Computational Physics
Physics 410 2014W
Assignment 4: ODE Solver Specifications
Due: Wednesday, November 5, 2014 6PM

Design and code an ODE integrater that can solve a N-dimensional first order ODE

$$\frac{d\vec{Y}}{dt} = \vec{f}(\vec{Y}, t)$$

It should take as input a vector ”Yi” $= \vec{Y}_i$ of initial data, the name of a function ”derivs” that returns the vector of derivatives $\frac{d\vec{Y}}{dt}$, a set of time points ”tpoints” (of length N_t) the solution is desired at, an integration method ”method” to advance each timestep, and a single parameter ”param” that can be used to adjust the accuracy of the solution for a given method. It should return the solution to the ODE as a N by N_t array ”Ysoln”. It should also separately return a count of the total number of times that ”derivs” was called while finding the solution.

You should include methods: Euler (euler), Second-Order Runge-Kutta (rk2), Fourth-Order Runge-Kutta (rk4), Fourth-Order Runge-Kutta with Adapative Stepzie (rk4adapt), Leapfrog (leap), Verlet (verlet), Modified Midpoint (mm), and Bulirsch-Stoer (bs). Please use these conventions when selection your method name.

For the fixed-step methods like the euler, rk2, rk4, leap, verlet, and mm methods ”param” should be a factor indicating how many substeps to divide the problem in between each adjacent set of time points. (i.e., for param $= 1$ you have a single step between tpoints[i] and tpoints[i+1], while for param $= 5$ you have a five substeps between tpoints[i] and tpoints[i+1] even though you only return the values at tpoints[i] and tpoints[i+1]). For fixed-accuracy methods like rk4adapt or bs ”param” should be the inverse of the target fractional accuracy of each step between tpoints[i] and tpoints[i+1] (i.e., for param $= 100$ you demand 1% accuracy on each full step).

Note: For the verlet method please take the convention that the first $N/2$ variables are position-like and the second $N/2$ variables are velocity-like.

Please include comments in your code on how to call your routine, and a short paragraph describing how to use your routine in your report. Code your routine so that it takes parameters in the order:

odeintegrator(Yi,derivs,tpoints,method,param)