Lab exercises
Degree in mechanical engineering
AY 2016-17
Prof. S. De Marchi
Padova, 28th March 2017

Useful to know

- The pre-defined variable `varargin` allows to specify a variable number of parameters to a function. For example, if we define a function

```
function myplot(x,varargin)
plot(x,varargin{:});
return
```

we could call it as

```
myplot(sin(0:.1:1),'color',[.5 .7 .3],'linestyle',':');
```

To know the number of input parameters, Matlab has the variable

```
nargin
```

Hence, we may check and modify `myplot` as follows:

```
function myplot(x,varargin)
if nargin==0
    error('bad number of parameters')
    return
elseif nargin==1 plot(x) else
    plot(x,varargin{:})
end
return
```

There are also the variable

```
varargout, nargout
```

that allows to have a variable number of output parameters and count them, respectively.

- In Matlab there exist functions useful for finding zeros of a function: `roots`, `fzero` and `fsolve`.

  (i) `roots` computes the zeros (also complex ones) of polynomials. Call it as `roots(a)`, with `a` the vector of the polynomial coefficients in reverse order: from the coefficent of higher order to that of the constant term.

  (ii) `fzero` can be called in the following way: `x=fzero(fun,x0,opt)` with `fun` specified using the symbol `@`.
  For example,

  ```
  f=@(x,c) sin(x^3/c); x0=2;
  sol = fzero(f,x0,[],9)
  ```

  here `9` is the value of the parameter `c`.

  (iii) `fsolve` is more general since it works on systems of non linear equations but can be used also on a single equation. A typical call is `x = fsolve(fun,x0)` where `fun` can be specified using "@".
  For example

  ```
  x = fsolve(@myfun,[2 3 4],optimoptions('fsolve','Display','iter'))
  ```

  where myfun is a MATLAB function such as:

  ```
  function F = myfun(x)
  F = sin(x);
  ```

  (iv) To know which input and output parameters requires a function, write in the Command window, `help fzero`, `help fsolve` or `help roots`.

Solve the following problems

1. Write the functions `Bisection.m` and `FixedPoint.m` for computing the zeros of a functions with the bisection's method and fixed point iteration, respectively.

2. Take the function $f(x) = x^2 - \sin(x+1)$ of which we want to compute its zeros.

   - By plotting the graph of f, individuates the `two` real roots of $f(x) = 0$ and the corresponding `separation intervals`, I_{α_1} e I_{α_2}.
   - Find two convergent `iterative methods`, with iteration functions $g_i(x)$, $i = 1, 2$. For each one of them determine the number of necessary iterations. Consider `kmax=100`, as max number of iterations and, for the test on the relative error `tol=1.e-6`.
   - Compare the results with the ones obtained with `fzero`.

3. Using a suitable iteration function, compare the `bisection method`, and the fixed point iteration method for computing the only real root of

$$1 = \frac{g}{2x^2}(\sinh(x) - \sin(x)), \qquad g = 9.81.$$

As before, take `kmax=100` and the tolerance `tol=1.0e-6` for the relative error.