Reordering

• It is strategic to reorder the (row,column) indices of the coefficient matrix to make LU factorization most efficient

• Basic Idea:
  – Move the buses that have the fewest connections to the top of $[Y]$. This reduces the number of fills when factorizing.
  – Constraints:
    • Buses of known voltage must stay at the bottom.
    • Augmented equations with zero main diagonal should remain toward the bottom to guarantee a fill of the main diagonal.
Reordering: (in-situ methods)

For: LU factorization
Gaussian Elim
Gauss-Jordan Elim

\[ A \times x = B \]

Strategy:
- Move rows with most zeros to top (i.e., least-connected buses to top)
- Augmented Gauss:
  Move rows with zero diagonal to bottom (and hope for a fill before normalization).

\[ x \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \end{bmatrix} \]
Helpful MatLab Commands

• General
  – save, load
  – who – lists variables
  – clear a, b, ..
  – inv(A) – inverts matrix
  – zeros, ones
  – find
  – help xxx to find help on a specific command
  – HTML help desk

• LU, sparse matrices
  – issparse, sparse, full
  – nnz, nonzeros
  – spy – shows topology
  – [L,U] = lu(A)
  – Reordering: colmd, symmmd, symrcm, colperm, randperm, dmperm
Use of the spy function

57-Bus
IEEE System

Sparsity:
93.44%
Exercise to Investigate Reordering

% COLMMD reordering:
flops(0);       %Reset floating point operations counter
%Reorder and fill new Y and I:
Y1=sparse(nbus,nbus);       %Change to Y1=zeros(nbus,nbus)to give slower full matrix
I1=zeros(nbus,1);
Reord1=colmmd(Y);
for n=1:nbus
    I1(n) = I(Reord1(n));
    for m=1:nbus
        Y1(n,m)=Y(Reord1(n),Reord1(m));
    end
end
BV1 = inv(Y1)*I1;

% Un-Reorder and recover correctly-indexed bus voltages:
BV_1 = zeros(nbus,1);
for n=1:nbus
    BV_1(Reord1(n)) = BV1(n);
end
nflops(1)=flops;

} We would not in a real case create duplicate matrices for the reordering. Improve this part