



General Rules

- Due date is 30 October 2002, Wednesday and **not subject to postpone**. Since this is your first time working with MATLAB, we keep the submission period so long (i.e. 2 weeks).
- Late submission is **not** allowed.
- There is **no** teaming up. The homework has to be done/turned in individually.
- In case of cheating, all parts involved (source(s) and receiver(s)) get zero.
- You will submit your homework electronically and also you are supposed to return a hard copy of your programs to our office (A-401). To submit your homework electronically you will use `submit371` command through a tar archive file (`hw1.tar`). This file will contain your program source codes and a `readme.txt` file that describes each of the program files.
- In your program codes, you are required to write comments.

Homework

1. Write a function to implement Fixed-Point Iteration method:

The function prototype is as follows:

```
function[ite, abserr, relerr, x] = fixedpoint(fun, guess, tol, max)
where fun      string containing name of the function
      guess    initial guess point for the fixed point
      tol      tolerance value used as stopping criterion
               (according to relative error)
      max      maximum number of iterations
      ite      the iteration number where a fixed point is found
      abserr   vector of absolute errors
      relerr   vector of relative errors
      x        vector of approximated fixed point
```

To help you in coding, a basic Secant Method MATLAB code can found in the following Appendix Section.

2. Use function in part1 to solve the equation $e^{-x} - x = 0$
 - a. By using Fixed-Point Iteration with $x = e^{-x}$
 - b. By using Fixed-Point Iteration with $x = -\ln x$

3. Write a function to implement Newton-Raphson Method.

The function prototype is as follows:

```
function[ite, abserr, relerr, x, y] = newton(f1, f2, guess, tol, max)
where  f1      string containing name of the function
       f2      string containing the name of the function which is the
               derivative function of f1
       guess    initial guess point for the root
       tol      tolerance value used as stopping criterion
               (according to relative error)
       max      maximum number of iterations
       ite      the iteration number where a root is found
       abserr   vector of absolute errors
       relerr   vector of relative errors
       x        vector of approximated root
       y        vector of function values f1(x)
```

4. Use function in part 3 to solve the equation $e^{-x} - x = 0$.

5. Write a simple report about this homework. The report should contain the following sections:

- Your program source codes
- Error vs. # of iterations graphs, and
- Comment on the results you obtained in Part 2 and 4.

Appendix

```
function [x, y] = secant(fun, a, b, tol, max)
% Find a zero using secant method.
%   fun    string containing name of function
%   a, b    first two estimates of the zero
%   tol     tolerance for change in computed zero
%   max     maximum number of iterations
%   x       vector of approximations to zero
%   y       vector of function values fun(x)
x(1) = a;          x(2) = b;
y(1) = feval(fun, x(1));  y(2) = feval(fun, x(2));
for i = 2 : max,
    x(i+1) = x(i) - y(i) * (x(i) - x(i-1))/(y(i) - y(i-1));
    y(i+1) = feval(fun, x(i+1));
    if (abs(x(i+1)-x(i)) < tol)
        disp('method has converged'); break;
    end
    if y(i) == 0.0
        disp('exact zero found'); break;
    end
    iter = i;
end
if (iter >= max)
    disp('zero not found to desired tolerance');
end
n = length(x);      k = 1:n;      out = [k' x' y'];
disp('step      x      y'), disp(out)
```