Numerical Computing

(or . . . Putting the Numbers Back into Maths)

D. Burgess

Astronomy Unit

School of Mathematical Sciences

Goldsmiths' Mathematics Course July 2009

The Motivation: Applying mathematics

Google: “Need for Speed”, or “Grid damage”

Why It's Important

- Want to use mathematical models of the real world
- Standard functions and analytical solutions are just a drop in the ocean!
- Everyday systems exhibit complex behaviour
  - sensitive to initial conditions
  - governing parameters have complex behaviour
  - systems usually have many interacting parts
- Solution: Work with numbers!

A Long History

- Computation of comet's trajectory - Clairaut 1758.
- Trigonometric and logarithmic tables
- The first computers were human!

Computing floor of the Mathematical Tables Project. National Archives and Records Administration, Schlesinger Library, Harvard.

Computers (and Computation) are Finite

Finite precision arithmetic

$$1.0 + 1.0 \times 10^{-10} = 1.0$$

Also affects initial conditions etc.

Need to know something about how numbers are stored

Errors from finite precision cannot be escaped!

From Continuous to Discrete Processes

Real world models start from continuous processes – integration and differentiation

Ordinary Differential equations for equations of motion

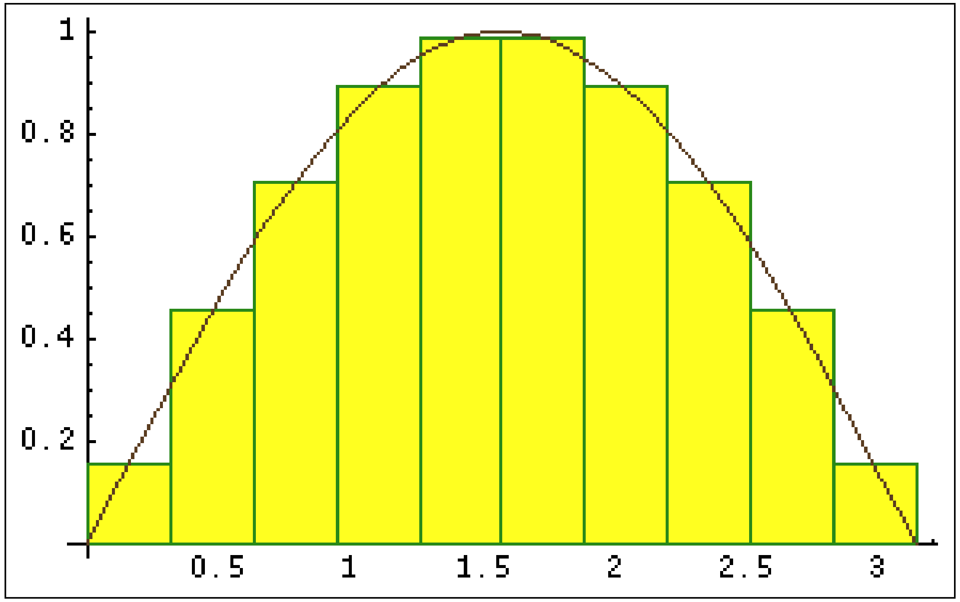
Partial differential equations for field (continuous) quantities

Riemann Integration: where ΔA is area of strip width Δx under curve of function $f(x)$

$$\int f(x)\,dx = \lim_{n\to\infty} \sum_n \Delta A$$

Numerical Integration:

- Just keep n finite
- . . . and compute the sum
- Value of sum should converges as n increases
- Precision increases
- Errors?



http://www.hostsrv.com/webmaa/app1/MSP/webm1010/riemann

Google: MSP riemann

Numerical Integration – 2

Error is introduced by making approximation of finite strips

- There are many different ways of approximation
- Behaviour of error changes with method
- Finite precision errors will always (eventually) be important

$$\text{Define Error } E_n : \qquad E_n = \frac{|I_\infty - I_n|}{I_\infty}$$

Order m of error E_n as a function of strip width Δx

$$E_n \propto (\Delta x)^m$$

$m = 1$ for First order method, etc.

Numerical Differentiation – 1

For quantity x – only deal with finite set of discrete values x_i,

$$x_i = i\,\Delta x, \qquad \text{where } \Delta x \text{ is called stepsize.}$$

Function $f(x)$ only evaluated at points x_i, so $f_i = f(x_i)$.

Finite Difference Approximation (FDA):

Simplest kind is just the chord approximation:

$$\frac{df}{dx} \approx \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

Numerical Differentiation – 2

But . . . to which point do we assign derivative:

$$\left.\frac{df}{dx}\right|_{x=x_i} = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} \qquad \text{Forward difference}$$

$$\left.\frac{df}{dx}\right|_{x=x_i+1} = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} \qquad \text{Backward difference}$$

Notice that related to Taylor series expansion of f.

- Finite differences of different orders: 1st order, 2nd order etc
  - corresponding to error behaviour

Solving Ordinary Differential Equations – Finite Difference Method

$$\frac{dy}{dx} = f(x, y) \qquad \text{solve for } y(x) \text{ given } y(0)$$

Replace derivative (at x_i) by finite (forward) difference approximation

$$\frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f_i$$

Solve for y_{i+1}

$$y_{i+1} = y_i + \Delta x f_i \qquad i = 0, \ldots \text{ and given } y_0 = y(0)$$

This is a recursive definition of a sequence $\{y_i\}, i = 0, 1, \ldots$

Euler's Method for ODE – Example

$$\frac{dy}{dt} = -y \qquad \text{solve for } y(t) \text{ given } y(0) = 1$$

Exact solution is $y = e^{-t}$.
Euler method gives sequence $y_{i+1} = y_i(1 - \Delta t)$, $y_0 = 1$ for $i = 0, \ldots$.

```
EulerEx1 := proc( t_0, y_0, n_steps, delta_t )

t_0      initial time  (use 0.0)
y_0      initial y value (use 1.0)
n_steps  number of steps to follow solution  (start with eg 5)
delta_t  step size
```

Investigate what happens for different values of step size!

Things flying through the air (Finally!)

Newton's equation of motion

$$\frac{d^2y}{dt^2} = -m \qquad y(0) = 0, v_y(0)$$

$$\frac{d^2x}{dt^2} = 0 \qquad x(0) = 0, v_x(0)$$

Second equation for horizontal motion: $x = tv_x(0)$

Solve second order ODE as *coupled system of first order ODE*.

$$\frac{dV_y}{dt} = -m \qquad \Rightarrow \quad V_{y,(n+1)} = V_{y,(n)} - m\Delta t$$

$$\frac{dy}{dt} = V_y \qquad \Rightarrow \quad y_{n+1} = y_n + V_{y,(n)}\Delta t$$

Note: the RHS of equation of motion can depend on time, or be as complicated as required, eg. add air resistance, variable mass, extra forces etc.

Projectiles – Example

Solve coupled sequences for vertical velocity V_y and position y

$$
\begin{aligned}
V_{y,(n+1)} &= V_{y,(n)} - m\Delta t \\
y_{n+1} &= y_n + V_{y,(n)}\Delta t
\end{aligned}
$$

```
EulerEx2 := proc( m, vx_0, vy_0, n_steps, delta_t )
```

```
The function EulerEx2 takes 5 arguments:
m         mass of object,
vx_0      initial x velocity,
vy_0      initial y velocity,
n_steps   number of steps to follow trajectory
delta_t   step size to use
```

Wrap Up

- The key advantage of numerical methods
  - arbitrarily complicated models (parameters, functions, etc) can be used, provided they can be represented computationally
- What is next?
  - More accurate finite approximations
  - Approximations which conserve physical quantities
  - Follow many parts of a system simultaneously
  - Allow for interaction between parts
  - Simulate the Universe!

    `http://www.mpa-garching.mpg.de/galform/gadget/index.shtml#movies`