CSE276C - Integration of ODEs

Henrik I. Christensen

Computer Science and Engineering
University of California, San Diego
http://cri.ucsd.edu

October 2020

Literature

- Numerical Recipes: Chapter 16
- Numerical Renaissance: Chapter 9

Outline

Introduction

- For integration of a set of ordinary differential equations you can always reduce it into a set of first order differential equations.
- Example

$$\frac{d^2y}{dx^2} + q(x)\frac{dy}{dx} = r(x)$$

- which can be rewritten

$$\begin{array}{rcl} \frac{dy}{dx} & = & z(x) \\ \frac{dz}{dx} & = & r(x) - q(x)z(x) \end{array}$$

- where z is a new variable

Small example

- Consider a simple motion of a mass when actuated by a mass

$$F(u_1) = m\frac{d^2 u_1}{dt^2}$$

- We can rewrite this as

$$\frac{d^2 u_1}{dt^2} = \frac{1}{m}F(u_1)$$

- We can introduce $u_2 = \frac{du_1}{dt}$ to generate

$$\begin{array}{rcl} \frac{du_1}{dt} & = & u_2 \\ \frac{du_2}{dt} & = & \frac{1}{m}F(u_1) \end{array}$$

OR

$$\frac{du}{dt} = f(u,t) \text{ with } u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

where

$$f = \begin{pmatrix} u_2 \\ \frac{F(u_1)}{m} \end{pmatrix}$$

Introduction (cont)

- The generic problem is thus a set of couple 1st order differential equations

$$\frac{dy_i(x)}{dx} = f_i(x_i, y_1, y_2, \ldots, y_n)$$

- There are three major approaches:
  1. Runge-Kutta: Euler type propagation
  2. Richardson extrapolation / Burlirsch-Stoer: extrapolation type estimation with small step sizes
  3. Predictor-Corrector: extrapolation with correction.
- Runge-Kutta most widely adopted for "generic" problems. Great if function evaluation is cheap
- Burlirsch-Stoer generates higher precision
- Predictor-Corrector is historically interesting, but rarely used today

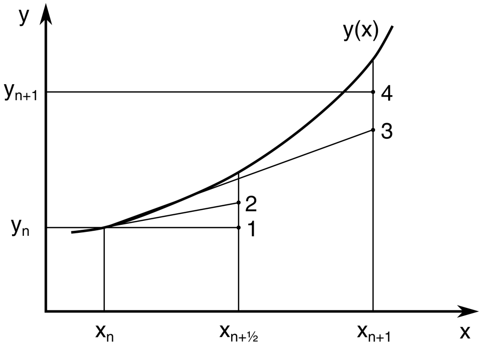
Outline

Runge-Kutta

- The forward Euler method is specified as

$$y_{n+1} = y_n + hf(x_n, y_n)$$

  with $x_{n+1} = x_n + h$
- A problem is that the integration is asymmetric



Runge-Kutta - Stepped Up

- We can use a mid-point to get a closer estimate, i.e.,

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1) \\ y_{n+1} &= y_n + k_2 + O(h^3) \end{aligned}$$

4th order Runge-Kutta

- We can easily extend to richer models. A typical example is the fourth order model

$$
\begin{aligned}
k_1 &= hf(x_n, y_n) \\
k_2 &= hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1) \\
k_3 &= hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_2) \\
k_4 &= hf(x_n + h, y_n + k_3) \\
y_{n+1} &= y_n + \tfrac{1}{6}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{6}k_4 + O(h^5)
\end{aligned}
$$

- By far the most frequently used RK method for ODE integration
- Requires four function evaluations for every step

Adaptive Runge-Kutta

- Could we adjust the step-size?
- Estimation of performance adds an overhead
- What would be an obvious solution?

Adaptive Runge-Kutta

- Could we adjust the step-size?
- Estimation of performance adds an overhead
- What would be an obvious solution?
  1. Do a full step
  2. Do a half step
  3. Compare (could be recursive)
  4. Next
- In general no one goes beyond 5th order Runge-Kutta

PI step control of RK

- Could we use PI control to track stepsize?

PI step control of RK

- Could we use PI control to track stepsize?
- How about

$$h_{n+1} = S h_n \mathrm{err}_n^{\alpha} \mathrm{err}_{n-1}^{\beta}$$

  where S is a scale factor. α and β are gain factors
- Typical default values $\alpha = \frac{1}{k} - 0.75\beta$ and $\beta = \frac{0.4}{k}$ and k is an integer that designates order of the integrator

Outline

Richardson Extrapolation / Burlirsch-Stoer

- Aimed at smooth functions
- Generates best precision with minimal effort
- Things to consider
  1. Does not do well on functions w. table lookup or interpolation
  2. Not well suited for functions with singulaties within intg range
  3. Not well suited for "expensive" functions
- The approach is based on three ideas
  1. Final answer is based on selection of (adaptive) stepsize just like Romberg
  2. Use of rational functions for extrapolation (allow larger h)
  3. Integration method reply on use of even functions
- Typically the steps size H is large and h is 100+ steps

Burlirsch-Stoer - The details

- Consider a modified mid-point strategy

$$x_{n+1} = x_n + H$$

but with sub-steps

$$h = \frac{H}{n}$$

- We can rewrite the integration

$$\begin{array}{rcl} z_0 & = & y(x_n) \\ z_1 & = & z_0 + hf(x_n, z_0) \\ z_{m+1} & = & z_{m-1} + 2hf(x_n + mh, z_n) \ m = 1, 2, 3, \dots n-1 \\ y(n_n + H) & = & \frac{1}{2}[z_n + z_{n-1} + hf(x + H, z_n)] \end{array}$$

- Centered mid-point or centered difference method
- The error can be shown to be

$$y_n - y(x + H) = \sum_{i=0}^{\infty} \alpha_i h^{2i}$$

- The power series implies that we can potentially do less evaluation.

Burlirsch-Stoer - How good is it?

- Suppose n is even and $y_{n/2}$ is the results of half as many steps
- Then

$$y(x+H) = \frac{4y_n - y_{n/2}}{3}$$

- which is arccurate to the 4th order as Runge-Kutta but with $2/3$ less derivative evaluation?
- How do you choose good step sizes for refinement?

Burlirsch-Stoer - How good is it?

- Suppose n is even and $y_{n/2}$ is the results of half as many steps
- Then

$$y(x+H) = \frac{4y_n - y_{n/2}}{3}$$

- which is arccurate to the 4th order as Runge-Kutta but with $2/3$ less derivative evaluation?
- How do you choose good step sizes for refinement?
- One strategy could be

$$n = 2, 4, 6, 8, 12, 16, 24, 32, \ldots \;\; n_{=} 2n_{j-2}$$

more recently a suggestion

$$n - 2, 3, 6, 8, 10, 12, 14, \ldots \;\; n_j = 2(j+1)$$

Step size control for Burlirsch-Stoer

- The error estimate can be tabulated as

$$\begin{array}{ccc} T_{00} & & \\ T_{10} & T_{01} & \\ T_{20} & T_{11} & T_{22} \end{array}$$

- where T_{ij} is the Lagrange interpolation of order i with j points. The relation between the polynomials is

$$T_{k,j+1} = \frac{2T_{k,j} - T_{k-1,j}}{(n_k/n_{k-j-1})^2 - 1} \; j = 0, 1, \ldots, k-1$$

- Each stepsize starts a new row. The difference $T_{kk} - T_{kk-1}$ is an an error estimate
- We can pre-compute the error estimates and use them to decide on step-size selection

Outline

Variable Dynamics

- Sometimes the variable dynamics are very different
- Consider

$$\begin{aligned} u' &= 998u + 1998v \\ v' &= -999u - 1999v \end{aligned}$$

- with u(0) = 1 and v(0) = 0 we can get

$$u = 2y - z \qquad v = -y - z$$

  We can solve and find

$$\begin{aligned} u &= 2e^{-x} - e^{-1000x} \\ v &= -e^{-x} + e^{-1000x} \end{aligned}$$

- The differneces in dynamics would generate challenging step sizes

Outline

Partial Differential Equations

- Huge topics that has its own course - MATH 110/MATH 231 A-C
- Widely used for studies of physical systems - simulation / analysis
- Three main categories
  1. Hyperbolic (wave equation)

  $$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2}$$

  where v is the speed of wave propagation
  2. Parabolic (diffusion equation)

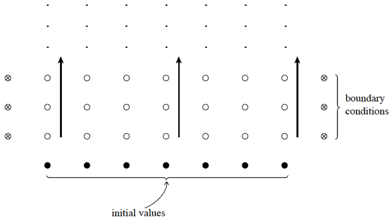
  $$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D \frac{\partial u}{\partial x}\right)$$

  where D is the diffusion coefficient
  3. Elliptic (Poisson equation)

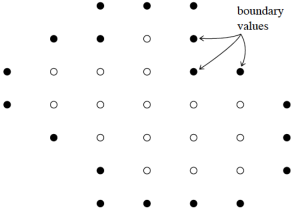
  $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$$

  where $\rho()$ is the source term.

Computational Considerations for PDEs



Initial Value

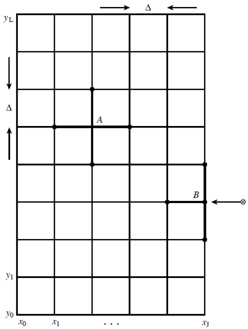


Boundary Value

Source - Numerical Recipes.

Finite difference calculations

- In most cases grid propagation
- Finite differences is a basic approximation
- Final structure is a sparse matrix
- Numerous models and packages to address



Outline

Summary

- We can organize ODEs as a set of coupled 1st order ODEs
- Runge-Kutta is ideal for "cheap" functions, especially 4th order approximation
- Buerlirsch-Stoer is ideal for high-accuracy integration
- It is important to consider the variable dynamics in integration of functions.
- Adaptive stepsize is often valuable as a way to generate realistic complexity