Computer problems from the textbook contain "CP" in the exercise number. For these problems, remember to adequately label all plots and include code that you have written along with your solutions. All code that you include should be properly explained. Do all other problems by hand and make sure to your work. A clear and complete presentation of your solutions is required for full credit.

1. (Sauer §2.4, #2) Find the $PA = LU$ factorization (using partial pivoting) of the following matrices:

   (a) $\begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix}$

   (b) $\begin{bmatrix} 0 & 1 & 3 \\ 2 & 1 & 1 \\ -1 & -1 & 2 \end{bmatrix}$

   (c) $\begin{bmatrix} 1 & 2 & -3 \\ 2 & 4 & 2 \\ -1 & 0 & 3 \end{bmatrix}$

   (d) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ -2 & 1 & 0 \end{bmatrix}$

2. (Sauer §2.4, #4) Solve the system by finding the $PA = LU$ factorization and then carrying out the two-step back substitution.

   (a) $\begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$

   (b) $\begin{bmatrix} -1 & 0 & 1 \\ 2 & 1 & 1 \\ -1 & 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 17 \\ 3 \end{bmatrix}$

3. Modify your function for LU factorization from last week's homework to implement the $PA = LU$ factorization using partial pivoting. Your function should take only a matrix A as its input and return the matrices P, L, and U as its outputs (*i.e.* its declaration should be `function [P,L,U] = LUPartialPivot(A)`). You should display an error if the matrix is singular. Check your function on the matrices from Exercise 1, and include a listing of your function in your solutions.

4. Write a function which uses the $PA = LU$ factorization you wrote in the previous exercise, along with two-step back substitution, to solve a linear system of equations $Ax = b$. Your function should take a matrix A and a vector b as its inputs and return the solution x as its output (*i.e.* its declaration should be `function x = LUPartialPivotSolve(A,b)`). Check your function on the systems from Exercise 2.

Now, read **Reality Check 2** starting on page 102 and complete the following exercises.

5. Write a Matlab program to define the structure matrix A in (2.34). Then, using the Matlab \ command or code of your own code from Exercise 4, solve the system for the displacements y_i using $n = 10$ grid steps.

6. Plot the solution from Exercise 5 against the correct solution

$$y(x) = \frac{x^2\left(x^2 - 4Lx + 6L^2\right)}{24EI}\, f,$$

where $f = f(x)$ is the constant defined above. Check the error at the end of the beam, $x = L$ meters. In this simple case the derivative approximations are exact, so your error should be near machine roundoff.

7. Rerun the calculation in Exercise 5 for $n = 10 \times 2^k$, where $k = 1, \ldots, 11$. Make a table of the errors at $x = L$ for each n. For which n is the error smallest? Why does the error begin to increase with n after a certain point? You may want to make an accompanying table of the condition number of A as a function of n to help answer the last question. To carry out this step for large k, you may need to ask Matlab to store the matrix A as a sparse matrix to avoid running out of memory. To do this, just initialize A with the command `A = sparse(n,n)`, and proceed as before. We will discuss sparse matrices in more detail next week.