CHEMICAL ENGINEERING 2E04

Chapter 4 – Differentiation and Integration

Module 4A: Numerical Differentiation

Dr. Jake Nease

Kieran McKenzie

Steven Karolat

Cynthia Pham

Chemical Engineering

McMaster University

Updated November 12, 2019

Contents

Supplementary Material

Suggested Readings

Gilat. V. Subramaniam: *Numerical Methods for Engineers and Scientists.* Wiley. Third Edition (2014)

- Finite Difference Differentiation
  - Chapter 8 → *8.2, 8.3, 8.4*
- Differentiation Using Lagrange Polynomials
  - Chapter 8 → *8.5*
- Partial Differentiation
  - Chapter 8 → *8.10*
- Differentiation Using Curve Fitting
  - Chapter 8 → *8.6*

Overview/Applications

You are likely used to taking the derivatives of equations by hand (analytically). What if we don't know the equation of our function, but still want to find the derivative, or the derivative is *too complicated* to take analytically?

Numerical differentiation has the potential to compute an *efficient approximation* of a derivative, whether we have a known function or not!

This module will cover a variety of methods to approximate the derivative using the following methods:

1) Finite Difference Differentiation
2) Differentiation using Lagrange Polynomials

Primary Learning Outcomes

- Discuss *finite difference differentiation* and the *error* associated with each of the methods.
- Derive the general representations for finite difference formulas.
- Recognize the specific applications of finite difference formulas.
- Discuss differentiation using *Lagrange polynomials* and differentiation using *curve fitting*.
- Expand our finite difference formulas to tackle *partial differentiation*.
- Discuss *computational error, truncation error, and total error* for numerical methods of differentiation.

Applications of Numerical Differentiation

- Determining fuel consumption (MPG) from gas tank levels.
- Production rates of units or mass produced (CO_2 emissions, for example).
- Determining reaction rate coefficients, k.
- Process control (derivative term) – infrequently used, mathematically we look at why.
- Optimization (analyzing where the derivative equals 0).
- Rate-based mass and heat transfer problems (solving for steady states).

Finite Difference Differentiation

The most common method of numerical differentiation uses *finite difference formulas (FDFs)* – these are a series of formulas *derived using a Taylor Series*.

Remember, the n^{th} order Taylor Series $T_n(x)$ of a function $f(x)$ centered on a specific point a is:

$$f(x) \approx T_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \frac{f'''(a)(x-a)^3}{3!} + \cdots + \frac{f^{(n)}(a)(x-a)^n}{n!}$$

Instead of approximating the value of a function at a point as we did in the NL equations section, *we isolate for* $f'(a)$ to get the approximated derivative at the centering point a. It is important to remember that in this case, we *know the values of* $f(x-a)$ *and* $f(a)$. There are several ways to do this, but they are all related. All methods except for the two-point methods are derived from the case where the data is *evenly spaced*.

Two-Point Forward Difference for First Derivative

If we want to take the derivative of a function at a point a using forward difference, it means we use the function information *ahead* of a to help determine the slope. Consider two points: a and $x = (a+h)$, where h is some small step size we can select. We may write out the Taylor Series approximating $f(x) \equiv f(a+h)$ as:

$$f(a+h) = f(a) + f'(a)([a+h]-[a]) + \frac{f''(a)([a+h]-[a])^2}{2!} + \cdots + \frac{f^{(n)}(a)([a+h]-[a])^n}{n!}$$

After truncating the series at *first order* and simplifying, we also get a term that is representative of error:

$$f(a+h) = f(a) + f'(a)(h) + \frac{f''(\xi)(h)^2}{2!}$$

Remember, ξ *is some value between* a *and* $(a+h)$ *that represents maximum truncation error as seen in Module 2!*
To determine the derivative of $f(a)$, we simply rearrange for $f'(a)$:

$$f'(a) = \frac{f(a+h)-f(a)}{h} - \underbrace{\frac{f''(\xi)(h)}{2!}}_{Truncation\ Error}$$

Looking at the error term, its magnitude is proportional to $([a+h]-[a])$. Because of this, *the error of the derivative approximation is said to be "on the order of* $([a+h]-[a]) = (h)$*" written as,* $\mathcal{O}(h)$. Typically, we don't compute this term, but showing that error is relative to h helps compare methods and know how the accuracy of the method changes if h changes. We also assume that we start off with a small enough separation between points, such that $h < 1$; this means the error term is *smaller* when using a truncation term with a higher order h; *i.e.* $\mathcal{O}(h^2) < \mathcal{O}(h)$. So, the formal way to approximate $f'(a)$ using a forward divided difference is:

$$\boxed{f'(a) = \frac{f(a+h)-f(a)}{(a+h)-a} - \mathcal{O}\big((a+h)-a\big)} \quad \text{AKA} \quad \boxed{f'(a) = \frac{f(a+h)-f(a)}{h} - \mathcal{O}(h)}$$

Two-Point Backward Difference for First Derivative

Similar forward difference, this time using a point behind a. Using a and $x = a-h$ in our Taylor series:

$$f(a-h) = f(a) - f'(a)(a-[a-h]) + \frac{f''(x_i)(a-[a-h])^2}{2!} - \cdots - \frac{f^{(n)}(x_i)(a-[a-h])^n}{n!}$$

Notice the rearrangement to have the sign changes highlighted in blue above.

After truncating the series at two terms and rearranging:

$$f'(a) = \frac{f(a) - f(a-h)}{a - [a-h]} + \underbrace{\frac{f''(\xi)(a - [a-h])^2}{2!}}_{Truncation\ Error}$$

$$\boxed{f'(a) = \frac{f(a) - f(a-h)}{h} + \mathcal{O}(h)}$$

Workshop: Two-point difference formulas

You've just installed a new ball valve in your plant. You've obtained some flow rate measurements while at different opening positions (see the plot below). A valve is considered fully open when the stem/orifice is $\mathbf{90°}$ from the perpendicular and closed when at $\mathbf{0°}$. It is important to know what the expected flow change per change in valve opening, $\frac{d(Flow\ Rate)}{d(Valve\ Opening)}$ to help control the process. At a position of $\mathbf{60°}$, plot tangent lines representing the two-point forward and backward difference derivatives.

Why? Many process control algorithms operate on the principle of $+$ or $-$ rather than "set it to x" - it is important to know what kind of change to expect while increasing or decreasing the opening when starting at any given position.

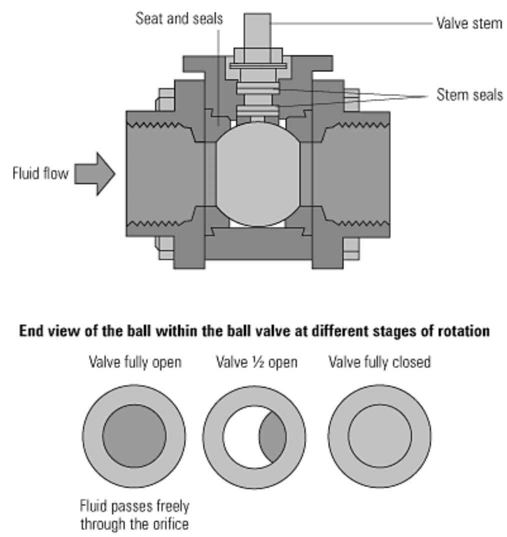
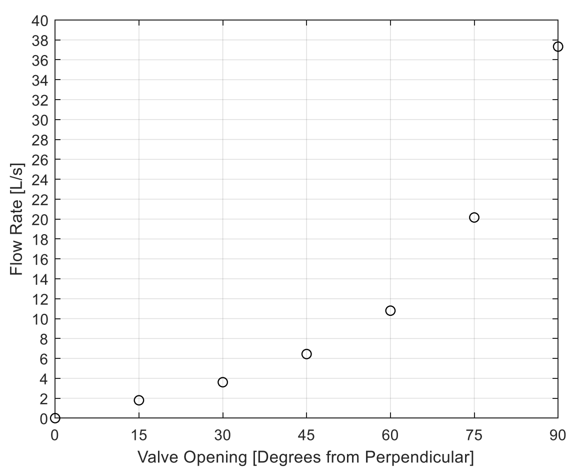


Figure 1. Ball Valve[1]



Three-Point Forward and Backward Difference for First Derivative

What if we truncate the Taylor Series after the second order term? More terms included in a Taylor Series means closer representation of the underlying function – so, by adding more terms, we can *further reduce our error*.

Deriving the three-point difference formulas involves combining two Taylor Series – this is where it becomes a pain to derive if the distance between points is not uniform, therefore, for this and all further finite difference formulas we will be focusing on formulas that handle *evenly spaced data*, with a constant spacing of h.

Using three points in either direction reduces error by including an additional Taylor Series term in the derivation. This time forward difference we will use points a, $a+h$, and $a+2h$ and backward difference uses, $a-2h$, $a-h$, and a.

Setting up the Taylor series for *forward difference*:

(1)	Setting up the Taylor series to find $f(a+h)$	$f(a+h) = f(a) + f'(a)([a+h]-[a]) + \frac{f''(a)([a+h]-[a])^2}{2!} + \frac{f'''(\xi_1)([a+h]-[a])^3}{3!}$ $\boxed{f(a+h) = f(a) + f'(a)\boldsymbol{h} + \frac{f''(a)\boldsymbol{h}^2}{2!} + \frac{f'''(\xi_1)\boldsymbol{h}^3}{3!}}$
(2)	Setting up the Taylor series to find $f(a+2h)$	$f(a+2h) = f(a) + f'(a)([a+2h]-[a]) + \frac{f''(a)([a+2h]-[a])^2}{2!} + \frac{f'''(\xi_1)([a+2h]-[a])^3}{3!}$ $\boxed{f(a+2h) = f(a) + f'(a)\boldsymbol{2h} + \frac{f''(a)(\boldsymbol{2h})^2}{2!} + \frac{f'''(\xi_2)(\boldsymbol{2h})^3}{3!}}$

We used two Taylor Series, each for different points, to give us *two distinct equations* and which have *the same two unknowns*, $f'(a)$ and $f''(a)$ – this results in a *system of linear equations* with $DOF = 0$. Since we are interested in finding $f'(a)$, we can eliminate $f''(a)$! By multiplying Equation (1) by 4, and subtracting Equation (2) we cancel out the $f''(a)$ term. After rearranging, this results in:

$$\boxed{f'(a) = \frac{-3f(a) + 4f(a+h) - f(a+2h)}{2h} + O(h^2)}$$

Workshop: Deriving the three-point backwards difference formula

The *three-point backward difference* formula is derived almost identically, with the only change being the starting Taylor Series. Let's derive it!

Two-Point Centered Difference for First Derivative

To derive the *center difference* formula, we *combine* the two-point forward and the two-point backward difference series. We'll also hold on to the $f'''(a)$ terms for the time-being.

(1)	Two-point Forward Difference:	$f(a+h) = f(a) + f'(a)h + \dfrac{f''(a)h^2}{2!} + \dfrac{f'''(\xi_1)h^3}{3!}$
(2)	Two-point Backward Difference:	$f(a-h) = f(a) - f'(a)h + \dfrac{f''(a)h^2}{2!} - \dfrac{f'''(\xi_2)h^3}{3!}$

We combine the two formulas by *subtracting* equation (2) from (1). Notice, *the* f'' *term disappears:*

$$f(a+h) - f(a-h) = 2f'(a)h + \frac{f'''(\xi_1)h^3}{3!} + \frac{f'''(\xi_2)h^3}{3!}$$

This can be rearranged to the following equation with an improved error term, now $\boldsymbol{\mathcal{O}(h^2)}$:

$$\boxed{f'(a) = \frac{f(a+h) - f(a-h)}{2h} + \mathcal{O}(h^2)}$$

Because a longer Taylor Series used in this derivation, the end formula inherently has a smaller error term, this time on the *order of* h^2 while only needing two points ($f(a+h)$ and $f(a-h)$) to calculate. *It has the same level of accuracy as the three-point forward/backward difference with less effort*.

In the example plot to the right, the centered difference almost perfectly matches the true slope of the function – an impressive feat considering we only used two points to find it!

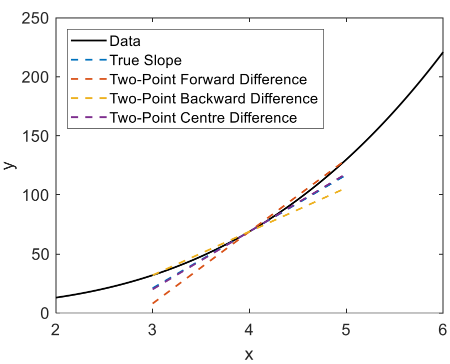


Figure 2: Two-point derivative approximations for a function and the true slope

Workshop: Three-point Difference Formulas and Two-point Center Difference

Return to the valve problem and compute/plot the three-point forward, three-point backward and two-point center difference approximations of the slope when at an opening of $60°$. These should be much closer to each other since we have improved error term.

Summary of Finite Difference Formulas

The above section covered the process of deriving formulas to calculate the first derivative - these methods can be continued and expanded upon to develop formulas to calculate the second derivative and beyond!

Finite Difference Formulas

Where h is the (equal) distance between evenly spaced data or a constant selected value:

Method	**Formula**	**Truncation Error**
First Derivative		
Two-Point Forward	$f'(a) = \dfrac{f(a+h) - f(a)}{h}$	$\mathcal{O}(h)$
Three-Point Forward	$f'(a) = \dfrac{-3f(a) + 4f(a+h) - f(a+2h)}{2h}$	$\mathcal{O}(h^2)$
Two-Point Backward	$f'(a) = \dfrac{f(a) - f(a-h)}{h}$	$\mathcal{O}(h)$
Three-point Backward	$f'(a) = \dfrac{f(a-2h) - 4f(a-h) + 3f(a)}{2h}$	$\mathcal{O}(h^2)$
Two-point Centered	$f'(a) = \dfrac{f(a+h) - f(a-h)}{2h}$	$\mathcal{O}(h^2)$
Four-point Centered	$f'(a) = \dfrac{f(a-2h) - 8f(a-h) + 8f(a+h) - f(a+2h)}{12h}$	$\mathcal{O}(h^4)$
Second Derivative		
Three-point Forward	$f''(a) = \dfrac{f(a) - 2f(a+h) + f(a+2h)}{h^2}$	$\mathcal{O}(h)$
Four-point Forward	$f''(a) = \dfrac{2f(a) - 5f(a+h) + 4f(a+2h) - f(a+3h)}{h^2}$	$\mathcal{O}(h^2)$
Three-point Backward	$f''(a) = \dfrac{f(a-2h) - 2f(a-h) + f(a)}{h^2}$	$\mathcal{O}(h)$
Four-point Backward	$f''(a) = \dfrac{-f(a-3h) + 4f(a-2h) - 5f(a-h) + 2f(a)}{h^2}$	$\mathcal{O}(h^2)$
Three-point Central	$f''(a) = \dfrac{f(a-h) - 2f(a) + f(a+h)}{h^2}$	$\mathcal{O}(h^2)$
Five-point Central	$f''(a) = \dfrac{-f(a-2h) + 16f(a-h) - 30f(a) + 16f(a+h) - f(a+2h)}{12h^2}$	$\mathcal{O}(h^4)$

Remarks: Finite Difference Formulas

Benefits

- There are many ways of calculating $f'(x)$: forwards, backwards, and central difference formulas, each considering a different set of neighboring points to construct the Taylor Series.
- Thanks to the Taylor Series derivations, we have a proportional term to define error – this comes in handy when comparing FDFs, helping decide which is most appropriate for the given situation.

Drawbacks

- We must *know* f(x) for any point at which we wish to know the derivative (the only exception being two-point center difference), and equal spacing between points is required for reliable answers
  - This *limits us -* if we have data at $x = 5$ and 7, we can't determine any (reliable) derivatives at $x = 6.15$).
- FDFs can be *sensitive to noisy data*
  - FDFs relies on a limited amount of neighboring data points – if this data is error filled, the resulting derivative approximation will also be.
    - Even if one of three points used has excessive noise, the entire calculation can be thrown off.
  - Depending on the data set, it may be wise to *selectively choose* the given data points used to calculate the derivative and *"skip" over the erroneous data* – bear in mind you still need to maintain evenly spaced data!
  - *Note:* This isn't a trait solely of FDFs, other methods can fall victim too.

Which Method Should You Use?

- At the starting point, only a forward difference formula can be used.
- At the end point, only a backward difference formula can be used.
- Therefore, a combination of FDFs are required to determine the derivative across the entire set of data points.
  - Generally, center difference is used for all interior points – for the same amount of points used, center difference results in a smaller error ($\mathcal{O}(h^2) < \mathcal{O}(h)$), giving us a better approximation.

The three plots below give an example of each type (all of which use two-point methods), compared to the "true" tangent determined analytically:

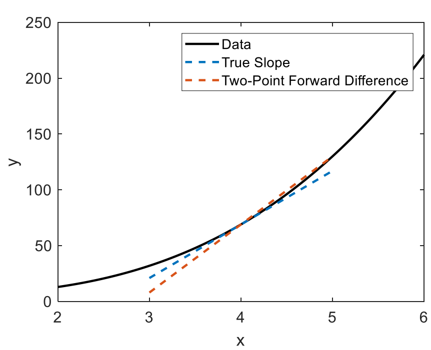


Figure 3: Two-point forward difference with an error of $[\mathcal{O}(h)]$ *plotted along with the true slope.*

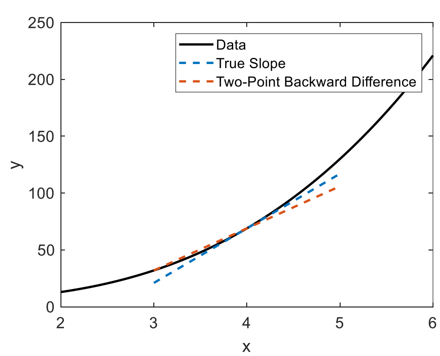


Figure 4: Two-point backward difference with an error of $[\mathcal{O}(h)]$ *plotted along with the true slope.*

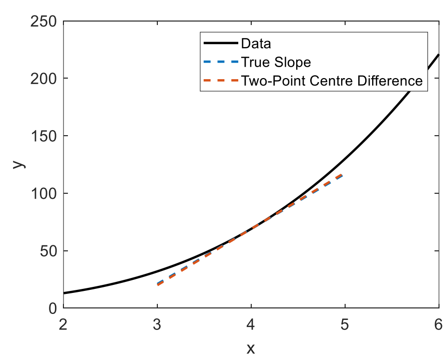


Figure 5: Center difference with an error of $[\mathcal{O}(h^2)]$ *plotted along with the true slope.*

Differentiation using Lagrange Polynomials

An alternative method to determining the derivative at a point uses Lagrange Polynomials. Using Lagrange polynomials has its benefits, arguably the biggest is that we *can now interpolate the approximate slope at any point* provided we have three or more points in the neighborhood.

Starting with the general formula for a Lagrange polynomial which passes through points $(x_1, f(x_1))$, $(x_2, f(x_2))$and $(x_3, f(x_3))$, we can approximate the function value $f(x)$ at some *new* value of x as:

$$f(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} f(x_1) + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} f(x_2) + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} f(x_3)$$

If we *take the derivative* of the formula above with respect to x, we get the following:

Differentiation using Lagrange Polynomials

The general formula for the derivative of a *second-order* Lagrange Polynomial is:

$$f'(x) = \frac{2x - x_2 - x_3}{(x_1 - x_2)(x_1 - x_3)} f(x_1) + \frac{2x - x_1 - x_3}{(x_2 - x_1)(x_2 - x_3)} f(x_2) + \frac{2x - x_1 - x_2}{(x_3 - x_1)(x_3 - x_2)} f(x_3)$$

Using the Lagrange derivative formula, we can *calculate the derivative of any point between the bounds of* x_1 *and* x_3. Also, the three-points used to create the polynomial *do not need to be evenly spaced*. This makes the Lagrange method incredibly robust, allowing it to handle uneven data, yet still return a reasonable estimate of slope within the bounds.

Let's say we want to use our Lagrange Polynomial to find the slope at x_i, one of the points used in deriving it, and the points are evenly spaced – *the Lagrange formula reduces (after simplifying) to one of the three-point FDF formulas, depending on the point being found!*

We can also take the Lagrange polynomial equation for $f'(x)$ and continue taking the derivative to find $f''(x)$ and so on, to derive more alternative ways to calculate derivatives rather than FDFs. The major downfall to using Lagrange is we have *no way to estimate the magnitude of error*. There's no Lagrange version of truncation error. Moreover, Lagrange only takes a few points into account when calculating, making it vulnerable to noisy data.

Workshop: Differentiation using Lagrange Polynomials

Returning to the valve example, suppose the valve position is 70°. What is the expected slope at the given setting?

Partial Differentiation

So far, we've only covered differentiation dealing with one dependent and one independent variable. Now we will look at handling functions in multiple dimensions (*i.e.* $f(x, y)$) with partial derivatives. Luckily, very little needs to change! If we want to take the derivative of $f(x, y)$ with respect to x $\left(\frac{\partial f}{\partial x}\right)$ all we need to do in our formulas is *hold* y *constant* - this is shown in the example below, which uses two-point forward difference:

$$\frac{\partial f}{\partial x} = \frac{f(x + h_x, y_i) - f(x_i, y_i)}{h_x}$$

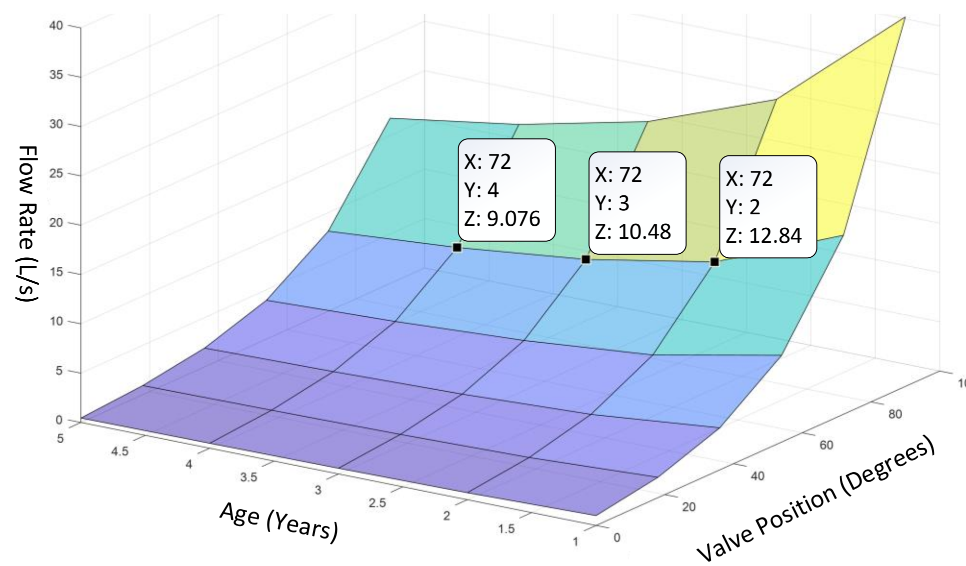
On the other hand, if we want to find $\frac{\partial f}{\partial y}$, change y and keep x as a constant:

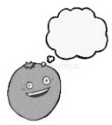
$$\frac{\partial f}{\partial y} = \frac{f\left(x_i, y + h_y\right) - f(x_i, y_i)}{h_y}$$

If it doesn't change across your desired derivative, then don't change it - easy as that!

Workshop: Partial differentiation

Let's make our valve problem multi-dimensional! We can add another axis that shows the valve's age. Over time, the valve will foul (gunk-up) creating a higher pressure drop/restricting flow. Using the three-point forward difference formula, find the derivative with respect to age at 2 years old, while holding the valve open at 72°.





Imagine This:

Imagine that you wanted to create your own version of the `fsolve` function in `MATLAB`. Can you use this information for computing the JACOBIAN of a function? Does that then relate back to solving systems of nonlinear equations if I don't know the TRUE derivatives of those functions at a given point?

Numerical Differentiation Error

For our FDFs, we can see error is proportional to h, so by reducing h (the interval between points) we reduce error associated with our approximations – but only so far…

- We are really trying to minimize the *total error* – a combination of two types of errors: the *inherent formula error* and the *error from using computers*.
  - Shown on the plot below, as h gets smaller, so does the associated truncation error of the Taylor series approximation.
  - However, there is a point when the computational error *takes over* as the predominant error source.
    - As we reach higher precision, we ask the computer to keep track of numbers to the n^{th} digit which can cause errors. The computer may begin to make guesses!
    - This is the same problem we ran into when dealing with matrices with high condition numbers – things get so relatively close to each other that the computer has a hard time telling them apart.
- As a result, we are better off to use a formula with a good (but not miniscule) truncation error to avoid confusing the computer.
  - As seen in the plot below, there is an *optimum point* where total error is minimized – this is the h we want to use to give the most reliable results.

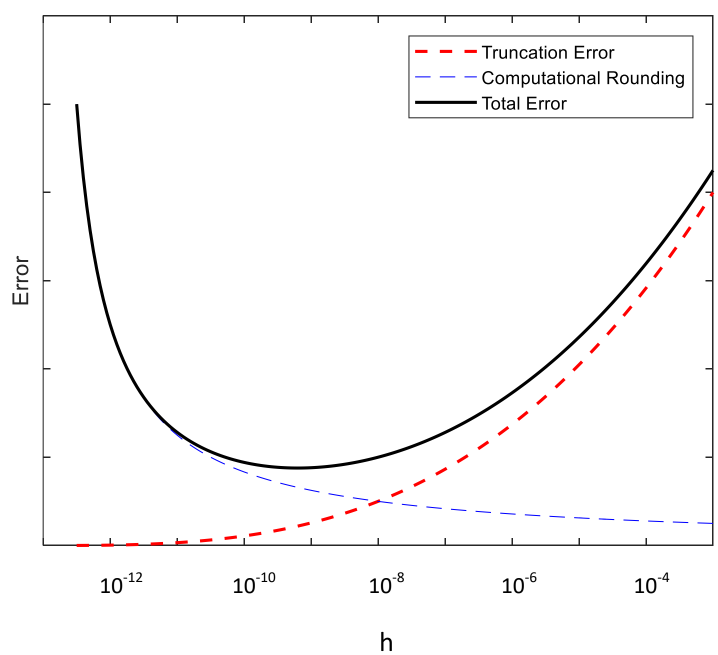


Figure 6: Total error, truncation error and computational error as a function of h.

Conclusion

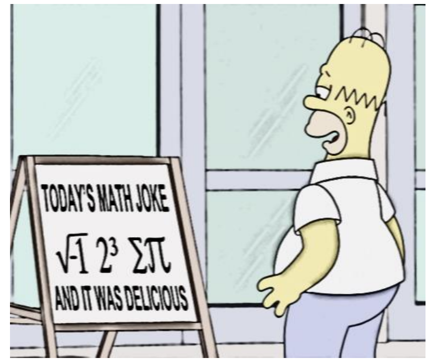


Figure 7: Today's math joke!

- You've now learned how to take raw data and calculate its derivatives without requiring the underlying equation.
- The biggest factors in determining whether your results are "accurate" enough is your choice of h, along with your choice of method.
    - Contrary to popular belief, using the smallest h isn't necessarily the best option – instead, it's about keeping a balance between a small error term and computational capabilities.

To find out more, visit the *supplementary module* which covers:

- Higher order partial derivatives

Next up: Integration!

[1] https://www.google.ca/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiPheOqo6vcAhVo34MKHepVBTEQjRx6BAgBEAU&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DZRe_ACYrJ0A&psig=AOvVaw0OMNVZAFoj8XQ7gZ-Rl9vN&ust=1532093356427937