

Contents

- [display the # of row interchanges](#)
- [compute & display the det.](#)

```
function [Amod,ord,nuroich,detA]=Gel(A,b,verbose)
```

```
% HW 1 Problem 3 (a)
% Function that performs Gaussian elim.
% Also outputs determinant

% [Amod,ord,nuroich,detA] = Gel(A,b,verbose)
%
% This function perform elimination with partial pivoting and scaling as
% described in Section 1.3.2 in the Hoffman textbook (viz. it does Gaussian
% elimination). Note that the ordering which preserves upper triangularity
% is stored in the ord output variable, such that the upper triangular output
% is given by row-permuted matrix Amod(ord,:). The verbose flag can be set to
% true or false (or omitted, default=false) in order to print out what the algrithm
% is doing for each elimination step.

%Parse the inputs, throw an error if something is obviously wrong with input data
narginchk(2,3);
if (nargin<3)
    verbose=false;
end %if

%Need to error check for square input.

%Allocation of space and setup
Amod=cat(2,A,b);      %make a copy of A and modify with RHS of system
n=size(A,1);          %number of unknowns
ord=(1:n)';           %ord is a mapping from input row being operated upon to the actual row that represents in the matrix ordering
nuroich = 0;          %store the number of row interchange for determinant calculation

%Elimination with scaled, partial pivoting for matrix Amod; note all row
%indices must be screen through ord mapping.
for ir1=1:n-1
    if (verbose)
        disp('Starting Gauss elimination from row: ');
        disp(ir1);
        disp('Current state of matrix: ');
        disp(Amod(ord,:));
    end %if

    %check scaled pivot elements to see if reordering should be done
    pivmax=0;
    ipivmax=ir1;       %max pivot element should never be higher than my current position
    for ipiv=ir1:n      %look only below my current position in the matrix
        pivcurr=abs(Amod(ord(ipiv),ir1))/max(abs(Amod(ord(ipiv),:))); %note that columns never get reordered...
        if (pivcurr>pivmax)
            pivmax=pivcurr;
            ipivmax=ipiv; %this stores the index into ord for row having largest pivot element
        end %if
    end %for

    %reorder if situation calls for it
    if (ipivmax ~= ir1)
        itmp=ord(ir1);
        ord(ir1)=ord(ipivmax);
        ord(ipivmax)=itmp;
        nuroich = nuroich + 1; % update the number of row interchange

        if (verbose)
            disp('Interchanging rows: ');
            disp(itmp);
            disp(' and: ');
            disp(ord(ir1));
            disp('Current matrix state after interchange: ');
            disp(Amod(ord,:));
        end %if
    end %if

    %perform the elimination for this row, former references to ir1 are now
    %mapped through the ord array
    for ir2=ir1+1:n
        fact=Amod(ord(ir2),ir1);
        Amod(ord(ir2),ir1:n+1)=Amod(ord(ir2),ir1:n+1)-fact/Amod(ord(ir1),ir1).*Amod(ord(ir1),ir1:n+1); %only need columns ahead of where we are in matrix
    end %for

    if (verbose)
        disp('Following elimination for row: ');
        disp(ir1);
    end
end
```

```
        disp(' matrix state: ');
        disp(Amod(ord,:));
    end %if
end %for
```

Error using Gel (line 18)
Not enough input arguments.

display the # of row interchanges

```
disp('The total number of row interchanges is: ');
disp(nuroich)
```

compute & display the det.

```
detA = (-1)^nuroich;
Amododr = Amod(ord,:);
for k = 1 : n
    detA = detA*Amododr(k,k);
end %for
disp('The determinant of A is: ');
disp(detA);
```

```
end %function
```