Course Instructor
Dr. Raymond C. Rumpf
Office: A-337
Phone: (915) 747-6958
E-Mail: rcrumpf@utep.edu

Computational Methods in EE

# Topic 3 – Review of Linear Algebra

*EE 4386/5301 Computational Methods in EE*

## Outline

Computational Methods in EE

- Solving systems of equations
- Matrices and matrix operations
- Common linear algebra problems
- Naïve Gauss elimination
- Gauss-Jordan method (direct)
- LU decomposition
- Jacobi method (iterative)
- Gauss-Seidel method
- Calculating matrix inverse

2

Computational Methods in EE

# Solving Systems of Equations

3

## Systems of Linear Equations

Computational Methods in EE

Very often in science and engineering, problems can be reduced to a system of linear equations.

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}$$

$a_{ij} \equiv$ constant coefficient (usually known)

$x_i \equiv$ unknown values

$b_i \equiv$ constants (usually excitation)

4

# Direct Analytical Solution

Computational Methods in EE

Suppose we wish to solve the following system of equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{aligned}
$$

Step 1 – Solve first equation for $x_1$.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \quad \rightarrow \quad x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2 - \frac{a_{13}}{a_{11}}x_3$$

$$
\begin{aligned}
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{aligned}
$$

5

# Direct Analytical Solution

Computational Methods in EE

Step 2 – *Forward Substitution* – Substitute this new equation into 2nd and 3rd equations to eliminate $x_1$.

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\
a'_{22}x_2 + a'_{23}x_3 &= b'_2 \\
a'_{32}x_2 + a'_{33}x_3 &= b'_3
\end{aligned}
$$

$$a'_{22} = a_{22} - \frac{a_{21}a_{12}}{a_{11}} \qquad a'_{23} = a_{23} - \frac{a_{21}a_{13}}{a_{11}} \qquad b'_2 = b_2 - \frac{a_{21}b_1}{a_{11}}$$

$$a'_{32} = a_{32} - \frac{a_{31}a_{12}}{a_{11}} \qquad a'_{33} = a_{33} - \frac{a_{31}a_{13}}{a_{11}} \qquad b'_3 = b_3 - \frac{a_{31}b_1}{a_{11}}$$

6

# Direct Analytical Solution

Computational Methods in EE

Step 3 – Solve second equation for $x_2$.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 &= b'_2 \quad \rightarrow \quad x_2 = \frac{b'_2}{a'_{22}} - \frac{a'_{23}}{a'_{22}}x_3 \\ a'_{32}x_2 + a'_{33}x_3 &= b'_3 \end{aligned}$$

7

# Direct Analytical Solution

Computational Methods in EE

Step 4 – *Forward Substitution* – Substitute this new equation into 3rd equation to eliminate $x_2$.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 &= b'_2 \\ a''_{33}x_3 &= b''_3 \end{aligned}$$

$$a''_{33} = a'_{33} - \frac{a'_{32}a'_{23}}{a'_{22}} \qquad b''_3 = b'_3 - \frac{a'_{32}b'_2}{a'_{22}}$$

8

# Direct Analytical Solution

Computational Methods in EE

Step 5 – Solve third equation for $x_3$. Since this is the last equation, we get the final answer for $x_3$.

$$x_3 = \frac{b_3''}{a_{33}''}$$

9

# Direct Analytical Solution

Computational Methods in EE

Step 6 – *Backward Substitution* – Given $x_3$, calculate $x_2$ using equation from Step 3.

$$x_2 = \frac{b_2' - a_{23}' x_3}{a_{22}'}$$

10

# Direct Analytical Solution

Computational Methods in EE

Step 7 – *Backward Substitution* – Given $x_2$ and $x_3$, calculate $x_1$ using equation from Step 1.

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

11

Computational Methods in EE

# Matrices & Matrix Operations

12

# Systems of Linear Equations

Computational Methods in EE

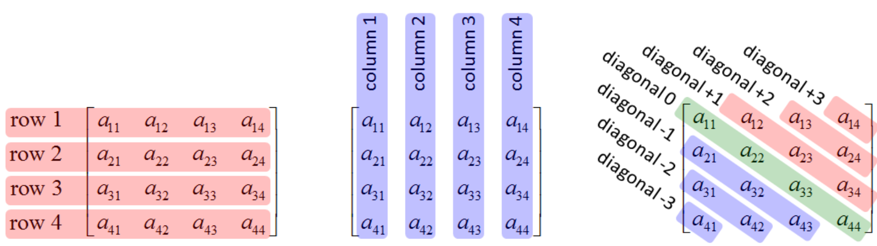
Systems of equations can be written in matrix form.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \quad \rightarrow \quad \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix}}_{\substack{[A] \\ \text{or} \\ \mathbf{A}}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\substack{[x] \\ \text{or} \\ \mathbf{x}}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\substack{[b] \\ \text{or} \\ \mathbf{b}}}$$



# Rows, Columns, and Diagonals

Computational Methods in EE




The center diagonal is usually just called *the diagonal*.

The elements along the diagonal are sometimes called the *pivot elements*.

# Special Matrices (1 of 2)

Computational Methods in EE

Symmetric Matrix

$$[A]=\begin{bmatrix}1 & 2 & 9 & 4\\ 2 & 6 & 5 & 8\\ 9 & 5 & 7 & 0\\ 4 & 8 & 0 & 3\end{bmatrix}$$

Diagonal Matrix

$$[A]=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 6 & 0 & 0\\ 0 & 0 & 7 & 0\\ 0 & 0 & 0 & 3\end{bmatrix}$$

Identity Matrix

$$[I]=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Zero Matrix

$$[0]=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$$

15

# Special Matrices (2 of 2)

Computational Methods in EE

Upper Triangular Matrix

$$[A]=\begin{bmatrix}1 & 2 & 9 & 4\\ 0 & 6 & 5 & 8\\ 0 & 0 & 7 & 5\\ 0 & 0 & 0 & 3\end{bmatrix}$$

Lower Triangular Matrix

$$[A]=\begin{bmatrix}1 & 0 & 0 & 0\\ 2 & 6 & 0 & 0\\ 9 & 5 & 7 & 0\\ 4 & 8 & 1 & 3\end{bmatrix}$$

Banded Matrix

$$[A]=\begin{bmatrix}1 & 2 & 0 & 0\\ 4 & 6 & 5 & 0\\ 0 & 8 & 7 & 5\\ 0 & 0 & 10 & 3\end{bmatrix}$$

Bandwidth of 3

16

# Block Matrices

Computational Methods in EE

Block matrices are "matrices of matrices."

$$[F]=\begin{bmatrix}[A] & [B]\\ [C] & [D]\end{bmatrix}=\begin{bmatrix}a_{11} & a_{12} & b_{11} & b_{12}\\ a_{21} & a_{22} & b_{21} & b_{22}\\ c_{11} & c_{12} & d_{11} & d_{12}\\ c_{21} & c_{22} & d_{21} & d_{22}\end{bmatrix}$$

$$[A]=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\quad [B]=\begin{bmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{bmatrix}\quad [C]=\begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix}\quad [D]=\begin{bmatrix}d_{11} & d_{12}\\ d_{21} & d_{22}\end{bmatrix}$$

17

# Sparse Matrices

Computational Methods in EE

Many matrices contain 99.9% zeros.

$$H=\begin{bmatrix}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
1&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0\\
0&1&0&0&1&0&0&0&0&0&0&1&0&0&1&0\\
0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&1\\
0&0&0&1&0&0&1&0&0&1&0&0&1&0&0&0
\end{bmatrix}$$

It is not efficient use of memory to store all these zeros. Instead, we store only the non-zero elements along with their positions in the matrix.

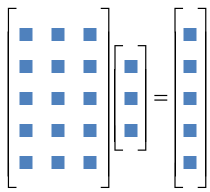
The opposite of a "sparse" matrix is a "full" matrix.

18

# Matrix Problem Size

Computational Methods in EE

**# Equations > # Unknowns**

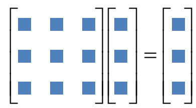


Usually occurs when the equations are derived from samples.

Solution is obtained as a *best fit* and is not exact.

Applications
- Curve fitting

**# Equations = # Unknowns**

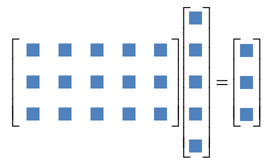


Most usual case.

Many standard algorithms exist to obtain an *exact* solution.

Applications
- Circuit theory
- Solving ODEs

**# Equations < # Unknowns**



Usually occurs when little is know about the problem or solution.

Solution is obtained by *optimization* and is not exact.

Applications
- Topology optimization

19

# Health of a Matrix (1 of 3)

Computational Methods in EE

Is this system of equations solvable?

$$\begin{aligned} x+2y+z&=8 \\ x+2y+z&=8 \\ 3x-y+z&=4 \end{aligned} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 3 & -1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\begin{bmatrix} 8 \\ 8 \\ 4 \end{bmatrix}$$

No!
The 1st and 2nd equations are the same.
The 2nd equation does not provide any new information to the problem.

$$\begin{aligned} x+2y+z&=8 \\ 2x+4y+2z&=16 \\ 3x-y+z&=4 \end{aligned} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 3 & -1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\begin{bmatrix} 8 \\ 16 \\ 4 \end{bmatrix}$$

No!
The 2nd equation is just 2× the 1st equation.
The 2nd equation is still not providing any new information.

$$\begin{aligned} x+2y+z&=8 \\ 4x+y+2z&=12 \\ 3x-y+z&=4 \end{aligned} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 4 & 1 & 2 \\ 3 & -1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\begin{bmatrix} 8 \\ 12 \\ 4 \end{bmatrix}$$

No!
The 2nd equation is the sum of the 1st and 3rd equation, thus the 2nd equation still does not provide any new information.

20

# Health of a Matrix (2 of 3)

Computational Methods in EE

Is this system of equations solvable?

$$x+z=8 \qquad x+2z=7 \qquad 3x+z=4$$

$$\begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 2\\ 3 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}=\begin{bmatrix}8\\ 7\\ 4\end{bmatrix}$$

No!
None of these equations contain any information about $y$.

So how do we know if a problem is solvable?

- All rows must be linearly independent – this ensures they provide new information to the problem.
- No rows can be all zeros – This would not provide any information.
- No columns can be all zeros – This would be ignoring information from one of the unknowns.

$$[A][x]=[b] \text{ is solvable if } \det[A]\neq 0$$

21

# Health of a Matrix (3 of 3)

Computational Methods in EE

Is the following system of equations solvable?

$$x+2y+z=8 \qquad 1.0001x+2y+z=8.0001 \qquad 3x-y+z=4$$

$$\begin{bmatrix}1 & 2 & 1\\ 1.0001 & 2 & 1\\ 3 & -1 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}=\begin{bmatrix}8\\ 8.0001\\ 4\end{bmatrix}$$

Technically yes, but we would expect the solution to be somewhat "touchy" and unstable. This is a poorly conditioned matrix.

Condition Number of a Matrix

The condition number $\kappa(\mathbf{A})$ of matrix $\mathbf{A}$ is a measure of how numerically "stable" it is.

Matrices with high condition numbers are less stable. Small changes in the element values of $\mathbf{A}$ will result in large changes in the elements of $\mathbf{b}$.

$$\kappa(\mathbf{A})=\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}$$

22

# Matrix Math (1 of 4)

Computational Methods in EE

Addition:

$$[A]+[B]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}+\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}=\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \end{bmatrix}$$

Subtraction:

$$[A]-[B]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}-\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}=\begin{bmatrix} a_{11}-b_{11} & a_{12}-b_{12} & a_{13}-b_{13} \\ a_{21}-b_{21} & a_{22}-b_{22} & a_{23}-b_{23} \\ a_{31}-b_{31} & a_{32}-b_{32} & a_{33}-b_{33} \end{bmatrix}$$

23

# Matrix Math (2 of 4)

Computational Methods in EE

Multiplication by a Scalar:

$$s[A]=s\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}=\begin{bmatrix} sa_{11} & sa_{12} & sa_{13} \\ sa_{21} & sa_{22} & sa_{23} \\ sa_{31} & sa_{32} & sa_{33} \end{bmatrix}$$

Multiplication by a Matrix

$$[A][B]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}=\begin{bmatrix} a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31} & \# & \# \\ \# & \# & \# \\ \# & \# & \# \end{bmatrix}$$

$$[A][x]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} a_{11}x_1+a_{12}x_2+a_{13}x_3 \\ a_{21}x_1+a_{22}x_2+a_{23}x_3 \\ a_{31}x_1+a_{32}x_2+a_{33}x_3 \end{bmatrix}$$

24

# Matrix Math (3 of 4)

Computational Methods in EE

Matrix Transpose:

$$[A]^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \qquad a'_{ij} = a_{ji}$$

Hermitian Transpose:

$$[A]^H = \left( \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}^T \right)^* = \begin{bmatrix} a_{11}^* & a_{21}^* & a_{31}^* \\ a_{12}^* & a_{22}^* & a_{32}^* \\ a_{13}^* & a_{23}^* & a_{33}^* \end{bmatrix} \qquad a'_{ij} = a_{ji}^*$$

25

# Matrix Math (4 of 4)

Computational Methods in EE

Determinants:

$\det[A]$ Think of this as the “magnitude” of a matrix.

Matrix Inverse:

$$[A]^{-1}[A] = [I]$$

Matrix Division:

$[A]^{-1}[B]$ predivide

$[B][A]^{-1}$ postdivide

26

# Matrix Algebra (1 of 3)

Computational Methods in EE

Commutative Laws

$$[A]+[B]=[B]+[A]$$
$$[A][B]\neq[B][A]$$

Associative Laws

$$([A]+[B])+[C]=[A]+([B]+[C])$$
$$([A][B])[C]=[A]([B][C])$$

Matrix Inverses and Transposes

$$[A]^{-1}[A]=[A][A]^{-1}=[I]$$
$$\left([A]^{-1}\right)^{-1}=[A]$$
$$([A][B])^{-1}=[B]^{-1}[A]^{-1}$$
$$\left([A]^{T}\right)^{-1}=\left([A]^{-1}\right)^{T} \qquad \left([A]^{T}\right)^{T}=[A]$$
$$([A]+[B])^{T}=[A]^{T}+[B]^{T} \qquad ([A][B])^{T}=[B]^{T}[A]^{T}$$

Distributive Laws

$$([A]+[B])[C]=[A][C]+[B][C]$$
$$[A]([B]+[C])=[A][B]+[A][C]$$

27

# Matrix Algebra (2 of 3)

Computational Methods in EE

Addition with a Scalar

$$\alpha+[A]=\text{doesn't make sense}$$

$$\alpha[I]+[A]=\begin{bmatrix} \alpha+a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \alpha+a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & & \alpha+a_{nn} \end{bmatrix}$$

Multiplication with a Scalar

$$\alpha([A]+[B])=\alpha[A]+\alpha[B]$$
$$\alpha([A][B])=(\alpha[A])[B]=[A](\alpha[B])$$

28

# Matrix Algebra (3 of 3)

Computational Methods in EE

Operations with Special Matrices

$$[0][A]=[A][0]=[0]$$
$$[I][A]=[A][I]=[A]$$
$$[0]+[A]=[A]+[0]=[A]$$
$$[A]-[A]=[0]$$

29

# Example of Matrix Algebra

Computational Methods in EE

Simplify the Following Equation

$$\left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}+\mathbf{D}=\mathbf{BC}+\mathbf{D}$$

Step 1 – Subtract $\mathbf{D}$ from both sides

$$\left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}+\mathbf{D}-\mathbf{D}=\mathbf{BC}+\mathbf{D}-\mathbf{D}$$
$$\left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}+\mathbf{0}=\mathbf{BC}+\mathbf{0}$$
$$\left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}=\mathbf{BC}$$

Step 2 – Inverse both sides

$$\left\{\left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\right\}^{-1}=\left\{\mathbf{BC}\right\}^{-1}$$
$$\mathbf{C}^{-1}\mathbf{A}=\mathbf{C}^{-1}\mathbf{B}^{-1}$$

Step 3 – Premultiply both sides by $\mathbf{C}$.

$$\mathbf{CC}^{-1}\mathbf{A}=\mathbf{CC}^{-1}\mathbf{B}^{-1}$$
$$\mathbf{IA}=\mathbf{IB}^{-1}$$
$$\mathbf{A}=\mathbf{B}^{-1}$$

30

Computational Methods in EE

# Common Linear Algebra Problems



Computational Methods in EE

## $[A][x] = [b]$

This problem arises when a problem $[A]$ is given some excitation $[b]$ and produces a solution $[x]$.

Examples: (1) waves scattering from an object, (2) heat through a device.

It produces a single solution.

Step 1 – Differential equation

$$\frac{d^2 f}{dx^2} + \gamma \frac{df}{dx} + f = b$$

Step 2 – ODE is converted to system of equations using finite-differences, finite elements, etc.

$$\begin{aligned} a_{11} f_1 + a_{12} f_2 + \cdots + a_{1n} f_n &= b_1 \\ a_{21} f_1 + a_{22} f_2 + \cdots + a_{2n} f_n &= b_2 \\ &\vdots \\ a_{n1} f_1 + a_{n2} f_2 + \cdots + a_{nn} f_n &= b_n \end{aligned}$$

Step 3 – System of equations is put into matrix form.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Step 4 – Matrix problem is solved for [$f$]

$$[f] = [A]^{-1}[b]$$

Step 5 – [$f$] is post processed to learn something.

# Eigen-Value Problems

Computational Methods in EE

Eigen-value problems arise when multiple solutions exist. No excitation is needed.

Examples: (1) resonating modes on a string, (2) electromagnetic modes in a waveguide, (3) electronic bands in a semiconductor.

$[A][x] = \lambda[x]$ Standard eigen-value problem

$[A][x] = \lambda[B][x]$ Generalized eigen-value problem

$[A]$ is the linear operation

$[x]$ is the unknown (eigen-vector)

$\lambda$ is the eigen-value and is just a scalar number

$[A]$ is potentially another part of the linear operation

33

# Determinants

Computational Methods in EE

The determinant is an important number associated with square matrices.

It is sort of a magnitude.

Unique solutions to systems of equations do not exist when the determinant is zero.

<u>2×2 Matrices</u>

$$\det[A] = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

<u>3×3 Matrices</u>

$$\det[A] = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

This can be calculated by walking across any of the rows.

34

# Cramer's Rule

Computational Methods in EE

Cramer's rule provides a methodical approach for calculating the unknowns of a system of equations.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$x_1 = \frac{1}{D} \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix} \qquad x_2 = \frac{1}{D} \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix} \qquad x_3 = \frac{1}{D} \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

$$D = \det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

35

Computational Methods in EE

# Naïve Gauss Elimination

36

# What is the Gauss-Jordan Method?

Computational Methods in EE

Naïve Guass elimination (GE) is the simplest method for solving a system of equations.

$$\begin{bmatrix} & A & \end{bmatrix}\begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

While simple, it can be unstable and “pivoting” is required to stabilize it.  The Naïve algorithm ignores this.

In fact, we sort of already implemented this when solving our first system of linear equations, we just did not do it in matrix forms.

37

# Step 1

Computational Methods in EE

We start with a matrix problem

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

38

# Step 2

Computational Methods in EE

Eliminate $x_1$ from rows 2 and 3.

$$(\text{New Row 2}) = (\text{Old Row 2}) - \frac{a_{21}}{a_{11}}(\text{Row 1})$$

$$(\text{New Row 3}) = (\text{Old Row 3}) - \frac{a_{31}}{a_{11}}(\text{Row 1})$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \end{bmatrix}$$

39

# Step 3

Computational Methods in EE

Eliminate $x_2$ from row 3.

$$(\text{New Row 3}) = (\text{Old Row 3}) - \frac{a'_{32}}{a'_{22}}(\text{Row 2})$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix}$$

40

# Step 4

Computational Methods in EE

Now we know $x_3$.

$$x_3 = \frac{b_3''}{a_{33}''}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}' & a_{23}' \\ 0 & 0 & a_{33}'' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2' \\ b_3'' \end{bmatrix}$$

41

# Step 5

Computational Methods in EE

We back-substitute to find $x_1$ and $x_2$.

$$x_2 = \frac{b_2'}{a_{22}'} - \frac{a_{23}'}{a_{22}'} x_3 \qquad x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2 - \frac{a_{13}}{a_{11}} x_3$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}' & a_{23}' \\ 0 & 0 & a_{33}'' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2' \\ b_3'' \end{bmatrix}$$

42

# Triangular Matrices

Computational Methods in EE

Notice we get an upper triangular matrix from Gauss elimination.

Triangular matrices represent systems of equations that are "almost" solved.

It is usually a very quick and easy procedure to solve triangular matrices.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix}$$

43

# Observation

Computational Methods in EE

For GE, we calculated three special factors.

These will arise again later in LU decomposition.

$$l_{21} = \frac{a_{21}}{a_{11}}$$

$$l_{31} = \frac{a_{31}}{a_{11}}$$

$$l_{32} = \frac{a'_{32}}{a'_{22}}$$

44

Computational Methods in EE

# Gauss-Jordan Method



## What is the Gauss-Jordan Method?

Computational Methods in EE

The Gauss-Jordan method is a technique to solve

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}\begin{bmatrix} \\ x \\ \\ \end{bmatrix}=\begin{bmatrix} \\ b \\ \\ \end{bmatrix}$$

or to calculate matrix inverses.

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}^{-1}$$

It is an excellent technique for solving these problems by hand!

# Step 1

Computational Methods in EE

We start with a matrix problem

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

47

# Step 2

Computational Methods in EE

We construct an "augmented" matrix.

$$\Big[ [A] \quad [b] \Big] \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix}$$

48

# Step 3

Computational Methods in EE

Normalize the first row by dividing by the diagonal element $a_{11}$.

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix} 1 & a_{12}/a_{11} & a_{13}/a_{11} & b_1/a_{11} \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix}$$

49

# Step 4

Computational Methods in EE

Eliminate $x_1$ from all other rows. For the second row, we subtract $a_{21}$*(row 1) from row 2. For the third row, we subtract $a_{31}$*(row 1) from row 3.

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & a'_{32} & a'_{33} & b'_3 \end{bmatrix}$$

Row 1

$$a'_{12} = a_{12}/a_{11}$$
$$a'_{13} = a_{13}/a_{11}$$
$$b'_1 = b_1/a_{11}$$

Row 2

$$a'_{22} = a_{22} - a_{21}a'_{12}$$
$$a'_{23} = a_{23} - a_{21}a'_{13}$$
$$b'_2 = b_2 - a_{21}b'_1$$

Row 3

$$a'_{32} = a_{32} - a_{31}a'_{12}$$
$$a'_{33} = a_{33} - a_{31}a'_{13}$$
$$b'_3 = b_3 - a_{31}b'_1$$

50

## Step 5

Computational Methods in EE

Normalize the second row by dividing by the diagonal element $a'_{22}$

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a'_{23}/a'_{22} & b'_2/a'_{22} \\ 0 & a'_{32} & a'_{33} & b'_3 \end{bmatrix}$$

51

## Step 6

Computational Methods in EE

Eliminate $x_2$ from all other rows. For the first row, we subtract $a_{12}$*(row 2) from row 1. For the third row, we subtract $a_{32}$*(row 2) from row 3.

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & a''_{13} & b''_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & 0 & a''_{33} & b''_3 \end{bmatrix}$$

Row 2

$$a''_{23} = a'_{23}/a'_{22}$$
$$b''_2 = b'_2/a'_{22}$$

Row 1

$$a''_{13} = a'_{13} - a'_{12}a''_{23}$$
$$b''_1 = b'_1 - a''_{23}b''_2$$

Row 3

$$a''_{33} = a'_{33} - a'_{32}a''_{23}$$
$$b''_3 = b'_3 - a''_{23}b''_2$$

52

# Step 7

Computational Methods in EE

Normalize row 3 by dividing by the diagonal element $a''_{33}$

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & a''_{13} & b''_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & 0 & 1 & b''_3/a''_{33} \end{bmatrix}$$

53

# Step 8

Computational Methods in EE

Eliminate $x_3$ from all other rows. For the first row, we subtract $a_{13}$*(row 3) from row 1. For the second row, we subtract $a_{23}$*(row 3) from row 3.

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & b'''_1 \\ 0 & 1 & 0 & b'''_2 \\ 0 & 0 & 1 & b'''_3 \end{bmatrix}$$

Row 3

$$b'''_3 = b''_3/a''_{33}$$

Row 1

$$b'''_1 = b''_1 - a''_{13}b'''_3$$

Row 2

$$b'''_2 = b''_2 - a''_{23}b'''_3$$

54

# Step 9

Computational Methods in EE

Extract solution from augmented matrix.

$$\begin{bmatrix}[A] & [b]\end{bmatrix} \rightarrow \begin{bmatrix}1 & 0 & 0 & b_1''' \\ 0 & 1 & 0 & b_2''' \\ 0 & 0 & 1 & b_3'''\end{bmatrix}$$

⬇

$$\begin{bmatrix}x_1 \\ x_2 \\ x_3\end{bmatrix} = \begin{bmatrix}b_1''' \\ b_2''' \\ b_3'''\end{bmatrix}$$

55

# Example

Computational Methods in EE

$[A] = \begin{bmatrix}1 & 2 & 3 \\ 0 & 4 & 1 \\ 1 & 1 & 0\end{bmatrix} \quad [b] = \begin{bmatrix}16 \\ 12 \\ 2\end{bmatrix}$ Step 1 – Define problem

$[A \;\; b] = \begin{bmatrix}1 & 2 & 3 & 16 \\ 0 & 4 & 1 & 12 \\ 1 & 1 & 0 & 2\end{bmatrix}$ Step 2 – Form augmented matrix

$\begin{bmatrix}1 & 2 & 3 & 16 \\ 0 & 4 & 1 & 12 \\ 1 & 1 & 0 & 2\end{bmatrix}$ Step 3 – Normalized row 1

$\begin{bmatrix}1 & 2 & 3 & 16 \\ 0 & 4 & 1 & 12 \\ 0 & -1 & -3 & -14\end{bmatrix}$ Step 4 – Subtract row 1 from rows 2 and 3

$\begin{bmatrix}1 & 2 & 3 & 16 \\ 0 & 1 & 0.25 & 3 \\ 0 & -1 & -3 & -14\end{bmatrix}$ Step 5 – Normalize row 2

$\begin{bmatrix}1 & 0 & 2.5 & 10 \\ 0 & 1 & 0.25 & 3 \\ 0 & 0 & -2.75 & -11\end{bmatrix}$ Step 6 – Subtract row 2 from row 1 and 3

$\begin{bmatrix}1 & 0 & 2.5 & 10 \\ 0 & 1 & 0.25 & 3 \\ 0 & 0 & 1 & 4\end{bmatrix}$ Step 7 – Normalize row 3

$\begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4\end{bmatrix}$ Step 8 – Subtract row 3 from row 1 and 2

$[x] = \begin{bmatrix}0 \\ 2 \\ 4\end{bmatrix}$ Step 9 – Extract answer from augmented matrix

56

# Algorithm for Any Size Matrix

Computational Methods in EE

1. Define [A] and [b]
2. Construct augmented matrix

   $$[U] = \big[[A] \quad [b]\big]$$

   ```
   U = [ A b ];
   ```

3. Iterate through all rows ($m$)
   a) Normalize $m$th row by dividing by diagonal element.

   $$[U]_{\text{row } m} = [U]_{\text{row } m} \div a_{mm}$$

   ```
   U(m,:) = U(m,:)/U(m,m);
   ```

   b) Iterate through all other rows ($r$), skipping the $m$th row
      i. Subtract row $m$ from row $r$

      $$[U]_{\text{row } r} = [U]_{\text{row } r} - a_{rm} \cdot [U]_{\text{row } m}$$

      ```
      U(r,:) = U(r,:) - U(r,m)*U(m,:);
      ```

4. Extract $[x]$ from augmented matrix

   $$[x] = [U]_{\text{column } M+1}$$

   ```
   x = U(:,M+1);
   ```

57

# How to Find Matrix Inverses

Computational Methods in EE

1. Define $[A]$
2. Construct augmented matrix

   $$[U] = \big[[A] \quad [I]\big]$$

3. Perform Gauss-Jordan method (iterate through all rows)
4. Extract $[A]^{-1}$ from augmented matrix

   $$[U'] = \big[[I] \quad [A]^{-1}\big]$$

58

Computational Methods in EE

# LU Decomposition



## Determining the Upper Triangular Matrix $[U]$

Computational Methods in EE

We start with

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \rightarrow \quad [A][x]=[b]$$

Using GE, we can write this system of equations using an upper-triangular matrix $[U]$.

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad \rightarrow \quad [U][x]=[d]$$

# Determining the Lower Triangular Matrix $[L]$

Computational Methods in EE

There exists a lower-triangular matrix $[L]$ such that

$$[A]=[L][U]$$

This equation is why the method is called LU decomposition.

$$[L]=\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

Recall these special terms.

We now substitute this decomposition into our original equation.

$$[A][x]=[b]$$
$$[A][x]-[b]=[0]$$
$$[L][U][x]-[b]=[0]$$
$$[L]\left([U][x]-[L]^{-1}[b]\right)=[0]$$
$$[L]\left([U][x]-[d]\right)=[A][x]-[b]=[0] \qquad \text{where } [d]=[L]^{-1}[b]$$

61

# Algorithm

Computational Methods in EE

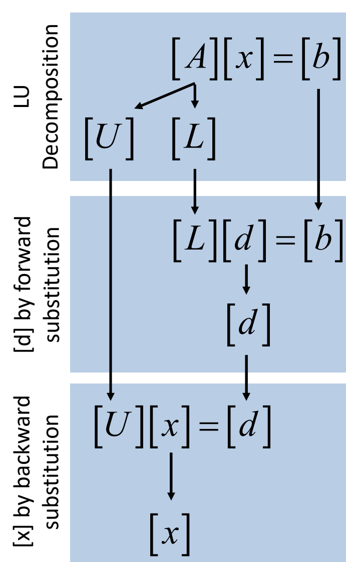
Step 1 – Decompose $[A]$ into $[L]$ and $[U]$.

a) Use GE to calculate $[U]$
b) Store $l$ terms during GE
c) Build $[L]$ from $l$ terms.
d) store $[L]$ and $[U]$ together as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ l_{21} & a'_{22} & a'_{23} \\ l_{31} & l_{32} & a''_{33} \end{bmatrix}$$

Step 2 – Solve $[L][d]=[b]$ using simple forward substitution.

Step 3 – Solve $[U][x]=[d]$ using simple backward substitution.




62

Computational Methods in EE

# Jacobi Method



## What is the Jacobi Method?

Computational Methods in EE

The Gauss-Jordan method was a direct solution of $[A][x]=[b]$.

This can be inefficient for large matrices, especially when a good initial guess $[x]$ is known.

We can create an iterative algorithm that improves the initial guess every iteration.

The method only converges for diagonally dominant matrices.

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

This means the $a_{ii}$ diagonal element must be larger than the sum of all other elements in the $i$th row.

The algorithm is very picky about this!

# Formulation (1 of 2)

Computational Methods in EE

Our matrix problem is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

We expand this into its component equations.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

65

# Formulation (2 of 2)

Computational Methods in EE

We solve the first equation for $x_1$, the second equation for $x_2$, and the third equation for $x_3$.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \quad \rightarrow \quad x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \quad \rightarrow \quad x_2 = \frac{b_2 - a_{21}x_1 - a_{23}x_3}{a_{22}}$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \quad \rightarrow \quad x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}}$$

These are the equations that we will iterate.

66

# Implementation (1 of 2)

Computational Methods in EE

Step 1 – Define Problem

$$[A]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad [b]=\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Step 2 – Come up with an initial guess for $[x]$.

$$x_1^{(1)},\ x_2^{(1)},\ \text{and } x_3^{(1)}$$

67

# Implementation (2 of 2)

Computational Methods in EE

Step 3 – Iterate $[x]$ until convergence

a) Calculate new values of $[x]$

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

b) Check how much the values have changed

$$\varepsilon_1^{(i+1)}=\left|\frac{x_1^{(i+1)}-x_1^{(i)}}{x_1^{(i+1)}}\right| \qquad \varepsilon_2^{(i+1)}=\left|\frac{x_2^{(i+1)}-x_2^{(i)}}{x_2^{(i+1)}}\right| \qquad \varepsilon_3^{(i+1)}=\left|\frac{x_3^{(i+1)}-x_3^{(i)}}{x_3^{(i+1)}}\right|$$

c) Continue to iterate until all values of $\varepsilon$ are sufficiently small.

68

# Matrix Formulation (1 of 2)

Computational Methods in EE

Let's rearrange our update equations this way

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}} \quad \rightarrow \quad x_1 = \frac{1}{a_{11}}\left[b_1 - \left(a_{12}x_2 + a_{13}x_3\right)\right]$$

$$x_2 = \frac{b_2 - a_{21}x_1 - a_{23}x_3}{a_{22}} \quad \rightarrow \quad x_2 = \frac{1}{a_{22}}\left[b_2 - \left(a_{21}x_1 + a_{23}x_3\right)\right]$$

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}} \quad \rightarrow \quad x_3 = \frac{1}{a_{33}}\left[b_3 - \left(a_{31}x_1 + a_{32}x_2\right)\right]$$

By inspecting these equations, we can write the update equation as

$$[x]_{i+1} = \left(\text{diag}[A]\right)^{-1} \cdot \left\{[b] - \left([A] - \text{diag}[A]\right)[x]_i\right\}$$

$$\text{diag}[A] = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{MM} \end{bmatrix}$$

69

# Matrix Formulation (2 of 2)

Computational Methods in EE

For simplicity, let

$$[D] = \text{diag}[A] \qquad [R] = [A] - \text{diag}[A]$$

Rearrange the "update equation" so that it makes more intuitive sense.

$$\begin{aligned}
[x]_{i+1} &= [D]^{-1} \cdot \left\{[b] - [R][x]_i\right\} \\
&= [D]^{-1} \cdot \left\{[b] - \left([A] - [D]\right)[x]_i\right\} \\
&= [D]^{-1}[b] - [D]^{-1}[A][x]_i + [D]^{-1}[D][x]_i \\
&= [x]_i + \underbrace{[D]^{-1}\left([b] - [A][x]_i\right)}
\end{aligned}$$

Improvement on $[x]_i$

Previous solution

70

# Matrix Implementation

Computational Methods in EE

1. Define $[A]$ and $[b]$
2. Make initial guess $[x]_1$
3. Extract diagonal of $[A]$

   $[D] = \text{diag}[A]$

   ```
   D = diag(A);
   ```

4. Iterate until convergence
   a) Calculate adjustment of $[x]_i$

   $$[\Delta x]_i = [D]^{-1}\left([b]-[A][x]_i\right)$$

   ```
   dx = D\(b - A*x);
   ```

   b) Adjust $[x]_i$

   $$[x]_{i+1} = [x]_i + [\Delta x]_i$$

   ```
   x = x + dx;
   ```

   c) Compute relative error

   $$\varepsilon = \max\left(\left|\frac{[\Delta x]_i}{[x]_i}\right|\right)$$

   ```
   err = max(abs(dx./x));
   ```

   d) Continue iteration until $\varepsilon$ is sufficiently small

71

Computational Methods in EE

# Gauss-Seidel Method

72

# Gauss-Seidel Method

Computational Methods in EE

The Jacobi method required $[A]$ to be diagonally dominant, which restricts what the method can be used to solve.

The Gauss-Seidel method is a modification to the Jacobi method to overcome this limitation.



# Formulation (1 of 4)

Computational Methods in EE

The matrix $[A]$ can be decomposed into the sum of a lower triangular matrix $[L']$ and an upper triangular matrix $[U]$.

$$[A]=[L']+[U]$$

$$[A]=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \\ a_{N1} & a_{N2} & & a_{NN} \end{bmatrix} \quad [L']=\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ a_{N1} & a_{N2} & & a_{NN} \end{bmatrix} \quad [U]=\begin{bmatrix} 0 & a_{12} & \cdots & a_{1N} \\ 0 & 0 & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & 0 \end{bmatrix}$$

Note: $[L']$ is not the same $[L]$ that we used in LU decomposition. These matrices are related through

$$[L']=[L]-[I]+\text{diag}[A]$$

# Formulation (2 of 4)

Computational Methods in EE

We are trying to solve $[A][x]=[b]$. Given that $[A] = [L'] + [U]$, we get

$$[A][x]=[b]$$

$$\left([L']+[U]\right)[x]=[b]$$

$$[L'][x]+[U][x]=[b]$$

75

# Formulation (3 of 4)

Computational Methods in EE

From our prior experience, we know that $[L'][x]=[b]$ is fast to solve for $[x]$ using forward-substitution.

We rearrange our matrix equation to take advantage of this.

$$[L'][x]+[U][x]=[b]$$

$$[L'][x]=[b]-[U][x]$$

$$[x]=\underbrace{[L']^{-1}\left([b]-[U][x]\right)}_{\text{This can be solved very fast!}}$$

76

# Formulation (4 of 4)

Computational Methods in EE

We derive our update equation from the last expression.

$$[x] = [L']^{-1}\left([b] - [U][x]\right)$$

$$\downarrow$$

$$\boxed{[x]_{i+1} = [L']^{-1}\left([b] - [U][x]_i\right)}$$

77

Computational Methods in EE

# Calculating Matrix Inverse

78

# Using Gauss-Jordan Method

Computational Methods in EE

Given matrix $[A]$, we form an augmented matrix

$$\left[[A] \quad [I]\right] \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & | & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & | & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & | & 0 & 0 & 1 \end{bmatrix}$$

Use the Gauss-Jordan method until the augmented matrix has the form

$$\begin{bmatrix} 1 & 0 & 0 & | & c_{11} & c_{12} & c_{13} \\ 0 & 1 & 0 & | & c_{21} & c_{22} & c_{23} \\ 0 & 0 & 1 & | & c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Here is the matrix inverse.

$$[A]^{-1} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

79

# Using LU Decomposition

Computational Methods in EE

We apply the LU decomposition method on a column-by-column basis. Each solution is a column in the inverse matrix.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_{12} \\ c_{22} \\ c_{32} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_{13} \\ c_{23} \\ c_{33} \end{bmatrix}$$

$$[A]^{-1} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

80