methods**
  - Concepts, Definitions and Convergence
  - Jacobi's method
  - Gauss-Seidel iteration
  - Stop Criteria
  - Example
  - Successive Over-Relaxation Methods
  - Gradient Methods and Krylov Subspace Methods
  - Preconditioning of $\mathbf{Ax=b}$

# Reading Assignment

- Chapter 11 of "Chapra and Canale, Numerical Methods for Engineers, 2006/2009."
  - Any chapter on "Solving linear systems of equations" in CFD references provided. For example: chapter 5 of "J. H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. Springer, NY, 3rd edition, 2002"

# Linear Systems of Equations: **Iterative Methods**
# **Element-by-Element Form of the Equations**

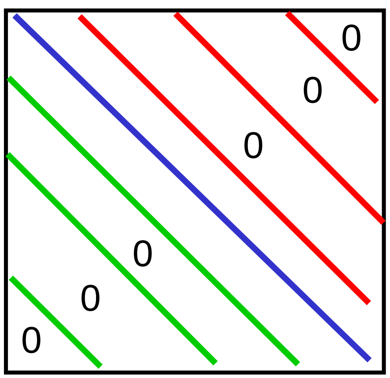



Sparse (large) Full-bandwidth Systems (frequent in practice)

Iterative Methods are then efficient

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

Rewrite Equations



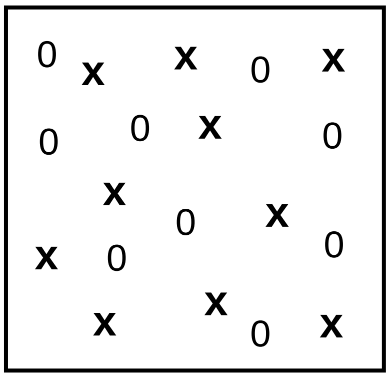

$$\overline{\overline{\mathbf{A}}}\overline{\mathbf{x}} = \overline{\mathbf{b}} \quad \Leftrightarrow \quad \sum_{j=1}^{n} a_{ij} x_j = b_i$$

$$a_{ii} \neq 0 \Rightarrow x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}} \ , \ i = 1, \ldots n$$

# Iterative Methods: Jacobi and Gauss Seidel

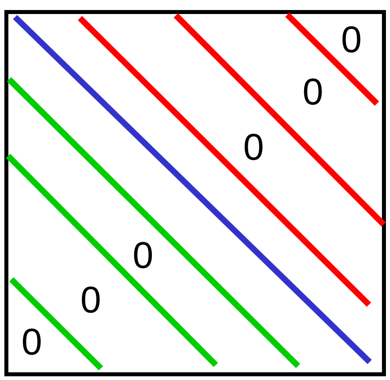
**Rewrite Equations:** $\overline{\overline{\mathbf{A}}}\overline{\mathbf{x}} = \overline{\mathbf{b}} \Leftrightarrow \sum_{j=1}^{n} a_{ij} x_j = b_i$

Sparse, Full-bandwidth Systems



$$a_{ii} \neq 0 \Rightarrow x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}, \quad i = 1, \ldots n$$

**=> Iterative, Recursive Methods:**



## Jacobi's Method

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, \quad i = 1, \ldots n$$

Computes a full new **x** based on full old **x**, i.e.
Each new $x_i$ is computed based on all old $x_i$'s

## Gauss-Seidel's Method

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, \quad i = 1, \ldots n$$

New **x** based most recent **x** elements, i.e.
Each new $x_{i-1}^{k+1}$ directly used to compute next element $x_i^{k+1}$

# Iterative Methods: Jacobi's Matrix form

Iteration – Matrix form

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

Decompose Coefficient Matrix

$$\overline{\overline{A}} = \overline{\overline{D}} + \overline{\overline{L}} + \overline{\overline{U}}$$

with

$$\overline{\overline{\mathbf{D}}} = \operatorname{diag}\, a_{ii}$$

$$\overline{\overline{\mathbf{L}}} = \begin{cases} a_{ij} & , \ i > j \\ 0, & i \le j \end{cases}$$

$$\overline{\overline{\mathbf{U}}} = \begin{cases} a_{ij} & , \ i < j \\ 0, & i \ge j \end{cases}$$

**Note: this is NOT LU-factorization**

Jacobi's Method

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}\ ,\ i = 1, \ldots n$$

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

Iteration Matrix form

$$\begin{cases} \overline{\overline{\mathbf{B}}} = -\overline{\overline{\mathbf{D}}}^{-1}\left(\overline{\overline{\mathbf{L}}} + \overline{\overline{\mathbf{U}}}\right) \\ \overline{\mathbf{c}} = \overline{\overline{\mathbf{D}}}^{-1}\overline{\mathbf{b}} \end{cases}$$

# Convergence of Jacobi and Gauss-Seidel

- Jacobi: $$\mathbf{A}\,\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{D}\,\mathbf{x} + (\mathbf{L} + \mathbf{U})\,\mathbf{x} = \mathbf{b}$$

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

- Gauss-Seidel: $$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, \; i = 1, \ldots n$$

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad (\mathbf{D} + \mathbf{L})\,\mathbf{x} + \mathbf{U}\,\mathbf{x} = \mathbf{b}$$

$$\mathbf{x}^{k+1} = -\mathbf{D}^{-1}\mathbf{L}\,\mathbf{x}^{k+1} - \mathbf{D}^{-1}\mathbf{U}\,\mathbf{x}^{k} + \mathbf{D}^{-1}\mathbf{b} \qquad \text{or}$$

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

- Both converge if **A** strictly diagonal dominant
- Gauss-Seidel also convergent if **A** symmetric positive definite matrix
- Also Jacobi convergent for **A** if
  – **A** symmetric and {**D**, **D** + **L** + **U**, **D** - **L** - **U**} are all positive definite

# Sufficient Condition for Convergence
# Proof for Jacobi

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{D}\,\mathbf{x} + (\mathbf{L} + \mathbf{U})\,\mathbf{x} = \mathbf{b}$$

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

Sufficient Convergence Condition $\left\|\overline{\overline{\mathbf{B}}}\right\| < 1$

Jacobi's Method $b_{ij} = -\frac{a_{ij}}{a_{ii}}\ ,\ i \neq j$

Using the ∞-Norm
(Maximum Row Sum)

$$\left\|\overline{\overline{\mathbf{B}}}\right\|_\infty = \max_i \sum_{j=1, j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|}$$

Hence, Sufficient Convergence Condition is:

$$\sum_{j=1, j\neq i}^{n} |a_{ij}| < |a_{ii}|$$

Strict Diagonal Dominance

# Illustration of Convergence (left) and Divergence (right) of the Gauss-Seidel Method

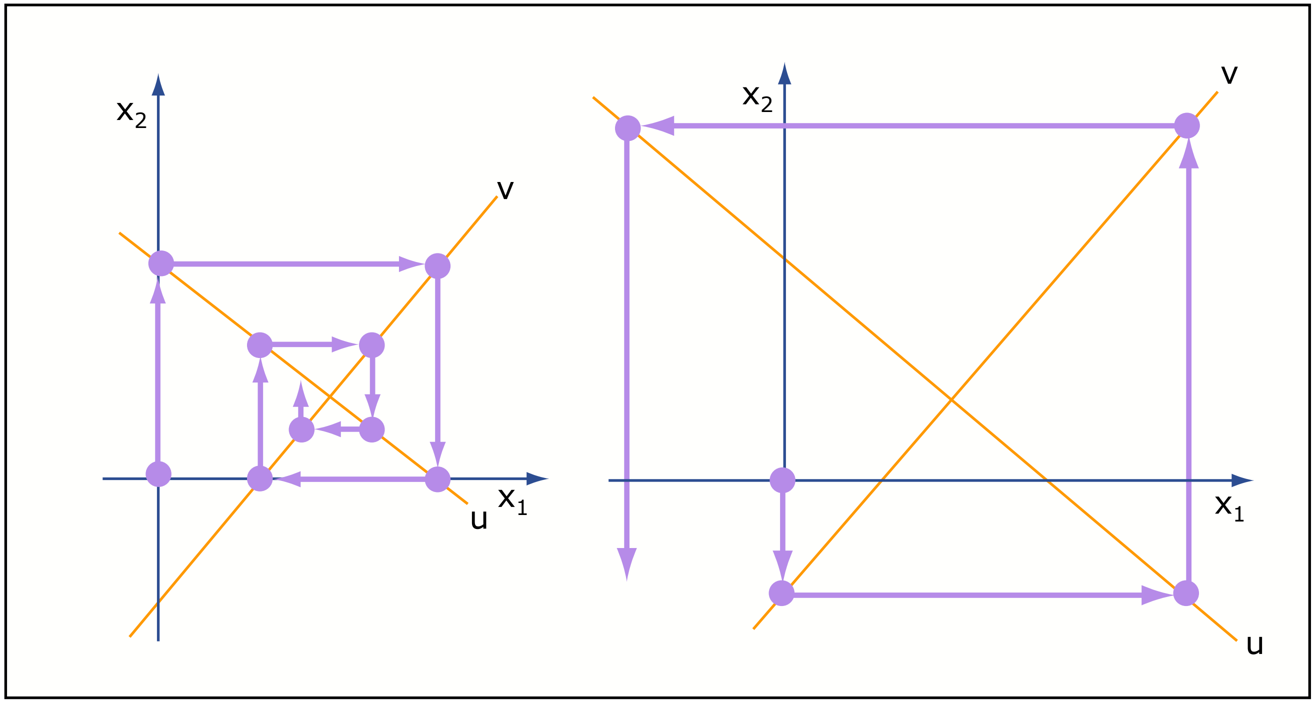



Image by MIT OpenCourseWare.

Since the same functions are plotted in each case, the order of implementation of the equations determines whether the method converges or diverges.

# Special Matrices: Tri-diagonal Systems

Finite Difference

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

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

Strict Diagonal Dominance?

$$kh > 2 \Rightarrow h > \frac{2}{k}$$

Considering Jacobi, a sufficient condition for convergence is:

With $\mathbf{B}=-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$: If $|a_{ii}| > \sum_{j=1,\, j\neq i}^{n} |a_{ij}| \quad \Rightarrow \quad \|\mathbf{B}\|_\infty = \max_{i=1\ldots n} \left(\sum_{j=1}^{n} |b_{ij}|\right) = \max_{i=1\ldots n} \left( \sum_{j=1,\, j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|}\right) < 1$

# vib_string.m (Part I)

```
n=99;
L=1.0;
h=L/(n+1);
k=2*pi;
kh=k*h

%Tri-Diagonal Linear System: Ax=b
x=[h:h:L-h]';
a=zeros(n,n);
f=zeros(n,1);
% Offdiagonal values
o=1.0

a(1,1) =kh^2 - 2;
a(1,2)=o;

for i=2:n-1
    a(i,i)=a(1,1);
    a(i,i-1) = o;
    a(i,i+1) = o;
end
a(n,n)=a(1,1);
a(n,n-1)=o;

% Hanning window load
nf=round((n+1)/3);
nw=round((n+1)/6);
nw=min(min(nw,nf-1),n-nf);
figure(1)
hold off

nw1=nf-nw;
nw2=nf+nw;
f(nw1:nw2) = h^2*hanning(nw2-nw1+1);
subplot(2,1,1); p=plot(x,f,'r');set(p,'LineWidth',2);
p=title('Force Distribution');
set(p,'FontSize',14)
```

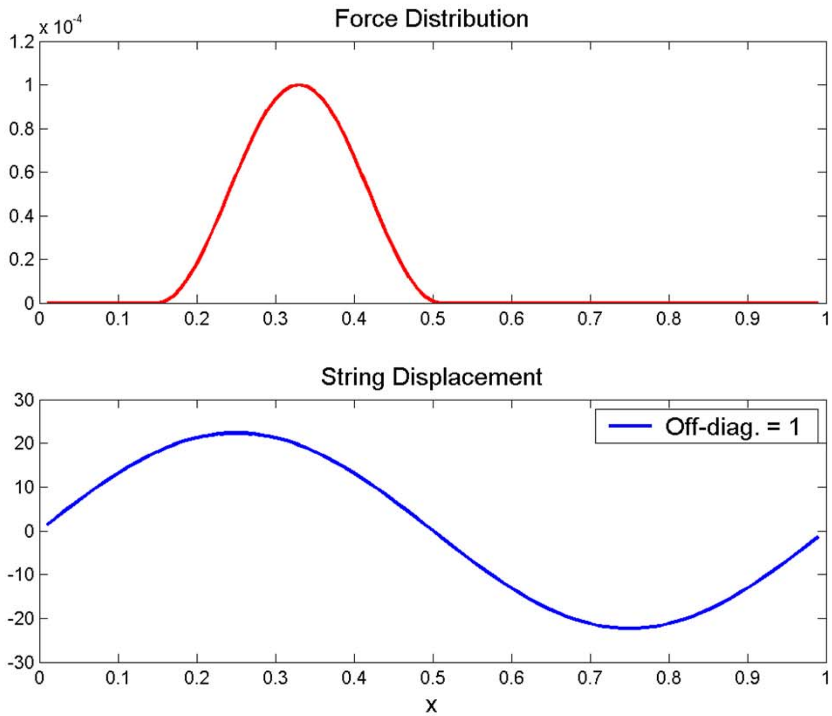
```
% Exact solution
y=inv(a)*f;
subplot(2,1,2); p=plot(x,y,'b');set(p,'LineWidth',2);
p=legend(['Off-diag. = ' num2str(o)]);
set(p,'FontSize',14);
p=title('String Displacement (Exact)');
set(p,'FontSize',14);
p=xlabel('x');
set(p,'FontSize',14);
```

# vib_string.m (Part II)

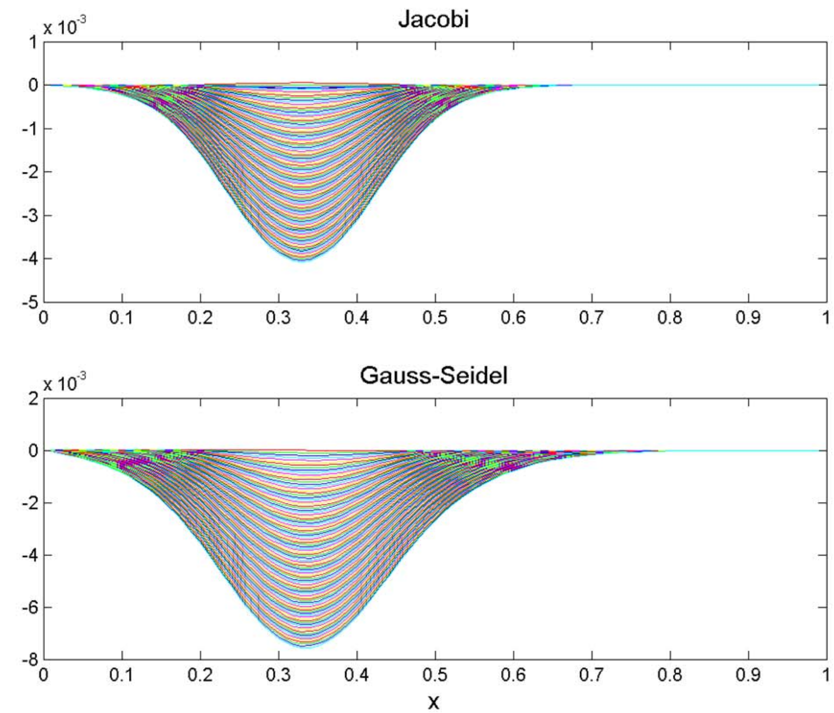
```
%Iterative solution using Jacobi's and Gauss-Seidel's methods
b=-a;
c=zeros(n,1);
for i=1:n
    b(i,i)=0;
    for j=1:n
        b(i,j)=b(i,j)/a(i,i);
        c(i)=f(i)/a(i,i);
    end
end

nj=100;
xj=f;
xgs=f;

figure(2)
nc=6
col=['r' 'g' 'b' 'c' 'm' 'y']
hold off
for j=1:nj
    % jacobi
    xj=b*xj+c;
    % gauss-seidel
    xgs(1)=b(1,2:n)*xgs(2:n) + c(1);
    for i=2:n-1
        xgs(i)=b(i,1:i-1)*xgs(1:i-1) + b(i,i+1:n)*xgs(i+1:n) +c(i);
    end
    xgs(n)= b(n,1:n-1)*xgs(1:n-1) +c(n);
    cc=col(mod(j-1,nc)+1);
    subplot(2,1,1); plot(x,xj,cc); hold on;
    p=title('Jacobi');
    set(p,'FontSize',14);
    subplot(2,1,2); plot(x,xgs,cc); hold on;
    p=title('Gauss-Seidel');
    set(p,'FontSize',14);
    p=xlabel('x');
    set(p,'FontSize',14);
end
```

# vib_string.m
# o=1.0, , k=2*pi, h=.01, kh<2

## Exact Solution



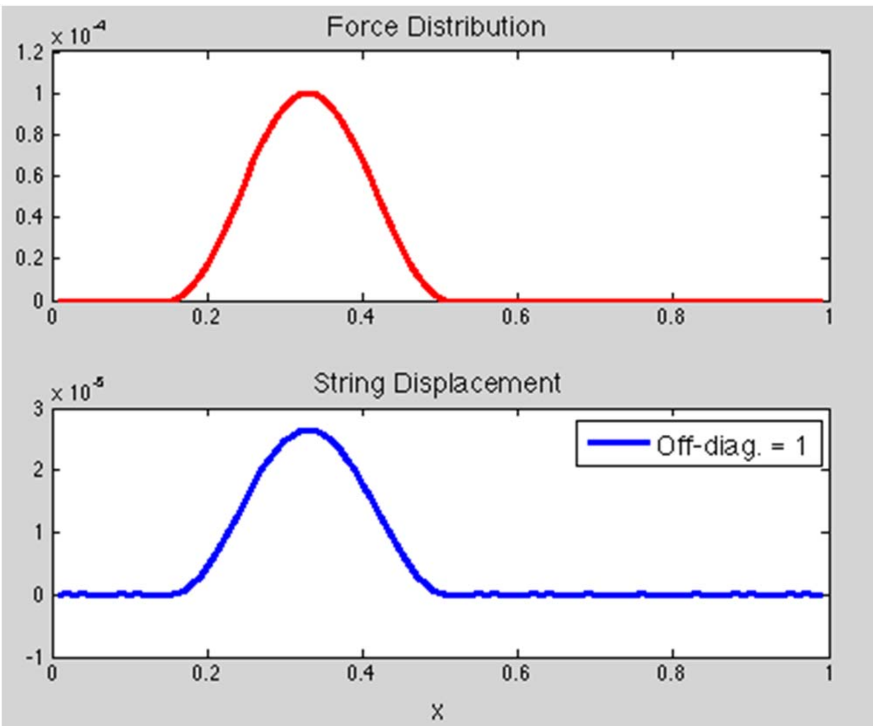

## Iterative Solutions



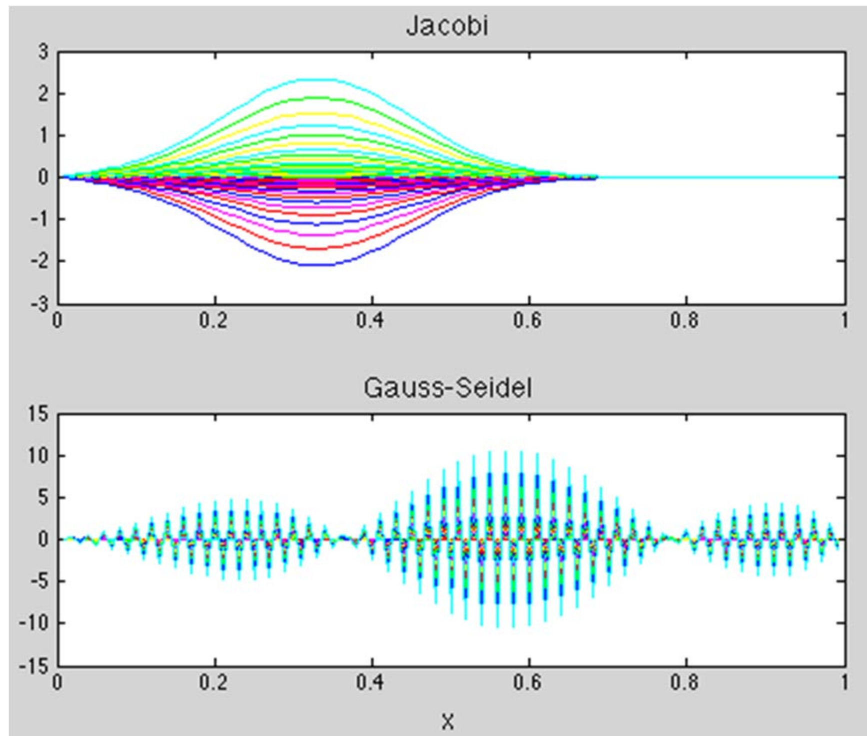

Coefficient Matrix Not Strictly Diagonally Dominant

# vib_string.m
# o=1.0, , k=2*pi*31, h=.01, kh<2

Exact Solution



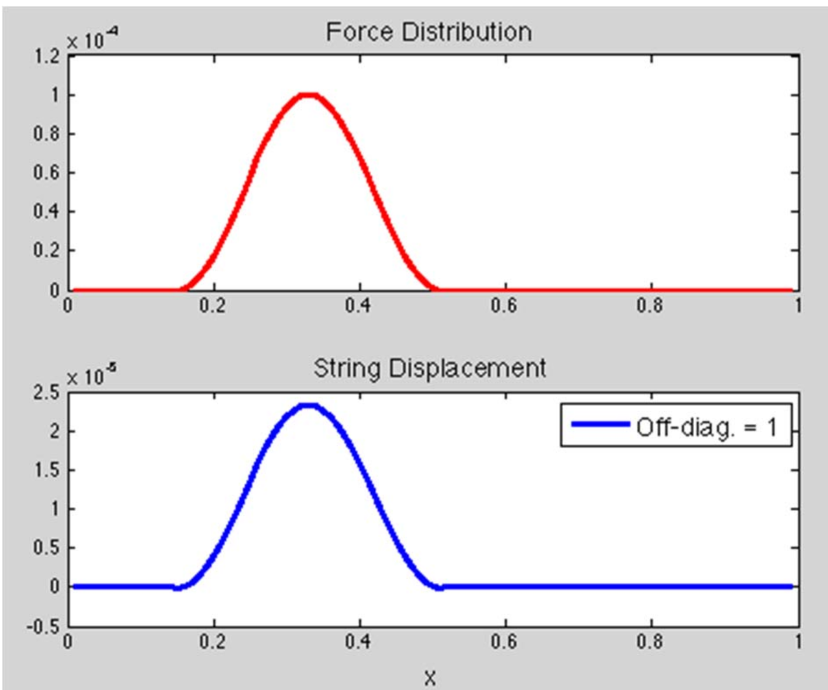

Iterative Solutions



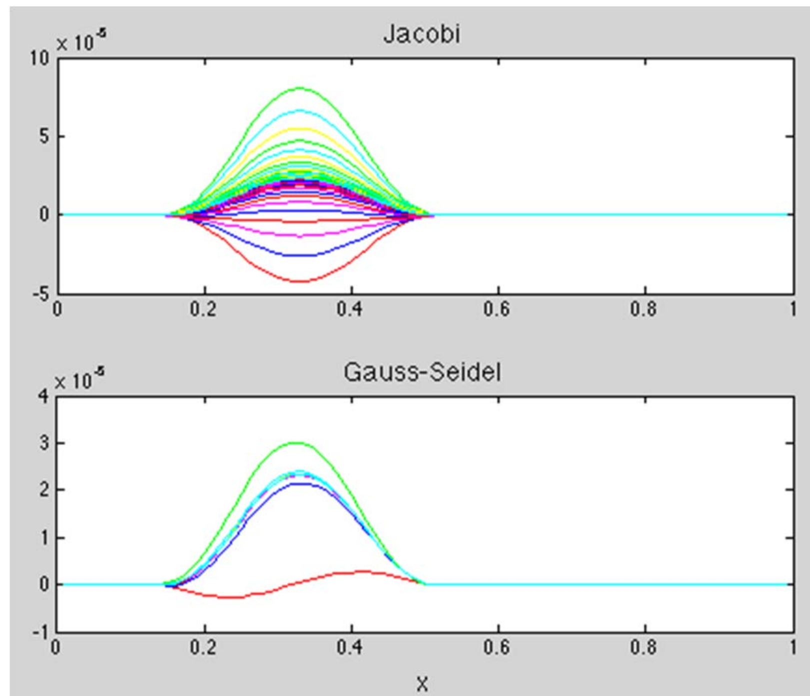

Coefficient Matrix Not Strictly Diagonally Dominant

# vib_string.m
# o=1.0, k=2*pi*33, h=.01, kh>2

Exact Solution



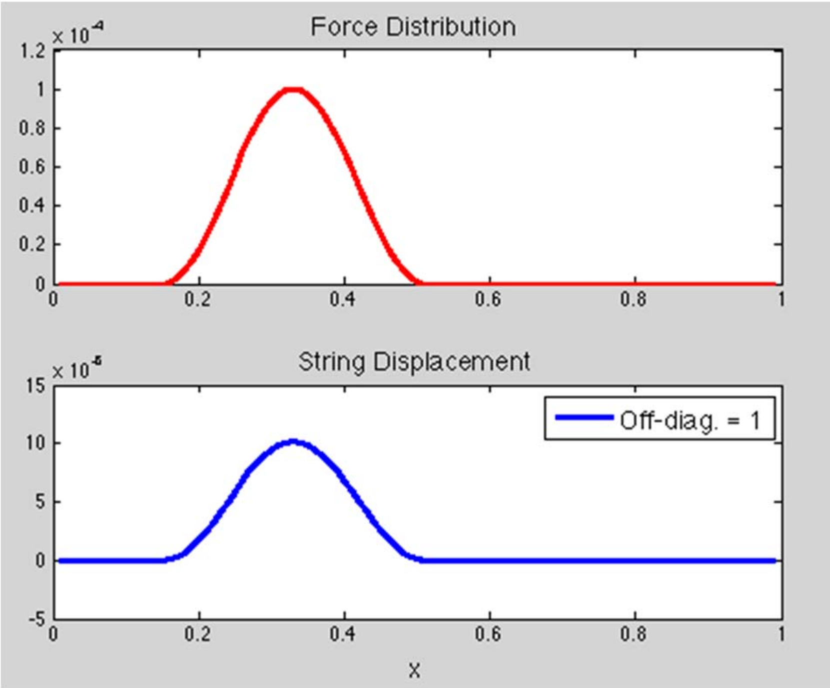

Iterative Solutions



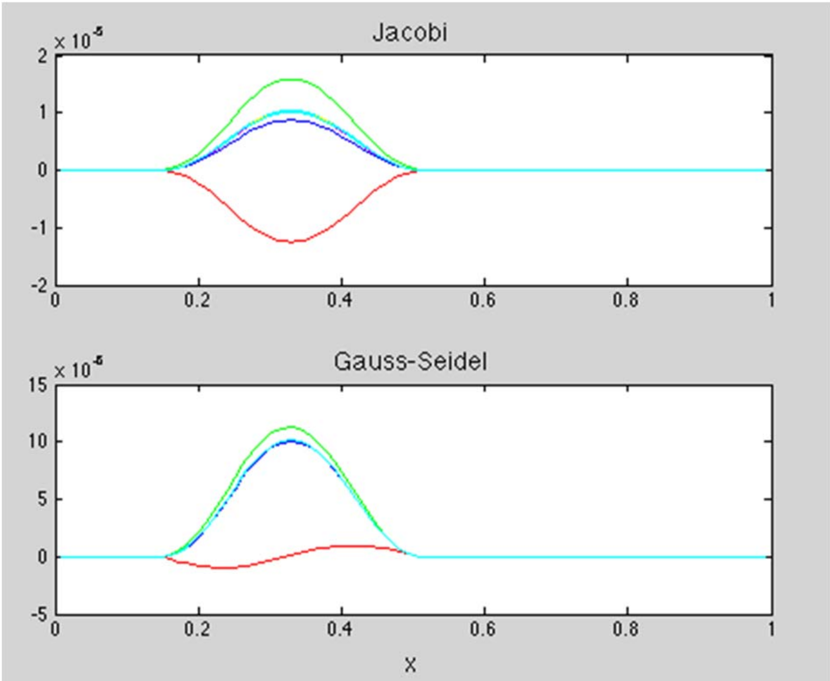

Coefficient Matrix Strictly Diagonally Dominant

# vib_string.m
# o=1.0, k=2*pi*50, h=.01, kh>2

Exact Solution



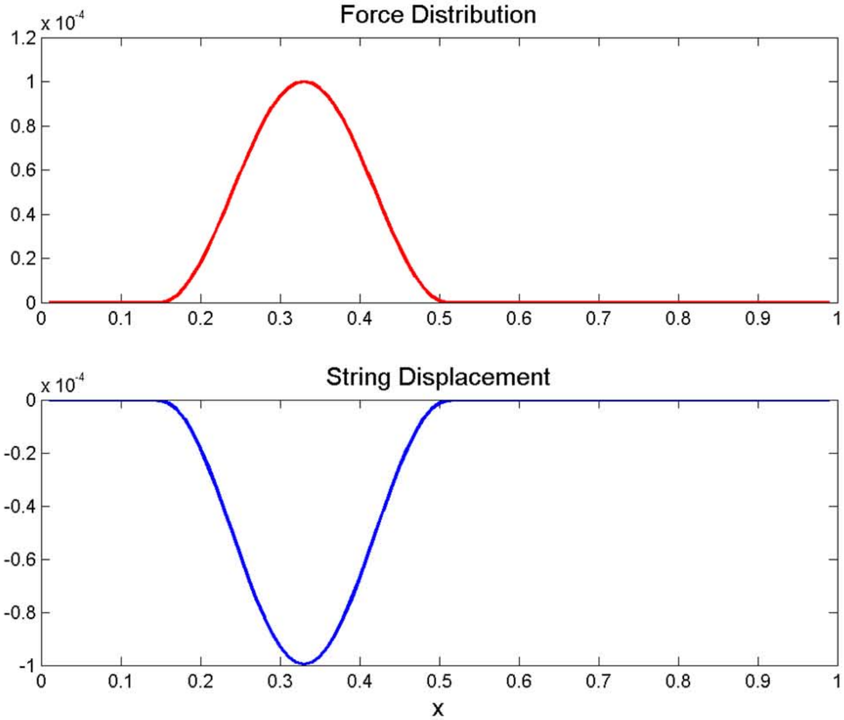

Iterative Solutions



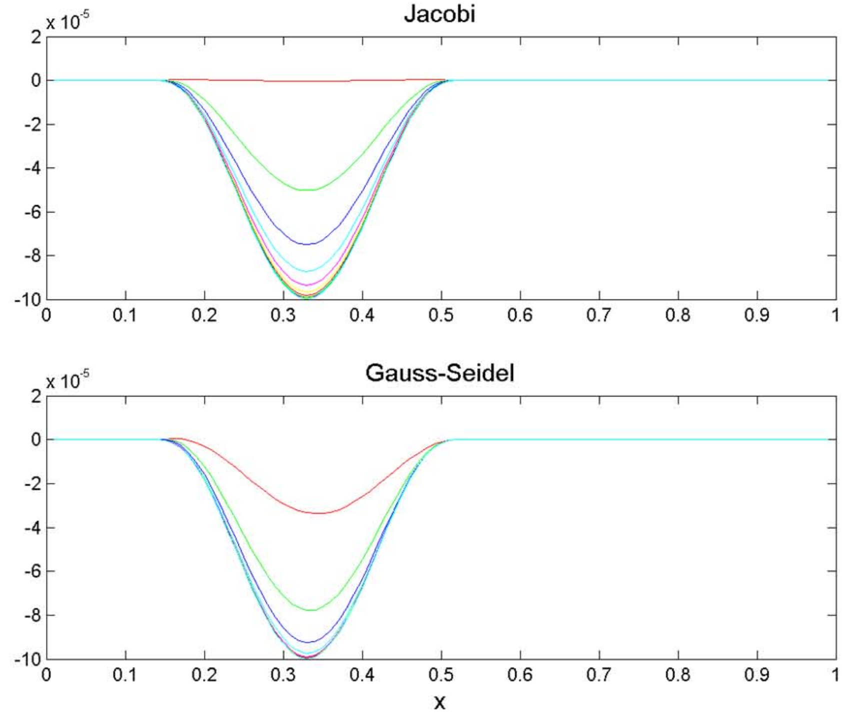

Coefficient Matrix Strictly Diagonally Dominant

# vib_string.m
# o = 0.5, k=2*pi, h=.01

Exact Solution




Iterative Solutions




Coefficient Matrix Strictly Diagonally Dominant

# Successive Over-relaxation (SOR) Method

- Aims to reduce the spectral radius of B to increase rate of convergence
- Add an extrapolation to each step of Gauss-Seidel

$$\mathbf{x}_i^{k+1} = \omega \overline{\mathbf{x}}_i^{k+1} + (1-\omega)\mathbf{x}_i^k \text{, } where \quad \overline{\mathbf{x}}_i^{k+1} \;\; computed\; by\; Gauss - Seidel$$

$$\omega = 1 \Rightarrow \; SOR \equiv \text{Gauss-Seidel}$$
$$1 < \omega < 2 \Rightarrow \; \text{Over-relaxation (weight new values more)}$$
$$0 < \omega < 1 \Rightarrow \; \text{Under-relaxation}$$

- If "A" symmetric and positive definite $\Rightarrow \; converges\; for\; 0 < \omega < 2$

- Matrix format:

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

- Hard to find optimal value of over-relaxation parameter for fast convergence (aim to minimize spectral radius of $\mathbf{B}$) due to BCs, etc.

$$\omega = \omega_{opt} = ?$$

2.29 Numerical Fluid Mechanics
Fall 2011