

**Course 6.336 - Introduction to Numerical Algorithms (Fall 2003)**  
**Solutions to Problem Set #2**

**Problem 2.1**

a) Assume, for the sake of simplicity, that all resistors in the line are of resistance  $R$ . The structure of the  $N \times N$  conductance matrix  $\mathbf{G}$  is:

$$\mathbf{G} = \begin{bmatrix} \frac{2}{R} & -\frac{1}{R} & 0 & \cdots & 0 \\ -\frac{1}{R} & \frac{2}{R} & -\frac{1}{R} & \ddots & \vdots \\ 0 & -\frac{1}{R} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\frac{1}{R} \\ 0 & \cdots & 0 & -\frac{1}{R} & \frac{2}{R} \end{bmatrix}$$

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

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

Suppose now that the  $j$ -th entry of the vector  $\mathbf{i}$  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  $\mathbf{i}$  multiplies only the  $j$ -th column of  $\mathbf{G}^{-1}$ . So a change in all the node voltages in  $\mathbf{v}$  will be algebraically possible only if the  $j$ -th column of  $\mathbf{G}^{-1}$  consists of all nonzero entries, i.e.  $\mathbf{G}^{-1}_{i,j} \neq 0$  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  $\mathbf{G}^{-1}$  is full, i.e. will have  $N^2$  nonzero entries.

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

After the first elimination step we get:

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

And after the second:

$$\mathbf{G}^{(2)} = \begin{bmatrix} \frac{2}{R} & -\frac{1}{R} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{3}{2R} & \ddots & \ddots & & \vdots \\ 0 & 0 & \frac{4}{3R} & \ddots & \ddots & \vdots \\ \vdots & \ddots & -\frac{1}{R} & \frac{2}{R} & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -\frac{1}{R} \\ 0 & \cdots & \cdots & 0 & -\frac{1}{R} & \frac{2}{R} \end{bmatrix}$$

Each elimination step targets only one row in the tridiagonal matrix  $\mathbf{G}$ . 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  $\mathbf{L}$  matrix will feature ones on the main diagonal, and the  $N - 1$  multipliers on the sub-diagonal. The  $\mathbf{U}$  matrix will also be bidiagonal, with the pivots on the main diagonal, and  $-R$ 's on the super-diagonal. It follows that the number of nonzero entries in  $\mathbf{L}$  or  $\mathbf{U}$  is thus  $N + (N - 1) = 2N - 1$ .

For  $N = 1000$  the number of nonzero entries in  $\mathbf{G}^{-1}$  is 1,000,000 while  $\mathbf{L}$  and  $\mathbf{U}$  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) To determine the smallest entry in the resistance matrix, let's use the fact that an entry  $r_{ij}$  if the resistance matrix is simply a voltage at node  $i$  caused by a unit current source connected to node  $j$ . And we are looking for a smallest entry, i.e. the case where voltage is minimal. You can easily figure out that we should put current source at one ends of the line and examine the voltage at the other end of the line. In other words, the smallest element for an  $N \times N$  resistance matrix is always  $r_{1N}$  or  $r_{N1}$  - they are equal, since matrix inversion preserves symmetry.

Now imagine our line with current source connected to the first node. The voltage at node  $N$  is evidently

$$r_{1N} = \frac{1}{N + 1} \tag{1}$$

Note, that for  $N \times N$  matrix correspond to  $N + 1$ -length resistor line.

## Problem 2.2

a) It is very easy to find counter-example, just remember the fomula for, say, second pivot:

$$\tilde{a}_{22} = a_{22} - a_{21} \frac{a_{12}}{a_{11}} \tag{2}$$

Evidently, there is no guarantee that  $|\tilde{a}_{22}| \leq |a_{22}|$ .

b) The statement is true. For example, (2) implies that  $|\tilde{a}_{22}| \leq |a_{22}|$ . Therefore we have proven the statement for order 2. To prove the statement in general case, we need to use

mathematical induction. Let's assume that we have proven our statement for the order  $N - 1$ . Now we need to show, that for the order  $N$  of the matrix, after eliminating first row from all subsequent rows, the resulting  $(N - 1) \times (N - 1)$  submatrix:

1. will have all positive diagonal entries, no larger than original diagonal entries
2. will have negative off-diagonals
3. will be strictly diagonally dominant
4. will be tridiagonal

This first statement we have already shown, since all multipliers except for the  $M_{21}$  are zero. Second one is also trivial. Same is the last one.

Now, let's show that the third statement holds. Since initial matrix is strictly diagonally dominant, we know that  $|a_{22}| > |a_{21}| + |a_{23}|$ , and  $|a_{11}| > |a_{12}|$ . The only thing we need to show is that the number we subtract from  $a_{22}$  is less than  $a_{21}$ , which is also evident. Therefore we have:

$$|\tilde{a}_{22}| > a_{22} - |a_{21}| > |a_{21}| + |a_{23}| - |a_{21}| = |\tilde{a}_{23}| \quad (3)$$

We have proved the statement in general case.

### Problem 2.3

a) For  $\mathbf{N}(\mathbf{y}, k) = \mathbf{I} + \mathbf{y}\mathbf{e}_k^T$  and given  $k$ , the matrix  $\mathbf{N}$  structurally looks like:

$$\mathbf{N}(\mathbf{y}, k) = \begin{bmatrix} 1 & \cdots & y_1 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 1 + y_k & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & y_n & \cdots & 1 \end{bmatrix}$$

where  $\mathbf{y} = [y_1 \cdots y_n]^T$ . A simple check will help you verify that  $\mathbf{N}^{-1}$  is structurally similar to  $\mathbf{N}$ . Let

$$\mathbf{N}^{-1} = \begin{bmatrix} 1 & \cdots & w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & w_k & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & w_n & \cdots & 1 \end{bmatrix}$$

Furthermore, since  $\mathbf{N}\mathbf{N}^{-1} = \mathbf{I}$  we get the following system in  $n$  unknowns:

$$\left\{ \begin{array}{l} w_1 + y_1 w_k = 0 \\ w_2 + y_2 w_k = 0 \\ \vdots = 0 \\ (1 + y_k)w_k = 1 \\ \vdots = 0 \\ w_n + y_n w_k = 0, \end{array} \right.$$

or equivalently  $\mathbf{N}\mathbf{w} = \mathbf{e}_k$ . Solving this system for  $\mathbf{w} = [w_1 \cdots w_n]^T$  gives a formula for obtaining  $\mathbf{N}^{-1}$ :

$$\mathbf{N}^{-1} = \begin{bmatrix} 1 & \cdots & -\frac{y_1}{(1+y_k)} & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \frac{1}{(1+y_k)} & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & -\frac{y_n}{(1+y_k)} & \cdots & 1 \end{bmatrix}$$

b) Using the result in part a) we see that

$$\mathbf{N}(\mathbf{y}, k)\mathbf{x} = \mathbf{e}_k$$

or equivalently

$$(\mathbf{I} + \mathbf{y}\mathbf{e}_k^T)\mathbf{x} = \mathbf{e}_k$$

from where we get successively

$$\mathbf{x} + x_k \mathbf{y} = \mathbf{e}_k$$

and

$$\mathbf{y} = \frac{1}{x_k}(\mathbf{e}_k - \mathbf{x}).$$

c) Say we wish to find the inverse of the matrix  $\mathbf{A}$ ,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Let us write  $\mathbf{A}$  as

$$\mathbf{A} = \begin{bmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)} \end{bmatrix}$$

where  $x_j^{(0)}$  denotes the  $j$ th column of  $A$  at step 0 of our (yet-to-be derived) matrix inversion algorithm.

In part (b) we showed how to find a matrix  $N$  such that  $Nx = e_k$  for a given  $x$ . Let us notate this matrix by  $N(x, k)$  so that  $N(x, k)x = e_k$ . In the first step of the algorithm we compute  $N(x_1^{(0)}, 1)$  and multiply it into  $A$ , so that

$$N(x_1^{(0)}, 1)\mathbf{A} = \begin{bmatrix} 1 & & & \\ 0 & x_2^{(1)} & \cdots & x_n^{(1)} \\ 0 & & & \\ 0 & & & \end{bmatrix}$$

where

$$x_j^{(1)} = N(x_1^{(0)}, 1)x_j^{(0)}$$

We now multiply  $N(x_1^{(0)}, 1)\mathbf{A}$  by  $N(x_2^{(1)}, 2)$ , and so forth, so that after  $k$  steps, we have

$$\prod_{j=1}^k N(x_j^{(j-1)}, j)\mathbf{A} = \begin{bmatrix} 1 & 0 & & & \\ 0 & 1 & & & \\ 0 & 0 & \cdots & x_{k+1}^{(k)} & \cdots & x_n^{(k)} \\ \vdots & \vdots & & & & \\ 0 & 0 & & & & \end{bmatrix}$$

where the column vectors at step  $k$  are

$$x_j^{(k)} = \prod_{j=1}^k N(x_j^{(j-1)}, j)x_j^{(0)}$$

It is fairly easy to convince yourself that multiplication by  $N(x_k^{(k-1)}, k)$  does not affect the first  $k - 1$  columns of the matrix, since the first  $k - 1$  columns of both  $N(x_k^{(k-1)}, k)$  and  $\prod_{j=1}^k N(x_j^{(j-1)}, j)\mathbf{A}$  are the identity vectors  $e$ . Thus,

$$\mathbf{A}^{-1} = \prod_{j=1}^n N(x_j^{(j-1)}, j)$$

What makes the algorithm in-place is that since  $x_{j \leq k}^{(k)} = e_j$ , we no longer have to store those vectors. Also, column  $l$  of  $\prod_{j=1}^k N(x_j^{(j-1)}, j)$  is  $e_l$  for  $l > k$ . Thus, we can accumulate  $A^{-1}$  in the space occupied by the columns of  $A$  that have already been reduced to unit vectors.

Note that if at some step  $k$ ,  $x_k^{(k-1)}[k] = 0$ , then this algorithm will fail. The solution is to pivot by swapping columns in the matrix. I haven't included code that performs pivoting, because that is the subject of the next problem set.

Some of you noticed that this is essentially the procedure known as "Jordan elimination." Jordan elimination is in some sense an extension of Gaussian elimination to the extent that at each point in the elimination the elements on previous pivotal rows are also eliminated.

The following code implements these ideas by expliciting forming all the required products.

```
%in-place invert a matrix A
n = size(A,1);
for i=1:n,
y = -A(:,i);
y(i) = y(i) + 1.0;
y = y / A(i,i);
for k = i+1:n;
m = A(i,k);
for j=1:n;
A(j,k) = A(j,k) + m * y(j);
end;
end;
A(:,i) = y;
A(i,i) = A(i,i) + 1.0;
for k = 1:i-1;
m = A(i,k);
for j=1:n;
A(j,k) = A(j,k) + m * y(j);
end;
end;
end;
```

On a Sun Sparc-10, this routine is about 500 times slower than Matlab's builtin `inv()` function.

A more efficient code would operate on the matrix by columns:

```
%in-place invert a matrix A; vectorized
n = size(A,1);
for i=1:n,
y = -A(:,i);
y(i) = y(i) + 1.0;
y = y / A(i,i);
for k = i+1:n;
A(:,k) = A(:,k) + A(i,k) * y;
end;
A(:,i) = y;
A(i,i) = A(i,i) + 1.0;
for k = 1:i-1;
```

```
A(:,k) = A(:,k) + A(i,k) * y;  
end;  
end;
```

This routine is significantly faster; it is only a factor of 10-15 slower than Matlab. Regardless of how fast the machine is, the fact that direct matrix inversion takes  $O(N^3)$  operations limits the size of the problem we can solve in a reasonable time. Matlab took roughly 2.7 seconds to invert an  $N = 250$  matrix. In a month, it could probably do  $N = 25000$ , in a year, about  $N = 50000$ . The first algorithm above could probably handle only  $N = 3000$  in a month,  $N = 7000$  in a year.