Lecture 27

Numerical ODEs

L. Olson

Department of Computer Science
University of Illinois at Urbana-Champaign

April 24, 2006

Wiki of the day

Highlight some interesting progress:

Caching in Matlab

- interesting as it highlights an essential component of our linear algebra section: level 1, 2, and 3 BLAS routines
- dot products, matrix-vector multiplies, matrix-matrix multiplies
- some intriguing matlab tests that you can use on your own machine to test caching
- +5 extra wiki points

On to ODEs

A basic Ordinary differential equation (ODE) is

$$\frac{dy}{dt} = f(t, y)$$

where

$$\begin{aligned} t &= \text{independent variable} \\ y &= \text{dependent variable only depends on } t \end{aligned}$$

Need one more piece of information (since we have only one derivative):

$$y(t_0) = y_0$$

ODEs: an example

$$\frac{dy}{dt} = -\frac{1}{2y}$$

How to solve? ...lots of ways. Try separation of variables:

$$2ydy = -dt$$
$$\Rightarrow \int 2ydy = -\int dt$$
$$\Rightarrow y^2 = -t + c$$
$$\Rightarrow y = \sqrt{c - t}$$

Example

Equation	Initial Value	Solution
$\frac{dy}{dt} = y + 1$	$y(0) = 0$	$y = e^t - 1$
$\frac{dy}{dt} = 6t - 1$	$y(1) = 6$	$y = 3t^2 - t + 4$
$\frac{dy}{dt} = \frac{t}{x+1}$	$y(0) = 0$	$y = \sqrt{t^2 + 1} - 1$

ODEs

- ODEs arise in every field of science and in many other disciplines such as economics, finance, etc.
- problem: ODEs are not always easily solved analytically:

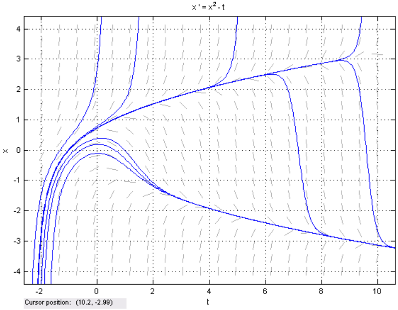
$$\frac{dy}{dt} = f(y,t)$$
$$y(t_0) = y_0$$

  yields

$$y(t) = y_0 + \int_{t_0}^{t} f(s,y)\,ds$$

- what to do?

Vector Fields



- $y' = y^2 - t$
- vector fields provide some insight (dfield7.m)
- another approach: numerically approximate the solution

Goals in ODEs

Big Methods:

- Euler's Method: Use the first term of a Taylor Series expansion
- Midpoint Method: improve on Euler
- Runge-Kutta Methods: large improvement

Big Ideas:

- In each of the three, we improve accuracy...at a cost
- What is the accuracy and how is it measured?
- How do we implement these routines?
- What's used in practice? (RK variants $\rightarrow$ ode23 and ode45 in Matlab)

Today: Euler's Method

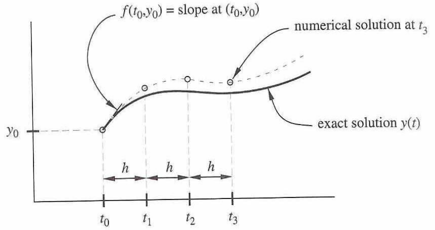


Figure 12.3 Conceptual idea behind the numerical approximation to the exact solution of $dy/dt = f(t, y)$, $y(t_0) = y_0$.

- Numerical ODEs: take small steps in time, making an approximation to the solution along the way
- let t be given by $t_j = t_0 + jh$
- t_j are the disrete points, h is the stepsize
- we want approximations to $y(t)$ at t_j: y_j
- error: $e_j = y_j - y(t_j)$

Euler

- We have the solution at t_0: $y(t_0) = y_0$
- And we know how it changes from there: $y' = f(y, t)$
- Let's use this information to make our first step
- Taylor Series:

$$y(t) = y(t_0) + (t - t_0)y'(t_0) + \frac{(t - t_0)^2}{2}y''(t_0) + \dots$$

- using only what we know (y and y') we have

$$y(t) \approx y(t_0) + (t - t_0)y'(t_0)$$

or

$$y(t) \approx y(t_0) + (t - t_0)f(t_0, y_0)$$

Euler

- Then the solution at the next point (t_1) becomes

$$y(t_1) \approx y(t_0) + (t_1 - t_0) f(t_0, y_0)$$

  or

$$y_1 = y_0 + h f(t_0, y_0)$$

- This is Euler's Method:

Euler's Method

$$y_j = y_{j-1} + h f(t_{j-1}, y_{j-1})$$

Euler

Example

$$\frac{dy}{dt} = t - 2y, \qquad y(0) = 1$$

Exact solution: $y = (1/4)(2t - 1 + 5e^{-2t})$

(demoEuler.m)(odeEuler.m)(rhs1.m)

Euler: Error

```
Max error =  3.12e-001 for h = 0.400000
Max error =  1.12e-001 for h = 0.200000
Max error =  5.02e-002 for h = 0.100000
Max error =  2.40e-002 for h = 0.050000
Max error =  1.17e-002 for h = 0.025000
```

- What do we notice about the error?
- halve the step size $\Rightarrow$ halve the error
- hmmmmm, so do we have $max(e_h) = ch^p$?

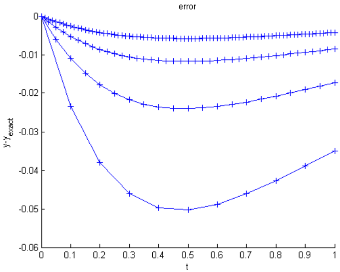
$$\frac{max(e_h)}{max(e_{2h})} \approx \frac{1}{2}$$

$$\Rightarrow \frac{ch^p}{c(2h)^p} = \frac{1}{2}$$

$$\Rightarrow \frac{1}{2^p} = \frac{1}{2}$$

$$\Rightarrow p = 1$$

Euler: Error



- we're concerned with *discretization* error: related to truncation error
- what is the error from step-to-step? *local* LDE
- what is the error over the interval? *global* GDE