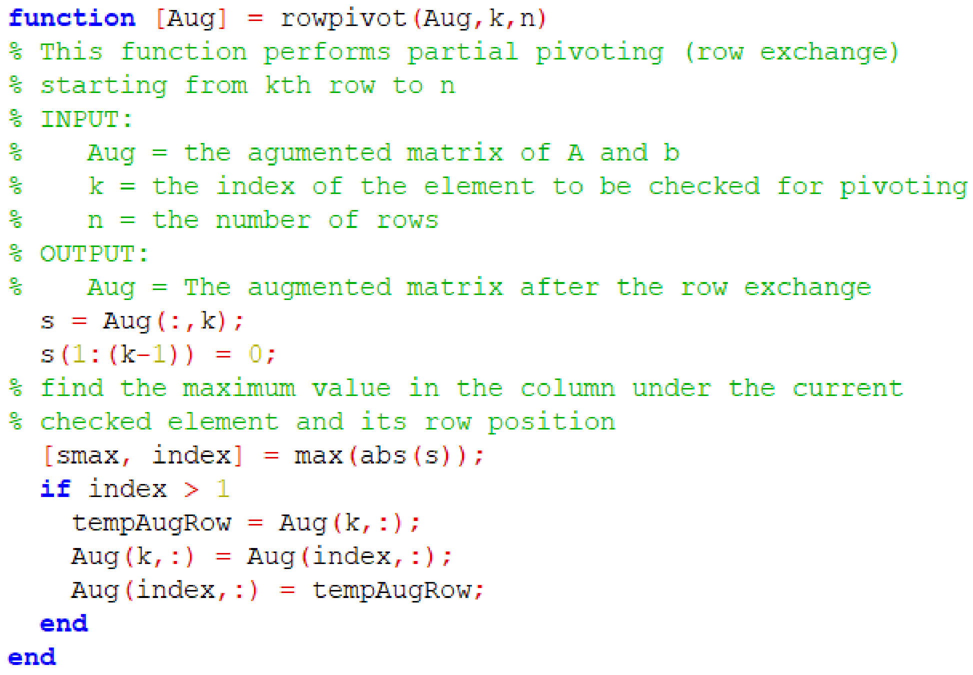
Gaussian Elimination with Partial Pivoting:

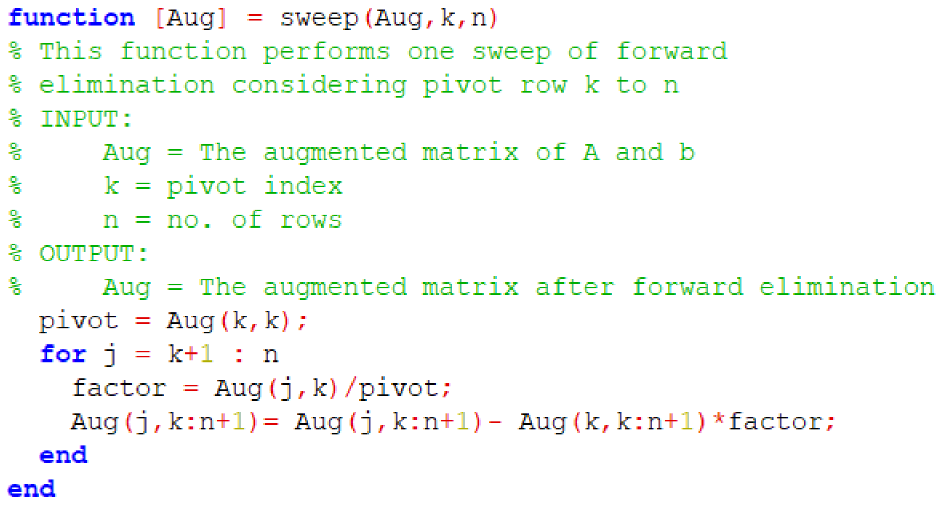
The gaussian elimination with pp will be broken down to those components:

1-Pivoting/Row Exchange function:



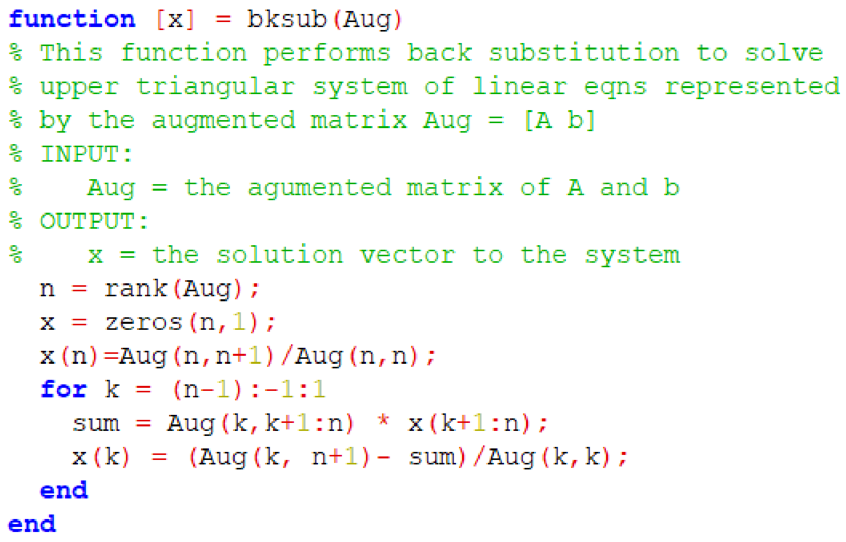
```
function [Aug] = rowpivot(Aug,k,n)
% This function performs partial pivoting (row exchange)
% starting from kth row to n
% INPUT:
%    Aug = the agumented matrix of A and b
%    k = the index of the element to be checked for pivoting
%    n = the number of rows
% OUTPUT:
%    Aug = The augmented matrix after the row exchange
  s = Aug(:,k);
  s(1:(k-1)) = 0;
% find the maximum value in the column under the current
% checked element and its row position
  [smax, index] = max(abs(s));
  if index > 1
    tempAugRow = Aug(k,:);
    Aug(k,:) = Aug(index,:);
    Aug(index,:) = tempAugRow;
  end
end
```

2-Forward Elimination function:



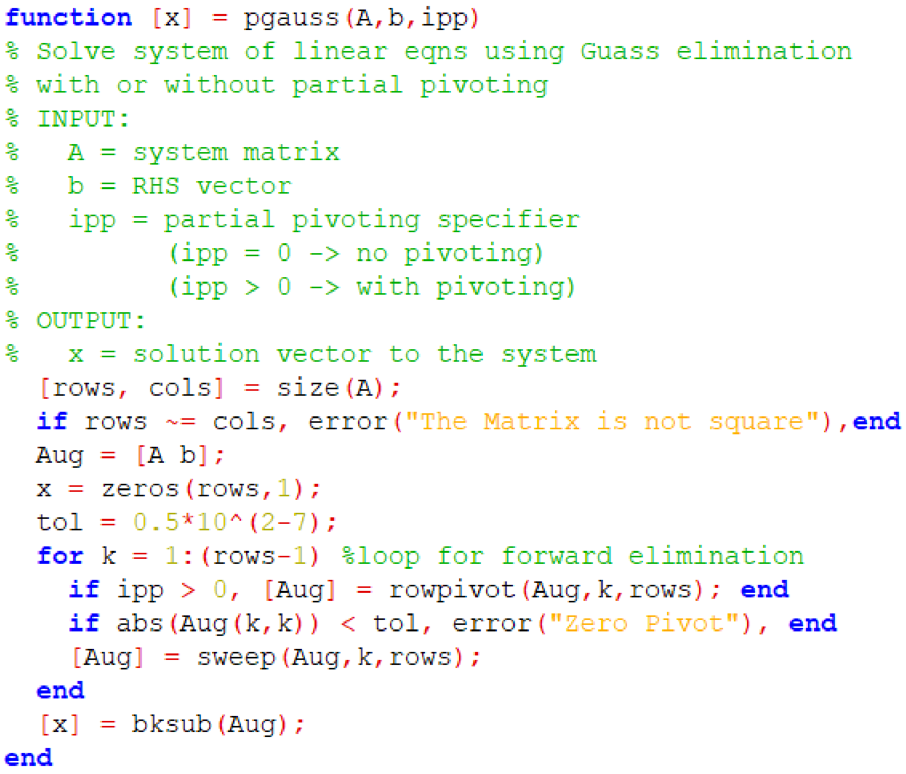
```
function [Aug] = sweep(Aug,k,n)
% This function performs one sweep of forward
% elimination considering pivot row k to n
% INPUT:
%     Aug = The augmented matrix of A and b
%     k = pivot index
%     n = no. of rows
% OUTPUT:
%     Aug = The augmented matrix after forward elimination
  pivot = Aug(k,k);
  for j = k+1 : n
    factor = Aug(j,k)/pivot;
    Aug(j,k:n+1)= Aug(j,k:n+1)- Aug(k,k:n+1)*factor;
  end
end
```

3-Back Substitution function:



```
function [x] = bksub(Aug)
% This function performs back substitution to solve
% upper triangular system of linear eqns represented
% by the augmented matrix Aug = [A b]
% INPUT:
%    Aug = the agumented matrix of A and b
% OUTPUT:
%    x = the solution vector to the system
  n = rank(Aug);
  x = zeros(n,1);
  x(n)=Aug(n,n+1)/Aug(n,n);
  for k = (n-1):-1:1
    sum = Aug(k,k+1:n) * x(k+1:n);
    x(k) = (Aug(k, n+1)- sum)/Aug(k,k);
  end
end
```

-The Gaussian Elimination could then be achieved by integrating those components as follows:



```
function [x] = pgauss(A,b,ipp)
% Solve system of linear eqns using Guass elimination
% with or without partial pivoting
% INPUT:
%   A = system matrix
%   b = RHS vector
%   ipp = partial pivoting specifier
%         (ipp = 0 -> no pivoting)
%         (ipp > 0 -> with pivoting)
% OUTPUT:
%   x = solution vector to the system
  [rows, cols] = size(A);
  if rows ~= cols, error("The Matrix is not square"),end
  Aug = [A b];
  x = zeros(rows,1);
  tol = 0.5*10^(2-7);
  for k = 1:(rows-1) %loop for forward elimination
    if ipp > 0, [Aug] = rowpivot(Aug,k,rows); end
    if abs(Aug(k,k)) < tol, error("Zero Pivot"), end
    [Aug] = sweep(Aug,k,rows);
  end
  [x] = bksub(Aug);
end
```