Assignment #6

Due Wednesday 6 November, 2012 at the start of class

In your solutions to all problems, remember to:

- show me your calculations,
- show me the MATLAB code or commands, if used, and
- check that you have answered the question.

Problem I. In MATLAB or OCTAVE, enter this command to get a 4×4 matrix A:

```
>> A = magic(4)
```

Now do Gauss elimination with partial pivoting *by-hand*, keeping track of the multipliers L along the way, to compute U and L. (Don't worry about keeping track of P, but do the partial pivoting row swaps.) Now get

`http://www.dms.uaf.edu/~bueler/mylu.m`

and do this:

```
>> [p, Z] = mylu(A)
```

Recover U and L from Z, and show that your by-hand result for these triangular matrices is the same.

Problem II. **(a)** Consider the example *permutation matrix*

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

By applying it to two vectors of your choice, demonstrate that $P\mathbf{v}$ is simply the entries of $\mathbf{v}$ in a different order, that is, *permuted*.

(b) The matrix P^{-1} is easy to find: it is the permutation that undoes the effect of the P permutation. Using that idea only, and without doing arithmetic, compute P^{-1}. (*Hint*: Of course you can confirm your answer with MATLAB, but give me a clear explanation of why P^{-1} has those particular entries.)

(c) I claim that if I enter

$$A = \begin{pmatrix} 0 & 0 & 0 & 3 \\ 3 & 2 & 1 & 4 \\ 0 & 0 & 2 & 1 \\ 0 & 5 & 4 & 3 \end{pmatrix}$$

into MATLAB or OCTAVE then the LU decomposition from Gauss elimination with partial pivoting is obvious. In particular, without running any codes or doing any by-hand arithmetic, explain how to get U and P and L so that $PA = LU$, and do it concretely. (*Hint*: Only P will require any thought or care, but no arithmetic. You will see that L is trivial; explain why.) Finally, confirm your understanding with MATLAB, namely enter A and do

```
>> [L,U,P] = lu(A)
```

Problem III. **(a)** Assume A, L, and U are $n \times n$ matrices, and that L is lower triangular while U is upper triangular. I claim that doing a forward substitution to solve $L\mathbf{y} = \mathbf{b}$ *and* a back substitution to solve $U\mathbf{x} = \mathbf{y}$ is essentially the same cost as just doing the multiplication $A\mathbf{v}$ for some vector $\mathbf{v}$. By referring to previous result or statements in the book, explain this claim about cost quantitatively.

(b) *Why the above?* In class there was the very good question: "If I have multiple right hand sides $\mathbf{b}_1, \dots, \mathbf{b}_k$ isn't it efficient to compute A^{-1} and then multiply to solve the systems, $\mathbf{x}_1 = A^{-1}\mathbf{b}_1, \dots, \mathbf{x}_k = A^{-1}\mathbf{b}_k$?" My answer was that it is even more efficient to compute $A = LU$ and then solve the systems $L(U\mathbf{x}_1) = \mathbf{b}_1, \dots, L(U\mathbf{x}_k) = \mathbf{b}_k$ by forward and back substitutions. Explain quantitatively why this is quicker.

Problem IV. Set up an experiment in MATLAB to measure the 2-norm condition number of random 100×100 matrices with entries which are normal random variables with zero mean and variance 1. Can you find such a matrix with condition number at least 10^4? (*Hint*: This is two or three lines of MATLAB only! Start with `help randn` and `help cond`. Of course I do not want to see the entries of any of the matrices!)

The remaining problems are from Chapter 7 of the textbook *Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms* by Greenbaum and Chartier:

Exercise 9 on page 176.

Exercise 14 on page 178.