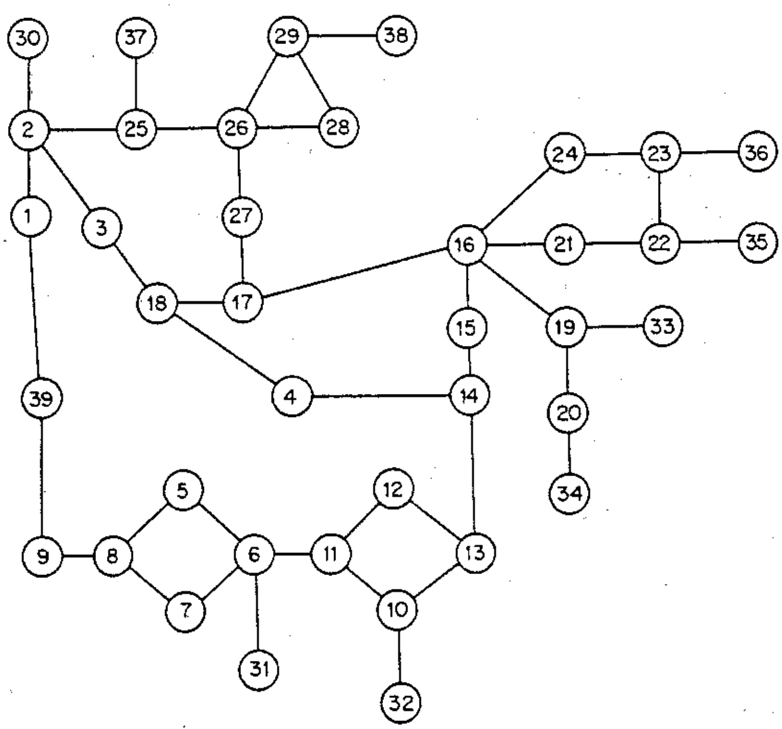
ELEN 118
A. I. Zecevic

PROJECT # 1: Sparse matrices and m - files in Matlab

The graph shown below corresponds to a model of the New England power system.



PROBLEM 1.

a) Form a symmetric matrix that corresponds to this graph, and store it as an ASCII file in the Matlab sparse matrix format. To keep things simple, you can assume that all diagonal entries are equal to 10, and that all off-diagonal entires have value 0.1.

b) Load the matrix into Matlab using function ***spconvert***( ). Obtain a spy plot of the matrix and print it out.

c) Perform an LU factorization of the matrix, using function **[L,U]** = ***lu***( ). Record the number of floating point operations (function ***flops***) and the number of fill in elements, and plot matrix L + U.

PROBLEM 2.

a) Find a minimal degree ordering for this matrix and record the resulting permutation vector. Show your work, and *do not* use Matlab in this step.

b) Use the permutation vector obtained in part a) to reorder the matrix. Obtain a spy plot of the reordered matrix and print it out.

c) Perform an LU factorization of the matrix, using function **[L,U] = *lu*( )**. Record the number of floating point operations and the number of fill in elements, and plot matrix L + U.

PROBLEM 3.

a) Use the Matlab function ***symmmd*( )** to obtain a minimal degree ordering for the matrix.

b) Use the permutation vector obtained in part a) to reorder the matrix. Obtain a spy plot of the reordered matrix and print it out.

c) Perform an LU factorization of the matrix, using function **[L,U] = *lu*( )**. Record the number of floating point operations and the number of fill in elements, and plot matrix L + U.

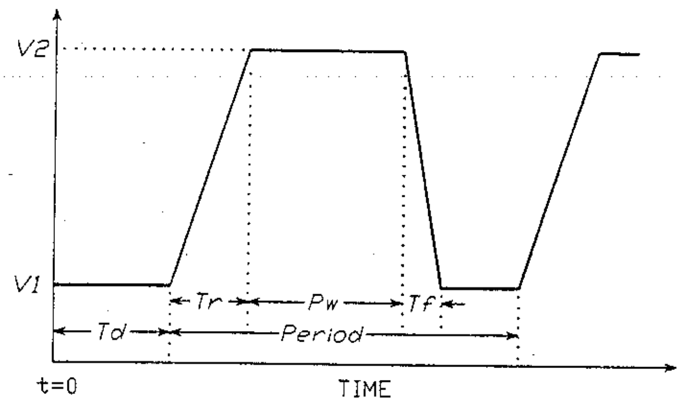
d) Compare your minimal degree ordering with the one in Matlab, in terms of the number of fill ins and the number of floating point operations. Which one does better?

PROBLEM 4.

In SPICE, a pulse waveform is described by a set of seven parameters.

Parameter number	**Parameter symbol**	**Parameter interpretation**
$a(1)$	*V1*	Initial voltage
$a(2)$	*V2*	Peak voltage
$a(3)$	*Td*	Initial delay time
$a(4)$	*Tr*	Rise time
$a(5)$	*Tf*	Fall time
$a(6)$	*Pw*	Pulse width
$a(7)$	*Period*	Pulse period

Schematically, we can represent these parameters as



a) Create an m - file that will produce a pulse waveform (call this function *pulse.m*). The input arguments to this function should be a 1×7 vector a which corresponds to the SPICE parameters (a = [a(1) a(2) ... a(7)]) and *time*, and the output should be *voltage v*. In other words, function ***v = pulse(a, t)*** should return the voltage of the pulse waveform at time *t*.

b) Use function **[*out, tout*] = *tstpls*(*a, h, tend*)** to test the function you created in part a). Choose t_{end} so that you capture at least *two cycles* of the waveform. Plot the resulting waveform.

```
function [out,tout]=tstpls(a,h,tend)

% This is a function to test pulse.m

time=h;

while time<=tend
   v=pulse(a,time);
   out=[out v];
   tout=[tout time];
   time=time+h;
end
```