Computational Physics (PHYS6350)

Review and concluding remarks

April 27, 2023

Instructor: Volodymyr Vovchenko (vvovchenko@uh.edu)

Course materials: https://github.com/vlvovch/PHYS6350-ComputationalPhysics

Reminders

The **Final Project** is due today(!)

- So far about 50% have been received
- Late submissions: allowed but with an increasing penalty (try to get it in this week)

You are encouraged to submit feedback through Faculty/Course evaluation
https://uh.edu/measurement-evaluation-center/faculty-course-evaluation/student_info/

Grading

- Homework (40%) *(completed)*
- Final project (20%)
- Mid-term (15%) *(completed)* and Final (25%)
  - Multiple choice, short and long answer questions

Overview: What we covered

Tentative Schedule (Last update 4/24/2023)

1/17	Introduction, Syllabus, Technical Details
1/19	Visualization of Data, Machine Precision
1/24	Function Interpolation
1/26, 1/31	Nonlinear Equations
2/2, 2/7, 2/9	Numerical Calculus
2/14, 2/16	Numerical Differential Equations
2/21, 2/23	Problems in Classical Mechanics
2/28	Molecular Dynamics
3/2, 3/7	Linear Algebra and Matrices
3/9	Midterm Exam
3/14, 3/16	Spring Break – no classes
3/21, 3/23	Partial Differential Equations
3/28, 3/30	Random Numbers and Monte Carlo Methods
4/4, 4/6	Problems in Statistical Physics
4/11, 4/13	Problems in Quantum Mechanics
4/18	Fourier Transform
4/20	Introduction to Machine Learning
4/25	Introduction to Parallel Computing
4/27	Review
5/9	Final Exam

Overview

The course has been focused on concepts used in computational physics, hence

- Jupyter notebooks and python
- (Re)implementation of many standard routines
- Little focus on performance

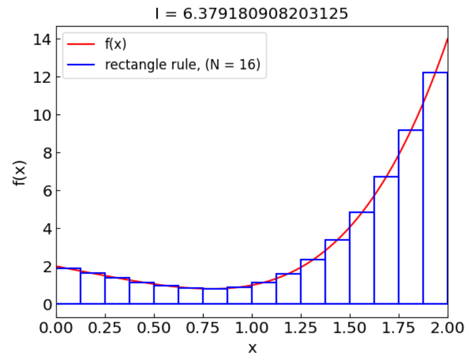
Overview: Numerical integration and differentiation

Numerical integration:

- For quick and dirty calculations
  - Rectangle or trapezoidal rules

- For a single and precise calculation
  - Methods with control over error estimate
  - E.g. Romberg method

- For repeated calculations of the same kind of integral
  - Gaussian quadratures

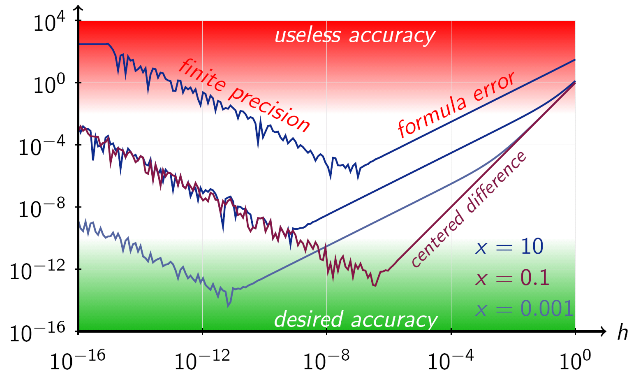


$$\int_a^b f(x)\,dx \approx \sum_k w_k f(x_k)$$

Numerical derivatives:

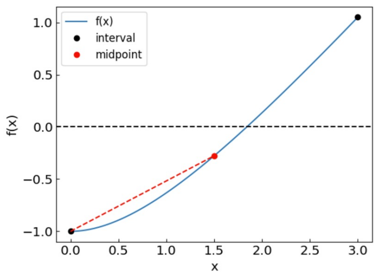
- Use central difference schemes
  - Lower order o.k. in many cases

- Beware of round-off error for small h and/or high-order derivatives



Overview: Non-linear equations

The main methods are



- Bisection method
  - If root is bracketed, guaranteed to converge

- Newton-Raphson method (or secant method if cannot compute f')
  - Converges fast when the root is nearby
  - Good to use for refining the root

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Systems of non-linear equations

- Newton's method
  - Fast convergence but requires Jacobian matrix calculation and inversion

- Broyden's method
  - No need for matrix inversion

Overview: Ordinary differential equations

Lots of physics problems formulated as systems of ODEs

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t).$$

- Good starting point: fourth-order Runge-Kutta (RK4)
  - If not, use adaptive time step
- Time-reversal symmetry and energy conservation
  - Leap-frog method (e.g. Molecular dynamics)
  - Bulirsch-Stoer method (control over error)
- Systems of ODEs
  - Same a single ODE component-by-component
- 2^{nd} or high-order ODE
  - Convert to a system of first-order ODEs
- Stiff/unstable systems
  - Use implicit methods

$$\frac{d^2\mathbf{x}}{dt^2} = \mathbf{f}(\mathbf{x}, d\mathbf{x}/dt, t),$$

$$\Downarrow$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{v},$$
$$\frac{d\mathbf{v}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{v}, t),$$

Overview: Linear algebra

- **Use standard/library routines whenever possible**

- LU-decomposition
  - Solve systems of linear eqs. with the same matrix **A** but varied **v**

$$\mathbf{A}\mathbf{x} = \mathbf{v}$$

- Tri- and band-diagonal systems
  - Appear often, in particular when discretizing PDEs
  - Use specific algorithms since they have linear complexity in

- Eigenvalues and eigenvectors
  - QR algorithm and diagonalization
  - Eigenvalues only is easier than eigenvalues + eigenvectors
  - Use standard/library routines

Overview: Partial differential equations

Lots of physics problems formulated as systems of PDEs

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

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

$$-\frac{\hbar^2}{2m}\frac{\partial^2 \psi}{\partial x^2} = i\hbar\frac{\partial \psi}{\partial t}$$

- Discretization of spatial derivatives
  - Central difference typically o.k.

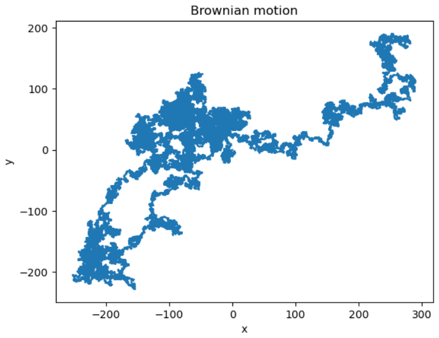
- Time propagation
  - Explicit scheme (FTCS)
    - Often has stability issues
    - More easily generalized to multiple dimensions

  - Crank-Nicolson scheme
    - More stable and conserves the wave amplitude
    - Implicit, but only involves a tridiagonal system solution
    - Multiple dimensions are more challenging

- Further reading
  - Finite element method (irregular spatial shapes)
  - Finite volume method (conservation laws and fluxes)

Overview: Random numbers (Monte Carlo method)

Powerful method where other approaches fail (typically when there is a large amount of degrees of freedom involved)



- The base generator (uniform numbers in [0,1] range) must be good
  - Mersenne Twister is o.k., linear congruential (old compilers) is not
  - Use different seed when doing parallel calculations

- Common applications
  - Numerical integration in multiple dimensions
  - Statistical physics
  - Intrinsically random processes (quantum mechanics)

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

- The answer is always approximate
  - The error typically scales as $1/\sqrt{N}$

$$\delta I = (b-a)\sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}$$