CS 101.3: Numerical Geometric Integration

Homework Assignment #1
Due date: Jan 21st 2009 at the beginning of class.

Abstract

In this assignment, you are asked to expand on some of the derivations shown in class, and implement a few basic integrators for a simple dynamical system, namely, a pendulum. Please note that the honor code applies: please do the derivations yourself even though many are available on the internet. If you have questions email patrickm@cs.caltech.edu - I will be happy to meet upon request.

1 Theory Part

1.1 Accuracy of Quadrature Rules

Remember that we defined the *trapezoidal rule* for numerical integration as:

$$\int_a^b f(x)\,dx \approx T := \frac{f(a)+f(b)}{2}\,(b-a), \tag{1}$$

while we defined the *midpoint rule* as:

$$\int_a^b f(x)\,dx \approx M := f(\frac{a+b}{2})\,(b-a).$$

- While they are both exact for linear function and only approximate for higher order polynomials, they provide a different numerical answer in general. Prove that:

  $$\int_a^b f(x)\,dx - T = -\frac{(b-a)^3}{12} f''(a) + O((b-a)^4)$$

  and

  $$\int_a^b f(x)\,dx - M = \frac{(b-a)^3}{24} f''(a) + O((b-a)^4).$$

  Hint: Use Taylor expansions around a.

- Recall Simpson's rule S:

  $$\int_a^b f(x)\,dx \approx S := \left[\frac{1}{6}f(a) + \frac{4}{6}f(\frac{a+b}{2}) + \frac{1}{6}f(b)\right]\,(b-a).$$

Leveraging your above proofs, derive the (overly conservative) error bound for Simpson's rule:

$$\int_a^b f(x)\, dx - S = O((b-a)^4).$$

1.2 Butcher's Tableaus

- Find Butcher's notation for the "implicit trapezoidal" scheme to solve an ODE of the type $y' = f(y)$:

$$y_{k+1} = y_k + h\frac{f(y_k) + f(y_{k+1})}{2}.$$

- Find Butcher's notation for the Runge scheme:

$$y_{k+1} = y_k + h\frac{f(y_k) + f(y_k + hf(y_k))}{2}.$$

- Create one implicit and one explicit time integration scheme based on Simpson's rule. Write them both in terms of $y_{k+1} = ...$ and using Butcher's notation.
- How can you tell quickly by looking at a tableau whether a scheme is implicit or explicit?

2 Implementation part

In Mathematica, implement the following schemes to simulate the motion of a pendulum in time:

- explicit Euler
- implicit Euler
- RK4
- symmetrized Euler, *i.e.*, one step of explicit Euler followed by one step of implicit Euler
- your two integrators based on Simpson's rule above

Remember that the pendulum's angle θ with the vertical satisfies the second-order ODE: $\ddot{\theta} = -\frac{g}{L} sin(\theta)$, where g is gravity and L is the pendulum's length. This equation can be written as two first-order ODEs: $\dot{\theta} = v$ and $\dot{v} = -\frac{g}{L} sin(\theta)$ where v is the velocity of the pendulum. Hence each timestep will require using the integrator to update both θ and v. Start the pendulum horizontal ($\theta = \pi/2$) and with zero velocity. Take $g = L = 1$ and run your simulations with timestep sizes of .001, .01 and .1 long enough for the pendulum to go through at least 3 periods.

Plot the various trajectories you get out these integrators in *phase-space*, that is, in the plane (θ, v). For the .01 timestep size, also plot the energy as a function of time, where the energy is defined as $\frac{1}{2}v^2 - \frac{g}{L}cos(\theta)$. Finally, for one example of your choice create an animation of the pendulum swinging.