CSE276C - Integration of Functions

Henrik I. Christensen

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

October 2024

Outline

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

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

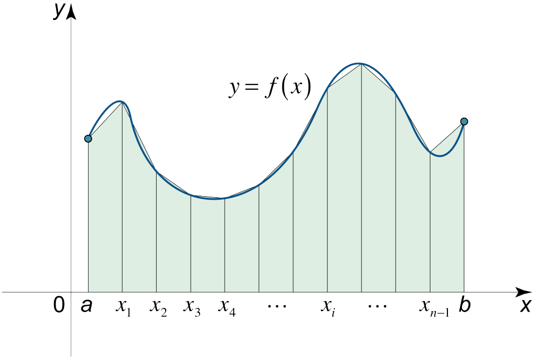
$$\frac{\partial y}{\partial x} = f(x)$$

- 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

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

- 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

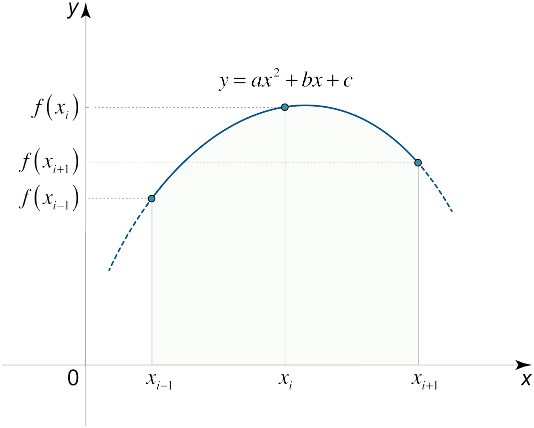
$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f_0 + 4f_1 + f_2) + O(h^5 f^{(4)})$$

- 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
- $(x_0, x_1), (x_1, x_2), \ldots, (x_{N-1}, x_N)$ 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?
- 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?

Handling of improper function

- What is an improper function?
  1. Integrand goes to a finite value but cannot be evaluated at a point, such as
  $$\frac{sinx}{x} \text{ at } x = 0$$
  2. Upper limit is ∞ or lower limit is $-\infty$
  3. Has a singularity at a boundary point, e.g.,
  $$x^{-1/2} \text{ at } x = 0$$
  4. Has a singularity within the interval at a known location
  5. Has a singularity within the interval at an unknown location
- If the value is infinite, e.g.,
$$\int_0^\infty x^{-1} dx \text{ or } \int_{-\infty}^\infty cosxdx$$
it is not improper but impossible

The Euler-Maclaurin Summation Formula

- We can write the basic Simpson's rule as

$$\begin{aligned}\int_a^b f(x)dx &= \tfrac{h}{2}\left[f(a)+2\sum_{k=1}^{N-1} f(a+kh)+f(b)\right] \\ &\quad -\sum_{k=1}^{N/2} \tfrac{h^{2k}B_{2k}}{(2k)!}\left[f^{(2k-1)}(b)-f^{(2k-1)}(a)\right] \\ &\quad -\sum_{k=0}^{N-1} \tfrac{h^{2k+1}B_{2k}}{(2k)!} f^{(2k)}(a+kh+\theta h)\end{aligned}$$

- 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

$$\begin{aligned}B_0 &= 1 \\ B_2 &= \tfrac{1}{6} \\ B_4 &= -\tfrac{1}{30}\end{aligned}$$

- 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

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

- 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

$$x = \frac{1}{2}(b+a) + \frac{1}{2}(b-a)\tanh(t) \; x \in [a,b] \text{ and } t \in [-\infty, \infty]$$

- where

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

- *sech*() converges very rapidly for $t \to \infty$ 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.

$$\begin{aligned} I &= \int_a^b f(x)dx \\ &= \int_a^m f(x)dx + \int_m^b f(x)dx \end{aligned}$$

- 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.

Questions

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

$$\begin{aligned} \tfrac{dy}{dx} &= z(x) \\ \tfrac{dz}{dx} &= r(x) - q(x)z(x) \end{aligned}$$

- where z is a new variable

Small example

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

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

- 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

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

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

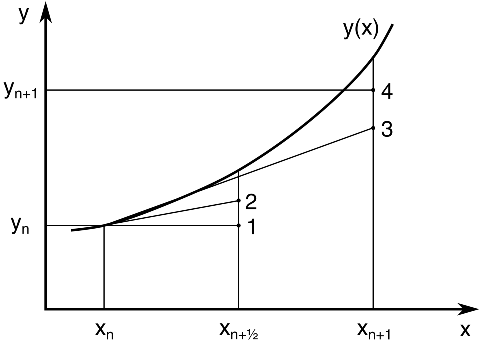
Outline

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?
  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?
- How about

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

  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

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

- 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 accurate 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

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

more recently a suggestion

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

Step size control for Burlirsch-Stoer

- The error estimate can be tabulated as

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

- 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 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 differences 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)

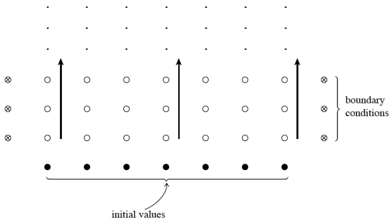
     $$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D \frac{\partial u}{\partial x}\right)$$

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

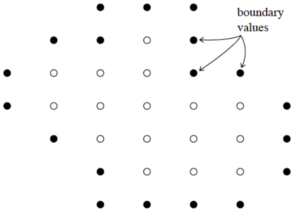
     $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$$

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

Computational Considerations for PDEs



Initial Value

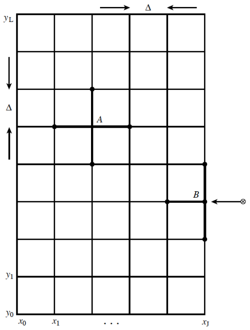


Boundary Value

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