Topics for Today:

- Any Remaining Startup Questions?
- Recap of Mesh and NODE equations from Lecture 1:
  - Symmetric about main diagonal
  - \([Y_{BUS}]\) is invertible, usually
- More on node equation formulations, sparse storage, etc.
- Possible Solution Methods
  - Brute Force Inversion and pre-multiplication
  - in situ methods:
    - Gauss Elimination
    - Gauss-Jordan Elimination
    - LU Factorization
- Matrix “manipulations”
  - Kron Reduction
  - Augmentation
    - Adding constraints (add’l variables) to system of equations
    - Adding a source, short circuit, ideal transformer, etc.
• Homework #1 -- To get started:
  • 1) Go thru videotaped EE5200 MatLab tutorials, refer to Matlab online help for matrix operations and take basic notes for your future reference.
  • 2) Use MatLab to solve the matrix equations for the mesh and the node problems in Lecture 1.
  • 3) Go thru the terminology listed in assignment, look up in text or other references. Find corresponding MatLab functions or capabilities. Take notes on MatLab syntax and application “in’s and out’s”
  • 4) Find out how to enter a sparse matrix into MatLab and document the procedure. Learn how to view network topology via the Matlab spy function.
• Don’t hesitate to send e-mail our group e-mail forum,
• it can be helpful for everyone to contribute questions and comments.
Implications of symmetry:

i.e. if $y_{nk} = y_{kn}$

Bilateral vs. non-Bilateral

\[
\begin{cases}
y_{kn} \neq y_{nk} \\
op 
\end{cases}
\]

P.S. xfrmr or dependent source

Transfer admittances:

\[
-y_{nk} = \frac{I_{nk}}{V_n}
\]

\[
-y_{kn} = \frac{I_{kn}}{V_k}
\]
- If symmetric about main diagonal, then might get by with staring only lower half of off-diagonal terms.

- Careful! a) in situ methods will produce "fills" Can't look statically at storage requirements!
  b) Produce errors in solution if non-bilateral.

Remaining topics:
- Linked list storage
- Thev \rightarrow Norton for gen's & \{Y_{bus}\}
- Augmenting \{Y_{bus}\}
- Partitioning (Kron Reduction)
Ex: Coefficient matrix

<table>
<thead>
<tr>
<th>3</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Full storage: \( A = \begin{bmatrix} 3 & 0 & 0 & 2 & 0 \\ 0 & 5 & 1 & 0 & 0 \\ 2 & 0 & 7 & 1 & 0 \\ 0 & 0 & 9 & 0 & 4 \end{bmatrix} \)

25 numbers.

Single precision: 8

Real: 4 bytes

Complex: 8 bytes \( \Rightarrow 8 \times 25 = 200 \text{ bytes} \)

\[ 10,000 \text{- bus} \]

\( \Rightarrow [Y]: 100,000,000 \text{ entries} \)
Linked List: Storage

Actual Values

A ←

<table>
<thead>
<tr>
<th>ICOL</th>
<th>NEXT</th>
<th>NBEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

5x2

↑ INT pointers to start of each row

If NBUS < 32,000, then we can use 2-byte Int.

KEY: Vital to understand data structure.

104
52
10

166 Bytes

Instructor: Bruce Mork Phone: (906) 487-2857 Email: bamork@mtu.edu
Data Type [A]
- Single-prec Complex A
- " " Int. ICOL
- " " Int. NEXT

\[ A(1) = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \]
Ex:

Thevenin equivalent of Gen

Convert to admittances w/ Norton equivs
Node 1

\[ \vec{I}_1 = (\vec{V}_1 - \vec{V}_2) \cdot j \cdot 3 + (\vec{V}_1 - \vec{O})(-j \cdot 10 s) \]

\[ \vec{I}_2 = (\vec{V}_2 - \vec{V}_1) \cdot j \cdot 3 + (\vec{V}_2 - \vec{V}_3) \cdot j \cdot 1 + (\vec{V}_2 - \vec{V}_3) \cdot j \cdot 2 \]

\[ \vec{I}_3 = (\vec{V}_3 - \vec{V}_2) \cdot (j \cdot 1 - j \cdot 2) + (\vec{V}_3 - \vec{O}) \cdot (j \cdot 4) \]

\[
\begin{bmatrix}
\vec{I}_1 \\
\vec{I}_2 \\
\vec{I}_3 \\
\end{bmatrix} = \begin{bmatrix}
\text{or, build} \\
\text{[4]} \text{ by} \\
\text{inspection.} \\
\end{bmatrix} \begin{bmatrix}
\vec{V}_1 \\
\vec{V}_2 \\
\vec{V}_3 \\
\end{bmatrix}
\]

Continue, see homework.
Admittance Equations

General Form:

\[
\begin{bmatrix}
Y_{\text{bus}}
\end{bmatrix}
\begin{bmatrix}
V_{\text{node}}
\end{bmatrix}
= 
\begin{bmatrix}
I_{\text{inj}}
\end{bmatrix}
\]

We can add constraints:
- V source Bus-Bus
- Short
- XFMR
- DEPENDENT SOURCES (OP-AMP)
Kroên Reduction - System Reduction - Kroên Elimination


Possible to reduce to equiv system of fewer nodes.
Goal: Only buses of interest need be observable.

Constraint: Must retain source nodes (nodes at which current is being injected).

Steps:
1) Reorder system to move buses to kept to top, i.e. 1,...,K Remaining L,...,Z nodes are absorbed into system.
2) Perform Kramm Reduction.
\[ \begin{bmatrix} K & L \\ L^T & M \end{bmatrix} \begin{bmatrix} V_A \\ V_B \end{bmatrix} = \begin{bmatrix} I_A \\ I_x \end{bmatrix} \]

**Y_{bus} \mathbf{v} = \mathbf{i}**

1. \[ I_A = KV_A + LV_B \]
2. \[ I_x = L^TV_A + MV_B \]

Since \[ I_x = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \]
(3) \(-L^T V_A = M V_B\) \(<\) From Egn. (2) for \(I_x = 0\).

(4) \(-M^{-1} L^T V_A = V_B\) \(<\) premultiply both sides by \(M^{-1}\).

Substituting \(V_B\) into Egn. (1),

\[
I_A = K V_A - L M^{-1} L^T V_A
\]

\[
[I_A] = [K-LM^{-1}L^T][V_A]
\]

The \([Y_{bus}]\) for this reduced system is thus implied to be \([K-LM^{-1}L^T]\).

Derivation assumes bilateral system (note \(L, L^T\))
System

Then this eg. only valid for bilateral

If L & L are off-diagonals

IMPORTANT OBSERVATION:

\[
\begin{bmatrix}
\text{Reduced} \\
\text{Ybus}
\end{bmatrix} = K - L M_1 L^T
\]

Reduced [Ybus] is