PHYS 512 Lecture Notes

Thomas J Bolder

September 4, 2025

1 Lecture 1

testing

$$f'(x) \approx \frac{f(x+dx) - f(x)}{dx} \tag{1}$$

Computational cost of this: there are two function evaluations. Want to get as much done as you can with as few function evaluations as possible.

Can do better than this though. We can rewrite $f(x+dx)$ as:

$$f(x+dx) = f(x) + dxf'(x) + \frac{(dx)^2}{2}f''(x) + ... \tag{2}$$

Consider $f(x-dx)$, and take $f'(x) \approx \frac{f(x+dx)-f(x-dx)}{2dx}$ and use the above Taylor series expansion to sub in.

So $f(x+dx) - f(x-dx) = 2dxf'(x) + 2\frac{(dx^3)f'''(x)}{6}$. So inputting it into $f'(x) \approx \frac{f(x+dx)-f(x)}{dx}$, get the Taylor series error term as $\frac{(dx)^2 f'''(x)}{6}$.

Want to think about roundoff error vs Taylor Series error. Check the code he wrote on it. Called 'deriv_class.py'

Want to get numerical derivatives super accurate because the errors can push up FAR. This type of thing can become a real pain in the ass later.

2 Lecture 2, 01.09

Lectures 2 and 3 are on polynomial interpolation.

Say we get a bunch of x points for a function and gives you a bunch of y output values also. We want to know what the best guess for that function at values inbetween x values (called interpolation) or outside of the x value interval (called extrapolation)

Most basic thing we can do is just take nearest neighbour. Pretty sucky because usually you have a smooth function you want to approximate.

So instead take a weighted average between the two nearest neighbours, which visually looks like connecting the dots. This is called a linear interpolation (obviously). By the way, smooth means well defined by a Taylor Series.

Next thing is to try quadratics. More interesting. For a quadratic, need three points to define it. Don't want to pick the closest data points to the

point you want to interpolate, because it kind of sucks. Get discontinuous function for two different points you want to interpolate.

For a cubic XXXXXXX

For Higher Order: what is a polynomial we can write down that is zero at a certain number of points? Want it to be zero at some of these points? Say we want it to be zero at $x_1, x_2, ...$ etc. Then we know that $(x - x_1)(x - x_2)...$ is exactly that. But now say we want it to be of value e.g. 1 at some x_0. We evaluate the value of the polynomial at x_0 and divide by its value to make it work. Pretty straightforward.

Can then stack a bunch of these to go through all of your points that you want at certain values. Given a set of x's and y's, the output thing is a polynomial that goes through the set that you have and takes care of one point but doesn't mess up the other ones.

They don't look great: they wiggle a good amount at high order. The jiggles get larger: multiplying many together. So going to higher order will mess thing up: small change in one side of data (due to noise, etc) can cause a big change on the other side.

Cubic was best: didn't need to do poly fit for each interpolated value. Can fit every interval and store coefficients. Also was continuous.

The problem is the cubics together were not continuous as a whole (double check?). Splines is when the interpolation function forces the function and the first n derivatives to be continuous. Most common is second derivative to be continuous (so first is guaranteed smooth (why?)).

Usually just called splines, even though we mean cubic splines.

There's a whole algorithm logic here, but need to add outline: XXXXXX XXXXXX

Never going to write this, exists everywhere in libraries.

Also have to be careful because if you get a funky datapoint, it will introduce ripples and wiggles to keep things smooth, which can really mess the fit up.

If function is smooth we can do quite well with a function that assumes it's smooth (such as a spline). Smooth functions interpolating with polynomials works great.

Can interpolate with rational functions also. That's ratio of two polynomials by the way.

Sidenote: what is a pole? to do with Taylor Series and the complex domain. Sometimes very sensible functions can be badly behaved for taylor series.

$$f(x) = \frac{P(x)}{1 + Q(x)} \tag{3}$$

Polynomials always blow up when you go outside the region you are working in. Rational functions don't necessarily do that: since a ratio of two polynomials. So better behaved further out, more flexibility here. For example to approximate a gaussian, you need something that dies down when you go out.

But how to determine P and Q?

$$\begin{aligned} \frac{P_0 + P_1 x + P_2 x^2 ...}{1 + Q_1 x + Q_2 x^2 ...} &= y(x), \\ P_0 + P_1 x + P_2 x^2 ... &= y + y Q_1 x + y Q_2 x^2 ..., \\ P_0 + P_1 x + P_2 x^2 ... + y Q_1 x + y Q_2 x^2 ... &= y \end{aligned} \tag{4}$$

So can solve linearly (write down how).