FLORIDA INSTITUTE OF TECHNOLOGY
MECHANICAL AND AEROSPACE ENGINEERING DEPARTMENT

MAE 3150: Aerospace Computational Techniques

Spring 2016
Homework 3 – Due February 18, 2016

For this assignment, you will need to write some short codes to complete the following problems. You are free to write in any language you prefer, including MATLAB, provided you do not use any built-in commands or intrinsic functions to shortcut the methods you are asked to investigate (though you may use them to verify your answers). For instance, you may not use MATLAB's `fzero` command to find the roots of an equation. The only restrictions on your codes are that they must write your output to a file, and numbers in your results must be properly formatted with an appropriate number of significant figures. Use exponential notation where (and only where) appropriate. Please turn in both hard and electronic copies of both your codes, and your hardcopies of your results.

1. Root finding has its place in several applications in the areas of vibration (modal analysis of wing flutter), acoustics (microphone and speaker placement for acoustic signature measurements), and computational fluid dynamics (matching governing equations to proper numerical schemes for aerodynamic calculations).

Determine the real root $0 < x_{root} < 2$, out to exactly three decimal points, of the polynomial

$$f\left(x\right) = x^3 - 8x^2 + 6x + 6$$

a) graphically (no code required; simply plot; you may use any function of any software to do this) *Hint: Plot a separate line y = 0 to show where the function crosses the axis. The suggested plot range is x = 0 to x = 6.*
b) using fixed-point iteration (five iterations, $x_o = 2$)
c) using Newton-Raphson (five iterations, $x_o = 2$)
d) using Secant method (five iterations, $x_{-1} = 1$, $x_o = 2$)

Compute the approximate percent relative errors out to exactly four decimal points for each iteration of your solutions as described in the notes.

If a method appears to diverge, provide a mathematical reason why that is.

The file name for your fixed-point iteration code should be:	*firstname_lastname.hmwk3_fpi.sp16.ext*
The file name for your Newton-Raphson code should be:	*firstname_lastname.hmwk3_nr.sp16.ext*
The file name for your Secant method code should be:	*firstname_lastname.hmwk3_sec.sp16.ext*

2. There are many applications where numerical integration is required in aerospace engineering. Examples include integrating a pressure distribution about an airfoil to find lift and drag, integrating a time-dependent drag profile on a rocket to determine time of flight and burnout altitude, and integrating a probability distribution function to find the average fuel drop diameter in an aircraft engine. In many instances, the function to be integrated may be a set of discrete data points rather than being analytical, but the technique is similar.

Integrate the following function both analytically (you may want to look this up in a table of integrals) and numerically:

$$I(f) = \int_0^{\pi} e^{-x} \cos^2 x \; dx$$

Use both the trapezoidal and Simpson's 1/3 rules to numerically integrate the function out to exactly seven decimal points. For both cases, start with $n = 1$, then double the number of segments until your last computation includes 512 segments. Compute the percent relative errors, out to exactly four decimal points, for the numerical results (as compared to the analytic answer).

$$\%\ \text{Error} = \frac{\text{Numerical Answer - Analytic Answer}}{\text{Analytic Answer}}$$

The file name for your Trapezoidal Rule code should be: *firstname_lastname.hmwk3_trap.sp16.ext*
The file name for your Simpson's Rule code should be: *firstname_lastname.hmwk3_simp.sp16.ext*