

CAAM 551 Problem set 2

Due Mon. 21 Sep. 09.

Note: If you are a CAAM major, you must do problem 2 **instead** of problem 1. If you are not a CAAM major, you may do problem 1 or problem 2 (DO NOT DO BOTH).

Problem 1

- a. Write a Matlab code that will convert a sparse matrix in Matlab format $[i, j, A_{ij}]$ to CSR storage. You should store the pointers to row starts in an array IA , the column indices of the nonzero elements in an array JA and the nonzero matrix elements of A in an array AA .

For example the matrix of EX3.7 p 89 of Saad is:

```
% CSR Storage
%
% AA = [ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. ]
% JA = [ 1 4 1 2 4 1 3 4 5 3 4 5]
%      ^   ^   ^   ^   ^
% IA = [ 1 3 6 10 12 13]
```

- b. Write a Matlab code that will multiply a square matrix A in CSR storage times a vector x . i.e. code $w = Ax$.
- c. Repeat parts 1 and 2 using Jagged Diagonal Storage

For the Saad example this would be

Hint: You might find the matlab command

```
[rows,cols,vals] = find(A(i,:));
```

to be useful.

You should write a test routine for each of the codes above and give computational evidence that they are working correctly as you did for the codes on the first problem set.

- d. Write a Matlab code that will first solve $Lc = b$ and will then solve $L^T x = c$, where L is a sparse non-singular real $n \times n$ lower triangular matrix. (b is a given n -vector and x is solution vector).

The result x will be the solution to $Ax = b$ where $A = LL^T$. The matrix L should be traversed by rows in both cases. Thus, the first code (involving L) should be row oriented and the second code (involving L^T) should be column oriented.

Your Matlab function should be of the form

```
function [x] = CHsolve(AA,JA,IA,b);
```

where the lower triangular matrix L is stored in the AA, JA, IA CSR data structure. (You can use L in place of A in the notation if you wish.)

Write a test function to demonstrate your solver works correctly and test it on a random sparse lower triangular L (set the diagonals to 1 to assure non-singularity, but don't assume this in the code).

Problem 2

The following function will create a struct in which you can create the CSR storage scheme and then just refer to a single entity instead of having to pass three arrays.

```
function [s] = csr(nz,m,n);
%
% This sets up a struct s to hold a matrix A in CSR format
%
s = struct('nz',nz,'m',m,'n',n,'ia',[1:n+1],'ja',[1:nz],'aa',randn(nz,1));
%
```

```

% s has fields
%
%   s.nz   % length of arrays ja and aa
%   s.m   % number of rows
%   s.n   % number of columns
%   s.ia  % array to hold pointers to row starts
%   s.ja  % array to hold row indices
%   s.aa  % array to hold matrix values

```

Thus if you call

```
L = csr(nz,n,n)
```

you may fill in the values $L.ia \leftarrow IA$, $L.ja \leftarrow JA$, $L.aa \leftarrow AA$ and refer to the nonzero entries in i -th row of L as

```

for jj = L.ia(i): L.ia(i+1) - 1,
    j = L.ja(jj);
    value_ij = L.aa(jj);
end

```

Use this structure to write all of the the codes of Problem 1 above. Thus the function `CHsolve` will be re-written so that you can call it in the form

```
x = CHsolve(L,b)
```

and this should produce the solution to $LL^T x = b$.

Problem 3

Do problems 4 a,b,c,d and 12 on pages 98-100 of Saad.

For problem 4, you can save some time by creating the matrices A and B in Matlab putting the numerical value 1 in place of $*$. Then you can do the matrix multiplications in Matlab. You may display the matlab results instead of writing them out by hand.

For problem 12, you may wish to use a consequence of the Cayley-Hamilton theorem which says that if a matrix A is nonsingular (hence square) then

$$A^{-1} = \gamma_0 I + \gamma_1 A + \gamma_2 A^2 + \dots + \gamma_{n-1} A^{n-1},$$

where the γ_j are scalars. We shall prove this later in the course but you may assume it is true for this problem.

Problem 4

Given an $m \times n$ matrix A with $m > n$ and a positive number λ , a *regularized* least squares solution x_λ may be computed by solving

$$\min \left\| \begin{pmatrix} A \\ \mu I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2,$$

where $\mu = \sqrt{\lambda}$.

- Derive the normal equations for the *regularized* least squares problem given above.
- Show $A^T A + \lambda I$ is symmetric and positive definite for every positive value of λ . Prove that the regularized least squares solution x_λ is unique for every positive value of λ .
- Use the Singular Value Decomposition of A to express the solution x_λ to the problem

$$\min \left\| \begin{pmatrix} A \\ \mu I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

where $b \in \mathbf{R}^m$ and $\mu^2 = \lambda$.

- Prove that $\lim_{\lambda \rightarrow 0^+} x_\lambda = x_{LS}$ the minimum norm least squares solution (Regardless of the rank of A).