Assignment 4, due before class, Monday June 26, 2023.

Haowen He heh4@rpi.edu

1. Text exercises 2(a), 2(b), and 4 on page 97. Hint: it may be helpful to recall the following formulas valid for 2×2 matrices

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \qquad A^{-1} = \frac{1}{ad-bc}\begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

(2a)

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \|A\|_\infty = \max\{3, 7\} = 7$$

$$A^{-1} = \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix} \qquad \|A^{-1}\|_\infty = \max\{3, 2\} = 3$$

$$\Rightarrow \kappa_\infty(A) = 7 \times 3 = \boxed{21}$$

(2b)

$$A = \begin{pmatrix} 1 & 2.01 \\ 3 & 6 \end{pmatrix} \qquad \|A\|_\infty = \max\{3.01, 9\} = 9$$

$$A^{-1} = \begin{pmatrix} -200 & 67 \\ 100 & -100/3 \end{pmatrix} \qquad \|A^{-1}\|_\infty = \max\{267, 400/3\} = 267$$

$$\Rightarrow \kappa_\infty(A) = 9 \times 267 = \boxed{2403}$$

(4a) $\left(\begin{array}{cc|c} 1 & 2 & 1 \\ 2 & 4.01 & 2 \end{array}\right) \xrightarrow{R_2 \to R_2 - 2R_1} \left(\begin{array}{cc|c} 1 & 2 & 1 \\ 0 & 0.01 & 0 \end{array}\right) \Rightarrow \begin{cases} 0.01 x_2 = 0 \Rightarrow x_2 = 0 \\ x_1 + 2x_2 = 1 \Rightarrow x_1 = 1 \end{cases}$

Actual solution: $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$\Rightarrow \quad x_c = (-1, 1)$

Forward error: $\|x - x_c\|_\infty = \max\{2, 1\} = 2$

Backward error: $\|b - Ax_c\|_\infty = \max\{0, 0.01\} = 0.01$

$$\text{Error magnification factor} = \frac{\text{rel. forward err}}{\text{rel. backward err}}$$

$$= \frac{2/1}{0.01/2}$$

$$= \boxed{400}$$

(4b) $\Rightarrow \quad x_c = (3, -1)$

Forward error: $\|x - x_c\|_\infty = \max\{|1-2|, 1\} = 2$

Backward error: $\|b - Ax_c\|_\infty = \max\{0, 0.01\} = 0.01$

$$\text{Error magnification factor} = \frac{\text{rel. forward err}}{\text{rel. backward err}}$$

$$= \frac{2/1}{0.01/2}$$

$$= \boxed{400}$$

(4c) $\Rightarrow x_c = (2, -1/2)$

Forward error : $\|x - x_c\|_\infty = \max\{|-1|, 1/2\} = 1$

Backward error : $\|b - Ax_c\|_\infty = \max\{0, 0.005\} = 0.005$

$$\text{Error magnification factor} = \frac{\text{rel. forward err}}{\text{rel. backward err}}$$

$$= \frac{1/1}{0.005/2}$$

$$= \boxed{400}$$

2. Computer problem 1 on page 98. For this problem, use $A_{ij} = 3/(i+2j-1)$ instead of the formula for A_{ij} given in the text. Be sure to include a brief comment on the result of the comparison. Is the result expected and why.

```
function [] = p2(n)
A=zeros(n);
x=ones(n,1);
for i=1:n
    for j=1:n
        A(i,j)=3/(i+2*j-1);
    end
end
b=A*x;
xc=A\b;

fe=norm(x-xc, "inf");
rfe=fe/norm(x, "inf");

be=norm(b-A*xc, "inf");
rbe=be/norm(b, "inf");

emf=rfe/rbe;

condnum=cond(A, inf);

disp(["forward error: ", num2str(fe)])
disp(["relative forward error: ", num2str(rfe)])
disp(["backward error: ", num2str(be)])
disp(["relative backward error: ", num2str(rbe)])
disp(["Error Magnification factor: ", num2str(emf)])
disp(["Condition number of A: ", num2str(condnum)])
```

```
p2(6)
p2(10)
```

n = 6

```
"forward error: "     "1.8479e-09"

"relative forward error: "     "1.8479e-09"

"backward error: "     "2.2204e-16"

"relative backward error: "     "6.042e-17"

"Error Magnification factor: "     "30584566.425"

"Condition number of A: "     "70342013.954"
```

n = 10

```
"forward error: "     "0.0028717"

"relative forward error: "     "0.0028717"

"backward error: "     "4.4409e-16"

"relative backward error: "     "1.0108e-16"

"Error Magnification factor: "     "28410438423259.55"

"Condition number of A: "     "131321934554336.8"
```

3. Text exercises 2(a), and 4(b) on page 106. These problems are to be done using hand calculations.

(2a) $A = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ -1 & 1 & -1 \end{pmatrix} \Rightarrow$ Row 2 has the largest pivot value $\Rightarrow P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

$$PA = \begin{pmatrix} 2 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & 1 & 0 \end{pmatrix} \xrightarrow[R_3 \to R_3 - \frac{1}{2}R_1]{R_2 \to R_2 + \frac{1}{2}R_1} \begin{pmatrix} 2 & 1 & -1 \\ 0 & 3/2 & -3/2 \\ 0 & 1/2 & 1/2 \end{pmatrix} \xrightarrow{R_3 \to R_3 - \frac{1}{3}R_2}$$

$$\begin{pmatrix} 2 & 1 & -1 \\ 0 & 3/2 & -3/2 \\ 0 & 0 & 1 \end{pmatrix} = U \quad \Rightarrow \quad L = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & 1/3 & 1 \end{pmatrix}$$

$$PA = LU$$

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & 1/3 & 1 \end{pmatrix}\begin{pmatrix} 2 & 1 & -1 \\ 0 & 3/2 & -3/2 \\ 0 & 0 & 1 \end{pmatrix}$$

(4b) $A = \begin{pmatrix} -1 & 0 & 1 \\ 2 & 1 & 1 \\ -1 & 2 & 0 \end{pmatrix} \Rightarrow$ Row 2 has the largest pivot value $\Rightarrow P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

$$PA = \begin{pmatrix} 2 & 1 & 1 \\ -1 & 2 & 0 \\ -1 & 0 & 1 \end{pmatrix} \xrightarrow[R_3 \to R_3 + \frac{1}{2}R_1]{R_2 \to R_2 + \frac{1}{2}R_1} \begin{pmatrix} 2 & 1 & 1 \\ 0 & 5/2 & 1/2 \\ 0 & 1/2 & 3/2 \end{pmatrix} \xrightarrow{R_3 \to R_3 - \frac{1}{5}R_2}$$

$$\begin{pmatrix} 2 & 1 & -1 \\ 0 & 5/2 & 1/2 \\ 0 & 0 & 7/5 \end{pmatrix} = U \quad \Rightarrow \quad L = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/2 & 1/5 & 1 \end{pmatrix}$$

back substitution: $Pb = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} -2 \\ 17 \\ 3 \end{pmatrix} = \begin{pmatrix} 17 \\ 3 \\ -2 \end{pmatrix}$

$$Ly = Pb \Rightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 17 \\ -1/2 & 1 & 0 & 3 \\ -1/2 & 1/5 & 1 & -2 \end{array}\right) \Rightarrow \begin{cases} y_1 = 17 \\ y_2 = 23/2 = 11.5 \\ y_3 = 21/5 = 4.2 \end{cases}$$

$$Ux = y \Rightarrow \left(\begin{array}{ccc|c} 2 & 1 & -1 & 17 \\ 0 & 5/2 & 1/2 & 11.5 \\ 0 & 0 & 7/5 & 4.2 \end{array}\right) \Rightarrow \begin{cases} x_3 = 3 \\ x_2 = 4 \\ x_1 = 5 \end{cases}$$

$$x = \begin{pmatrix} 5 \\ 4 \\ 3 \end{pmatrix}.$$

4. Write a `Matlab` code (following the algorithm discussed in class) that takes an $n \times n$ matrix A as input and returns P, L and U such that $PA = LU$. Your code should use row elimination with partial pivoting. Test your code using the matrix in text exercise $2(a)$ above and on a matrix generated using `A=rand(6,6)`. In both cases, compute `norm(P*A-L*U)` to verify that your code is working correctly.

```
function [P,L,U] = myPLU(A)

[m,n]=size(A);
if m~=n
    error("Matrix A is not square!")
end

P=eye(n);
L=eye(n);
U=A;

for k=1:n-1
    [pv,p] = max(abs(U(k:n, k)));
    p=p+k-1;
    if p~=k
        P([p,k],:)=P([k,p],:);
        U([p,k],:)=U([k,p],:);
        if k>1
            L([p,k],1:k-1)=L([k,p],1:k-1);
        end
    end

    for i=k+1:n
        m=U(i,k)/U(k,k);
        L(i,k)=m;
        for j=k:n
            U(i,j)=U(i,j)-U(k,j)*m;
        end
    end
end
```

```
% Problem 4 Part (i)
A=[1,1,0;2,1,-1;-1,1,-1];
[P,L,U]=myPLU(A)
fprintf("norm(P*A-L*U) = %g\n", norm(P*A-L*U))
```

```
P =

     0     1     0
     0     0     1
     1     0     0


L =

    1.0000         0         0
   -0.5000    1.0000         0
    0.5000    0.3333    1.0000


U =

    2.0000    1.0000   -1.0000
         0    1.5000   -1.5000
         0         0    1.0000

norm(P*A-L*U) = 0
```

```
% Problem 4 Part (iI)
A=rand(6,6);
[P,L,U]=myPLU(A)
fprintf("norm(P*A-L*U) = %g\n", norm(P*A-L*U))
```

```
P =

     0     0     1     0     0     0
     0     0     0     1     0     0
     0     1     0     0     0     0
     1     0     0     0     0     0
     0     0     0     0     0     1
     0     0     0     0     1     0


L =

    1.0000         0         0         0         0         0
    0.3766    1.0000         0         0         0         0
    0.2621    0.5457    1.0000         0         0         0
    0.8763    0.4409   -0.2799    1.0000         0         0
    0.2702    0.6511   -0.3456    0.6993    1.0000         0
    0.2116    0.7315    0.7278   -0.9116    0.4772    1.0000
```

```
U =

    0.9293    0.3517    0.7572    0.5308    0.0119    0.1656
         0    0.6984    0.4685    0.5793    0.3326    0.5396
    0.0000   -0.0000   -0.1683   -0.4013    0.2847    0.1906
         0         0         0   -0.7570    0.4915   -0.0185
         0         0         0         0    0.3292    0.3368
         0         0         0         0         0   -0.4831

norm(P*A-L*U) = 1.74528e-16
```

5. Consider the tridiagonal matrices

$$A = \begin{bmatrix} 2 & 3 & 0 \\ -1 & -5/2 & 2 \\ 0 & -1/4 & 3/2 \end{bmatrix}, \qquad B = \begin{bmatrix} 4 & -2 & 0 \\ -2 & 3 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

(a) Use direct factorization (as discussed in class) to find (by hand calculations) a lower triangular matrix L (with ones on the main diagonal) and an upper triangular matrix U such that $A = LU$.

(b) Show that the symmetric matrix B is positive definite by writing $\mathbf{x}^T B\mathbf{x}$ as a sum of squares involving the three components, (x_1, x_2, x_3), of $\mathbf{x}$. Be sure to show that $\mathbf{x}^T B\mathbf{x} = 0$ implies that $x_1 = x_2 = x_3 = 0$.

(a) $$A = \begin{pmatrix} 2 & 3 & 0 \\ -1 & -5/2 & 2 \\ 0 & -1/4 & 3/2 \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & u_{22} + l_{21}u_{12} & u_{23} + l_{21}u_{13} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{pmatrix}$$

$$u_{11} = 2 \Rightarrow \begin{cases} l_{21}u_{11} = -1 \Rightarrow l_{21} = -1/2 \\ l_{31}u_{11} = 0 \Rightarrow l_{31} = 0 \end{cases}$$

$$u_{12} = 3 \Rightarrow \begin{cases} u_{22} + l_{21}u_{12} = -\frac{5}{2} \Rightarrow u_{22} = -1 \\ l_{31}u_{12} + l_{32}u_{22} = -\frac{1}{4} \Rightarrow l_{32} = \frac{1}{4} \end{cases}$$

$$u_{13} = 0 \Rightarrow \begin{cases} u_{23} + l_{21}u_{13} = 2 \Rightarrow u_{23} = 2 \\ l_{31}u_{13} + l_{32}u_{23} + u_{33} = \frac{3}{2} \Rightarrow u_{33} = 1 \end{cases}$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 1/4 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = \begin{pmatrix} 2 & 3 & 0 \\ 0 & -1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

(b) $x^T B x = 4x_1^2 - 4x_1x_2 + 3x_2^2 + 2x_2x_3 + x_3^2$

$$= (4x_1^2 - 4x_1x_2 + x_2^2) + x_2^2 + (x_2^2 + 2x_2x_3 + x_3^2)$$

$$= (2x_1 - x_2)^2 + x_2^2 + (x_2 + x_3)^2 \geqslant 0.$$

Now as $x^T B x$ is the sum of squares, it follows that $x^T B x = 0$ if and only if $2x_1 - x_2 = x_2 = x_2 + x_3 = 0$, which implies that all three components of x are zero.

6. Let

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} \frac{1}{4}(x_1 - x_2 - 1) \\ \alpha(x_1^2 - 1) + x_2 \end{bmatrix}, \qquad \mathbf{F}(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}x_1 - x_2 - 1 \\ x_1^2 - x_2^2 - 1 \end{bmatrix}.$$

(a) Show that $\mathbf{G}(\mathbf{x})$ has a fixed point $\mathbf{x}^\star = (-1, 2)$ for any value of the constant α. Analyze the eigenvalues of the Jacobian matrix $J = \partial\mathbf{G}/\partial\mathbf{x}$ to determine whether the fixed-point iteration $\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k)$ is locally convergent when $(i)\alpha = -1/4$ or when (ii) $\alpha = 1/2$.

(b) Let $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ be the two component functions of $\mathbf{F}(\mathbf{x})$. Plot the graphs of $f_1 = 0$ and $f_2 = 0$ in the $x_1 - x_2$ plane to estimate the roots $\mathbf{x} = \mathbf{r}$ of $\mathbf{F}$. Write a `Matlab` code to determine all roots of $\mathbf{F}$ using Newton's method. Output $\mathbf{x}_k$, the k^{th} iterate in Newton's method, and $||\mathbf{x}_k - \mathbf{x}_{k-1}||$ to verify quadratic convergence.

(a) $G(x^*) = \begin{pmatrix} \frac{1}{4}(-1-2-1) \\ \alpha(1-1)+2 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix} = x^*$, thus x^* is a fixed point of $G(x)$ regardless of the value of constant α

$$J = \begin{pmatrix} \frac{\partial G_1}{\partial x_1} & \frac{\partial G_1}{\partial x_2} \\ \frac{\partial G_2}{\partial x_1} & \frac{\partial G_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1/4 & -1/4 \\ 2\alpha x_1 & 1 \end{pmatrix} \Rightarrow J_{(-1,2)} = \begin{pmatrix} 1/4 & -1/4 \\ -2\alpha & 1 \end{pmatrix}$$

At $\alpha = -\frac{1}{4}$, $J_1 = \begin{pmatrix} 1/4 & -1/4 \\ 1/2 & 1 \end{pmatrix}$

$$\det(J_1 - \lambda I) = \begin{vmatrix} \frac{1}{4}-\lambda & -1/4 \\ 1/2 & 1-\lambda \end{vmatrix} = \lambda^2 - \frac{5}{4}\lambda + \frac{1}{4} + \frac{1}{8} = \lambda^2 - \frac{5}{4}\lambda + \frac{3}{8}$$

set $\lambda^2 - \frac{5}{4}\lambda + \frac{3}{8} = 0 \Rightarrow (\lambda - \frac{1}{2})(\lambda - \frac{3}{4}) = 0$

$$\lambda_1 = \frac{1}{2}, \quad \lambda_2 = \frac{3}{4}$$

$$\max\{|\lambda_i|\} < 1$$

Therefore, when $\alpha = -1/4$, the fixed-point iteration is locally convergent.

At $\alpha = \frac{1}{2}$, $J_2 = \begin{pmatrix} 1/4 & -1/4 \\ -1 & 1 \end{pmatrix}$

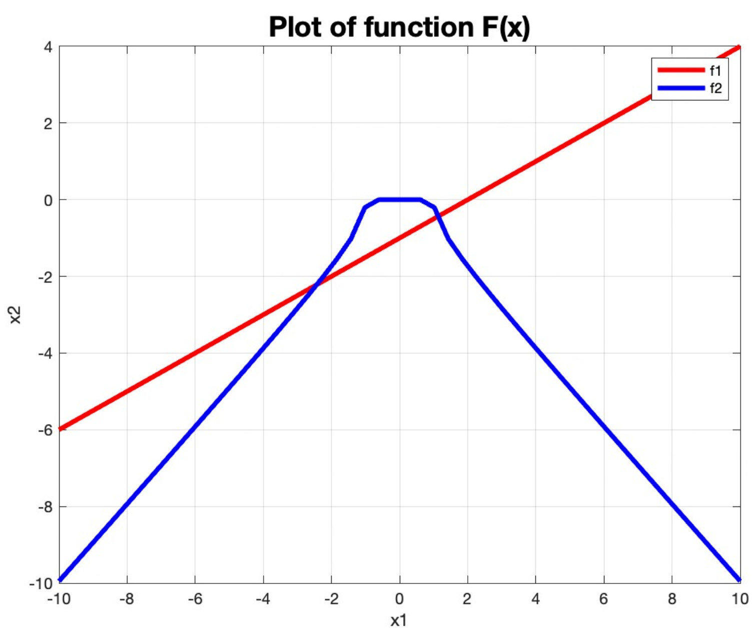
$$\det(J_2 - \lambda I) = \begin{vmatrix} \frac{1}{4} - \lambda & -1/4 \\ 1/2 & 1-\lambda \end{vmatrix} = \lambda^2 - \frac{5}{4}\lambda + \frac{1}{4} - \frac{1}{4} = \lambda^2 - \frac{5}{4}\lambda$$

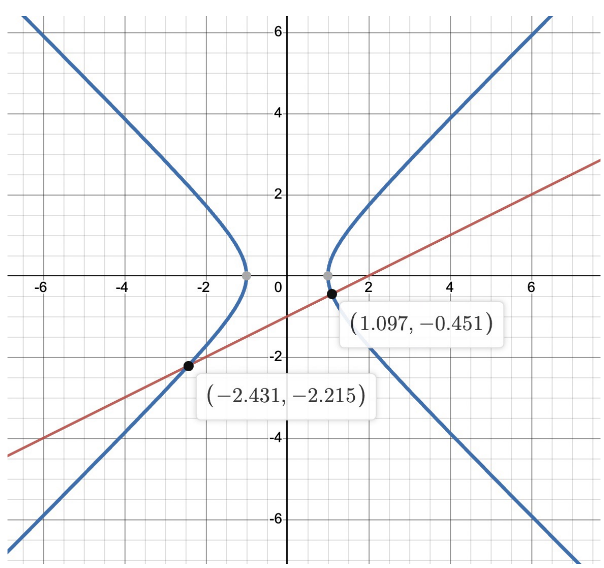
set $\lambda^2 - \frac{5}{4}\lambda = 0 \Rightarrow \lambda(\lambda - \frac{5}{4}) = 0$

$$\lambda_1 = 0, \quad \lambda_2 = \frac{5}{4}$$

$$\max\{|\lambda_i|\} > 1$$

Therefore, when $\alpha = \frac{1}{2}$, the fixed-point iteration is NOT locally convergent.





```
function x=myNewtonNL(f,fp,x0)

tol=1e-12;
nMax=10;
xpre=[0;0];
x=x0;
n=0;

while n<nMax && abs(norm(x-xpre))>tol
    xpre=x;
    s=fp(x)\-f(x);
    x=s+x;
    n=n+1;
    fprintf("iteration %d x=[%3.5f; %3.5f;] err=%8e\n",n,x(1,:),x(2,:),norm(x-
xpre))
end
end
```

```
f=@(x)[1/2*x(1)-x(2)-1; x(1)^2-x(2)^2-1];
fp=@(x)[1/2 -1; 2*x(1) -2*x(2)];
x0=[1; -0.5];
myNewtonNL(f,fp,x0)
x0=[-2.5; -2];
myNewtonNL(f,fp,x0)
```

```
iteration 1 x=[1.10000; -0.45000;] err=1.118034e-01
iteration 2 x=[1.09717; -0.45142;] err=3.164247e-03
iteration 3 x=[1.09717; -0.45142;] err=2.538619e-06
iteration 4 x=[1.09717; -0.45142;] err=1.633781e-12
iteration 5 x=[1.09717; -0.45142;] err=2.482534e-16

ans =

    1.0972
   -0.4514

iteration 1 x=[-2.41667; -2.20833;] err=2.243819e-01
iteration 2 x=[-2.43056; -2.21528;] err=1.552825e-02
iteration 3 x=[-2.43050; -2.21525;] err=6.113484e-05
iteration 4 x=[-2.43050; -2.21525;] err=9.476218e-10
iteration 5 x=[-2.43050; -2.21525;] err=4.440892e-16

ans =

   -2.4305
   -2.2153
```