Gaussian Elimination Algorithm
with Partial Pivoting

Forward Elimination Applied to Matrix

for k = 1:n ………………………………………………………………………….*The outer loop - this eliminates variable k*

  choose ip_k such that $|A_{ip_k,k}| = \max\{|A_{i,k}| : i \geq k\}$ …………………*Find the largest of the candidate pivots*

  if $A_{ip_k,k}$ = 0…………………………………………………………………………*if the largest is zero, no possible pivot*

    warning ('Pivot in Gaussian Elimination is zero')………………………..and maybe get out of here

  end

  swap $A_{k,k},\ldots,A_{k,n}$ **with** $A_{ip_k,k},\ldots,A_{ip_k,n}$ ………………………….swap the rows to get the pivot into position

  swap b_k **with** b_{ip_k} ………………………………………………..swap the corresponding right hand sides

  for i = k+1:n ……………………………………………………………………………………………*loop on the rows*

    $A_{i,k} = A_{i,k} / A_{k,k}$

    for j = k+1:n………………………………………………………………*loop on the columns – innermost loop*

      $A_{i,j} = A_{i,j} - A_{i,k} A_{k,j}$ …………………………………………………update the i,j element

    end

    $b_i = b_i - A_{i,k}\, b_k$ ………………………………………………… it's just like b was an extra column

  end

end

This results in the upper triangle of the eliminated system in the upper triangle of A, the multipliers in the strict lower triangle of A, and the swapping information in the ip array.

Solving …………………………………………………………………………………………

for i = n:-1:1……………………………………………………………………*here is where we solve the upper triangular system*

  for j = i+1:n…………………………………………….

    $b_i = b_i - A_{i,j} x_j$ ……………………………*this loop stores b(i) minus the summation A(i,j)*x(j) into b(i)*

  end…………………………………………………………

  $x_i = b_i / A_{i,i}$ ……………………………………………………………………………………*and divide by A(i,i) to get x(i)*

end

and the output is the solution x.