Rob's Spring 2011 Math313/513 at Penn, Computer Problem Set 2

The next MATLAB/Octave assignment focuses on how *sparsity* improves the performance of doing Gaussian elimination, and for working with matrices in general. (Please review Moler and read section 9.1 of Strang for some background.) First, let's (roughly) define what we mean:

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

Of course "many" means different things to different people: it is usually not a concern in practice, but if about 90% of the entries are zero, we'd likely say that matrix is sparse.

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

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

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

- We 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 doesn't have many zeros, then it isn't so good.

MATLAB has a command

```
>> whos
```

that tells us how much memory is being used to store each of our 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	?
100	100	50	?
100	100	100	?
100	100	200	?
100	100	500	?

What can we 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 $|i - j| \geq k$.

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. (Please see page 469 of Strang for a picture.)

Problem 4 Strang's 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. Show how the performance changes when the band gets really wide (so the matrix isn't very sparse).

And finally, as before, **please document your work above with a script!**