CS 101

Numerical Geometric Integration

Patrick Mullen
Mathieu Desbrun

Classical Numerical Integration

Discrete approximation of the real solution
- discrete in time (i.e., a set of positions)
- just like a movie (~30 frames a second)

Various techniques:
- Some ways are better than others
- **Never perfect**
- Always better if we know the type of solutions we should get...

Today's Show

"Classical" Numerical Integration
- ODE to make it simple
  - we will see PDEs next time
- a quick fly-though of key techniques
  - Euler, Adams, Runge-Kutta,...
- consider it a warm-up
  - but we will reuse some notions later

Math Reminder

The Taylor series expansion of *f(x)*

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f^{(2)}(x_0)}{2!}(x-x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x-x_0)^3 + \ldots$$

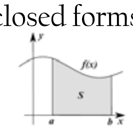
$$= \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(x_0)\,(x-x_0)^k$$

- how to write it for $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$?
  - Jacobian, Hessian..
- try: $\frac{f(x_0+h)-f(x_0-h)}{2h}$ $\frac{f(x_0-h)-2f(x_0)+f(x_0+h)}{h^2}$

Math Reminder

Quadrature rules
- integrals don't always have closed forms

$$\int_a^b f(x)\,dx$$



- rectangle rule: $\int \approx f(a)\,[b-a]$
- trapezoidal rule: $\int \approx \frac{f(a)+f(b)}{2}[b-a]$
- midpoint rule: $\int \approx f(\frac{a+b}{2})\,[b-a]$

same order, yet one is better than the other....

- higher order via Gauss, Newton-Costes

Initial Value Problem

Ordinary Diff. Equation of the form:

Find $y(t)$ such that

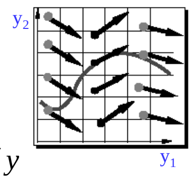
$$\begin{cases} \frac{dy}{dt} = f(y(t)) \\ y(0) = \alpha \end{cases}$$

- velocity fct of position
- *f* assumed smooth

Initial Value Problem

"Trace" forward

$$\begin{cases} \frac{dy}{dt} = f(y(t)) \\ y(0) = \alpha \end{cases}$$



- f is like a "current," driving the evolution of y
- you can find a symbolic solution
  - if you are really lucky
  - for instance, $\dot{y} = -ky$ (you have 10 seconds to solve this one)

Euler's Method

Simplest method (1768):

- introduce time steps $t_k = k\,h$
  - regular discretization of time
  - with time step size h
- name of the game: find $y(t_k) \sim y_k$
- by approximating the real "flow" over h

$$y_{k+1} = y_k + hf(y_k)$$

Euler's Method

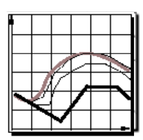
$\frac{dy}{dt} = f(y(t))$

Taylor series to make sense of it....

$$y(t_{k+1}) = y(t_k + h) = y(t_k) + hf(y_k) + O(h^2)$$

So *local* step error: $O(h^2)$

Can be really, really bad:



Euler's Method—backwards...

Side note

- Euler's obviously not time reversible
- what if you find y_{k+1} so that time reversal produces Euler's method?

$$y_k = y_{k+1} - hf(y_k) \qquad y_{k+1} = y_k + hf(y_{k+1})$$

- on pendulum, would now die down
  - instead of blowing up
- called **Implicit Euler**

Euler's Method

What about global error?

- to reach time T, it takes T/h steps
- so global error is O(h)

More generally

- for one-step methods with local error $O(h^{n+1})$, global error is $O(h^n)$
- how to get a higher-order method?

Taylor Expansion Methods

Euler revisited

$$\boldsymbol{y(t_k + h) = y(t_k) + h\dot{y}(t_k) + \frac{h^2}{2}\ddot{y}(t_k) + O(h^3)}$$

But: $\dot{y}(t) = f(y(t)) \rightsquigarrow \ddot{y}(t) = f'(y(t))f(y(t))$

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

And so on...

- but it requires higher derivatives of f....

Multistep Methods

Avoid higher-order derivatives
- do multiple evaluations instead

Main idea ("Adams methods", circa 1855+)

$$\int_{t_k}^{t_{k+1}} y'(\tau)d\tau = y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(y(\tau))d\tau$$

- use a quadrature rule to approx. integral

$$\int_{t_k}^{t_{k+1}} f(y(\tau))d\tau \approx h\,[\beta_{d-1}f(y_k) + \beta_{d-2}f(y_{k-1}) + \ldots + \beta_0 f(y_{k-d+1})]$$

Multistep Methods

What do you get?

$$y_{k+1} = y_k + h\,[\beta_{d-1}f(y_k) + \ldots + \beta_0 f(y_{k-d+1})]$$

- reuses past evaluations!
- quadrature made to be of order d
  - so integrator of local error $O(h^{d+1})$
- careful: global error can be tricky
  - Dahlquist, zero-instability

Slew of Variants

Many improvements can be done
- Implicit Adams
  - quadrature uses y_{k+1}
  - no longer an explicit update
- Predictor-Corrector
  - explicit as initial guess for implicit solve
- General linear multistep methods

$$\alpha_d y_{k+1} + \ldots + \alpha_0 y_{k-d+1} = h\,[\beta_{d-1}f(y_k) + \ldots + \beta_0 f(y_{k-d+1})]$$

Runge-Kutta Methods

What about recursive evaluations of f?
- notice that Euler's is based on

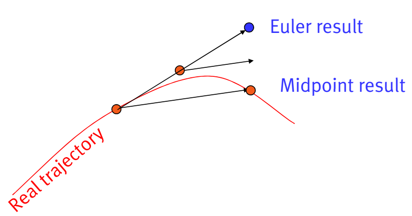
  $$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(y(\tau))d\tau$$

  with *rectangle quadrature*
- Runge's idea:
  - what about using midpoint instead?

$$y_{k+1} = y_k + hf(Y) \text{ with } \mathbf{Y = y_k + \frac{h}{2}f(y_k)}$$

Midpoint method

Geometric interpretation



Generalization: RK Methods

Butcher's notation

$$\mathbf{y_{k+1} = y_k + h\sum_{i=1}^{s} b_i f(Y_i)}$$

$$Y_i = y_k + h\sum_{j=1}^{s} a_{ij} f(Y_j)$$

An s-stage method has $s(s+1)$ params
- can be implicit, recursive, etc...

$$\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \cdots & \cdots & \ddots & \cdots \\ a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline b_1 & b_2 & \cdots & b_s \end{array}$$

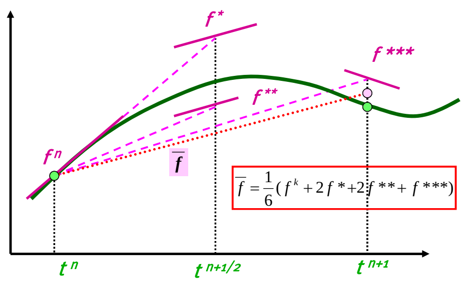
Runge Kutta 4

Quite famous...

$$y^{k+1} = y^k + \frac{h}{6}\underbrace{(f^k + 2f^* + 2f^{**} + f^{***})}_{\text{weighted-average "slope"}}$$

$$\begin{cases} Y_1 = y^k + \frac{h}{2} f^k, \quad f^* = f(Y_1) \\ Y_2 = y^k + \frac{h}{2} f^*, \quad f^{**} = f(Y_2) \\ Y_3 = y^k + h\, f^{**}, \quad f^{***} = f(Y_3) \\ y^{k+1} = y^k + h\,\overline{f}, \quad \overline{f} = \frac{1}{6}(f^n + 2f^* + 2f^{**} + f^{***}) \end{cases}$$

Runge-Kutta 4 Method



Butcher Tableau

Let's Connect the Dots

- Euler's: $\frac{0}{1}$
- Implicit Euler: $\frac{1}{1}$
- Midpoint method: $\frac{1/2}{1}$
- RK4:

$$\begin{array}{cccc} 0 & & & \\ 1/2 & & & \\ 0 & 1/2 & & \\ 0 & 0 & 1 & \\ \hline 1/6 & 2/6 & 2/6 & 1/6 \end{array}$$

Order of Accuracy

Analysis readily available

- involves conditions on *a* and *b*

$\sum_i b_i = 1$ for order 1 $\sum_{i,j,k} b_i a_{ij} a_{ik} = \frac{1}{3}$
$\sum_{i,j} b_i a_{ij} = \frac{1}{2}$ for order 2 $\sum_{i,j,k} b_i a_{ij} a_{jk} = \frac{1}{6}$ for order 3

- s-stages can be of order *2s* at best

Again, slew of variants

- for instance, *f* or time step fct of ***time***
- huge literature if you are interested

Important Terms

Consistency: local error /h $\rightarrow$ 0

Stability: errors are damped out

- definition varies based on type of eqn
- {A|L|Z|zero...}-stability

Convergence: global error $\rightarrow$ 0

- consistency + stability = convergence
  - for linear PDEs at least

More About ODEs

This was only for first order ODEs!

- whole world out there for other ODEs
- if first or second order, it's probably ok
  - or if it's in particular forms
    - homogenous equations, separable equations
- otherwise, you are on your own...

Monday: PDEs... and first homework