Assignment 3 - Solving linear systems
COMP 350 - Numerical Computing
Prof. Chang Xiao-Wen
Fall 2018

LE, Nhat Hung
McGill ID: 260793376
Date: October 9, 2018
Due date: October 11, 2018

1. a. b. (5 points) Solve the following system using GEPP:

$$\left[\begin{array}{cccc|c} 3 & 1 & -3 & 15 & 17 \\ -1 & -3 & -4 & 4 & 16 \\ 4 & 4 & -4 & 8 & 8 \\ 2 & -2 & -4 & 8 & 16 \end{array}\right] \sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ -1 & -3 & -4 & 4 & 16 \\ 3 & 1 & -3 & 15 & 17 \\ 2 & -2 & -4 & 8 & 16 \end{array}\right] \quad mult = \frac{-1}{4}$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ -\frac{1}{4} & -2 & -5 & 6 & 18 \\ 3 & 1 & -3 & 15 & 17 \\ 2 & -2 & -4 & 8 & 16 \end{array}\right] \quad mult = \frac{3}{4}$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ -\frac{1}{4} & -2 & -5 & 6 & 18 \\ \frac{3}{4} & -2 & 0 & 9 & 11 \\ 2 & -2 & -4 & 8 & 16 \end{array}\right] \quad mult = \frac{2}{4}$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ -\frac{1}{4} & -2 & -5 & 6 & 18 \\ \frac{3}{4} & -2 & 0 & 9 & 11 \\ \frac{1}{2} & -4 & -2 & 4 & 12 \end{array}\right]$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ \frac{1}{2} & -4 & -2 & 4 & 12 \\ \frac{3}{4} & -2 & 0 & 9 & 11 \\ -\frac{1}{4} & -2 & -5 & 6 & 18 \end{array}\right] \quad mult = \frac{-2}{-4}$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ \frac{1}{2} & -4 & -2 & 4 & 12 \\ \frac{3}{4} & \frac{1}{2} & 1 & 7 & 5 \\ -\frac{1}{4} & -2 & -5 & 6 & 18 \end{array}\right] \quad mult = \frac{-2}{-4}$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ \frac{1}{2} & -4 & -2 & 4 & 12 \\ \frac{3}{4} & \frac{1}{2} & 1 & 7 & 5 \\ -\frac{1}{4} & \frac{1}{2} & -4 & 4 & 12 \end{array}\right]$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ \frac{1}{2} & -4 & -2 & 4 & 12 \\ -\frac{1}{4} & \frac{1}{2} & -4 & 4 & 12 \\ \frac{3}{4} & \frac{1}{2} & 1 & 7 & 5 \end{array}\right] \qquad mult = \frac{1}{-4}$$

$$\sim \left[\begin{array}{cccc|c} 4 & 4 & -4 & 8 & 8 \\ \frac{1}{2} & -4 & -2 & 4 & 12 \\ -\frac{1}{4} & \frac{1}{2} & -4 & 4 & 12 \\ \frac{3}{4} & \frac{1}{2} & -\frac{1}{4} & 8 & 8 \end{array}\right]$$

Then,

$$\begin{bmatrix} 4 & 4 & -4 & 8 \\ 0 & -4 & -2 & 4 \\ 0 & 0 & -4 & 4 \\ 0 & 0 & 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 12 \\ 8 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -2 \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{4} & \frac{1}{2} & 1 & 0 \\ \frac{3}{4} & \frac{1}{2} & -\frac{1}{4} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 4 & 4 & -4 & 8 \\ 0 & -4 & -2 & 4 \\ 0 & 0 & -4 & 4 \\ 0 & 0 & 0 & 8 \end{bmatrix}$$

2. (3 points) Suppose we have a complex linear system $Ax = b$, where $A \in C^{n\times n}$ is nonsingular and $b \in C^n$. Can you solve it in real arithmetic operations? What is the cost?

A complex number

$$a + ib$$

can be written as

$$\begin{pmatrix} a & -b \\ b & a \end{pmatrix}.$$

And the complex linear system

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

has an equivalent $2n \times 2n$ real linear system

$$\begin{bmatrix} a^r_{1,1} & -a^i_{1,1} & a^r_{1,2} & -a^i_{1,2} & \cdots & a^r_{1,n} & -a^i_{1,n} \\ a^i_{1,1} & a^r_{1,1} & a^i_{1,2} & a^r_{1,2} & & a^i_{1,n} & a^r_{1,n} \\ a^r_{2,1} & -a^i_{2,1} & a^r_{2,2} & -a^i_{2,2} & \cdots & a^r_{2,n} & -a^i_{2,n} \\ a^i_{2,1} & a^r_{2,1} & a^i_{2,2} & a^r_{2,2} & & a^i_{2,n} & a^r_{2,n} \\ \vdots & & \vdots & & \ddots & \vdots & \\ a^r_{n,1} & -a^i_{n,1} & a^r_{n,2} & -a^i_{n,2} & \cdots & a^r_{n,n} & -a^i_{n,n} \\ a^i_{n,1} & a^r_{n,1} & a^i_{n,2} & a^r_{n,2} & & a^i_{n,n} & a^r_{n,n} \end{bmatrix} \begin{bmatrix} x^r_1 \\ x^i_1 \\ x^r_2 \\ x^i_2 \\ \vdots \\ x^r_4 \\ x^i_4 \end{bmatrix} = \begin{bmatrix} b^r_1 \\ b^i_1 \\ b^r_2 \\ b^i_2 \\ \vdots \\ b^r_4 \\ b^i_4 \end{bmatrix}.$$

We can solve the system with GEPP with the cost of

$$\frac{2}{3}(2n)^3\, flops + \frac{1}{2}(2n)^2\, comparisons = \frac{16}{3}n^3\, flops + 2n^2\, comparisons$$

3. Suppose $A \in R^{n\times n}$ is nonsingular.

a. (4 points) Given $B \in R^{n\times p}$, show how to use the LU factorization with partial pivoting to solve $AX = B$. What is the cost of your method?

This is equivalent to solving $Ax = b$, for p pairs of column vectors x and b .
We can modify the general GEPP algorithm:

Modifications are in bold.

```
for k = 1:n-1
  determine q such that
    |a_qk| = max{|a_kk|, |a_{k+1,k}|, ..., |a_nk|}
  for j = k:n
    do interchange: a_kj <-> a_qj
  end
  for j = 1:p
    do interchange: b_kj <-> b_qj
  end
  for i = k+1:n
    m_ik <- a_ik/a_kk
    for j = k+1:n
      a_ij <- a_ij - m_ik a_kj
    end
    for j = 1:p
      b_ij <- b_ij - m_ik b_kj
    end
  end
end
for k = 1:p
  x_nk <- b_nk/a_nn
```

```
  for i = n-1:-1:1
```
$$x_{ik} \leftarrow (b_{ik} - \sum_{j=i+1}^{n} a_{ij}x_{jk})/a_{ii}$$
```
  end
end
```

The cost of LUPP, same as GEPP, for a pair of column vectors $x,\ b$ is $O(n^3)$. Therefore, the cost for p pairs of x and b is $O(pn^3)$.

b. (5 points) Use lupp.m given in the lecture notes to solve $AX = B$, where

$$10 \times 10 \text{ Hilbert matrix } A = (a_{ij}),\ a_{ij} = 1/(i+j-1);$$
$$10 \times 5\ B = A{*}randn(10,\ 5).$$

First try:

$$\frac{||X_c - X_t||_F}{||X_t||_F} = 1.0253 \times 10^{-4}$$

$$\epsilon||A||_F||A^{-1}||_F = 0.0036$$

$$\frac{||B - AX_c||_F}{||A||_F||X_c||_F} = 4.7452 \times 10^{-17}$$

i. Run your code 10 times.

MATLAB code:

```
disp('||Xc - Xt||_F / ||Xt||_F   eps ||A||_F ||A^-1||_F      ||B - AXc||_F /
(||A||_F ||Xc||_F)     eps');

for k = 1:10
    A = hilb(10);
    X_t = randn(10, 5);
    B = A * X_t;

    [L, U, P] = lupp(A);
    PB = P * B;
    for i = 1:5
        Y(:,i) = gepp(L, PB(:,i));
    end
    for i = 1:5
        X_c(:,i) = gepp(U, Y(:,i));
    end

    a = norm(X_c - X_t,'fro') / norm(X_t,'fro');
```

```
    b = eps * cond(A,'fro');
    c = norm(B - A*X_c,'fro') / (norm(A,'fro') * norm(X_c,'fro'));

    res = [a, b, c, eps];
    g = sprintf('%d                    ', res);
    fprintf('%s\n', g)
end
```

Results:

||Xc - Xt||_F / ||Xt||_F	eps ||A||_F ||A^-1||_F	||B - AXc||_F / (||A||_F ||Xc||_F)	eps
1.614063e-04	3.626334e-03	4.179527e-17	2.220446e-16
7.203691e-05	3.626334e-03	2.528097e-17	2.220446e-16
1.038685e-04	3.626334e-03	4.450611e-17	2.220446e-16
1.182851e-04	3.626334e-03	1.092011e-16	2.220446e-16
9.403553e-05	3.626334e-03	5.763648e-17	2.220446e-16
7.107827e-05	3.626334e-03	4.116562e-17	2.220446e-16
6.308490e-05	3.626334e-03	3.375183e-17	2.220446e-16
2.394096e-04	3.626334e-03	6.035635e-17	2.220446e-16
1.875017e-04	3.626334e-03	6.716765e-17	2.220446e-16
2.435789e-04	3.626334e-03	4.222571e-17	2.220446e-16

Figure 1: Displayed result for program above

ii. Do you see any rough relation between $\|X_c - X_t\|_F / \|X_t\|_F$ and $\varepsilon\|A\|_F\|A^{-1}\|_F$?

$\|X_c - X_t\|_F / \|X_t\|_F$, which is the relative error, is always smaller than $\varepsilon\|A\|_F\|A^{-1}\|_F$, similar to the result seen in class.

iii. Do you see any rough relation between $\|B - AX_c\|_F / (\|A\|_F\|X_c\|_F)$ and ε ?

We observe that $\|B - AX_c\|_F / (\|A\|_F\|X_c\|_F) \lesssim \varepsilon$.
This is derived from the result

$$\|r\| \lesssim \varepsilon \|A\| \|X_c\|$$

with $\|r\| = \|B - AX_c\|$.

c. Computing A^{-1} .
i. (3 points) Show how to use the LU factorization with partial pivoting to compute the inverse of a general $n \times n$ nonsingular matrix A . What is the cost of your method?

We use LU factorization to solve $AX = I$, with $I =$ identity matrix.
This is equivalent to solving n equations $Ax_i = e_i$, where x_i is the i^{th} column of A^{-1} , and e_i is the i^{th} unit vector.

This requires one LU factorization and n pairs of forward and backward substitutions.
The cost is

$$\frac{2}{3}n^3 + n(2n^2) = \frac{8}{3}n^3 flops$$

or about

. $3n^3 flops$