MATH 513 Spring 2009, Homework 4, Addendum

This addendum focuses on how *sparsity* improves the performance of doing Gaussian elimination, and working with matrices in general.

First of all, here's a definition:

Definition A matrix is *sparse* if it has many zero entries.

Exactly what "many" means is usually not a concern in practice – roughly speaking 10% or so of the entries being nonzero is usually a good guess.

Sparse matrices are easier to multiply: you only worry about the nonzero entries! MATLAB knows about sparse matrices, but they're stored differently than regular "full" matrices. To get a sparse matrix, you have a few options:

- You can make a full matrix and convert it to a sparse one:

  ```
  >> A=rand(5)
  >> sparseA = sparse(A)
  ```

- You can also list the row/column/entry triples:

  ```
  >> sparseA=sparse([56 78 39],[43 65 2],[1 2 3],100,100)
  ```

  the above command makes a matrix $A = (a_{ij})$ whose only nonzero elements are $a_{56,43} = 1, a_{78,65} = 2$, and $a_{39,2} = 3$.

Sparse matrices store to row/column indices of the corresponding nonzero elements, and that's it. If the matrix is actually sparse (in the sense of our definition), this is good. However, if the matrix really doesn't have too many zeros, it isn't so good.

MATLAB has a command

```
>> whos
```

that tells you how much memory is being used to store each of your variables.

Problem 1 Create a few sparse matrices (possibly write a function) to fill out the following table:

Rows	Columns	Number of nonzeros	Size in memory
10	10	5	?
10	10	10	?
10	10	20	?
10	10	50	?
10	10	80	?
100	100	50	?
100	100	100	?
100	100	500	?
100	100	900	?

What can you conclude about the storage-efficiency of sparse matrices?

Definition A *banded* matrix $B = (b_{ij})$ is one for which there is a $k > 0$ for which $b_{ij} = 0$ when $|i - j| > k$.

All of the nonzero entries are confined to a band around the diagonal.

Problem 2 Why would a banded matrix be better for elimination than a more general sparse matrix?

Problem 3 Show that the LU factorization of a banded matrix is still banded. See page 453 for a picture.

Problem 4 The book points out that the number of operations needed to LU factor a banded matrix is proportional to the matrix size. Write a Matlab program to demonstrate this using some random sparse banded matrices of various sizes. You should show how the performance changes when the band gets really wide (so the matrix isn't so sparse).