0.1 Midterm Project

Due: Monday, October 19, 2007 at 10:00 pm

This midterm exam is worth 10% of the course grade. Remember that it should be treated as a take-home exam. You should not discuss the project with your classmates or anyone else, except for the course instructors. If you need help, please *do* ask your course instructors. When you do this, you must acknowledge in your report how you were helped. (actually, you should always do this.) Your codes should be strictly your own. You may use your own previous homework without limitation.

These problems focus entirely on combinations of methods that we have used. You do not need to use pointers, arrays, structures, xmgrace, or Philsplot to solve these problems (but you can if you want).

Your programs should separate the task into multiple functions, each with a documented purpose. Programs should be properly indented and commented, with variable names that communicate their purpose to the reader. Your report should discuss your solutions, including the numerical accuracy of your chosen step sizes.

Problem 1: Consider a ball thrown vertically into the air at an upward velocity V_0 from a height $y = 0$. The ball suffers a drag quadratic in the velocity with a terminal velocity of 15 m/s. Compute what the velocity V_0 must be to make the ball fall back to its initial height in exactly 3 seconds. What is the velocity when the ball returns to its initial point?

Compare your result to the exact solution in the drag-free case.

This is known as a mixed boundary problem: we have specified the position at two times, rather than position and velocity at one (initial) time. The position at a time of 3 seconds can be considered as a function of V_0. Search for the appropriate V_0 using the bisection algorithm. For any given V_0, you should solve the ball's motion with the second-order Runga-Kutta method.

Problem 2: Compute the integral

$$\int\int re^{-r^2}\;dx\;dy \tag{1}$$

where $r = \sqrt{x^2 + y^2}$, within the unit circle ($x^2 + y^2 < 1$) by two different methods.

First, convert to polar coordinates and solve the radial integral to high accuracy using either the trapezoidal or midpoint method. Quantify the accuracy of your solutions. Use enough bins that you can get an answer to 10^{-8}.

Second, leave the problem in Cartesian coordinates and solve it as a nested set of integrals, applying the trapezoidal or midpoint method to both integrals. Note that the integrand of the outer integral is itself an integral, so you have to solve the inner integral repeatedly to find the value of the outer integrand. Remember that the limits of the inner integral are non-trivial in Cartesian coordinates.

Use the same number of bins for both coordinates. How many bins per dimension do you need to get accuracy to 10^{-4}? How many total evaluations of the integrand is this?

Clearly multi-dimensional integrals can be expensive to compute if we don't have a symmetry to exploit!

For the Cartesian treatment, we are applying the numerical integration twice with two different integrands. There are various ways to code this, but we haven't discussed the best way (pointers to functions) yet. You may choose to write the midpoint/trapezoidal method twice, so as to employ different integrands in each. Or you might have the integrand function have an argument to say whether it is the inner or outer integrand, and if outer, call the numerical integrator again.

Problem 3: In piloting spacecraft around the solar system, as in sending spacecraft to Mars, the course is adjusted by briefly firing rocket motors to change the velocity. This problem illustrates what happens when you do this.

A satellite is in a circular orbit around the sun. As it crosses the x axis, its rocket engine is fired briefly. The satellite receives an impulse, or instantaneous increment in its momentum, in the direction of its motion. What is its subsequent orbit? Note that the satellite receives only a single impulse, not an impulse every time it crosses the x axis. Hint: how do you tell when it has completed one orbit? Use the same logic you used last week when you were checking the period of the orbit, which, in turn, is the same logic you used the week before in figuring out when the ball hit the ground. Also, you do not need to worry about kicking when the satellite is *exactly* at the x axis; if you are using a small time step, just using the first step after the satellite crosses the axis will be close enough.

a) Write a program to integrate the orbit of the satellite with the second-order Runga-Kutta method (note that last week's assignment should be a good starting point). Start the satellite at a distance of 1 AU along the x axis in a circular orbit. Set up the code so that after one revolution of a circular orbit, the satellite gets a kick; continue the integration for another two or three years. Print the x and y coordinates during this process (both before and after the impulse), and use `graph`, `simpleplot`, or `PhilsPlot` to plot the orbits. Make sure that the satellite only gets kicked once, instead of every time it crosses the x axis. Use a kick which increases the satellite's velocity by 10%. Include a plot of the orbits, before and after the firing of the rocket motor, in your homework submission.

b) Repeat the problem, now applying the kick in the radial direction. In other words, as the satellite crosses the positive x axis, it gets an increment in the x component of its velocity. Again, plot the orbit before and after the firing of the motor. How does the answer differ from the case of the tangential kick?