COMP 350 Numerical Computing

Assignment #3: Solving linear systems.

Date given: Monday, October 2. Date due: 5pm, Monday, October 16, 2017

(To be marked by Sitao Luan and Yangchao Yi (sitao.luan, yangchao.yi@mail.mcgill.ca))

1. (a) (3 points) Solve the following system using GEPP (Gaussian elimination with partial pivoting):

$$\begin{bmatrix} -6 & -4 & 46 & 32 \\ 12 & 24 & -12 & -24 \\ 6 & 36 & 6 & 12 \\ 6 & 4 & -1 & 31 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -6 \\ 0 \\ 18 \\ 15 \end{bmatrix}$$

   Show intermediate matrices, vectors and multipliers at each step.

   Note: Do the computations by your hands and don't consider any rounding errors.

   (b) (2 points) Compute the LU factorization of the matrix in the previous question with partial pivoting: $PA = LU$. You have to show intermediate results at each step. This question and the previous one can be answered together.

2. (MATLAB programming) Consider the system of $2n+1$ equations:

$$\begin{bmatrix} d_1 & & & & & & a_{2n+1} \\ & d_2 & & & & a_{2n} & \\ & & \ddots & & \iddots & & \\ & & & d_{n+1} & & & \\ & & \iddots & & \ddots & & \\ & a_2 & & & & d_{2n} & \\ a_1 & & & & & & d_{2n+1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n+1} \\ \vdots \\ x_{2n} \\ x_{2n+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n+1} \\ \vdots \\ b_{2n} \\ b_{2n+1} \end{bmatrix}$$

   (a) (8 points) Write two function M-files to solve this system using the GENP and GEPP approaches, respectively. Your programs have to make use of the structure of this system so that they do not do unnecessary computations and do not waste storage. How many flops are involved in your programs? How many memory locations are needed?

   (b) (3 points) Write a script M-file to test your two programs on the **same** data: random elements a_i, d_i produced by MATLAB built-in function `randn` (note that a_{n+1} should be equal to d_{n+1}), and b_i defined by $b_i = d_i + a_{2n+2-i}$ for $i \neq n+1$, and $b_{n+1} = d_{n+1}$. Notice that the exact solution $x = [1, 1, \ldots, 1]^T$. Let x_{np} and x_{pp} be the computed solutions by your two programs, respectively. Compute $\|x - x_{np}\|_2/\|x\|_2$, $\|x - x_{pp}\|_2/\|x\|_2$ and $\epsilon\|A\|_2\|A^{-1}\|_2$. Here for the test purpose in MATLAB A is used to store the coefficient matrix. In your test, you may take $n = 9$. Comment on your results.

   Note: Check MTALAB built-in functions or constants `norm`, `cond`, `ones` and `eps` to see how to compute or get related quantities.

   (c) (2 points) Now you change d_1 in your previous test example to 10^{-15}, and correspondingly you change b_1 so that b_1 still satisfies $b_1 = d_1 + a_{2n+1}$. Do the same thing as in 2(b).

   (d) (2 points) Now you make other changes in your test example: change a_1 to 10^{-8}, and correspondingly change b_{2n+1} so that $b_{2n+1} = a_1 + d_{2n+1}$ still holds. Again do the same thing as in 2(b).

Print out the data and computed results.