5. Gauss Partial Pivoting function main file:

function x = GaussPP(A,b)

n = size(A,1); %getting n

A = [A,b]; %produces the augmented matrix

%elimination process starts

for i = 1:n-1

p = i;

%comparison to select the pivot

for j = i+1:n

if abs(A(j,i)) > abs(A(i,i))

U = A(i,:);

A(i,:) = A(j,:);

A(j,:) = U;

end

end

%cheking for nullity of the pivots

while A(p,i)== 0 & p <= n

p = p+1;

end

if p == n+1

disp('No unique solution');

break

else

if p ~= i

T = A(i,:);

A(i,:) = A(p,:);

A(p,:) = T;

end

end

for j = i+1:n

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

for k = i+1:n+1

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

end

end

end

%checking for nonzero of last entry

if A(n,n) == 0

disp('No unique solution');

return

end

%backward substitution

x(n) = A(n,n+1)/A(n,n);

for i = n - 1:-1:1

sumax = 0;

for j = i+1:n

sumax = sumax + A(i,j)\*x(j);

end

x(i) = (A(i,n+1) - sumax)/A(i,i);

end

Command window:

>> A = [1.19 2.11 -100 1; 1.42 -0.122 12.2 -1; 0 100 -99.9 1; 15.3 0.110 -13.1 -1]

b = [1.12; 3.44; 2.15; 4.16]

sol=GaussPP(A,b)

A =

1.1900 2.1100 -100.0000 1.0000

1.4200 -0.1220 12.2000 -1.0000

0 100.0000 -99.9000 1.0000

15.3000 0.1100 -13.1000 -1.0000

b =

1.1200

3.4400

2.1500

4.1600

sol =

-0.0450 0.0100 -0.0530 -4.1523