ECE 521

HW #1 SOLUTIONS

1.a) MNA stamp

$$
\begin{array}{cc}
\begin{array}{cccccc} V_A & V_B & V_C & V_D & i_1 & i_2 \end{array} & \text{RHS} \\
\begin{bmatrix}
 & & & & 1 & \\
 & & & & -1 & \\
 & & & & & 1 \\
 & & & & & -1 \\
1 & -1 & -h_{12} & h_{12} & -h_{11} & 0 \\
0 & 0 & -h_{22} & h_{22} & -h_{21} & 1
\end{bmatrix}
&
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{array}
$$

b) A plain nodal analysis description is possible only if $h_{11} \neq 0$

$$\Rightarrow i_1 = \frac{1}{h_{11}} v_1 - \frac{h_{12}}{h_{11}} v_2$$

$$i_2 = \frac{h_{21}}{h_{11}} v_1 + \left(h_{22} - \frac{h_{21} h_{12}}{h_{11}}\right) v_2$$

NA stamp

$$
\begin{array}{cc}
\begin{array}{cccc} V_A & V_B & V_C & V_D \end{array} & \text{RHS} \\
\begin{bmatrix}
1/h_{11} & -1/h_{11} & -h_{12}/h_{11} & h_{12}/h_{11} \\
-1/h_{11} & +1/h_{11} & +h_{12}/h_{11} & -h_{12}/h_{11} \\
h_{21}/h_{11} & -h_{21}/h_{11} & \left(h_{22} - \frac{h_{21} h_{12}}{h_{11}}\right) & -\left(h_{22} - \frac{h_{21} h_{12}}{h_{11}}\right) \\
-h_{21}/h_{11} & h_{21}/h_{11} & -\left(h_{22} - \frac{h_{21} h_{12}}{h_{11}}\right) & \left(h_{22} - \frac{h_{21} h_{12}}{h_{11}}\right)
\end{bmatrix}
&
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{array}
$$

2. See page 142 (Vlach & Singhal) for a discussion. The correct solution is

MNA stamp

$$
\begin{array}{ccc}
 & \begin{array}{ccc} N+ & N- & I \end{array} & \text{RHS} \\
\begin{array}{c} N+ \\ N- \\ I \end{array} &
\begin{bmatrix}
 & & 1 \\
 & & -1 \\
L & -L & -(1-L)
\end{bmatrix}
&
\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\end{array}
$$

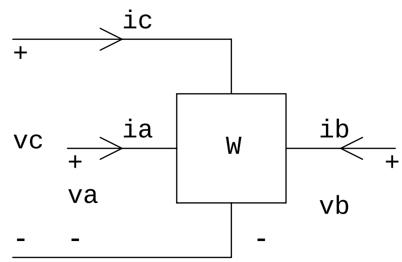
where $L = 1$ for switch on & $L = 0$ for off

Note Solutions to problems 3-5 courtesy Prof. Jacob White of MIT.

3 (a) In general, the constitutive relation for any circuit element can be written as:

$$Z_v v + Z_i i = s$$

where Z_v, Z_i are matrices that depend on the nature of the element, and s is a vector of source terms.



We have already seen this (in a simplied form) in the equation $Gv = i$. Consider a resistor of value $1/g$:

$$gv - i = 0,$$

a current source of value i_s:

$$0v + i = i_s,$$

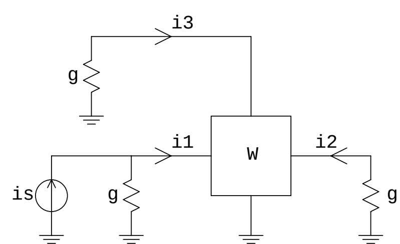
or a voltage-controlled current source:

$$g(v_1 - v_2) = i_{34}.$$

Each of the above constitutive relations can be written in the form $i = Gv$, i.e. we could solve for i on one side of the equation. In other words, the Z_i matrix can be inverted. This is the situation in which plain nodal analysis may be applied.

However, the Z_i matrix for the W-element is singular (the first column equals the sum of the second plus the third), so we cannot solve the constitutive relation for i, and thus cannot use NA for circuits featuring the W-element.

(b) We use the modified nodal analysis (MNA) here. MNA is in fact the most popular formulation approach today for circuit simulation programs like SPICE. The MNA matrix retains many of the nice numerical properties of the PNA (plain nodal analysis) matrix if most of the elements have constitutive relations with invertible Z_i.



We start by treating the branch currents i_1, i_2, i_3 associated with the W-element as *known* current sources, and simply writing down the plain nodal analysis equations:

$$gv_1 = i_s - i_1$$

$$gv_2 = -i_2$$

$$gv_3 = -i_3$$

Since the currents i_1, i_2, i_3 are really *unknowns*, we move them to the left-hand-side. There are now three equations and six unknowns. The additional three equations we need are just the constitutive relations for the W-element:

$$v_1 + i_1 + i_2 = 0$$

$$v_1 + v_2 + i_2 - i_3 = 0$$

$$v_1 + v_2 + v_3 + i_1 + i_3 = 0$$

Putting everything in matrix form, we get

$$\begin{bmatrix} g & & & 1 & & \\ & g & & & 1 & \\ & & g & & & 1 \\ 1 & & & 1 & 1 & \\ 1 & 1 & & & 1 & -1 \\ 1 & 1 & 1 & 1 & & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} i_s \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

Note that the first three rows are simply plain nodal analysis equations rearranged to treat the "fictitious" current sources as unknowns; the upper left corner of the matrix is exactly the nodal analysis matrix. The last three rows are the constitutive relations.

4 (a) The structure of the $N \times N$ conductance matrix $\boldsymbol{G}$ is:

$$\boldsymbol{G} = \begin{bmatrix} 2G & -G & 0 & \cdots & 0 \\ -G & 2G & -G & \ddots & \vdots \\ 0 & -G & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -G \\ 0 & \cdots & 0 & -G & 2G \end{bmatrix}$$

The matrix $\boldsymbol{G}$ is a tridiagonal matrix (i.e. a band matrix of bandwidth 2). By inspection, the number of nonzero entries in $\boldsymbol{G}$ is $N + 2(N-1) = 3N - 2$.

(b) The matrix problem for the resistor line, written in terms of the resistance matrix $\boldsymbol{G}^{-1}$ is $\boldsymbol{G}^{-1}\boldsymbol{i} = \boldsymbol{v}$ where $\boldsymbol{i}$ is the vector of current source currents flowing into each of the nodes, and $\boldsymbol{v}$ is the vector of node voltages. For our original resistor line, $\boldsymbol{i}$ is a zero vector.

Suppose now that the j-th entry of the vector $\boldsymbol{i}$ is nonzero. Physically, an injection of current into node j will cause a change in all the node voltages. The j-th entry of vector $\boldsymbol{i}$ multiplies only the j-th column of $\boldsymbol{G}^{-1}$. So a change in all the node voltages in $\boldsymbol{v}$ will be algebraically possible only if the j-th column of $\boldsymbol{G}^{-1}$ consists of all nonzero entries, i.e. ${\boldsymbol{G}^{-1}}_{i,j} \neq 0$ for all i.

By extending this argument to all entries of the current source vector (and all columns of the resistance matrix), we see that the $N \times N$ resistance matrix $\boldsymbol{G}^{-1}$ is full, i.e. will have N^2 nonzero entries.

(c) The factorization of the tridiagonal conductance matrix $\boldsymbol{G}$ produces two bidiagonal factors $\boldsymbol{L}$ and $\boldsymbol{U}$, such that $\boldsymbol{LU} = \boldsymbol{G}$. In order to see this, let's examine the first few elimination steps for the matrix $\boldsymbol{G}$.

After the first elimination step we get:

$$\boldsymbol{G}^{(1)} = \begin{bmatrix} 2G & -G & 0 & \cdots & 0 \\ 0 & 3G/2 & -G & \ddots & \vdots \\ 0 & -G & 2G & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -G \\ 0 & \cdots & 0 & -G & 2G \end{bmatrix}$$

And after the second:

$$\boldsymbol{G}^{(2)} = \begin{bmatrix} 2G & -G & 0 & \cdots & \cdots & 0 \\ 0 & 3G/2 & \ddots & \ddots & & \vdots \\ 0 & 0 & 4G/3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & -G & 2G & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -G \\ 0 & \cdots & \cdots & 0 & -G & 2G \end{bmatrix}$$

Each elimination step targets only one row in the tridiagonal matrix $\boldsymbol{G}$. In addition, the triangular block of zeros in the upper-right corner of the matrix remains untouched. Thus after all $N-1$ elimination steps, the $\boldsymbol{L}$ matrix will feature ones on the main diagonal, and the $N-1$ multipliers on the sub-diagonal. The $\boldsymbol{U}$ matrix will also be bidiagonal, with the pivots on the main diagonal, and $-G$'s on the super-diagonal. It follows that the number of nonzero entries in $\boldsymbol{L}$ or $\boldsymbol{U}$ is thus $N + (N-1) = 2N-1$.

For $N = 1000$ the number of nonzero entries in $\boldsymbol{G^{-1}}$ is $1,000,000$ while $\boldsymbol{L}$ and $\boldsymbol{U}$ will each contain only 1999 nonzero entries. It is not a good idea to use the inverse of a matrix for solving the matrix problem due to the excessive number of required multiplications proportional to the number of nonzero entries.

(d) Moral - avoid inverting matrices.

5 (a) Structurally $\boldsymbol{A} \in \Re^{N\times N}$ looks like:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & \cdots & -\alpha \\ -\alpha & 1 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & \cdots & -\alpha & 1 \end{bmatrix}$$

After the first step of elimination we get:

$$
\boldsymbol{A}^{(1)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & -\alpha \\ 0 & 1 & 0 & \cdots & -\alpha^2 \\ \vdots & -\alpha & 1 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & -\alpha & 1 \end{bmatrix}
$$

After $N-1$ steps of elimination we have:

$$
\boldsymbol{A}^{(N-1)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & -\alpha \\ 0 & \ddots & & & & & -\alpha^2 \\ \vdots & \ddots & \ddots & & & & \vdots \\ \vdots & & 0 & 1 & \cdots & & -\alpha^k \\ \vdots & & & 0 & 1 & & \\ \vdots & & & & \ddots & \ddots & \\ \vdots & & & & & 1 & -\alpha^{N-1} \\ 0 & & & & & -\alpha & 1 \end{bmatrix}
$$

We will have $N-2$ fill-ins, all in the last column.

We can see that the largest number generated during the elimination is α^N. Assuming β is the largest number representable in our computer, the largest value of N for which $\boldsymbol{A}$ can be factored without overflow satisfies:

$$
|1-\alpha^N| < \beta
$$

or equivalently

$$
N < \frac{\log(\beta+1)}{\log\alpha}
$$

(b) To avoid overflow in the elimination process described, regardless of the size of N we must use a reordered Gaussian elimination algorithm.

If at each elimination step we choose as pivot the largest element in the column, we are guaranteed that the multipliers used are smaller than one, therefore causing no growth in the matrix.

For our problem the sub-diagonal element is always the largest in the column (we are assuming $\alpha > 1$). Therefore we exchange rows and get:

$$
\boldsymbol{A}^{(1)} = \begin{bmatrix} -\alpha & 1 & \cdots & \cdots & 0 \\ 1 & 0 & 0 & \cdots & -\alpha \\ \vdots & -\alpha & 1 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & -\alpha & 1 \end{bmatrix}
$$

Eliminating the subdiagonal element in the second row leads to:

$$
\boldsymbol{A}^{(1)} = \begin{bmatrix} -\alpha & 1 & \cdots & \cdots & 0 \\ 0 & 1/\alpha & 0 & \cdots & -\alpha \\ \vdots & -\alpha & 1 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & -\alpha & 1 \end{bmatrix}
$$

We can see that there was no growth in the matrix. Furthermore we note that a fill-in was created in the diagonal element of the (now) second row. The next step is to eliminate the sub-diagonal element in the third row. Again our algorithm searches the second column for the largest element and uses that as a pivot. That leads to exchanging rows again to get:

$$
\boldsymbol{A}^{(1)} = \begin{bmatrix} -\alpha & 1 & \cdots & \cdots & 0 \\ 0 & -\alpha & 1 & & \vdots \\ 0 & 1/\alpha & 0 & \cdots & -\alpha \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & -\alpha & 1 \end{bmatrix}
$$

Clearly the elimination process will again generate no growth in the matrix and one single fill-in is generated in the diagonal position of the (now) third row.

Therefore we will again generate $N-2$ fill-ins in the matrix, but avoid overflow, regardless of the size of N.

You may argue that now there will be underflow, but in general that is gracefully handled and is not a problem.

Another interesting solution to the problem (although it hides the general strategy) is to simply exchange the order of the unknowns and put the last column first, to get:

$$
\boldsymbol{A'} = \begin{bmatrix} -\alpha & 1 & 0 & \cdots & 0 \\ 0 & -\alpha & 1 & \cdots & 0 \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 1 & 0 & & \cdots & -\alpha \end{bmatrix}
$$

from where it is clear that direct Gaussian-elimination will produce no growth in the matrix and will again generate $N-2$ fill-ins (the last row will be filled in as the elimination proceeds). You may argue that in this case each fill-in will be eliminated in the step that immediately follows its creation, so that in fact we only need storage for a single element, but zeros created by such exact cancellations are more difficult to handle than structural zeros, and so are not usually exploited in sparse matrix codes.