**12.010 Homework #4** **Due Thursday, November 17, 2011**

All solutions can be included in a single NoteBook with the results in different sections or sub-sections.

**Question (1):** (25-points) (a) Write Mathematica NoteBook which generates a table of error function (erf) and its derivatives for real arguments (*z*) between -3 and 3 in steps of 0.25. The error function is defined by the equation below (but is rarely evaluated by performing the integration).

$$\operatorname{erf}(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\,dt.$$

(see http://mathworld.wolfram.com/Erf.html for information the error function )
The values in the table should be given with 5 decimal places. The table should have headers explaining what the columns are. Explain how you designed the NoteBook and give an example of the output.
(b) How would you change this NoteBook if 10 significant digits were required?
Mathematica NoteBook should also be supplied

**Question (2):** (25-points).
Write a NoteBook that reads your name in the form <first name> <middle name> <last name> and outputs the last name first and adds a comma after the name, the first name, and initial of your middle name with a period after the middle initial. If the names start with lower case letters, then these should be capitalized. The NoteBook should not be specific to the lengths of your name (ie., the NoteBook should work with anyone's name.
As an example. An input of
thomas abram herring
would generate:
Herring, Thomas A.

**Question (3):** (50-points) Write a Mathematica NoteBook that will compute the motion of a bicyclist and the energy used cycling along an oscillating, sloped straight-line path. The path followed will be expressed as

$$H(x) = Sx + A\sin(2\pi x/\lambda) + B\cos(2\pi x/\lambda)$$

where *H(x)* is the height of the path above the starting height, *S* is a slope in m/m, *A* and *B* are amplitudes of sinusoidal oscillations in the path. The wavelength of the oscillations is λ. The forces acting on the bicycle are:

$$\text{Wind Drag} \quad F_d = 1/2 A_r C_d \rho V^2$$

$$\text{Rolling Drag} \quad F_r = M_r g C_r$$

where $A_r$ is the cross-sectional area of the rider, $C_d$ is the drag coefficient, r is the density of air and *V* is the velocity of the bike. For the rolling drag, $M_r$ is the mass of the rider and bike, *g* is gravitation acceleration and $C_r$ is rolling drag coefficient.

The bicyclist puts power into the bike by pedaling. The force generated by this power is given by

$$\text{Rider force} \quad F_r = P_r / V$$

where $F_r$ is the force produced by the rider, $P_r$ is power used by the rider and *V* is velocity that the bike is traveling (the force is assumed to act along the velocity vector of the bike). Your NoteBook can assume that the power can be used at different rates along the path. The energy used will be the integrated power supplied by the rider. Assume that there is maximum value to the rider force.

Your code should allow for input of the constants above (path and force coefficients). The NoteBook can assume a constant power scenario and constant force at low velocities.

As a test of your NoteBook use the following constants to compute:

(a) Time to travel and energy used to travel 10 km along a path specified by S=0.01, A=5.0 m, B=0.0 m and $\lambda$= 2km, with constant power use of $P_r$ =100Watts and a maximum force available of 20N.
(b) The position and velocity of the bike tabulated at a 100-second interval.
(c) Add graphics to your NoteBook which plots the velocity of the bike as a function of time and position along the path.

Assume the following values

$C_d$ = 0.9

$C_r$ = 0.007

$A_r$ = 0.67 m$^2$

r = 1.226 km/m$^3$

*g* = 9.8 m/s$^2$

$M_r$ = 80 kg

In this case, the Mathematica NoteBook will not be of the type used for fortran and C/C++. Look at the documentation on NDSolve for this problem.

Your answer to this question should include:

(a) The algorithms used and the design of your NoteBook
(b) The Mathematica NoteBook with your code and solution (I run your NoteBook).
(c) The results from the test case above.

12.010 Computational Methods of Scientific Programming
Fall 2011