Assignment # 8

Due Monday 27 April at start of class

Exercise 18.1.

Exercise 20.3(a).

Exercise 21.1.

Exercise 22.1.

Exercise 24.1.

P17. **(a)** Write an algorithm, and implement it as a function in MATLAB|OCTAVE, which takes as input a square $m \times m$ matrix A and computes $PA = LU$ by partial pivoting. However, unlike Algorithm 21.1, your algorithm will not actually move rows in memory. Instead the computation is "in place". No more memory will be used to store L and U than the memory already used to store A. Clearly some additional memory must be used to store P. However, your algorithm will use only m additional memory locations to store a permutation vector of integers.

Thus your algorithm should take A as input and produce a new matrix Z which has all of the numbers l_{jk} and u_{jk} (in a nontrivial pattern, generally). And it will return an m-vector p which can be used for indexing the rows of a matrix. In particular, assuming your function is called "`mylu`", you will be able to use it this way to recover matrices L, U, P:

```
>> [p,Z] = mylu(A);
>> P = eye(m,m);  P = P(p,:)
>> U = triu(Z(p,:))
>> L = tril(Z(p,:),-1) + diag(ones(m,1))
```

Demonstrate that your algorithm works by applying it to the matrix A given by (21.2) in the text. Compare the result to MATLAB|OCTAVE's `lu` applied to the same matrix: `[L,U,P]=lu(A)`.

(b) Now write another function "`myslash`" which solves square systems $Ax = b$ by using `mylu`, and without using more memory than is needed to store A. That is, `myslash` gets p and Z from `mylu` and solves the system without building L or U or P. Show it works by comparing performance on some nontrivial 4×4 system, comparing to result from "\":

```
>> x = myslash(A,b)
>> x = A\b
```

(c) By now you understand pretty well what the built-in backslash command does in the case of a generic square matrix A. What does the built-in backslash command to solve the system $Ax = b$ if

- A is square and triangular?
- A is symmetric and positive definite?
- A is nonsquare, $A \in \mathbb{C}^{m \times n}$ and $m > n$?

In each case, completely describe what is done by saying which algorithms in TREF & BAU are applied, in what order, and why. (What does the built-in backslash command do if A is symmetric, $A_{11} > 0$, but A is *not* positive definite? Which is to say, how does the backslash command determine if A is positive definite?)