Numerical Analysis Day 12: ODEs and IVPs

Jeff Parker

November 20th 2012

1 Approximating Initial Value Problems (IVP)

We introduce a number of approximation methods with Matlab implementations. We start with a first order differential equation, of the form

$$y'(t) = f(t, y(t))$$

We are given an initial value, such as $y(t_0)$, and with to know y at some future time t_n. Our methods will proceed one step at a time, computing values w_i which we hope are close to $y(t_i)$.

Euler's method:

$$w_{i+1} = w_i + hf(t_i, w_i)$$

```
% one step of Euler's Method
% Input: function, current time t, current value y,  stepsize h
% Output: approximate solution value at time t+h
function y=eulerstep(f, t, y, h)
   y = y + h*f(t,y);
```

Trapezoid method:

$$w_{i+1} = w_i + \frac{h}{2}(f(t_i, w_i) + f(t_i + h, w_i + hf(t_i, w_i)))$$

```
% One step of Trapazoid Method
% Input: function, current time t, current value y,  stepsize h
% Output: approximate solution value at time t+h
function y = trapstep(f, t, y, h)
```

```
f1 = f(t,y);
y1 = y + h*f1;
y  = y + (h/2)*(f1 + f(t+h, y1));
```

The **Runge-Kutta Methods** are a series of IVP solvers

The first of these is the midpoint method, a two-stage second-order method.

$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right)$$

The best known **Runge-Kutta Method** may be this fourth-order method

$$w_{i+1} = w_i + \frac{h}{6}(s_1 + 2s_2 + 2s_3 + s_4)$$

where

$s_1 = f(t_i, w_i)$
$s_2 = f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}s_1\right)$
$s_3 = f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}s_2\right)$
$s_4 = f(t_i + h, w_i + hs_3)$

```
% One step of the Runge-Kutta order 4 method
function y = rungakutta4step(f, t, w, h)
   s1 = f(t, w);
   s2 = f(t+h/2, w + h*s1/2);
   s3 = f(t+h/2, w + h*s2/2);
   s4 = f(t+h, w + h*s3);
   y = w + h*(s1 + 2*s2 + 2*s3 + s4)/6;
```

2 Systems of ODEs

To model real systems, we need to handle systems with second, third, and higher derivatives. We also need to handle more than one variable to capture events in the world around us.

First, we discuss how to deal with multiple variables, such as this example. We start with a system of first order differential equations.

$$y_1' = y_2^2 - 2y_1$$

$$y_2' = y_1 - y_2 - ty_2^2$$

$$y_1(0) = 0$$

$$y_2(0) = 1$$

To simulate the trajectory of (y_1, y_2), we take a step in each variable. In this example, we would follow the approximations $(w_{i,1}, w_{i,2})$ tracking (y_1, y_2) at time $t = i$.

$$w_{i+1,1} = w_{i,1} + h(w_{i,2}^2 - 2w_{i,1})$$

$$w_{i+1,2} = w_{i,2} + h(w_{i,1} - w_{i,2} - t_i w_{i,2}^2)$$

Next we discuss how to model systems with higher order derivatives. An object in free fall obeys the rule $a = g$ or $y'' = g$. This is a second order differential equation.

We turn this into the following first order system with two variables: y_1, the position, and y_2, the velocity.

$$y_1' = y_2$$

$$y_2' = g$$

Thus turns one second order differential equation into a system of two first order differential equations.