MIDTERM: NUMERICAL COMPUTING

The Total points is **108** and the maximum you can score is **100** points.
Date: **22th Feb 2023**

(1) (2+3+3+4+4+4+5=25 points) Write down the output of the following commands in octave.
   (a) w=linspace(3,10,11)'
   (b) A=diag(2:3:11).^2
   (c) A=diag(2:11:3).^2
   (d) B=(ones(2,4)-eye(2,4))*diag(2:5)*(ones(4,2)-eye(4.2))
   (e) p=[1 0 0 0 -1]; roots(p)
   (f) A=[1 2;3 4];B=[1 0;0 1]; disp(A.*B); disp(A*B)
   (g) str=sprintf(´pi is close to %1.3f and eps is %1.0e´,pi,eps); disp(str)

(2) (5+4+8+8=25 points) Write down a command or a short code to achieve the following goals:
   (a) Display real part, imaginary part and conjugate of a complex number z. Display transpose of a matrix A over complex numbers.
   (b) Display the plot of the function $f(x) = sin(x) + e^x$ for x between $-\pi$ and π.
   (c) Write a function using $fprintf$ to display a table consisting of 3 columns and n rows whose entries are the values of the functions x, $e^x + sin(x)$ and $e^x - sin(x)$ for a given vector of length n as given below.

x	$e^x + sin(x)$	$e^x - sin(x)$
1	3.5598	1.8768
1.5	5.4792	3.4842
...	...	...

   (d) Write a code to evaluate cube root of 7 correct upto five places of decimal using Newton's method.

(3) (15 points) In the Octave code written on the attached sheet, add comments in the code to explain the lines of the code which end with a % symbol. Also give a short description and write a synopsis for the code. (**Remember to attach the sheet with your answer script!!**)

(4) (20 points) Modify the algorithm of the secant method so that all the estimates lie in the given bracket. Write an Octave code for a function which approximates the roots of a function bracketted by a given vector using this new algorithm. Use absolute convergence criterion.

(5) (8 points) Explain the difference between partial pivoting and full pivoting in the Gauss elimination method.

(6) (15 points) Write a function whose input will be a tridiagonal matrix A and output will be the determinant of A using the special structure of the tridiagonal matrix.

```
function x = GEPivShow(A,b,ptol)
%
%
%
% Synopsis:
%
%
% Input:     A,b  = coefficient matrix and right hand side vector
%            ptol = (optional) tolerance for detection of zero pivot
%                   Default:  ptol = 50*eps
%
% Output:    x = solution vector, if solution exists

if nargin<3, ptol = 50*eps;  end                    %.....................................................
[m,n] = size(A);
if m~=n,  error('A matrix needs to be square');  end
nb = n+1;   Ab = [A b];                             %.....................................................
fprintf('\nBegin forward elmination with Augmented system:\n');  disp(Ab);

% ..........................................................................................
for i = 1:n-1                     %  loop over pivot row
  [pivot,p] = max(abs(Ab(i:n,i)));                  %.....................................................
  ip = p + i - 1;                                   %  p is ...............................................
  if ip~=i                                          %  ip is ..............................................
    fprintf('\nSwap rows %d and %d;  new pivot = %g\n',i,ip,Ab(ip,i));
    Ab([i ip],:) = Ab([ip i],:);                    % ....................................................
  end
  pivot = Ab(i,i);
  if abs(pivot)<ptol, error('zero pivot encountered after row exchange');  end
  %.........................................................................................
  for k = i+1:n
     Ab(k,i:nb) = Ab(k,i:nb) - (Ab(k,i)/pivot)*Ab(i,i:nb);
  end
  fprintf('\nAfter elimination in column %d with pivot = %f\n',i,pivot);
  disp(Ab);
end

%.....................................................................
x = zeros(n,1);           %.....................................................
x(n) = Ab(n,nb)/Ab(n,n);
for i=n-1:-1:1
  x(i) = (Ab(i,nb) - Ab(i,i+1:n)*x(i+1:n))/Ab(i,i);
end
```