Computer assignment 2

For Scientific Computing MA3012/MA7012

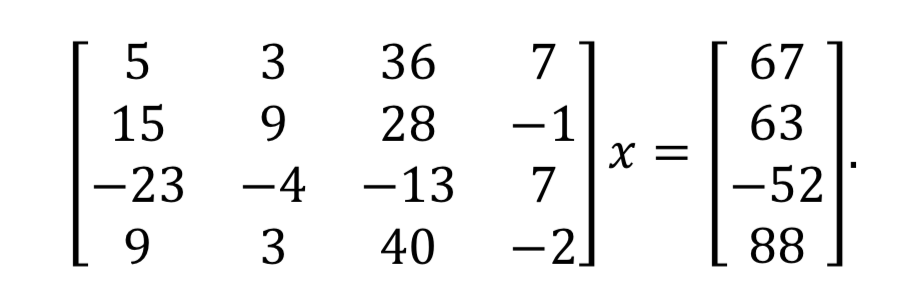
# Gleb Vorobchuk

# Leicester

# October 2017

# Task №1.

Consider the system of linear equations

A)Try to solve this system using Matlab function gaussel.

Code of the function:

function [x] = gaussel(A,b)

% [x] = gaussel(A,b)

%

% This subroutine will perform Gaussian elimination

% and back substitution to solve the system Ax = b.

% INPUT : A - matrix for the left hand side.

% b - vector for the right hand side

%

% OUTPUT : x - the solution vector.

N = max(size(A));

% Perform Gaussian Elimination

for j=2:N,

for i=j:N,

m = A(i,j-1)/A(j-1,j-1);

A(i,:) = A(i,:) - A(j-1,:)\*m;

b(i) = b(i) - m\*b(j-1);

end

A

% Perform back substitution

x = zeros(N,1);

x(N) = b(N)/A(N,N);

for j=N-1:-1:1,

x(j) = (b(j)-A(j,j+1:N)\*x(j+1:N))/A(j,j);

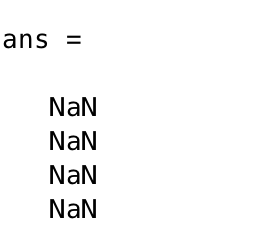
end

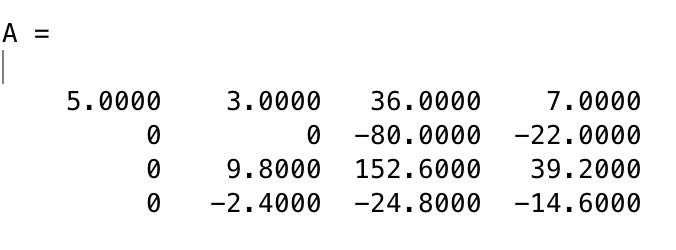
end

% End of function

Results of computation:

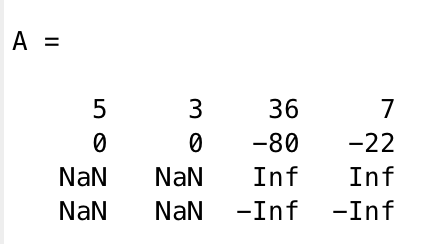
We can see that this function is not working properly.

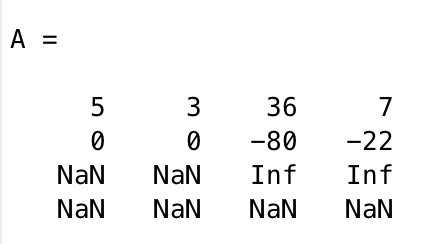
This is due to the fact that we need to eliminate the first element in the second row of the matrix. To do this, we need to multiply the first line by 3 and take it from the second line.

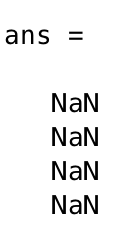
Matrix A after first iteration:

Now we have the second element in second row that equal to zero.

In order to continue elimination we need to multiply second row by which leads us to divide by zero.

Matrix A after second iteration:

Matrix A after third iteration:

And vector is equal to:

B)Modify the function gaussel to implement gaussian elimination with scaled partial pivoting.

Matlab code:

function [x,l] = gaussel\_spp(A,b)

format long

x\_hat=A\b

% [x] = gaussel(A,b)

%

% This subroutine will perform Gaussian elimination

% and back substitution to solve the system Ax = b.

% INPUT : A - matrix for the left hand side.

% b - vector for the right hand side

%

% OUTPUT : x - the solution vector

N = max(size(A));

%Define vector l with pivot rows on each step

for i=1:N,

l(i)=i;

end

%Define vector s

for i=1:N,

s(i)=max(abs(A(i,:)));

end

% Perform Gaussian Elimination

for j=1:N,

for i=1:N,

if i<j

p(i)=0; %Eliminate to chose new pivot every time

else

p(i)=abs(A(i,j)/s(i));% Find pivot element

end

[M,I] = max(p(:));%Get index of pivot element

end

A([I,j],:)=A([j,I],:) %Swap rows

b([I,j],:) = b([j,I],:)%Swap values of b

s([I j])=s([j I]) %Swap s values to prevent elimination of unused values.

l([I j])=l([j I]) %Swap values of l

for i=j:N-1,

m(i)=A(i+1,j)/A(j,j)

A(i+1,:) = A(i+1,:) - A(j,:)\*m(i);

b(i+1) = b(i+1) - m(i)\*b(j);

end

A

end

% Perform back substitution

x = zeros(N,1);

x(N) = b(N)/A(N,N);

for j=N-1:-1:1,

x(j) = (b(j)-A(j,j+1:N)\*x(j+1:N))/A(j,j);

end

l

%Calculating the error

x\_r=[1;-1;2;-1]

e\_1=abs(x-x\_r)%error with calculated values.

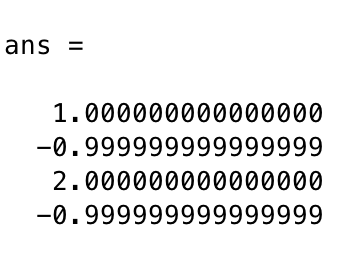
e\_2=abs(x\_hat-x\_r)%error with A\b values

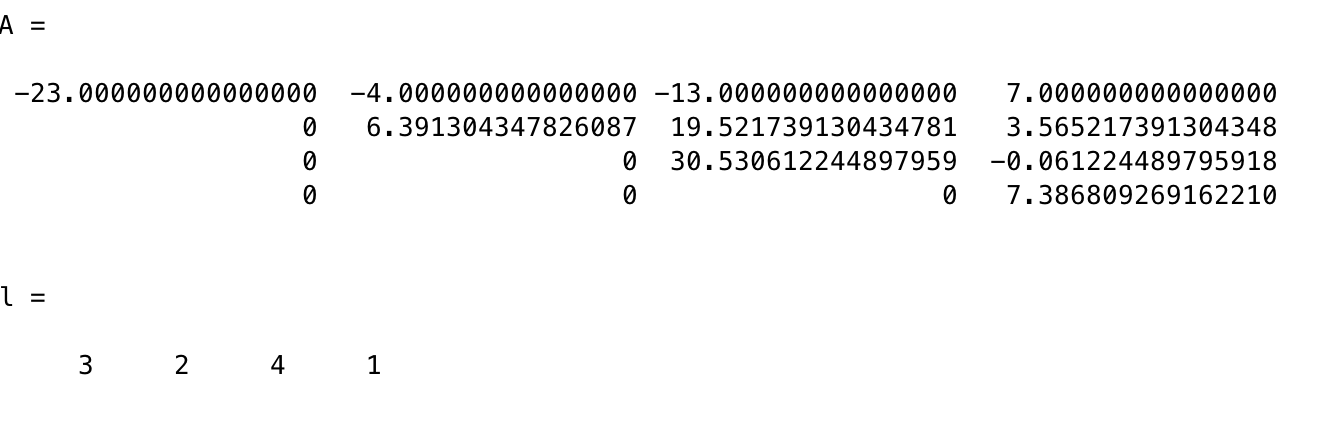
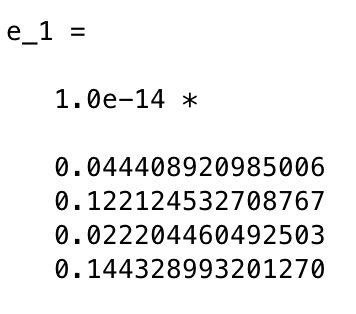
n\_e\_1 = sqrt(sum(e\_1.^2)) % norms of the errors

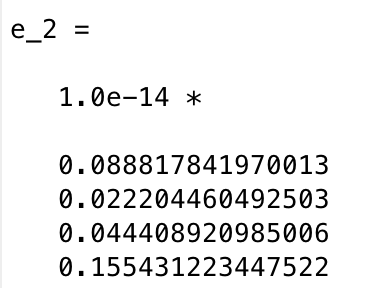
n\_e\_2 = sqrt(sum(e\_2.^2))

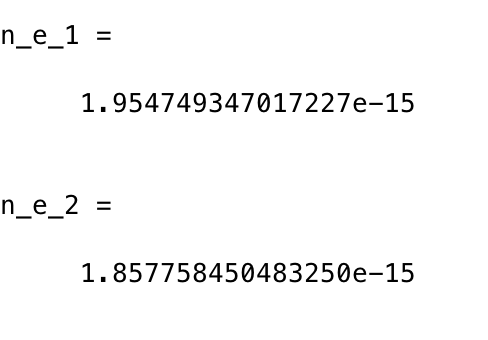
end

% End of function

Results:

The error of x.

The error of A\b:

Norms of errors X and A\b respectively:

As we can see the error is less in case of using A\b. This method is more accurate.

# Task №2.

Implement the Newton’s formula.

Matlab code:

function [x]=run\_newton(f,df,x0,N,sens)

x=x0;

i=0;

x\_hat = 1;%real root of polynom

while i<N

if abs(f(x)) < sens

x

i = N+1;

else if abs(df(x)) < sens

disp("error")

disp(x)

i = N+1;

else

x = x - (f(x)/df(x)); %implementation of Newton's formula

i = i + 1;

l(i) = f(x);

e\_k(i) = x - x\_hat;

end

end

end

%Calculate e\_k and e\_k+1

e\_k\_1 = [];

e\_k\_1 = e\_k;

e\_k(length(e\_k))=[];

%linear\_model=fitlm(log(e\_k),log(e\_k\_1));

%Coefficients of the linear model

%lin\_coef=linear\_model.Coefficients.Estimate;

%disp(lin\_coef)

%r = lin\_coef(2)%r is the second linear\_coeficient

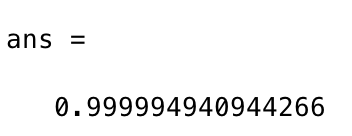
n\_e\_1 = sqrt(sum(e\_k.^2)) %norms of the errors

n\_e\_2 = sqrt(sum(e\_k\_1.^2))

end

Function was started with following parameters:

| Example | | | | | |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Alpha | 1 | 1 | 0 | 0 | 0 |
| Beta | 0 | 1 | 1 | 0 | 0 |
| Gamma | 0 | 1 | 1 | 1 | 0 |
| Workers per year | 50 | 100 | 50 | 30 | 0 |

Results:

