CSE276C - Integration of Functions





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

October 2021

Outline

- Introduction
- ODE Introduction
- 3 Runge-Kutta
- 4 Richardson / Burlirsch-Stoer
- 5 Variable Dynamics
- 6 Partial Differential Equations
- Summary

Introduction

- Interested in integration of function to allow estimation of future value
- Lots of potential applications in robotics
 - Position estimation
 - Path optimization
 - Image restoration
- Consider both end-point and boundary value problems, which anchors the problem

H. I. Christensen (UCSD) Math for Robotics Oct 2021 3 / 41

Introduction - Setting the stage

We are trying to solve

$$I = \int_a^b f(x) dx$$

• trying to solve I = y(b) for the equation

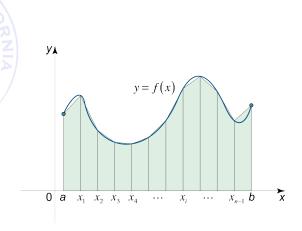
$$\frac{\partial y}{\partial x} = f(x)$$

• with the boundary condition

$$y(a) = 0$$

- Objective to generate a good estimate of y(b) with a reasonable number of evaluations
- Emphasis on 1D problems, but in most cases generalization is straight forward

Setting the stage



Basic use of Simpson's rule

Consider equally spaces data points

$$x_i = x_0 + ih \ i = 0, 1, ..., N$$

the function is evaluated at x_i

$$f_i = f(x_i)$$

The Newton-Cotes rules is then

$$\int_{x_0}^{x_1} f(x) dx = \frac{f_1 + f_0}{2} h + O(f''h^3)$$

The Simpson rules is

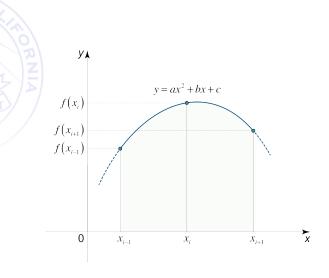
$$\int_{x_0}^{x_2} f(x) dx = \frac{h}{3} (f_0 + 4f_1 + f_2) + O(h^5 f^{(4)})$$

- which is exact to the 3rd degree
- The Simpson $\frac{3}{8}$ rule

$$\int_{x_0}^{x_3} f(x) dx = \frac{h}{8} (3f_0 + 9f_1 + 9f_2 + 3f_3)$$

• There are a series of rules for higher order, check literature

Simpson's Rule



Simpson / Trapezoid Rules

- Clearly the local rules can be chained into a longer evaluation
- \bullet $(x_0, x_1), (x_1, x_2), \dots, (x_{N-1}, x_N)$ to get an extended trapezoid form

$$\int_{x_0}^{x_N} f(x) dx = h(\frac{1}{2}f_0 + f_1 + f_2 + \ldots + f_{N-1} + \frac{1}{2}f_N)$$

• The error estimate is

$$O\left(\frac{(x_N-x_0)f''}{N^2}\right)$$

Trapezoid Rule - Strategy?

• How can you effective use the trapezoid rule?

Trapezoid Rule - Strategy?

- How can you effective use the trapezoid rule?
- Use of a coarse to fine strategy and watch convergence
- This is termed Romberg integration in numerical toolboxes
- In general these methods generate good accuracy for proper functions?

H. I. Christensen (UCSD) Math for Robotics Oct 2021 9 / 4:

Handling of improper function

• What is an improper function?

Handling of improper function

- What is an improper function?
 - Integrand goes to a finite value but cannot be evaluated at a point, such as

$$\frac{\sin x}{x}$$
 at $x = 0$

- **2** Upper limit is ∞ or lower limit is $-\infty$
- Has a singularity at a boundary point, e.g.,

$$x^{-1/2}$$
 at $x=0$

- 4 Has a singularity within the interval at a known location
- 4 Has a singularity within the interval at an unknown location
- If the value is infinite, e.g.,

$$\int_0^\infty x^{-1} dx \text{ or } \int_{-\infty}^\infty \cos x dx$$

it is not improper but impossible

The Euler-Maclaurin Summation Formula

We can write the basic Simpson's rule as

$$\int_{a}^{b} f(x)dx = \frac{h}{2} \left[f(a) + 2 \sum_{k=1}^{N-1} f(a+kh) + f(b) \right] \\ - \sum_{k=1}^{N/2} \frac{h^{2k} B_{2k}}{(2k)!} \left[f^{(2k-1)}(b) - f^{(2k-1)}(a) \right] \\ - \sum_{k=0}^{N-1} \frac{h^{2k+1} B_{2k}}{(2k)!} f^{(2k)}(a+kh+\theta h)$$

- where 2m first derivatives are continuous over (a,b). h = (a-b)/N and $\theta \in (0,1)$
- So what are the B's?
- They are Bernoulli numbers

$$\frac{t}{e^t - 1} = \sum_{n=0}^{\infty} B_n \frac{t^n}{n!}$$

example values

$$B_0 = 1$$
 $B_2 = \frac{1}{6}$
 $B_4 = -\frac{1}{30}$

- Enables you to compute an estimate of the error for a particular integration
- Other integration functions have similar error functions decreasing with

Extended Mid-point Formulation

In many cases using the mid-point is a valuable alternative

$$\int_{x_0}^{x_{N-1}} f(x) dx = h(f_{1/2} + f_{3/2} + \ldots + f_{N-3/2}) + O(\frac{1}{N^2})$$

When combined with the Euler-Maclaurin you get

$$\int_{x_0}^{x_{N-1}} f(x) dx = h(f_{1/2} + f_{3/2} + \dots + f_{N-3/2})
+ \frac{B_2 h^2}{4} (f'_{N-1} - f'_0) + \dots + \frac{B_{2k} h^{2k}}{(2k)!} (f_{N-1}^{(2k)} - f_0^{(2k)}) + \dots$$

• We can do this recursively to estimate convergence

Handling improper integrals

- A trick for improper integrals is to do variable substitution to eliminate a challenge
- Say one of the values is at $-\infty$ or ∞ we can substitute

$$\int_{a}^{b} f(x)dx = \int_{1/b}^{1/a} \frac{1}{t^{2}} f\left(\frac{1}{t}\right) dt$$

Variable substitution

More generally we can do variable substitution as

$$I = \int_a^b f(x)dx = \int_c^d f(x(t))\frac{dx}{dt}dt$$

An example is the Schwartz tanh rule

$$x=rac{1}{2}(b+a)+rac{1}{2}(b-a) anh(t)\ x\in [a,b]\ ext{and}\ t\in [-\infty,\infty]$$

where

$$\frac{\partial x}{\partial t} = \frac{1}{2}(b-a)sech^{2}(t) = \frac{2}{b-a}(b-t)(t-a)$$

ullet sech() converges very rapidly for $t o \infty$ which allows for integration close to singularities

Gauss Integration

- Sometimes uniform sampling is not ideal
- A Gauss model may be an alternative
- The idea is

$$\int_a^b W(x)f(x)dx \approx \sum_{j=0}^{N-1} W_j f(x_j)$$

- For polynomials this can be an exact approximation
- We can approximate f(x) with a Gaussian Mixture and choose weights to match

$$f(x) \approx \sum_{k=0}^{N} W_k N(x|x_k, \sigma_k)$$

Partitioned / Adaptive Integration

 If you have a function with variable dynamics it makes sense to partition the integration into intervals and use Romberg integration on each interval, i.e.

$$I = \int_a^b f(x)dx$$

=
$$\int_a^m f(x)dx + \int_m^b f(x)dx$$

• Rule 1 of data analysis understand your data

Starting

- Simple linear approximations are effective for well-behaved functions
- The order of your approximation can vary according to function complexity
- Using Bernoulli functions we can approximate the estimated error
- Recursive estimation with error monitoring is often effective
- Do a function analysis first to make sure function is proper
- Next we will discuss integration of ODE with standard methods such as Runga-Kutta, Step-size variation, etc.

H. I. Christensen (UCSD) Oct 2021 Math for Robotics

Questions



Questions

H. I. Christensen (UCSD) Math for Robotics Oct 2021 18 / 4:

Outline

- 1 Introduction
- ODE Introduction
- Runge-Kutta
- 4 Richardson / Burlirsch-Stoer
- Variable Dynamics
- 6 Partial Differential Equations
- Summary

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

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

• 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

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

where

$$f = \left(\begin{array}{c} u_2 \\ \frac{F(u_1)}{m} \end{array}\right)$$

Introduction (cont)

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

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

- There are three major approaches:
 - Runge-Kutta: Euler type propagation
 - Richardson extrapolation / Burlirsch-Stoer: extrapolation type estimation with small step sizes
 - 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

H. I. Christensen (UCSD) Math for Robotics Oct 2021 22 / 41

Outline

- 1 Introduction
- 2 ODE Introduction
- 3 Runge-Kutta
- 4 Richardson / Burlirsch-Stoer
- Variable Dynamics
- 6 Partial Differential Equations
- Summary

H. I. Christensen (UCSD) Math for Robotics Oct 2021 23 / 4:

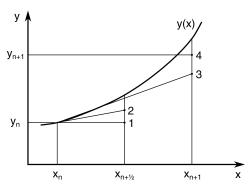
Runge-Kutta

The forward Euler method is specified as

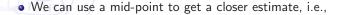
$$y_{n+1} = y_n + hf(x_n, y_n)$$

with
$$x_{n+1} = x_n + h$$

• A problem is that the integration is asymmetric



Runge-Kutta - Stepped Up



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

4th order Runge-Kutta

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

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

- 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?
 - Do a full step
 - O Do a half step
 - Ompare (could be recursive)
 - Mext
- 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

$$h_{n+1} = Sh_n \operatorname{err}_n^{\alpha} \operatorname{err}_{n-1}^{\beta}$$

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

- 1 Introduction
- 2 ODE Introduction
- Runge-Kutta
- 4 Richardson / Burlirsch-Stoer
- 5 Variable Dynamics
- 6 Partial Differential Equations
- Summary

Richardson Extrapolation / Burlirsch-Stoer

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

H. I. Christensen (UCSD) Math for Robotics Oct 2021 30 / 4:

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

$$z_{0} = y(x_{n})$$

$$z_{1} = z_{0} + hf(x_{n}, z_{0})$$

$$z_{m+1} = z_{m-1} + 2hf(x_{n} + mh, z_{n}) m = 1, 2, 3, ...n - 1$$

$$y(n_{n} + H) = \frac{1}{2}[z_{n} + z_{n-1} + hf(x + H, z_{n})]$$

- 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

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

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

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

- 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

- 1 Introduction
- 2 ODE Introduction
- Runge-Kutta
- 4 Richardson / Burlirsch-Stoer
- Variable Dynamics
- 6 Partial Differential Equations
- Summary

Variable Dynamics

- Sometimes the variable dynamics are very different
- Consider

$$u' = 998u + 1998v v' = -999u - 1999v$$

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

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

We can solve and find

$$u = 2e^{-x} - e^{-1000x}$$

$$v = -e^{-x} + e^{-1000x}$$

The differneces in dynamics would generate challenging step sizes

Outline

- 1 Introduction
- ODE Introduction
- Runge-Kutta
- A Richardson / Burlirsch-Stoer
- Variable Dynamics
- 6 Partial Differential Equations
- Summary

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
 - 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

Parabolic (diffusion equation)

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

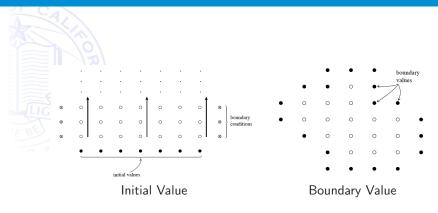
where D is the diffusion coefficient

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

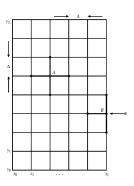


Source - Numerical Recipes.

H. I. Christensen (UCSD) Math for Robotics Oct 2021 38 / 41

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

- 1 Introduction
- ODE Introduction
- Runge-Kutta
- 4 Richardson / Burlirsch-Stoer
- Variable Dynamics
- 6 Partial Differential Equations
- Summary

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

H. I. Christensen (UCSD) Oct 2021 Math for Robotics