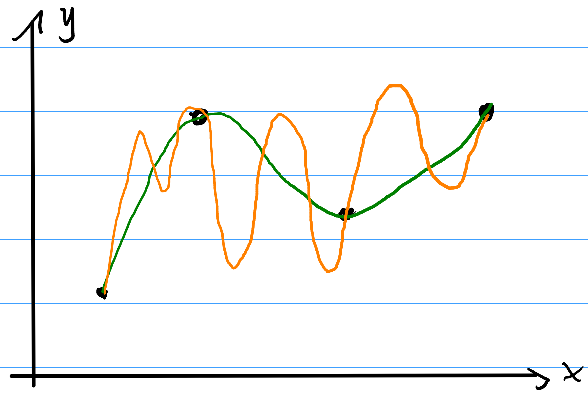
(10) Function Spaces

What is interpolation?

Finding a function through given x/y points.



given

x	y
x_1	y_1
x_2	y_2
x_m	y_m

But there are lots of possible functions through those points. Be more precise.

To make interpolation a well-specified problem, we require that the resulting function (the "interpolant") is a linear combination of a given basis of functions.

Given: $\varphi_1(x), \varphi_2(x), \dots, \varphi_m(x)$

#functions matches #points

Want: $f(x) = \alpha_1 \varphi_1(x) + \dots + \alpha_m \varphi_m(x)$

so that $f(x_i) = y_i$

So how do we find this interpolant?

By solving a linear system:

$$\begin{aligned}\alpha_1 \varphi_1(x_1) + \cdots + \alpha_m \varphi_m(x_1) &= y_1\\ &\vdots\\ \alpha_1 \varphi_1(x_m) + \cdots + \alpha_m \varphi_m(x_m) &= y_m\end{aligned} \quad \rightsquigarrow \quad \underbrace{\begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_m(x_1) \\ \vdots & & \\ \varphi_1(x_m) & \cdots & \varphi_m(x_m) \end{pmatrix}}_{V} \vec{\alpha} = \vec{y}$$

V is called the generalized Vandermonde matrix.

The general idea is

$$V \underbrace{\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}}_{\text{basis coefficients}} = \underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}}_{\text{values at interp. points}} \quad \rightsquigarrow \quad V\vec{\alpha} = \vec{y}$$

Can you give an example?

Suppose we've fixed m points (also called 'interpolation nodes'):

$$x_1 \quad \dots \quad x_m$$

We then fix a basis. In this case, we will use the monomials:

$$\underbrace{x^0}_{1}, x^1, x^2, x^3, x^4, x^5, \dots, x^{m-1}$$

number of basis functions
must match number of nodes

So we are looking for the coefficients α in

$$\alpha_0 1 + \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_{m-1} x^{m-1}$$

Set up the Vandermonde matrix:

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^4 \\ 1 & x_2 & & & \vdots \\ 1 & x_3 & & & \vdots \\ 1 & x_4 & & & \\ 1 & x_5 & & \cdots & x_5^4 \end{bmatrix}$$

The term "Vandermonde matrix" first appeared for monomials, so this is actually *the* Vandermonde matrix. We've earlier 'generalized' it by allowing different sets of functions.

All this should look relatively familiar: It's exactly what we've been doing in data fitting--except the matrices are square.

Can you give an example with actual numbers?

x_i	y_i
0	7
2	4
3	4

$$V = \begin{array}{c} \begin{array}{ccc} 1 & x & x^2 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{pmatrix} \end{array}$$

$$V\vec{\alpha} = \begin{pmatrix} 7 \\ 4 \\ 4 \end{pmatrix}$$

Solve

$$\vec{\alpha} = \begin{pmatrix} 7 \\ -2.5 \\ 0.5 \end{pmatrix}$$

$$f(x) = \alpha_0 1 + \alpha_1 x + \alpha_2 x^2$$

Demo: Interpolation with generalized Vandermonde matrices

Where is interpolation helpful?

There are lots of things you can do if you have a function that you cannot do if you only have point values.

Examples:

- Evaluate anywhere, not just at the given data points
- Find rate of change depending on x
- More generally, use calculus

If I interpolate a function I already know, will the interpolant be exactly that function?

Unless you're very lucky, no, that will not typically the case.

But under some assumptions, we can say when the interpolant will be "close", and in what sense.

Demo: Interpolation Error (Part 1)

Needed assumptions:

- Function f being interpolated is "smooth" (has many derivatives)
- Length h of the interpolation interval is "sufficiently small"

In this case:

h: length of the interpolation interval

$$|f(x)-\hat{f}(x)| \le C \cdot h^{n+1} \quad \text{for all x in the interval}$$

"true" function

interpolant

n: highest polynomial degree in interpolant

C: an unknown constant (depends on f and n, but not h)

The main purpose of estimates like this is to predict dependency of the error on the length of the interval h.

So what types of predictions can I make using the interpolation error estimate?

Suppose I have an interpolation error E for some interval length h
for interpolation with a quadratic function.

What would happen if I used h/2 instead of h?

$$\mathrm{Error}(h) \approx C \cdot h^3$$

$$\mathrm{Error}\left(\frac{h}{2}\right) \approx C \cdot \left(\frac{h}{2}\right)^3 = \frac{1}{8} \cdot C \cdot h^3 \approx \frac{1}{8} \cdot \mathrm{Error}(h)$$

That's a concrete (testable) prediction!

Demo: Interpolation Error, part 2

Generally, if a method has an error estimate of the form

$$\mathrm{Error} \leq C \cdot h^p$$

it is called p-th order convergent.

So interpolation with polynomials of degree n is (n+1)-th order convergent.

Shorter intervals (smaller h) seem to decrease the error. Why is that?

The function will vary less on a shorter interval.
That means it is easier to "capture" the behavior of the function using
a line/parabola/cubic/...

Interpolation allows several choices. What are good/bad choices?

Have:

- Choice of basis
- Choice of nodes

Let's look at choice of nodes first, then at choice of basis.

What is a "good" set of nodes for polynomial interpolation?

Demo: Choice of interpolation nodes

Observation: Best if nodes cluster towards interval ends

"Best" set of interpolation nodes on [-1,1]: "Chebyshev nodes"

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right) \qquad k=1\ldots n$$

What are some choices of interpolation basis?

(1) Monomial basis $1, x, x^2, \ldots, x^{m-1}$

Already discussed above

Demo: Monomial Interpolation

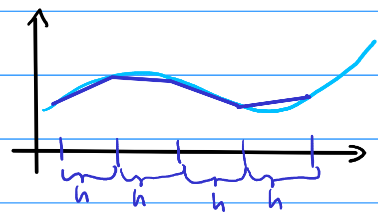
Observation: Works fine up to m=6 or so. Beyond that,
the conditioning of the Vandermonde matrix starts to hurt.
Using a better set of nodes helps a little, but not enough
for arbitrarily large n.

So here's an idea: Use multiple lower-degree pieces.

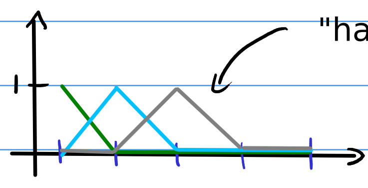
What are some choices of interpolation basis? (cont'd)

(2) Piecewise polynomials

For example: Piecewise linear, continuous



What would a basis for this interpolation scheme look like?



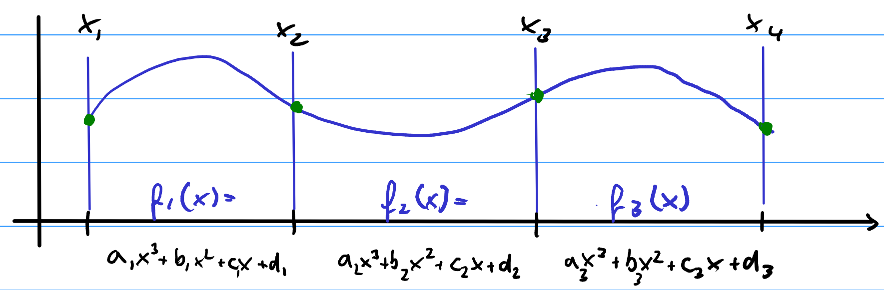
"hat functions", one per interpolation node

Problem: Need very small h (lots of intervals) to be accurate.

Idea: Use higher-degree polynomial on each piece.

This results in a so-called "spline".

With polynomials of degree 3, you would get a "cubic spline".



given data points

Need 4 coefficients per piece. -> More coefficients than data points.

Need to impose extra conditions to get a unique answer. → HW

What are some choices of interpolation basis? (cont'd)

(3) Sines and cosines ("Fourier basis")

What's the basis? $\cancel{\sin(0x)} \to 0$, $\cos(0x)$, $\sin(1x)$, $\cos(1x)$, $\sin(2x)$, $\cos(2x)$, ...

What to use as points? equispaced on [0,2π]... but:

$$2\pi\frac{0}{5},\ 2\pi\frac{1}{5},\ 2\pi\frac{2}{5},\ 2\pi\frac{3}{5},\ 2\pi\frac{4}{5},\ 2\pi\frac{5}{5}$$

no – $\cos(n\,2\pi) = \cos(0)$
$\sin(n\,2\pi) = \sin(0)$

So leave out the last point.

Write down a Vandermonde matrix for up to n=2:

$$\begin{matrix}
\cos(0\cdot 0) & \sin(1\cdot 0) & \cos(1\cdot 0) & \sin(2\cdot 0) & \cos(2\cdot 0)\\
\cos(0\,\frac{2\pi}{5}) & \sin(1\cdot\frac{2\pi}{5}) & \cos(1\cdot\frac{2\pi}{5}) & \sin(2\cdot\frac{2\pi}{5}) & \cos(2\cdot\frac{2\pi}{5})\\
\cos(0\,\frac{2\pi\cdot 2}{5}) & \sin(1\cdot\frac{2\pi\cdot 2}{5}) & \cos(1\,\frac{2\pi\cdot 2}{5}) & \sin(2\cdot\ ) & \cos(2\cdot\ )\\
\cos(0\,\frac{2\pi 3}{5}) & \sin(1\cdot\ ) & \cos(1\cdot\ ) & \sin(2\cdot\ ) & \cos(2\cdot\ )\\
\cos(0\,\frac{2\pi\cdot 4}{5}) & \sin(1\cdot\ ) & \cos(1\cdot\ ) & \sin(2\cdot\ ) & \cos(2\cdot\ )
\end{matrix}$$

Demo: Fourier Interpolation

Observations:

- works best for periodic functions
- generalized Vandermonde has orthogonal columns

An important motivation for the Fourier basis is that its coefficients correspond to frequencies.

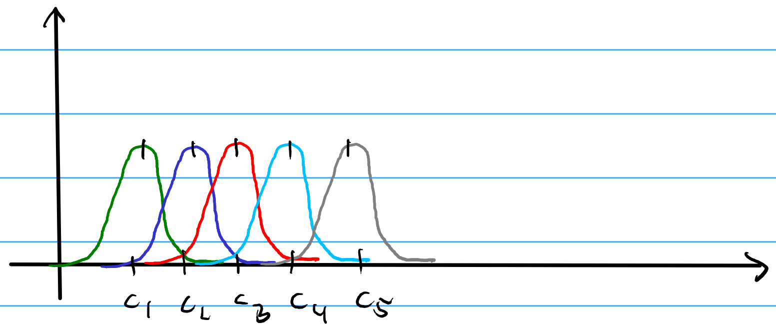
Demo: Audio synthesis with sines

Demo: Audio experiments

What are some choices of interpolation basis? (cont'd)

(4) Radial basis functions

Idea: Use multiple copies of the same function, centered at a number of locations on the real line.



$$\varphi_1(x) = \varphi(x - c_1)$$
$$\varphi_2(x) = \varphi(x - c_2)$$
$$\vdots$$
$$\varphi_5(x) = \varphi(x - c_5)$$

The "mother function" φ obeys

- $\varphi(x) = \varphi(|x|)$

φ only depends on "radius" |x|
(=distance from origin)

→ "radial" basis function

- $\varphi(x) \to 0$ as $|x| \to \infty$

Demo: Interpolation with Radial Basis Functions

⊕ Easy to construct

Easily generalizes to multiple dimensions

⊖ Can be tricky to stabilize

Quality of result depends on choice of radius, centers

Interpolant decays to zero away from nodes

So how would I use calculus on an interpolant?

$f(x_i) = \tilde{f}(x_i) \quad i=1\dots n$

Have: interpolant $\hat{f}(x) = \alpha_1 \varphi_1(x) + \dots + \alpha_n \varphi_n(x)$

Want: derivative $\tilde{f}'(x) = \alpha_1 \varphi_1'(x) + \dots + \alpha_n \varphi_n'(x)$

Easy because interpolation basis (φ_i) is known!

Have: function values at nodes (x_i) $f(x_i)$

Want: values of derivative at nodes $f'(x_i)$ ← hard to get

$\tilde{f}'(x_i)$ ← easy to get!

How?

1. Compute coefficients $\vec{\alpha} = V^{-1}\vec{f}$ $\quad \vec{f} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$

2. Build generalized Vandermonde with derivatives of basis

$$V' = \begin{pmatrix} \varphi_1'(x_1) & \cdots & \varphi_n'(x_1) \\ \vdots & & \vdots \\ \varphi_1'(x_n) & \cdots & \varphi_n'(x_n) \end{pmatrix}$$

3. Compute $V'\vec{\alpha} = \begin{pmatrix} \alpha_1 \varphi_1'(x_1) + \dots + \alpha_n \varphi_n'(x_1) \\ \vdots \\ \end{pmatrix} = \underbrace{\begin{pmatrix} \tilde{f}'(x_1) \\ \vdots \\ \tilde{f}'(x_n) \end{pmatrix}}_{\vec{\tilde{f}}'}$

Now all in one step: $\vec{\tilde{f}}' = \underbrace{V'V^{-1}}\vec{f}$

matrix to apply a derivative (!)

"Differentiation matrix"

So how would I use calculus on an interpolant? (cont'd)

If you want coefficients of the derivative, use

$$V^{-1} V' V^{-1} \vec{f}.$$

Demo: Taking derivatives with Vandermonde matrices

Give a matrix that takes two derivatives.

$$V' V^{-1} V' V^{-1}$$

What is the observed behavior of the error when taking a derivative?

$$|f'(x) - \tilde{f}'(x)| \leq C \cdot h^n \longleftarrow$$ one less power than interpolation (where n is the highest poly degree)

What do the entries of the differentiation matrix mean?

do not care

$$D = V' V^{-1} = \begin{pmatrix} \boxed{////////} \\ -\frac{1}{2h} \quad 0 \quad \frac{1}{2h} \\ \boxed{////////} \end{pmatrix} \qquad V' V^{-1} \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix} = \begin{pmatrix} \boxed{////////////} \\ -\frac{1}{2h} f(x_0) + \frac{1}{2h} f(x_2) \\ \boxed{////////////} \end{pmatrix}$$

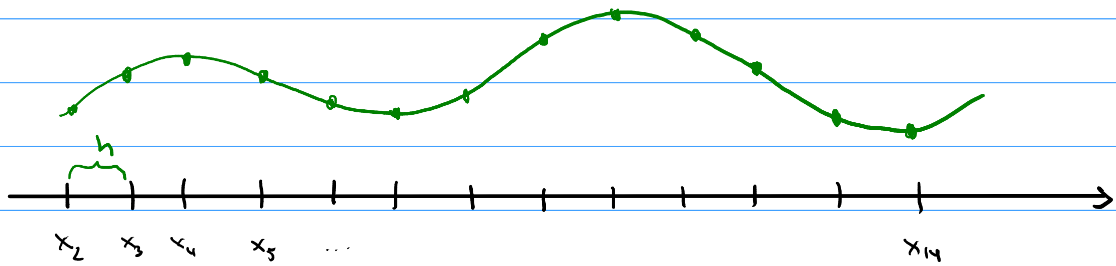
$$f'(x_1) \approx \frac{1}{2h} \cdot [f(x_2) - f(x_0)]$$

Demo: Finite differences, part 1

This looks a lot like the def. of the derivative: $\frac{f(x+h) - f(x)}{h} \longrightarrow f'(x) \qquad (h \to 0)$

This is called a (second-order) "centered difference".

Can we simplify the proccess for equispaced x-coordinates?

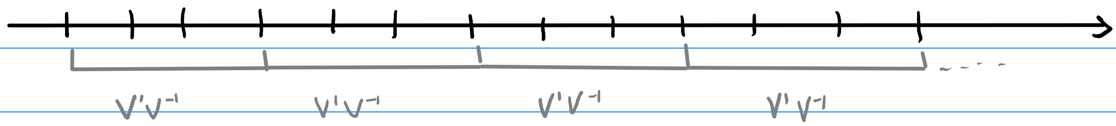


The process so far:

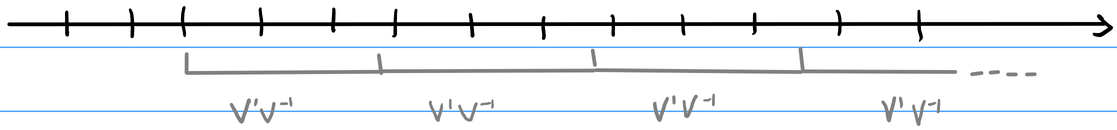
(1) Form group of n data points

(2) Compute interpolant

(3) Apply derivative



But why not like this?



The result could be different, but it will not be less accurate.

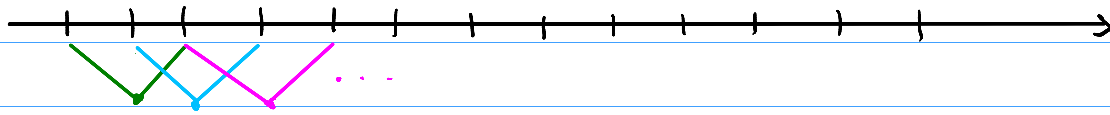
Lesson: For equispaced data, the 'grouping' for applying the differentiation matrix does not matter.

Reason: Differentiation is "translation-invariant".

$$\frac{d}{dx}\left[f(x-c)\right] = f'(x-c)$$

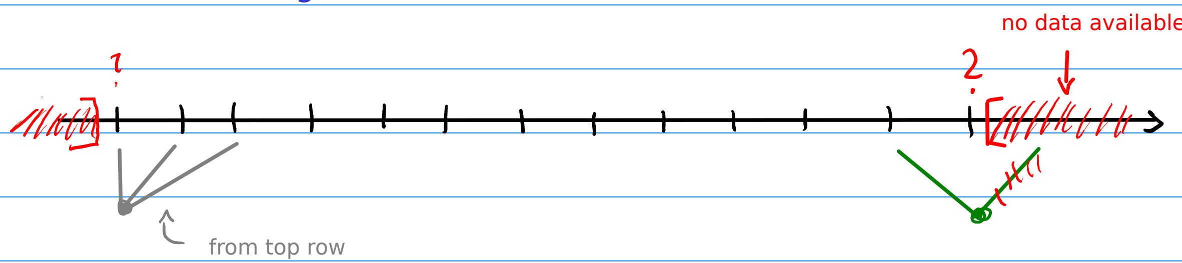
How would we use translation invariance with centered differences?

$$f'(x_i) \approx \frac{1}{2h}\left(f(x_{i+1}) - f(x_{i-1})\right)$$



This is why finite difference formulas are also often called finite difference 'stencils'. Like a real-word stencil, you move it to where you need it---and then apply it there.

What do we do at the edges?



Centered difference needs data from left *and* right -> unusable

Idea: Use a different finite difference formula

E.g top row of differentiation matrix contains a 'right-looking' finite difference formula.

How accurate are finite difference formulas?

$|f' - \tilde{f}'| \le C \cdot h^n \leftarrow$ as above! (It's the same thing after all.)

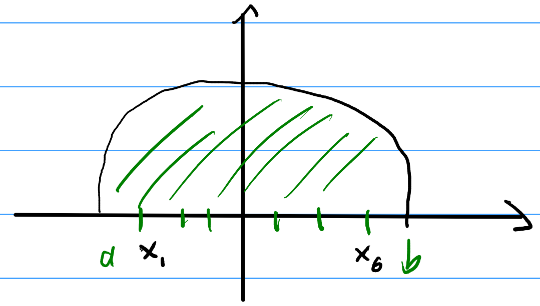
Can anything else go wrong when we numerically compute derivatives?

Demo: Finite differences vs. Noise

Lesson: Numerical differentiation amplifies noise, worse on finer grids

Can we use a similar process to compute (approximate) integrals?

The process of computing approximate integrals is called "quadrature".



E.g.: find area under the (half-)circle

Same idea as derivatives: Interpolate, then integrate.

Have: interpolant $\tilde{f}(x) = \alpha_1 \varphi_1(x) + \cdots + \alpha_n \varphi_n(x)$

$f(x_i) = \tilde{f}(x_i) \quad (i=1 \ldots n)$

x_i are called "quadrature nodes"

Want: integral $\int_a^b \tilde{f}(x)\,dx = \int_a^b \alpha_1 \varphi_1(x) + \cdots + \alpha_n \varphi_n(x)\,dx$

$$= \alpha_1 \int_a^b \varphi_1(x)\,dx + \cdots + \alpha_n \int_a^b \varphi_n(x)\,dx$$

Idea: $d_i := \int_a^b \varphi_i(x)\,dx$ can be computed ahead of time

So: $\int_a^b \tilde{f}(x) = \alpha_1 d_1 + \cdots + \alpha_n d_n = \vec{\alpha} \cdot \vec{d}$

Now: $\vec{\alpha} = V^{-1}\vec{f}$ $\qquad \vec{f} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$

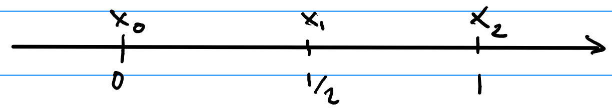
Altogether: $\int_a^b \tilde{f}(x) = (V^{-1}\vec{f}) \cdot \vec{d} = \vec{f}^T V^{-T} \vec{d} \qquad \vec{w} = V^{-T}\vec{d}$

$\vec{w}$'s entries are called the quadrature weights

So, once I know my nodes and my weights, what does quadrature look like?

$$\int_a^b \tilde{f}(x) = w_1 \cdot f(x_1) + \cdots + w_n \cdot f(x_n)$$

Can you do a full example?



$f(x_0) = 2 \qquad f(x_1) = 0 \qquad f(x_2) = 3$

1. Find coefficients $\qquad \vec{\alpha} = V^{-1}\begin{pmatrix}2\\0\\3\end{pmatrix}$

2. Compute integrals

$$\int_0^1 1 \, dx = 1$$

$$\int_0^1 x \, dx = \frac{1}{2}$$

$$\int_0^1 x^2 \, dx = \left[\frac{1}{3}x^3\right]_0^1 = \frac{1}{3}$$

3. Combine it all together

$$\underbrace{\begin{pmatrix}1 & \frac{1}{2} & \frac{1}{3}\end{pmatrix} V^{-1}}_{\text{"weights"}} \begin{pmatrix}2\\0\\3\end{pmatrix}$$

Demo: Computing the weights in Simpson's rule

$$\int_0^1 \tilde{f}(x)\,dx = \begin{pmatrix}.166\\.666\\.166\end{pmatrix} \cdot \begin{pmatrix}f(0)\\f(\frac{1}{2})\\f(1)\end{pmatrix}$$

It turns out that this has someone's name attached to it. It's called "Simpson's Rule".

What does Simpson's Rule look like on [0, 1/2]?

$$\frac{1}{2}\begin{pmatrix} .166 \\ .666 \\ .166 \end{pmatrix} \cdot \begin{pmatrix} f(0) \\ f(\frac{1}{4}) \\ f(\frac{1}{2}) \end{pmatrix}$$

What does Simpson's Rule look like on [5, 6]?

$$\begin{pmatrix} .166 \\ .666 \\ .166 \end{pmatrix} \cdot \begin{pmatrix} f(5) \\ f(5.5) \\ f(6) \end{pmatrix}$$

How accurate is Simpson's rule?

Demo: Accuracy of Simpson's rule

Quadrature: $\left|\int_a^b f(x)\,dx - \int_a^b \tilde{f}(x)\,dx\right| \le C \cdot h^{n+2}$

(Due to a happy accident, odd n produce an error that is even smaller than h^(n+2).)

Interpolation: $|f(x) - \tilde{f}(x)| \le C \cdot h^{n+1}$

Differentiation: $|f'(x) - \tilde{f}'(x)| \le C \cdot h^{n}$

Lesson: More derivatives => Worse accuracy