18.330 Problem set 5 (spring 2020)

Submission deadline: 11:59pm on Tuesday, March 10

Exercise 1: Finite differences via interpolation

Consider the simplest forward finite-difference approximation for $f'(x)$:

$$g(h) := \frac{f(x+h) - f(x)}{h}$$

When we calculate this numerically, there are *two* sources of error: truncation error, coming from approximating the exact Taylor expansion with a finite piece of it, *and* floating-point roundoff error.

1. Suppose that we perturb the input, h, by Δh. Calculate (analytically) an approximation to the (absolute) error Δg on the output to first order in Δh; you should find that it grows like h^{-1}.
2. Suppose that the input perturbation size is ϵ_{mach}; the error from [1] is then the **roundoff error**. Find an estimate for the value of h at which the truncation error balances with the roundoff error, and find the size of the error there. Compare this with the plot that we did in class.
3. Consider an interval $[a, b]$ and let m be the midpoint of the interval. Use Lagrange interpolation to find an analytical expression for the unique quadratic function that passes through $(a, f(a))$, $(m, f(m))$ and $(b, f(b))$.
4. Use your result from [3] to derive the centered difference approximation for the derivative $f'(t_k)$ in terms of equally-spaced points t_k separated by a distance h.
5. What approximation does it give for the second derivative $f''(t_k)$?
6. Use [3] to find a backward difference expression for $f'(t_k)$ using information at nodes t_{k-2} and t_{k-1}.
7. Find numerically the rate of convergence of the results from [3] and [4] for equally-spaced points separated by a distance h for the function $\sin(2x)$ at $x = \pi/4$, for values of h between 10^{-6} and 10^{-1}.

Exercise 2: Integration using Simpson's rule

In this problem we will derive the second-order Newton–Cotes quadrature rule, known as **Simpson's rule**, for calculating $\int_a^b f(x)\, dx$.

Suppose you are given an N-point quadrature rule with nodes $(t_k)_{k=0^N}$ and weights $(w_k)_{k=0}^N$ for integrating over the interval $[-1, 1]. That is, the t_k are $N+1$ points with $-1 \leq t_k \leq 1$, and the w_k are given to you such that

$$\int_{-1}^{1} f(x)\,dx \simeq \sum_{k=0}^{N} w_k\, f(t_k)$$

1. Construct a new quadrature rule for integrating over a general interval $[u, v]$. I.e., find t'_k and w'_k such that

$$\int_{u}^{v} f(x)\,dx \simeq \sum_{k=0}^{N} w'_k f(t'_k)$$

Derive the basic second-order Newton–Cotes quadrature rule for $\int_{-1}^{1} f(x)\,dx$, as follows:

2. Use your results from [Exercise 1] to find the degree-2 polynomial p_2 that agrees with f at the three points $x = -1, 0, 1$. (Leave your result in terms of the values $f(-1)$, $f(0)$ and $f(1)$.)
3. Integrate p_2 interval $[-1, 1]$ to approximate $\int f$ in terms of $f(-1$, $f(0)$ and $f(1)$. Express this result as a quadrature rule.
4. Combine your answers to [2] and [3] to write down the basic (not composite) Simpson's rule for integrating f over $[u, v]$.
5. Given an interval $[a, b]$, subdivide it into N equal-width subintervals, apply the basic Simpson's rule to integrate f over each subinterval, and sum the results to obtain the composite Simpson rule for integrating f over $[a, b]$. How many samples of f does this rule require? (Be careful not to overcount).

Exercise 3: Using Newton–Cotes methods

1. Implement the composite 0th (rectangle), 1st (trapezoid), and 2nd-order (Simpson) Newton–Cotes quadrature rules for integrating an arbitrary function over an arbitrary interval with $N+1$ points. Each should be a single function like `rectangle(f, a, b, N)`.

   Note that in the case of Simpson's rule, we are using a *total* of $N+1$ points; how many intervals does this correspond to?

[1] Basic quadrature rules are for nodes distributed over a single interval; composite quadrature rules are obtained by splitting up a large interval into subintervals and using a basic rule on each subinterval.

2. Calculate $\int_{-1}^{1} \exp(2x)\, dx$ using each method. Plot the relative error

$$E(N) := \frac{I_{\text{approx}}(N) - I_{\text{exact}}}{I_{\text{exact}}}$$

as a function of N for N in the range $[10, 10^6]$ (or use a higher or lower upper bound depending on the computing power you have available).

Do these errors correspond with the expectations from the arguments in lectures?

3. Do the same for $\int_{-1}^{2} \exp(-x^2)\, dx$. Use the `erf` function from the `SpecialFunctions.jl` package to calculate the "exact" result. [Hint: Check carefully the help for that function to make sure of the definition used.]

4. We showed that the trapezium rule has error at most $\mathcal{O}(h^2)$. Consider the following integral of a smooth, periodic function:

$$I = \int_0^{2\pi} \exp(\cos(\theta))\, d\theta$$

Plot the error in the trapezium rule in this case. How fast does it decay with N? [This will be important later in the course.]

Note that this integral can be calculated exactly as $2\pi I_0(1)$, where I_0 is a **modified Bessel function**, which can be evaluated at 1 using the `SpecialFunctions.jl` package as `besseli(0, 1)`.

Exercise 4: Euler method for ODEs

1. Implement the Euler method in a function `euler(f, x0, δt, t_final)`, assuming that $t_0 = 0$. Your code should work equally well if you put vectors in, to solve the equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$.

2. Use your code to integrate the differential equation $\dot{x} = 2x$ from $t = 0$ to $t = 5$ with initial condition $x_0 = 0.5$. Plot the exact solution and the numerical solution for values of $\delta t = 0.01, 0.05, 0.1, 0.5$. On a different plot show the relative error as a function of time, compared to the analytical solution.

3. Do the same for $\dot{x} = -2x$ with initial condition $x_0 = 3$.

4. For the above two cases, calculate the error at $t = 5$ when the time interval is split into N pieces for $N between 10 and 1000$. Plot the error as a function of N. What is the rate of convergence as $h \to 0$?

A pendulum satisfies the ODE $\ddot{\theta} + \sin(\theta) = 0$, where θ is the angle with the vertical.

5. Show analytically that the quantity ("energy") $E(\theta, \dot{\theta}) := \frac{1}{2}\dot{\theta}^2 - \cos(\theta)$ is **conserved** along a trajectory, i.e. that $\frac{d}{dt}[E(\theta(t), \dot{\theta}(t)) = 0]$, so that $E(\theta(t), \dot{\theta}(t)) = E(\theta(t_0), \dot{\theta}(t_0))$.
6. Solve this equation using the Euler method for initial conditions $(0, 1)$ to show that the energy is *not* conserved.
7. Draw the phase plane. Explain graphically what is happening in terms of what each step does.
8. Plot E as a function of time for different values of δt. How fast does it grow? Explain this in terms of what happens at each step.