

Homework 3

Mike Sutherland

May 4, 2022

1 Problem 1 - Lagrange Interpolation

First, we construct a function to generate Lagrange polynomials of a general order and for a general set of evaluated points. The results for this function on

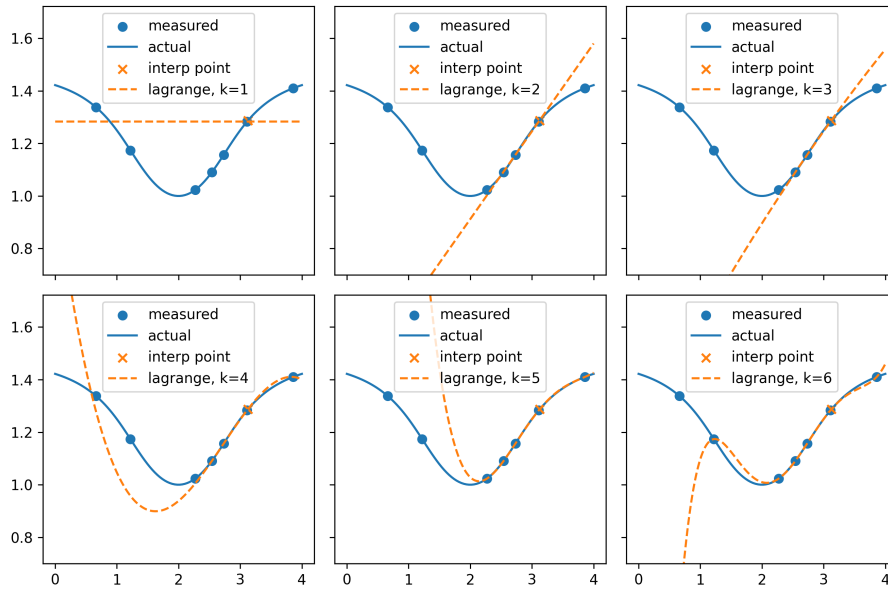


Figure 1: A series of lagrange plots of various order. The true function is the transcendental function $\arctan((x-2)^2)/\pi + 1$. We can see that as we add more points, the approximation becomes more accurate; however, there is also more overshoot.

some test data is shown in fig. 1. We can see the effect of increasing order on the accuracy of the approximation; with increased order, the approximation could become more accurate, but it also is subject to more extreme overshoot. We can also see how the order of the constructed polynomial affects the direction

of extrapolation: the sign of extrapolated points changes with odd/even orders of lagrange polynomials.

We turn to the example data of Problem 1. We can see from fig. 2 the

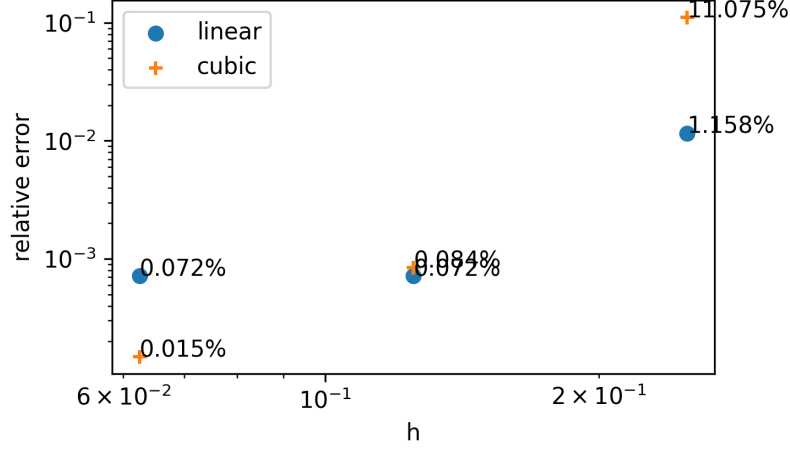


Figure 2: Lagrange interpolation of polynomials and their respective error. For small values of h , the error is smaller for the higher-order polynomial. However, for larger values of h , the error is smaller for the lower-order polynomial.

effect of the order of the polynomial on the error of the solution, as it relates to the step size of the sampled data. We can see how lower-order polynomials obtain worse interpolation for small step sizes, while higher-order polynomials obtain worse interpolation for large step sizes. As the order of the polynomial increases, the fluctuations in areas between data points become larger, due to the presence of higher-order terms.

The optimal order of a Lagrange polynomial varies depending on the data being sampled. If the structure of the data is unknown before performing the interpolation, lower-order polynomials (usually, cubic) are used, because they often obtain a better fit than linear interpolation, while limiting the overshoot. For functions that are polynomials, a Lagrange polynomial of the same order as the original function will produce an exact fit of the data.

Because of the overshoot issue with Lagrange polynomials, they are often not be suitable for noisy data. In those cases, it may be wise to either filter the data or use a different technique, like Gaussian Process Regression with a rational quadratic kernel.

We also notice that Lagrange polynomial interpolation is not suitable for extrapolation, since the polynomial is unbounded.

1.1 Problem 2

[to do]

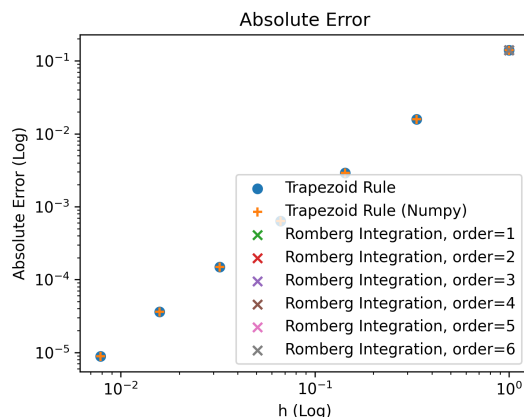


Figure 3: Log-Log plot of the absolute error vs the step size, as compared with the analytic solution of a known integral. We can see that as the step size decreases, the error decreases in turn, until the step size is so small that round-off error becomes an issue.

2 Problem 3 - Integration

To solve Problem 2, we created a python program that computes the integral of a given function over an interval using the Trapezoid rule.

The trapezoid rule computes the integral of a function by summing the area of trapezoids evaluated at points along a function. If a smaller step size is used, more linear segments of the original function are evaluated, which reduces the error. The relationship between error and step size is order $\mathcal{O}(h)$ because of this linear relationship. This is effect is visible in fig. 3.

We evaluate the runtime in fig. 4. We can see that the runtime increases linearly with a decrease in step size.

Romberg integration has a runtime comparable to naive trapezoid integration; its benefit is that as the algorithm runs, the evaluated integral is visible to the algorithm, which makes it useful when the programmer needs to specify expected error ahead of time, without knowing ahead of time what step size results in that expected error.

3 Problem 4 - Differentiation

We created a program which differentiates a given function using a 2nd order and 4th order central difference scheme.

Figure 5 shows the relationship between the step size and error of 2nd and 4th order numerical derivatives. The 4th order scheme is more accurate (its error is $\mathcal{O}(h^4)$) than the 2nd order scheme (its error is $\mathcal{O}(h^2)$). However, because of the additional multiplication of h in the denominator of the 4th order scheme,

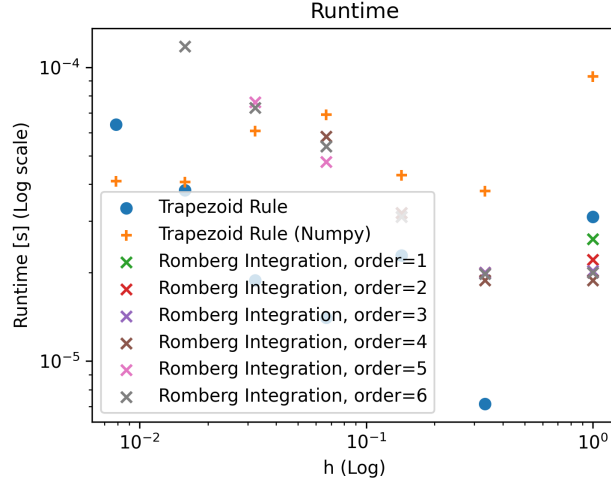


Figure 4: Log-Log Runtime of a naive trapezoid rule, `np.trapz`, and a Romberg integration, vs. h . We can see that as h decreases, runtime increases. The relationship is $\mathcal{O}(h)$. The naive python implementations have a higher overhead than their numpy variants.

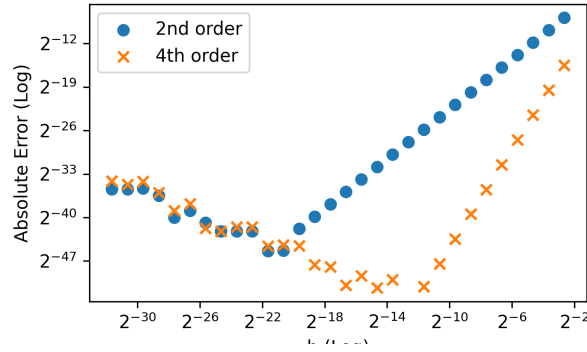


Figure 5: Differentiation of a function using a 2nd order and 4th order central difference scheme. The 2nd order central difference scheme is less accurate than the 4th order central difference scheme. The limit of numerical precision is visible for both schemes.

it runs into the limits of numerical precision faster than the 2nd order scheme. Despite this, it is ultimately more accurate.

4 Problem 6 - Differentiation

Explain how you would construct a 6th-order accurate central finite difference approximation for the first derivative on a uniform grid. We begin with a general expression for the Taylor series:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{1}{2}f''(x_i)h^2 \dots \quad (1)$$

In general, this yields a series of derivative terms. For an n th order term, we begin the Taylor series at $f^n(x_i)$. We can then rearrange these terms to obtain a system of equations:

$$f(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{h} f'(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i))}{h^2} f''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i))}{h^3} f'''(x_i) \dots \quad (2)$$

From the coefficients of these terms, we can construct a matrix and multiply to solve for the derivatives. We then evaluate the derivatives up to order $k + 1$, plug them into the original Taylor series (of order $k + 1$), and rearrange to solve for the derivative of the original function. The equations in this example were 2nd-order accurate, but the accuracy can be increased by taking higher-order terms from the original Taylor series.

Because a general formula for the Taylor series is possible, any order can be constructed.