CME 108/MATH 114
Introduction to Scientific Computing
Summer 2020

Introduction to Numerical Differentiation

Motivation/ guiding questions

1. Given a function f defined by (d_i) a discrete data table, how can I approximate the value of f' at a point?

2. Given a function f or an oracle which evaluates f (I know nothing of its derivate and I may not know how to differentiate the function if its given by a formula), how can I approximate values of f' by linear combinations of values of f?

Common in applications dealing with complex functions e.g. molecular dynamics, neural nets. (To be fair, ML frameworks like TensorFlow also provide oracles for evaluating gradients analytically.

There are two main approaches to this problem:

1. Given a data table, construct the interpolating polynomial (or spline) and evaluate the polynomial's derivative analytically (the **exact** derivative of the polynomial serves to **approximate** the derivative of the function defined by the data);

2. Use Taylor's theorem to derive a formula which approximates values of the derivative using linear combinations of function values.

Approach 1: Derivatives via interpolating polynomials (discussed earlier, in Lagrange interpolation notes).

Given a data table $(x_0, f_0), \ldots, (x_n, f_n)$ (here $f_j = f(x_j)$), recall that the Lagrange form of the interpolating polynomial is given by

$$p(x) = \sum_{j=0}^{n} f_j L_{n,j}(x).$$

So this approach dictates we take

$$f'(x_i) \approx p'(x_i) = \sum_{j=0}^{n} f_j L'_{n,j}(x_i) = \sum_{j=0}^{n} d_{ij} f_j,$$

where $d_{ij} = L'_{n,j}(x_i)$ In vector form, we have

$$\begin{bmatrix} f'(x_0) \\ \vdots \\ f'(x_n) \end{bmatrix} \approx \begin{bmatrix} p'(x_0) \\ \vdots \\ p'(x_n) \end{bmatrix} = \begin{bmatrix} \sum_j d_{0,j} f_j \\ \vdots \\ \sum_j d_{n,j} f_j \end{bmatrix} = D\vec{f}, \tag{1}$$

with $\vec{f} = [f_0 \cdots f_n]^T$.

It turns out that (try to show this!)

$$L'_{n,j}(x_i) = \begin{cases} \frac{w_j/w_i}{x_i - x_j}, i \neq j \\ \sum_{i \neq j} L'_n, j(x_i), i = j, \end{cases} \tag{2}$$

and this formula can be used to compute the entries of the differentiation matrix D.

Let's take a closer look, when n is small!

Example 0.0.1. *Finite difference approximation of f' using Lagrange interpolation, when $n = 2$.*

Let $n = 2$, consider the three equally spaced nodes x_0, x_1, and x_2, and set $h = x_{j+1} - x_j$.

In this case

$$p(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f_1 \tag{3}$$

$$+ \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f_2 \tag{4}$$

$$= \frac{(x - x_1)(x - x_2)}{2h^2} f_0 + \frac{(x - x_0)(x - x_2)}{-h^2} f_1 + \frac{(x - x_0)(x - x_1)}{2h^2} f_2 \tag{5}$$

and so

$$p'(x) = \frac{1}{2h^2}[(2x - x_1 - x_2)] f_0 - 2(2x - x_0 - x_2) f_1 + (2x - x_0 - x_1) f_2], \tag{6}$$

Then

$$p'(x_0) = \frac{1}{2h^2}[(2x_0 - x_1 - x_2)]f_0 - 2(2x - x_0 - x_2)f_1 + (2x_0 - x_0 - x_1)f_2], \tag{7}$$
$$\tag{8}$$

(recall e.g.

$$x_0 - x_1 = -h, \tag{9}$$
$$x_0 - x_2 = -2h) \tag{10}$$

$$= \frac{1}{2h^2}[-3hf_0 - 2(-2h)f_1 - hf_2] \tag{11}$$
$$= \frac{-3f_0 + 4f_1 - f_2}{2h}, \tag{12}$$

second order one-sided difference!
and

$$p'(x_1) = \frac{1}{2h^2}[(2x_1 - x_1 - x_2)f_0 - 2(2x_1 - x_0 - x_2f_1 + (2x_1 - x_0 - x_1)f_2] \tag{13}$$
$$= \frac{1}{2h^2}[-hf_0 - 2(0)f_1 + hf_2] \tag{14}$$
$$= \frac{f_2 - f_0}{2h} \tag{15}$$

second-order central difference!

In this (small) case (recall $n = 2!$) we get recognizable formulas which we will re-derive below using Taylor's theorem.

Approach2: Use Taylor's theorem.
In this approach we use Taylor's theorem to obtain a local polynomial approximation of f, and we combine approximations at different points "in the right way" to cancel error terms.

This technique is perhaps best illustrated by example.

Example 0.0.2. *If f has two continuous derivatives on (a, b), then Taylor's theorem guarantees that for every $x_0 \in (a, b)$ there exists $x_0 < \xi < x_0 + h$ such that*

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(\xi).$$

Isolating $f'(x_0)$ defines an approximation formula:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi). \tag{16}$$

*Forward differences (we take $h > 0$), so this steps forward to approximate derivative. $O(h)$ error term, so we say formula is **first order**.*

Similarly, $f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2} f''(\xi_2)$, with $x_0 - h < \xi < x_0$, so

$$f'(x_0) = \frac{(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi_2) \tag{17}$$

Backward difference (step backwards for approximation). $O(h)$ error term so first order.

In practice, when the error is proportional to h, we expect that halving the step size approximately cuts the error in half.

What if we want the error to decrease faster? We'll use a Taylor table to derive higher order formulas! (One more appearance of linear systems!).

Basic idea: Set $f_{j+k} = f(x_j + kh)$

Suppose we want to use f_{-1}, f_0, and f_1 to obtain a second order formula approximating $f'(x_j)$. Then we must find coefficients α_{-1}, α_0, and α_1 such that

$$f'(x_j) = \alpha_{-1} f_{j-1} + \alpha_0 f_j + \alpha_{-1} f_{j+1} + O(h^2). \tag{18}$$

(we call the set of points used to approximate $f'(x_j)$ the "stencil". In this case we use a three-point stencil $(x_j - h, x_j, x_j + h)$ which is centered about x_j.

You may think of the stencil as the footprint of the formula-where you need to step in order to approximate the derivative).

Remark 0.0.3. In general, to define finite difference formulas, we first need to choose a stencil. Larger stencils may lead to higher order formulas but they may become expensive to evaluate and more importantly less useful because they can only be applied on points sufficiently far from the grid boundaries.

Example 0.0.4. *(somewhat ridiculous, for illustration purposes only!). Forward difference using 7 point stencil on 10 points*

Can't use it to approximate $f'(x_4), f'(x_5), \ldots, f'(x_9)$

Back to our symmetrical 3-point stencil example.

Here we've **chosen** to use three points, and a stencil centered about the point at which we want to evaluate the derivative.

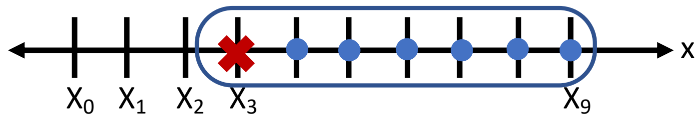

Figure 1: Geometric intuition of the MVT

Using Taylor's theorem,

$$f_{j+k} = f(x_j + jh) = f(x_j) + khf'(x_j) + \frac{(kh)^2}{2} f''(x_j) + O(h^3) \tag{19}$$

so

$$\begin{aligned}
f'(x_j) &= \alpha_{-1} + \alpha_0 f_j + \alpha_1 f_{j+1} + O(h^2) && (20)\\
&= \alpha_{-1}(f(x_j) - hf'(x_j) + \frac{h^2}{2} f''(x_j) + O(h^3)) && (21)\\
&+ \alpha_0(f(x_j)) && (22)\\
&+ \alpha_1(f(x_j) + hf'(x_j) + \frac{h^2}{2} f''(x_j) + O(h^2)) && (23)\\
&= (\alpha_{-1} + \alpha_0 + \alpha_1) f(x_j) && (24)\\
&+ (-h\alpha_{-1} + \frac{h^2}{2}\alpha_1) f'(x_j) && (25)\\
&+ (\frac{h^2}{2}\alpha_{-1} + \frac{h^2}{2}\alpha_1) f''(x_j) && (26)\\
&+ (\alpha_{-1} + \alpha_1) O(h^3) && (27)
\end{aligned}$$

Now we compare coefficients of each term on the LHS and RHS: $f(x_j)$ does not appear on LHS, so $\alpha_{-1} + \alpha_0 + \alpha_1 = 0$ coefficient of $f'(x_j)$ on LHS is 1, so $-h\alpha_{-1} + h\alpha_1 = 1$, $f''(x_j)$ does not appear on LHS, so $\frac{h^2}{2}\alpha_{-1} + \frac{h^2}{2}\alpha = 0$.

As a single matrix equation, our system reads

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1/2 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} = \frac{1}{h} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{28}$$

We can input A and e_2 into MATLAB and find $x = A\backslash e_2$. Then $\begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} = \frac{1}{h}x$ (here x has numerical values and h is a symbol we use to write our formula).

However, in this case we don't need MATLAB: the third equation implies $\alpha_{-1} + \alpha_1 = 0$, so $\alpha_{-1} = -\alpha_1$ and substituting into the first gives $\alpha_0 = 0$.

The second equation then implies $\alpha_1 = \frac{1}{2h}$, so that

$$f'(x_j) = \alpha_{-1} f_{j-1} + \alpha_0 f_j + \alpha_1 f_j + 1 + O(h^2) \tag{29}$$

$$= \frac{f_{j+1} - f_{j-1}}{2h} + O(h^2) \tag{30}$$

the second-order central differential derived using Lagrange polynomial earlier!(see p.3)

The system follows from Taylor table:

	f_j	f'_j	f''_j	f'''
f'_j	0	1	0	0
$\alpha_{-1} f_{j-1}$	α_{-1}	$h\alpha_{-1}$	$\frac{h^2}{2}\alpha_{-1}$	$-\frac{h^3}{6}\alpha_{-1}$
$\alpha_0 f_j$	α_0	0	0	0
$\alpha_1 f_{j+1}$	α_1	$h\alpha_1$	$\frac{h^2}{2}\alpha_1$	$\frac{-h^3}{6}\alpha_{-1}$

Each column in the table defines an equation:

1. First column $0 = \alpha_{-1} + \alpha_0 + \alpha_1$
2. $1 = -h\alpha_{-1} + h\alpha_1$
3. $0 = \frac{h^2}{2}\alpha_{-1} + \frac{h^2}{2}\alpha_1$

In general, when we seek n coefficients, we use the first n columns to build our linear system, and further columns provide information on the error term.

In this case (as before) $\alpha_{-1} = -\frac{1}{2h}, \alpha_1 = \frac{1}{2h}$, so the last column implies the error term is

$$\frac{-h^3}{6}\alpha_{-1} f'''(\xi_1) + \frac{h^3}{6}\alpha_1 f'''(\xi_2) = \frac{h^2}{6}\left(\frac{f'''(\xi_1) + f'''(\xi_2)}{2}\right) = \frac{h^2}{6} f'''(\xi) \tag{31}$$

(assuming f''' is continuous and using the IVT)

We can use this method to find finite difference formulas for higher order derivatives:

Example 0.0.5.

	f_j	f'_j	f''_j	f'''	$f^{(4)}$
f''_j	0	0	1	0	0
$\alpha_{-1}f_{j-1}$	α_{-1}	$h\alpha_{-1}$	$\frac{h^2}{2}\alpha_{-1}$	$-\frac{h^3}{6}\alpha_{-1}$	$\frac{h^4}{24}\alpha_{-1}$
$\alpha_0 f_j$	α_0	0	0	0	0
$\alpha_1 f_{j+1}$	α_1	$h\alpha_1$	$\frac{h^2}{2}\alpha_1$	$\frac{-h^3}{6}\alpha_{-1}$	$\frac{h^4}{24}\alpha_{-1}$

The system here is **similar** to the last one, but its not the same!

$$\alpha_{-1} + \alpha_0 + \alpha_1 = 0 \tag{32}$$

$$-h\alpha_{-1} + h\alpha_1 = 0 \tag{33}$$

$$\frac{h^2}{2}\alpha_{-1} + \frac{h^2}{2}\alpha_1 = 1 \tag{34}$$

or in matrix form

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1/2 & 0 & 1/2 \end{bmatrix} \vec{\alpha} = \frac{1}{h^2} e_3 \tag{35}$$

One based indexing for jth standard basis vector.

In this case we find that $\alpha_{-1} = \alpha_1 = \frac{1}{h^2}, \alpha_0 = -\frac{2}{h^2}$

The α_j's here are proportional to $\frac{1}{h^2}$, so we look at two more columns to discuss the error. The f'''_j column implies $\left(\frac{-h^3}{6}\alpha_{-1} + \frac{h^3}{6}\alpha_1\right) f'''(x_j) = 0$ so the error term is given by the $f^{(4)}$ column:

$$\frac{h^4}{24}\alpha_{-1} f^{(4)}(\xi_1) + \frac{h^4}{24}\alpha_{-1} f^{(4)}(\xi_2)\alpha_1 = \frac{h^2}{12}\left(\frac{f^{(4)}(\xi_1) + f^{(4)}(\xi_2)}{2}\right) \tag{36}$$

$$= \frac{h^2}{2} f^{(4)}(\xi), \tag{37}$$

for some $\alpha_1 < \alpha < \alpha_2$, assuming f is "smooth enough".

This gives the **second order central** difference for $f''(x_j)$:

$$f''(x_j) = \frac{f_{j-1} + f_{j+1}}{h^2} + O(h^2)$$

In general, when constructing finite difference formulas for $f^{(m)}$ using an n-point stencil, we end up with an $n \times n$ linear system of the form $A\vec{\alpha} = \frac{1}{h^{(m)}} e_{(m+1)}$ which can be solved with the aid of a computer.