ME 2016 - Computing Techniques
Fall 2018 – Section A

Homework 3
Due Thursday, September 27
Electronic submission part due at 2:55 PM (5 minutes before class time)
Hardcopy part: end of class

Root Finding: Bisection method
Please BOX your final results

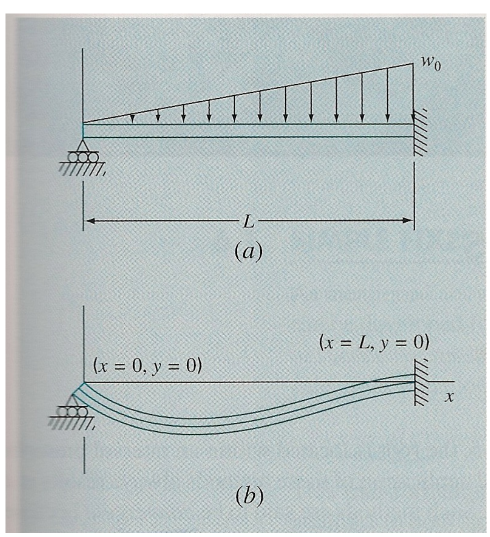
Problem 1: Textbook problem 5.1 **(15 points)**

5.1 Determine the real roots of $f(x) = -0.5x^2 + 2.5x + 4.5$:
(a) Graphically.
(b) Using the quadratic formula.
(c) Using three iterations of the bisection method to determine the highest root. Employ initial guesses of $x_l = 5$ and $x_u = 10$. Compute the estimated error ε_a and the true error ε_t after each iteration.

Solve this problem *by hand*, but for part (a), use MATLAB to plot the function f; include a printout of your plot with your homework. Make sure that your plot has a grid, a title and axis labels. Note that you can calculate the approximate relative percent error ε_a as defined in class starting at the second iteration only.

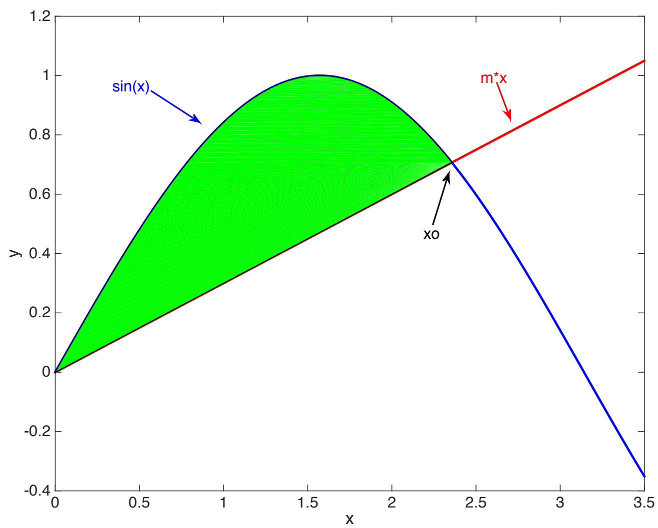
Problem 2: **(15 points)**
The figure below shows a uniform beam subjected to a linearly increasing distributed load w_o. The equation for the resulting elastic curve is

$$y = \frac{w_o}{120EIL}(-x^5 + 2L^2x^3 - L^4x)$$



Perform 2 iterations of the bisection method (by hand) to determine the point of maximum deflection (that is, the value of x where $dy/dx = 0$); use an initial interval of [2.6 2.8] and the following parameter values: L = 6 m, E = 5x10^8 kN/m^2, I = 3x10^{-4} m^4 and w_o = 0.25 kN/m.
Plot the function for this root-finding problem (using Matlab) and include a printout of the plot with your homework. Calculate the maximum deflection $y_{\max}$ using the approximate value of the root obtained at the second iteration.

Problem 3: **(20 points)**
The figure below represents the area A (in green) between the curve $y = \sin x$ and the line $y = mx$. You need to find the slope m necessary to obtain a specified area A.



(a) Formulate this problem as a root-finding problem in terms of x_0, the x-axis location where the line crosses the sine curve.
Hint: The resulting equation, of the form $f(x_0) = 0$, should only involve the given parameter A and the unknown x_0; to find $f(x_0)$, you will need to combine 2 equations.

(b) We need A = 1.4; estimate the root x_0 and the corresponding slope m by performing two iterations of the bisection method (i.e. compute x_r^1 and x_r^2, as well as their corresponding slope values) using $x_l^1 = \pi/2$ and $x_u^1 = \pi$ as the initial bounds. For the second iteration, give the absolute value of the approximate error $|E_a^2|$ for the root x_0.

(c) We want to know how many iterations n are needed in order to achieve an absolute error tolerance $|E_a|$ of at least 10^{-6}. Use your results from (b) and the fact that $|E_a^{i+1}| = \frac{1}{2}|E_a^i|$ (the linear convergence property of the bisection method) to calculate the required number of iterations n. Hint: Do NOT calculate $|E_a^i|$ for successive iterations until it is less than 10^{-6}; instead, derive a general relationship between $|E_a^n|$ and $|E_a^2|$.

Problem 4: MATLAB programming **(50 points)**

Write a generic (i.e. non-problem specific) MATLAB function that implements the bisection method. The function should have the form

`function [xr,iter,X] = Bisection(f,xl,xu,es,imax)`,

where `f` is a function handle (see below) that defines the root-finding problem $f(x) = 0$, `xl` and `xu` are the lower and upper initial values, `es` is the stopping criterion for the relative approximate percent error defined in class, and `imax` is the maximum number of allowable iterations. The function outputs the value of the root (`xr`), the number of iterations used to calculate that root (`iter`), and a vector (`X`) of length iter containing the successive approximate root values at each iteration (i.e. the first element of X contains the approximation obtained at iteration 1, and the last element of X is X(iter) = xr).

Use a `while` loop, and make sure that the number of times that the function `f` has to be evaluated is minimized.

At each iteration, the function should calculate the absolute value of the relative approximate percent error $|\varepsilon_a|$(as defined in class), starting with the second iteration (because two successive approximate root values are needed to compute ε_a).

At the end, the function should print the final root value (`xr`), the number of iterations used (`iter`), and the value of the relative approximate percent error for the last iteration in the command window in a nicely formatted way, using the `fprintf` command.

Finally, it should generate a plot of the absolute value of the relative approximate percent error $|\varepsilon_a|$ as a function of the iteration number (use a continuous line). The plot should have a grid, a title, and x and y labels. Since the magnitude of the error can vary greatly, plot it on a semi-logarithmic scale (see the `semilogy` command).

A function handle for an anonymous function can be created in the MATLAB command window (or inside an M-file) using the following syntax (type `help function_handle` for more info):

```
FUNHANDLE = @(ARGLIST)EXPRESSION
```

For example, typing

```
y = @(x) 1 + sqrt(x);
```

creates the function $y(x) = 1 + \sqrt{x}$, which can then be called in the following way:

```
>> y(4)
ans =
      3
```

Use your function `Bisection` to solve the falling parachutist problem for the drag coefficient (I gave you the equation $f(c) = 0$ in class when I introduced the chapter on root-finding, and it is also described in Example 5.1 of the textbook; use m = 68.1 kg, v = 40 m.s^{-1}, t = 10s and g = 9.81 m.s^{-2}) with a stopping criterion of 0.01%, a maximum number of iterations of 100, and an initial interval of [12,16].
Print the error plot and print the command window showing your formatted results (final root, number of iterations and approximate error).

Finally, you will use your function to re-visit problem 1 of this assignment and to obtain the highest root. To do that , write a script called `HW3_PB1`. The script should define the function handle corresponding to problem 1, as well as all the necessary input parameters to call `Bisection`: use a stopping criterion of 10^{-6} %, a maximum number of iterations of 100, and xl = 5, xu = 10. The script should then call `Bisection`, compute the absolute value of the true percent relative error $|\varepsilon_t|$ and plot $|\varepsilon_t|$ as a function of the iteration number on the same plot as the $|\varepsilon_a|$ plot generated by the function, using a continuous line of a different color. Include a legend on your plot.
Print the plot and print the command window showing your formatted results (final root, number of iterations and approximate error).

Instructions for Homework 3 submission:

1) Hand in paper copies of your handwritten solutions for problems 1, 2 and 3; include the plot asked for in some of those problems.

2) Hand in paper copies of 2 the plots and results asked for in problem 4

3) Include the Bisection.m and the HW3_PB1.m electronic files in a zipped folder named HW3_LASTNAME_FIRSTNAME and upload to Canvas. **Make sure your folder has a .zip extension. Please do not use any compression software that would result in a different compressed format with another extension.**

Make sure you follow the formatting instructions and that you include comments in your codes.

Please note the following, valid for all MATLAB assignments:

1. **Refrain from using the commands `clear`, `clc` and `close` (or any of their variations) in any of your m-files, as they interfere with the grading process.**

2. **When your code is run, it should produce only what is asked for in the assignment (for Bisection.m, only 1 plot and 3 formatted values in the command window). Please make sure to suppress any unnecessary intermediate result in any form: no other figure, curve or value printed in the command window should appear.**

You will lose points if you ignore these instructions.

Please note that Canvas will only accept a file with a ZIP extension: all of your M-files must be in that zipped folder. If we cannot open your files because you failed to follow these instructions, you will get a zero for this part of the assignment.

Finally, please keep in mind the policies about late assignments and collaboration outlined in the syllabus, and copied here for your convenience:

- Late assignments: homework and projects will be accepted up to 24 hours after the deadline with a 50% penalty. This policy will be strictly enforced and no submission will be accepted after 24 hours. Hard copies of homework will be due at the end of class. ***It is your responsibility to make sure that you have successfully uploaded and submitted all the required files in the required format to Canvas on time***: please double-check to avoid having to re-submit your work after the due date. The only acceptable proof that you have submitted your work on time is the time stamp of your Canvas submission. ***Files will not be accepted by email even if they show a "last modified date" that is before the due date.***

- Collaboration: students may *discuss* their assignments with each other, but homework and projects must be *completed individually* by each student. ***You must turn in your own work***. Copying someone else's work and submitting it as your own will not be tolerated. In particular, ***for Matlab assignments***, this policy means that students can discuss aspects such as the general approach to solve a problem or the syntax of a specific command, but ***they should not look at each other's codes***. If it is suspected that this has occurred, you will be reported to the Dean of Students for an honor code violation.