

---

## Table of Contents

Dummy .....	1
Part 1 .....	1
Part 2 .....	1
Local Functions .....	2

## Dummy

Ignore this. It is used to make the outputs align for the main code

```
A = [2 0 1 -1; 6 3 2 -1; 4 3 -2 3; -2 -6 2 -14];
b = [6; 15; 3; 12];
answer = GPP(A, b);
```

## Part 1

Use Gaussian elimination with partial pivoting

```
fprintf("Part 1: Use Gaussian elimination with partial pivoting\n");
```

```
A = [2 0 1 -1; 6 3 2 -1; 4 3 -2 3; -2 -6 2 -14];
b = [6; 15; 3; 12];
```

```
answer = GPP(A, b);
disp(answer);
```

```
Part 1: Use Gaussian elimination with partial pivoting
Row 2 was swapped with row 1 during iteration 1
Row 4 was swapped with row 2 during iteration 2
No row swap performed during iteration 3
    2.0000
         0
    1.0000
   -1.0000
```

## Part 2

Use Gaussian elimination with scaled partial pivoting

```
fprintf("Part 2: Use Gaussian elimination with scaled partial pivoting\n");
```

```
A = [pi -exp(1) sqrt(2) -sqrt(3); pi exp(1) -exp(2) 3/7; sqrt(5) -
sqrt(6) 1 -sqrt(2); pi^3 exp(2) -sqrt(7) 1/9];
b = [sqrt(11); 0; pi; sqrt(2)];
```

```
answer = GSPP(A, b);
```

---

```
disp(answer);
```

*Part 2: Use Gaussian elimination with scaled partial pivoting*

*No row swap performed during iteration 1*

*Row 3 was swapped with row 2 during iteration 2*

*No row swap performed during iteration 3*

*0.5551*

*-2.3493*

*-0.4903*

*2.3786*

## Local Functions

```
function x = GPP(A,b)
C = [A b];
for i=1:size(A, 1)-1
    C = PartialPivotRowSwap(i,C);
    C = Elimination(i,C);
end
x = BackSubstitution(C);
end
```

```
function x = GSPP(A,b)
C = [A b];
for i=1:size(A, 1)-1
    C = ScaledPartialPivotRowSwap(i,C);
    C = Elimination(i,C);
end
x = BackSubstitution(C);
end
```

```
function x = findGreatestRowPP(i, C)
greatestNum = C(i,i);
rowOfGreatestNum = i;
for j=i+1:size(C, 1)
    if abs(C(j,i)) > abs(greatestNum)
        greatestNum = C(j,i);
        rowOfGreatestNum = j;
    end
end
x = rowOfGreatestNum;
end
```

```
function x = PartialPivotRowSwap(i, C)
rowToSwap = findGreatestRowPP(i, C);
if (rowToSwap == i)
    fprintf("No row swap performed during iteration %.d\n", i);
else
    fprintf("Row %.d was swapped with row %.d during iteration %.d\n", rowToSwap, i, i);
    tempRow = C(rowToSwap,:);
```

---

```

        C(rowToSwap,:) = C(i,:);
        C(i,:) = tempRow;
    end
x = C;
end

function x = findGreatestRowSPP(i, C)
greatestNum = C(i,i)/max(abs(C(i,1:end-1)));
rowOfGreatestNum = i;
for j=i+1:size(C, 1)
    scaledRowValue = C(j,i)/max(abs(C(j,i:end-1)));
    if abs(scaledRowValue) > abs(greatestNum)
        greatestNum = scaledRowValue;
        rowOfGreatestNum = j;
    end
end
x = rowOfGreatestNum;
end

function x = ScaledPartialPivotRowSwap(i, C)
rowToSwap = findGreatestRowSPP(i, C);
if (rowToSwap == i)
    fprintf("No row swap performed during iteration %.d\n", i);
else
    fprintf("Row %.d was swapped with row %.d during iteration %.d\n", rowToSwap, i, i);
    tempRow = C(rowToSwap,:);
    C(rowToSwap,:) = C(i,:);
    C(i,:) = tempRow;
end
x = C;
end

function x = Elimination(i, C)
for j=i+1:size(C, 1)
    m = C(j,i)/C(i,i);
    C(j,:) = C(j,:) - (m*C(i,:));
end
x = C;
end

function x = BackSubstitution(C)
solution = zeros(size(C, 1),1);
solution(end) = C(end,end)/C(end,end-1);
for i=size(C, 1)-1:-1:1
    C(i,end) = C(i,end)-sum(C(i,i+1:end-1).*(solution(i+1:end))');
    solution(i) = (C(i,end))/C(i,i);
end
x = solution;
end

function x = Dummy()
fprintf("\n");
x = 0;

```

---

---

end

*Row 2 was swapped with row 1 during iteration 1  
Row 4 was swapped with row 2 during iteration 2  
No row swap performed during iteration 3*

*Published with MATLAB® R2020a*