module 02
all things errors + taylor

ahmad taha — spring 2026

ce 2989 — numerical methods in cee

email: ahmad.taha@vanderbilt.edu

webpage: http://lab.vanderbilt.edu/taha

VANDERBILT UNIVERSITY

January 22, 2026

outline

- approximation and round-off errors
- truncation errors, analysis, and bounds
- taylor series approximation
- taylor series approximations in higher dimensions
- error propagation and sensitivity analysis

approximation and round-off errors

- for many engineering problems, we cannot obtain analytical solutions
- numerical methods yield approximate results: results that are close to the exact solution
- we saw that in the derivative approximation in the previous module
- we cannot exactly compute the errors associated with numerical methods
- because we rarely know what the ground truth is
- and if we knew the ground truth, then this class is useless
- **ground truth**: **actual** solution to an ODE $\dot{x}(t) = f(x(t))$; solution to nonlinear system of equations $f(x) = 0$ or linear equations $Ax = b$
- but again, we usually don't know the ground truth so we need a ***numerical algorithm to approximate solutions***
- an algorithm usually introduces errors as well
- two important questions:
  - *how confident are we in an approximate result?*
  - *how much error is present in our calculation and is it tolerable?*

the concept of significant figures

- computers do not store infinite amount of decimals
- the more accurate a number is, the more digits (that follow the decimal point) will be correct
- only finitely many real numbers are representable
- most real numbers are replaced by nearby representable ones
- *significant figure:* a way to indicate precision
- significant digits of a number are those that can be used with confidence
- the numbers 0.00001845, 0.0001845, and 0.001845 all have four significant figures, all can be written as 1.845×10^{-k}
  1. all nonzero digits are significant
  2. zeroes between nonzero digits are significant
  3. zeroes to the left of the first nonzero digits are not significant
  4. zeroes to the right of a decimal point in a number are significant:
     - when a number ends in zeroes that are not to the right of a decimal point, the zeroes are not necessarily significant
- the numbers 4.53×10^4, 4.530×10^4, and 4.5300×10^4 have three, four, and five significant figures
- why are significant figures important for this class:
  - like in Euler's method, we need to know when to stop iterating so we need to know what kind of accuracy in terms of sig figs is required
  - some numbers like π, e, $\sqrt{5}$ have infinite number of digits, so we need to know when to stop
- how many sig figs to require? well this is application dependent

the concept of significant figures

- computers do not store infinite amount of decimals
- the more accurate a number is, the more digits (that follow the decimal point) will be correct
- only finitely many real numbers are representable
- most real numbers are replaced by nearby representable ones
- *significant figure:* a way to indicate precision
- significant digits of a number are those that can be used with confidence
- the numbers 0.00001845, 0.0001845, and 0.001845 all have four significant figures, all can be written as 1.845×10^{-k}
  1. all nonzero digits are significant
  2. zeroes between nonzero digits are significant
  3. zeroes to the left of the first nonzero digits are not significant
  4. zeroes to the right of a decimal point in a number are significant:
     - when a number ends in zeroes that are not to the right of a decimal point, the zeroes are not necessarily significant
- the numbers 4.53×10^4, 4.530×10^4, and 4.5300×10^4 have three, four, and five significant figures
- why are significant figures important for this class:
  - like in Euler's method, we need to know when to stop iterating so we need to know what kind of accuracy in terms of sig figs is required
  - some numbers like π, e, $\sqrt{5}$ have infinite number of digits, so we need to know when to stop
- how many sig figs to require? well this is application dependent

example on sig figs and when to stop

- suppose we want to compute the square root of 7
- one algorithm to compute square roots of numbers is newton's method (will learn about it later)
- problem can be written as $f(x) = 0 = x^2 - 7 = 0 \rightarrow$ solve for x
- newton's numerical iteration:

$$x_{k+1} = x_k - \frac{x_k^2 - 7}{2x_k} = \frac{1}{2}\left(x_k + \frac{7}{x_k}\right), \qquad k = 0, 1, 2, \ldots$$

iteration k	x_k	$\lvert x_k - \sqrt{7}\rvert$
0	3.0000000000	3.543×10^{-1}
1	2.6666666667	2.021×10^{-2}
2	2.6458333333	4.264×10^{-5}
3	2.6457513111	1.366×10^{-10}
4	2.6457513111	0
5	2.6457513111	0
6	2.6457513111	0
7	2.6457513111	0
8	2.6457513111	0
9	2.6457513111	0

- sig figs determine when to stop

definitions of errors

- we have different types of errors, need to be defined

absolute error

the absolute value of the difference between the true number x and its finite representation, or numerical approximation $\hat{x}$ is given as

$$\epsilon_a = \text{ae} = |x - \hat{x}| \geq 0$$

relative error

the relative value of the difference between the true number x and its finite representation, or numerical approximation $\hat{x}$ is given as

$$\epsilon_r = \text{re} = \frac{|x - \hat{x}|}{|x|} \geq 0$$

or $\text{re} \times 100\%$ in percentage points

what if we don't know the ground truth x?

- for numerical methods, true value is known only when we deal with analytical functions
- irl, we usually do not know the answer a priori
- so instead we use approximate relative error (are):

$$\epsilon_{ar} = \text{are} = \frac{|\text{approximate error}|}{|\text{approximation}|} \times 100\%$$

- in iterative methods like newton's method, we use

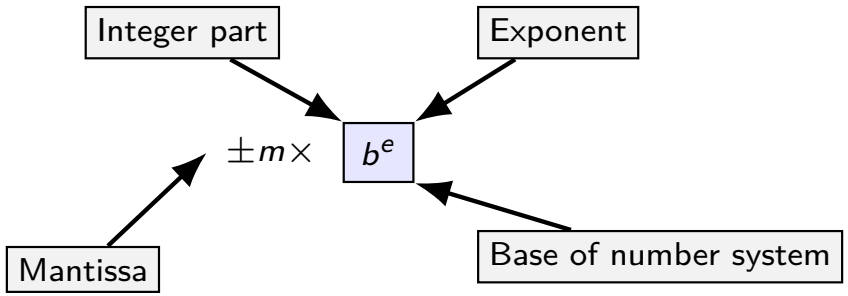
$$\epsilon_{ar} = \text{are} = \frac{|\text{current approximation} - \text{previous approximation}|}{|\text{current approximation}|} \times 100\%$$

- usually, we repeat computations until stopping criterion is satisfied:

$$\text{are} \leq \underbrace{\text{are}_{\text{max}}}_{\text{pre-defined tolerance}}$$

when to use ae or re?

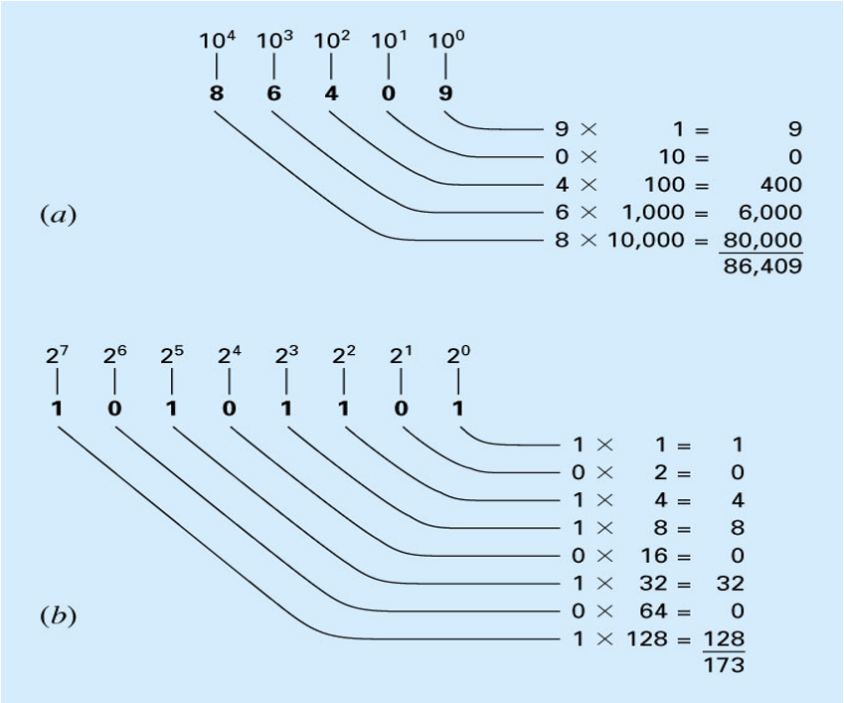
- when does one use one or the other definition of error?
- example: consider $x = 0.33, \hat{x} = 0.30$
- $\text{ae} = 0.03$ and $\text{re} = 0.03/0.33 = 0.09091 \approx 9.1\%$
- when $x = 0.33 \times 10^{-5}$ and $\hat{x} = 0.30 \times 10^{-5}$, we get $\text{ae} = 3 \times 10^{-7}$ or $\text{ae} = 3 \times 10^{-5}$ % but the $\text{re} = 0.09091 \approx 9.1\%$
- you now see that relative error is unchanged, while the absolute error changed by a factor of 10^{-5}
- notes:
  1. ae is strongly dependent on the magnitude of x
  2. ae is misleading unless it is stated what it is an error of
  3. re is a measure of the number of significant digits of x that are correct
- irrational numbers are typically represented in computer using *floating point* form (a scientific notation with finite digits and a floating exponent):

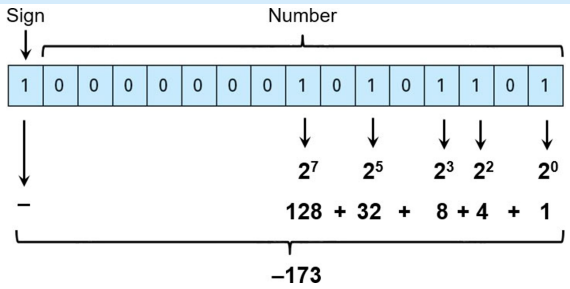


main computational blocks in computers

- computers are digital—they understand 0's and 1's, on/off
- smallest unit of computer information: the **bit** (short for *binary digit*)
- memory in a computer is via magnetic storage devices with optical storage
- using two levels of light reflectance: 0 and 1
- bits are the building blocks in computers and all digital information
- 1 byte $=$ 8 bits, 1 Kilobyte $=$ 8000 bits, 1 Megabyte $= 8 \times 10^6$ bits
- every word is usually 1–8 bytes, sometimes more depending on storage
- computers are classified by the number of bits they can process at one time
- computers often use 32 or 64 bits in their CPUs to process instructions
- because computers have finite capacity, not all numbers can be stored so there are numbers like π that cannot be precisely presented on a computer
- examples on floating point representations in base-10 and base-2

decimal vs binary system representation





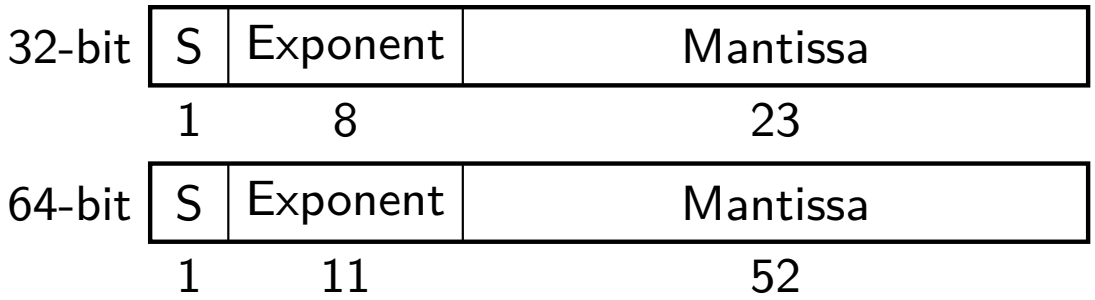
- computers are switches, understand binary numbers, base-2, 0 or 1, on or off
- this is in comparison with base-10 system that uses 0,1,2,3,...9 or base-8 or base-16
- converting from base-2 (0,1) to base-10 (0,1,2,...,9) system:

$$(1101)_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1$$

$$= (8) + (4) + (0) + (1) = 13$$

32 versus 64-bit computer architectures

- 32-bit vs 64-bit: difference in precision
- single precision (32-bit) has about 7 significant decimal digits
- item double precision (64-bit) has about 16 significant decimal digits
- more bits implies less rounding error accumulation
- 64-bit arithmetic allows reliable computation for much larger problem sizes where accuracy is needed
- difference between 32-bit vs 64-bit computers:



- mantissa being 52 digits on the 64-bit computer doesn't mean we can have 52 decimal points (remember, this number a bunch of zeroes and ones)

round-off errors

- when computers are used, unavoidable *round-off errors* happen especially with irrational numbers of quantities
- btw, which set is bigger? rational or irrational numbers?
- error arises because the arithmetic performed in a machine involves numbers with only a finite number of digits
- there are two major approaches: chopping and rounding
- when chopping a number to a specified number of decimal places, say n, the first n digits of the mantissa are retained
- when rounding a number, the computer chooses the closest number that is representable by the computer
- natural question: what error is committed when a number is chopped or rounded to n digits?

rounding versus chopping

- consider $x = 0.d_1 d_2 \cdots d_n d_{n+1} \cdots$ and **chopping** to n digits yields: $\hat{x} = 0.d_1 d_2 \cdots d_n$ with an error of

$$x - \hat{x} = 0.d_{n+1} d_{n+2} \cdots \times 10^{-n} < 0.999 \cdots \times 10^{-n} < 1 \times 10^{-n} \quad \leftarrow \quad \text{upper bound}$$

- now let's see what you do when you round instead of chop
- consider again that $x = 0.d_1 d_2 \cdots d_n d_{n+1} \cdots$ then if $d_{n+1} < 5$, round x to n digits and the error is

$$x - \hat{x} = 0.d_{n+1} d_{n+2} \cdots \times 10^{-n} \leq 0.499 \cdots \times 10^{-n} < 0.5 \times 10^{-n}$$

- you get some similar upper bound for the other case for $5 \leq d_{n+1} \leq 9$
- in short, the error committed by rounding a number x to n digits is 0.5×10^{-n}, which is half the upper bound (max error) committed by chopping

example: rounding vs chopping

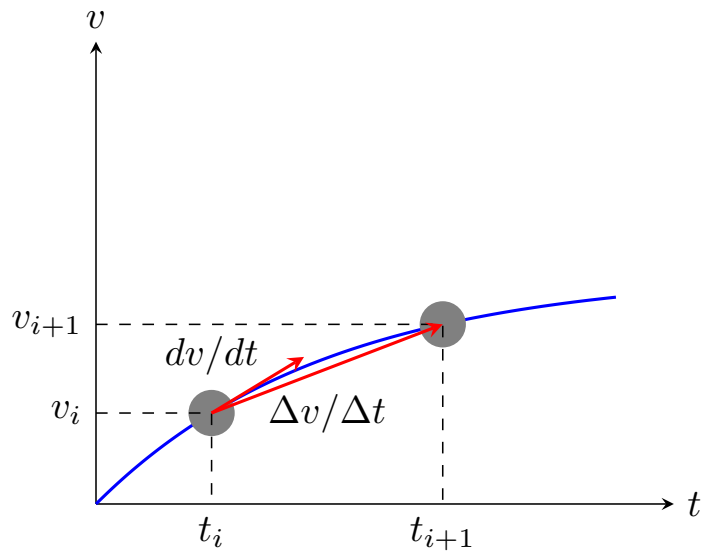
- storing $\pi = 3.14159265358$ using only 7 sig figs
- chopping: $\hat{\pi} = 3.141592$ then $\mathrm{re} = 0.00000065$
- rounding: $\hat{\pi} = 3.141593$ then $\mathrm{re} = 0.00000035$
- so chopping is worse than rounding?
- since number of significant figures is large enough, resulting chopping error is usually negligible

what are truncation errors?

- errors that yield from approximating functions with their approximations
- basically the difference between the actual value of a function and an approximated value
- this happens when an infinite series is replaced with a finite number of steps
- example: approximating the derivative of the velocity with its finite difference:

$$\dot{v}(t) = dv/dt \approx \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

  - how good is the approximation? *error analysis*
  - what's the impact of reducing Δt? *convergence analysis*



more motivation

- how do we approximate or exactly calculate different functions like $\cos(x), \sin(x), \tanh(x)$, etc.. via only basic operations (addition, division, etc..)?
- maclaurin series gives us a way of, for example, computing exponentials at x:

$$e^x = 1 + x + x^2/2! + x^3/3! + \cdots$$

- or even other nonlinear functions
- how do we derive such series for a given function or collection of functions?
- how many terms do we really need to add?
- how good is the approximation if we only consider k terms?
- knowing when to stop is important

taylor's theorem – different versions

- taylor's theorem: one of the most fundamental results in math + engineering
- good and bad news: many versions to this theorem
- unfortunate news:

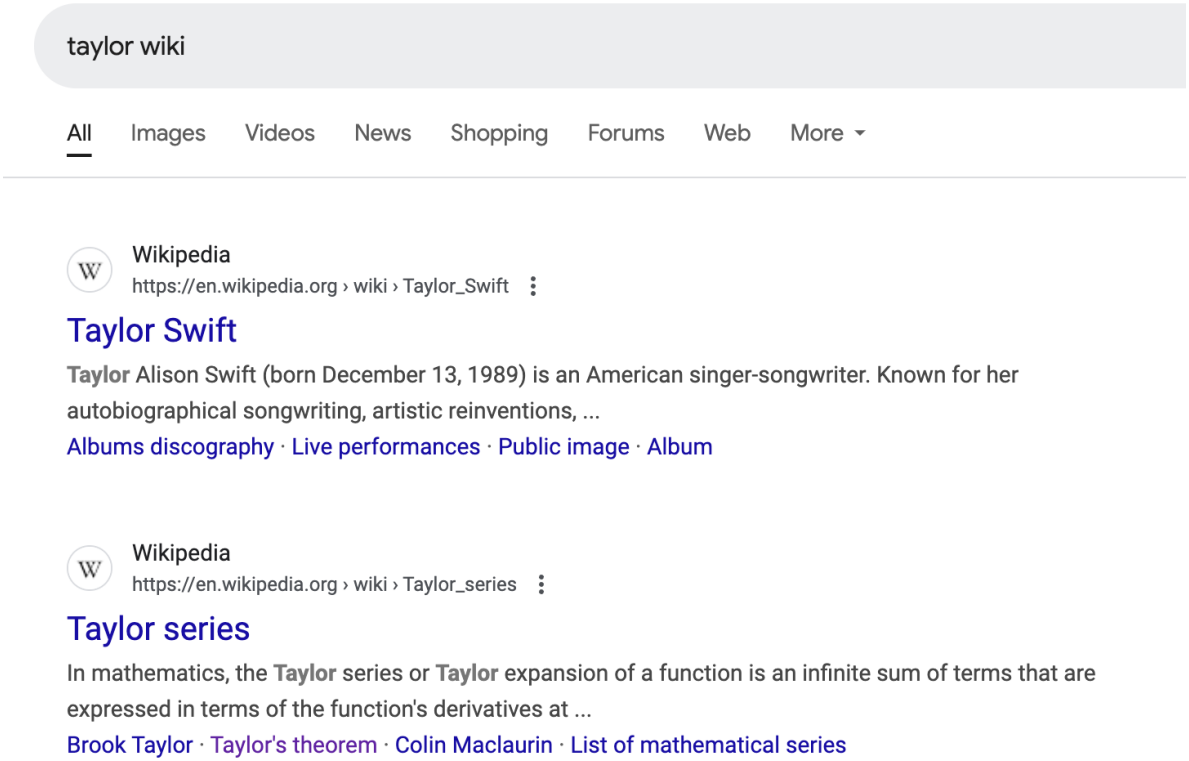


Figure: brook taylor would be heartbroken to know that google yields this

taylor's theorem

- taylor's theorem stated in 1715 but an earlier version exists (poor cauchy)

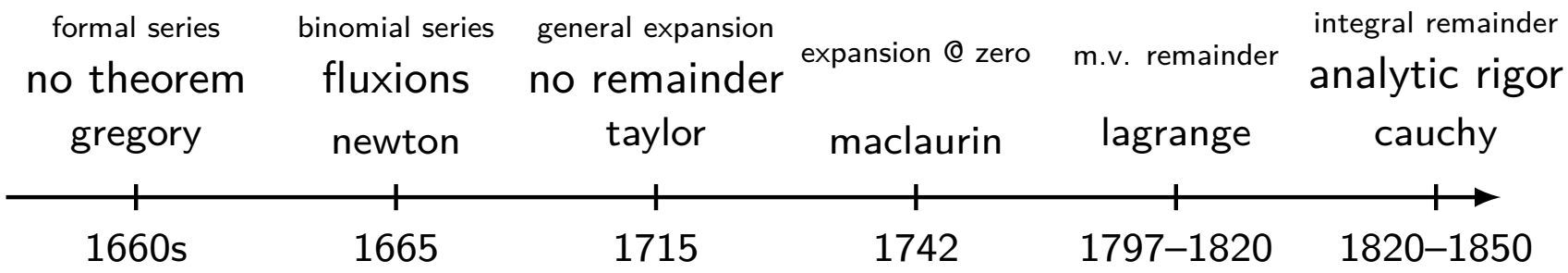
taylor's theorem – taylor's version – the integral form

if function f and its first $n+1$ derivatives are continuous on an interval containing a and x, then the function precisely can be written as

$$f(x) = \sum_{k=0}^{n} f^{(k)}(a) \frac{(x-a)^k}{k!} + \underbrace{\int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)\, dt}_{R_n}$$

where $f^{(k)}(a)$: kth derivative of $f(x)$ evaluated at $x = a$

taylor's theorem: historical timeline



- series existed before taylor, but only as formal algebraic objects; history:
  - gregory: gregory series for arctangent:

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{2k+1}, \qquad |x| \leq 1$$

  - general taylor expansion at an arbitrary point[1]:

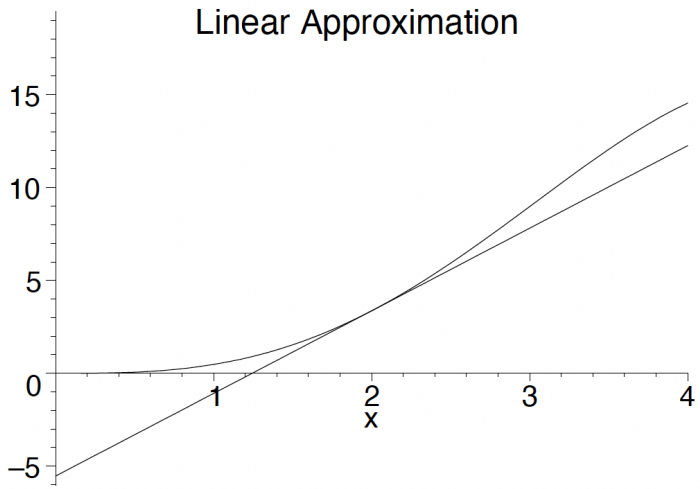
$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!} (x-a)^k + R_n(x)$$

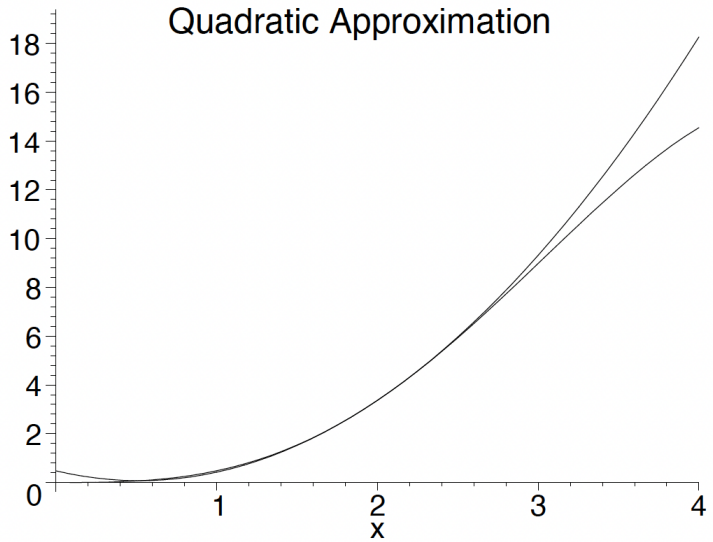
  - maclaurin (1698–1746): a special case around $x = 0$ for the taylor series
  - largrange and cauchy ...
- taylor (1715) was first to state a general expansion about an arbitrary point; rigor, error control, and convergence came later

[1] brook taylor, *methodus incrementorum directa et inversa* (1715)

taylor's theorem — why is it that important?

- taylor's theorem basically relates a function to its derivative and higher derivatives
- in fact, we will see how taylor's theorem is an extension of the mean value theorem (mvt)
- taylor's theorem is so powerful because it allows great approximation of functions via polynomials
- **example:** the following figure shows function $f(x) = x^2 \sin(x/2)$ and its linear approximation around $a = 2$
- notice how at $x = 2$, the function and the approximation are equal





taylor approximation about $x_0 = 2$: setup and first order

- consider the smooth function

$$f(x) = x^2 \ln x \, e^{\sin x}, \qquad x > 0$$

- evaluate at the expansion point:

$$f(2) = 2^2 \ln 2 \, e^{\sin 2} = 4 \ln 2 \, e^{\sin 2}$$

- compute the first derivative (product rule with three factors):

$$f'(x) = \frac{d}{dx}(x^2) \ln x \, e^{\sin x} + x^2 \frac{d}{dx}(\ln x) \, e^{\sin x} + x^2 \ln x \, \frac{d}{dx}(e^{\sin x})$$

$$= (2x) \ln x \, e^{\sin x} + x^2 \frac{1}{x} e^{\sin x} + x^2 \ln x \, e^{\sin x} \cos x$$

- factor common terms:

$$f'(x) = e^{\sin x} \left(2x \ln x + x + x^2 \ln x \cos x \right)$$

- evaluate the derivative at $x = 2$:

$$f'(2) = e^{\sin 2} \left(4 \ln 2 + 2 + 4 \ln 2 \cos 2 \right)$$

- first-order taylor approximation:

$$f_{linear}(x) = f(2) + f'(2)(x - 2)$$

second derivative and second-order approximation

- write the first derivative in product form:

$$f'(x) = e^{\sin x} g(x), \qquad g(x) = 2x \ln x + x + x^2 \ln x \cos x$$

- differentiate using the product rule:

$$f''(x) = \frac{d}{dx}(e^{\sin x}) g(x) + e^{\sin x} g'(x) = e^{\sin x} \Big(\cos x \, g(x) + g'(x) \Big)$$

- compute $g'(x)$ term by term:

$$\begin{aligned} g'(x) &= \frac{d}{dx}(2x \ln x) + \frac{d}{dx}(x) + \frac{d}{dx}(x^2 \ln x \cos x) \\ &= 2 \ln x + 2 + 1 + \cos x (2x \ln x + x) - x^2 \ln x \sin x \end{aligned}$$

- substitute back: $f''(x) =$

$$e^{\sin x} \Big[\cos x (2x \ln x + x + x^2 \ln x \cos x) + 2 \ln x + 3 + \cos x (2x \ln x + x) - x^2 \ln x \sin x \Big]$$

- evaluate at $x = 2$:

$$f''(2) = e^{\sin 2} \Big[\cos 2 (4 \ln 2 + 2 + 4 \ln 2 \cos 2) + 2 \ln 2 + 3 + \cos 2 (4 \ln 2 + 2) - 4 \ln 2 \sin 2 \Big]$$

- second-order taylor approximation:

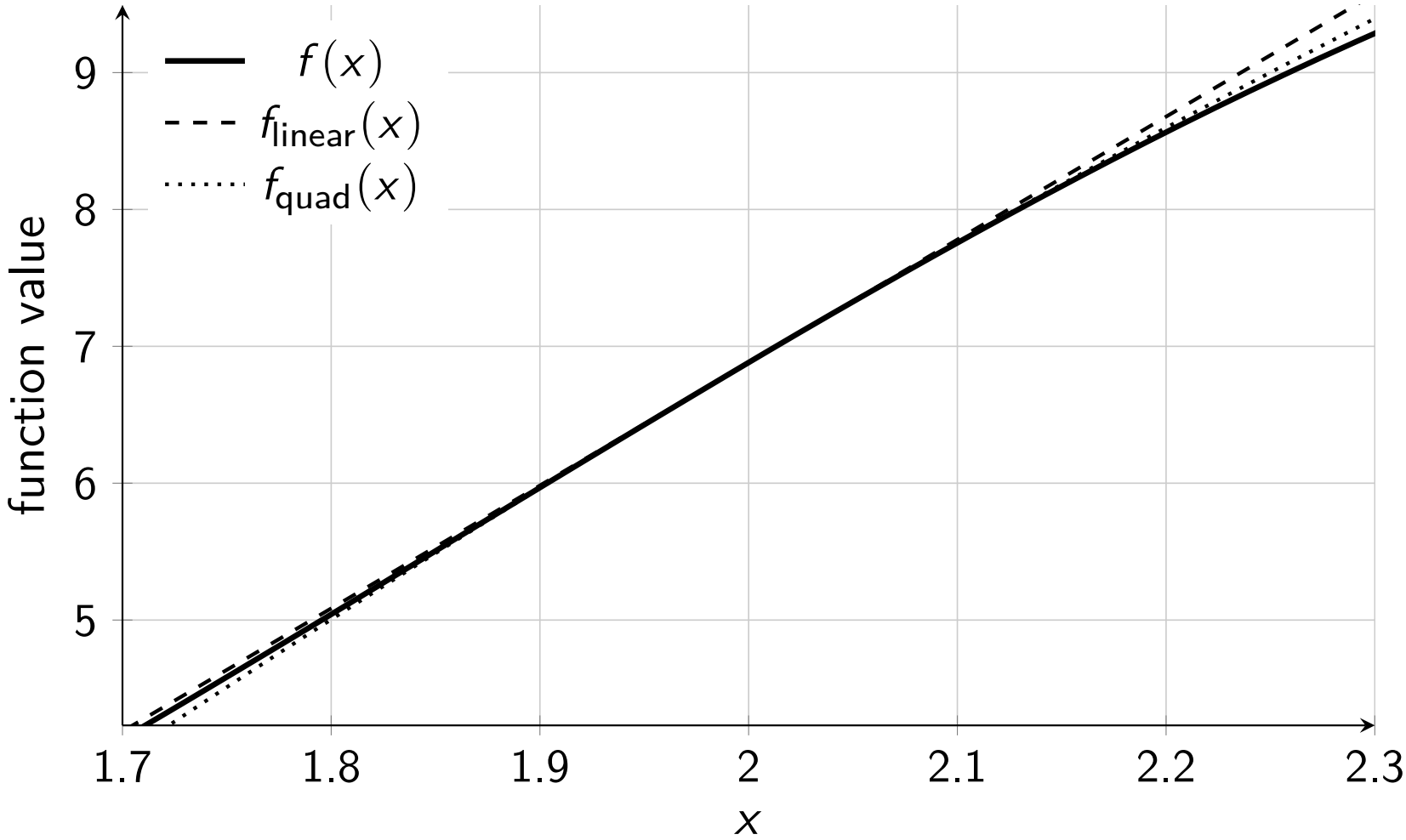
$$f_{quad}(x) = f(2) + f'(2)(x-2) + \frac{1}{2} f''(2)(x-2)^2$$

numerical comparison near $x_0 = 2$

x	$f(x)$	$f_{\text{linear}}(x)$	$f_{\text{quad}}(x)$	$\lvert f - f_{\text{linear}}\rvert$	$\lvert f - f_{\text{quad}}\rvert$
1.90	5.96920	5.98478	5.96458	1.56×10^{-2}	4.62×10^{-3}
1.95	6.42949	6.43397	6.42892	4.48×10^{-3}	5.72×10^{-4}
2.00	6.88317	6.88317	6.88317	0	0
2.05	7.32676	7.33236	7.32731	5.60×10^{-3}	5.56×10^{-4}
2.10	7.75698	7.78156	7.76136	2.46×10^{-2}	4.38×10^{-3}

- f_{linear} error slightly larger than f_{quad}
- second-order approximation reduces error by roughly one order of magnitude
- at the evaluation point $x_0 = 2$, any approximation recovers the function value—why?

taylor approximation near $x_0 = 2$



taylor series approximation

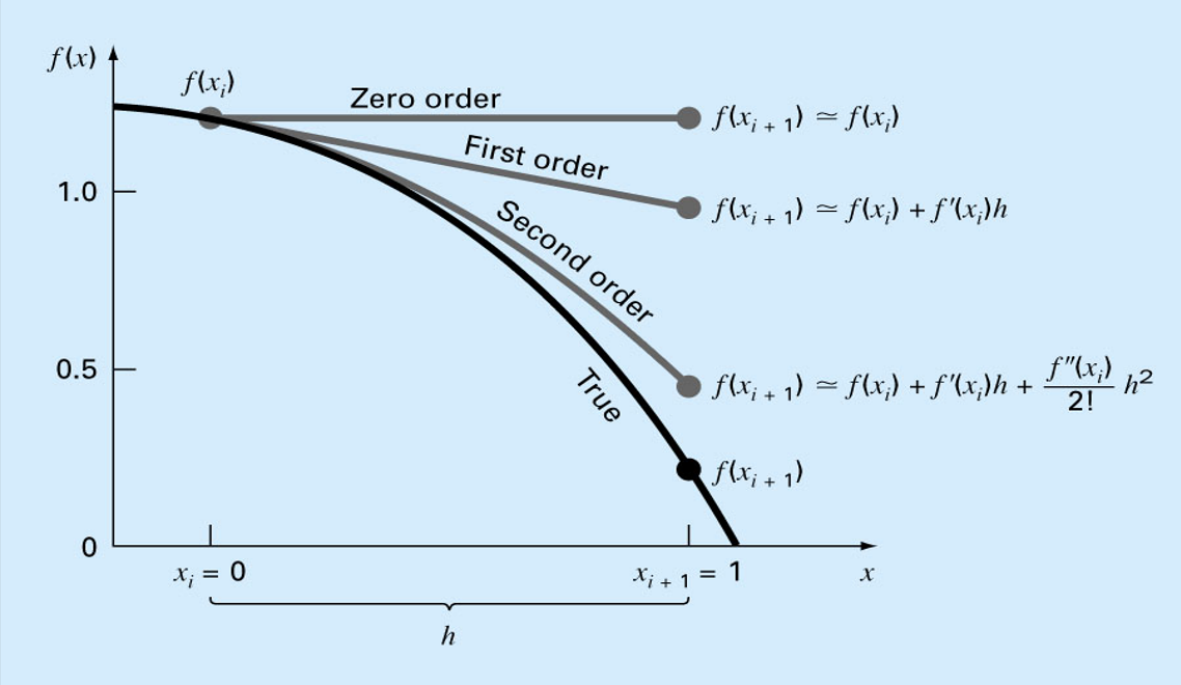
$$f(x) = \sum_{k=0}^{n} f^{(k)}(a)\frac{(x-a)^k}{k!} + \underbrace{\int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)\, dt}_{R_n}$$

- we usually ignore the residual R_n (or remainder)
- and we also ignore so many of the terms and consider an approximation of order $m << n$ because many functions anyway have zero-derivatives after $k > m$
- also, as we add more terms the m-order approximation around a (i.e., a local approx)

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2!}f''(a)(x-a)^2 + \frac{1}{3!}f^{(3)}(x-a)^3 + \cdots + \frac{1}{m!}f^{(m)}(x-a)^m$$

  becomes very accurate
- if $f(x)$ is a constant, then $f(x) = f(a)$...if $f(x)$ is a linear function, then?
- bad news: the R_n in the integral form is difficult to analyze...
- we need bounds and analysis on R_n

taylor series approximation



- functions such as trig, exponential, and others are expressed in an approximate fashion using taylor series
- note that any smooth function can be approximated as a polynomial
- in short, taylor series predicts the value of a function at one point (x) in terms of the function value $f(a)$ and its derivatives at another point (a)

mean value theorem saves the day

integral mean value theorem

for a differentiable function $f(x)$ defined over an interval $x \in [a, b]$, there exists c in (a, b) such that

$$\int_a^b f(x)\,dx = f(c)(b-a) \quad \text{or} \quad f'(c) = \frac{f(b)-f(a)}{b-a}$$

simiarly, if functions $g(x)$ and $h(x)$ are continuous and integrable on an interval $[a, b]$, and $h(x)$ does not change sign in the interval, then there exists a point c between a and x such that

$$\int_a^b g(x)h(x)dx = g(c)\int_a^b h(x)dx$$

- this allows us to write the following

$$R_n = \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)\,dt = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$$

  where ξ is a point between (a, x), unknown, but bounded between a and x
- this form is known as the *derivative or Lagrange form of remainder* R_n

why is this important?

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$$

- well, this makes analysis of truncation error R_n much easier as we don't have to integrate
- notice that in R_n now, we have a constant term $f^{(n+1)}(\xi)$ and the $(n+1)!$
- so basically, the accuracy of the nth order Taylor series approximation is a function of $(x-a)^{(n+1)}$
- in many instances, $x = x_{i+1}$ and $a = x_i$, so basically we end up with $x_{i+1} - x_i = h$ or the Δx, i.e., how far from x do we want to evaluate the function
- in that case, we say that the approximation is proportional to distance h raised to the $(n+1)$th power

example with $f(x) = e^x$

- problem: approximating e^x at $a = 0$ with $n = 2$
  - polynomial: $f_{quad}(x) = 1 + x + \frac{x^2}{2}$
  - derivatives: $f(x) = e^x$, so $f'''(x) = e^x$
- integral form (hard to use or analyze for error bounds):

$$R_2(x) = \frac{1}{2}\int_0^x e^t (x-t)^2 \, dt$$

- lagrange form (easy to use for bounds):

$$R_2(x) = \frac{e^c}{3!}x^3 \quad \text{for some } c \in (0, x)$$

- why is this helpful:
  - if we want error at $x = 1$, we know $c \in (0, 1)$
  - max error is easy to bound: $\frac{e^1}{6}(1)^3 \approx \frac{2.718}{6}$
- problem: analyze the truncation error for function $f(x) = e^x$ at $a = 0$

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1} = \frac{e^\xi}{(n+1)!}x^{n+1} \leq \frac{e^x}{(n+1)!}x^{n+1}$$

- for $n = 7, x = 1$, the upper bound is $R_7 \leq |e/8!| = 0.67 \times 10^{-4}$
- what happens with $n = 8, 9, 10$?

another example

- say we wanna approximate $e^{0.01}$ with an error of less than 10^{-12}, how many terms n are needed?
- well recall that $0 < \xi < 0.01$, so

$$R_n = \frac{e^{\xi}}{(n+1)!}x^{n+1} \leq \frac{e^{0.01}}{(n+1)!}0.01^{n+1} < \frac{1.1}{(n+1)!}0.01^{n+1} < 10^{-12}$$

- testing different values of n, yields $n = 4$ is needed
- changing x to $x = 0.5$, yields $n = 12$ to achieve the desired accuracy
- and to approximate $e^{10.5}$, $n = 55$ is required...too much

approximation of derivatives and finite differences

- taylor series is extremely useful in estimating truncation errors of known functions
- but sometimes we don't know what the function is but we know the derivatives of it
- recall the example of the hot air balloon...we can predict velocity (instead of solving the ODE for $v(t)$):

$$v(t_{i+1}) = v(t_i) + v'(t_i)(t_{i+1} - t_i) + 0.5v''(t_i)(t_{i+1} - t_i)^2 + \cdots + R_n$$

- we can also consider only first order approximation given by euler:

$$v(t_{i+1}) = v(t_i) + v'(t_i)(t_{i+1} - t_i) + R_1 \quad \Rightarrow \quad v'(t_i) = \underbrace{\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}}_{\text{approximation}} - \underbrace{\frac{R_1}{t_{i+1} - t_i}}_{\text{truncation error}}$$

- recall that

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - a)^{n+1}$$

so this entails that the truncation error for forward euler

$$\frac{R_1}{t_{i+1} - t_i} = \frac{v''(\xi)}{2!}(t_{i+1} - t_i) = O(h) \quad \text{is proportional to the step size}$$

forward, backward, and central difference schemes

- objective here: to compute derivatives of functions $f'(x)$ at arbitrary points with different equations
- we learned about forward euler, but two more methods can be similarly derived
- one of them yields similar results to forward euler, but the other one is so much better!!!!
- backward euler (or backward divided difference) can be written as:

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

compared to the forward euler (or forward divided difference)

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

- a third method is the central difference:

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

where you can show that the error or remainder is proportional to the square of step-size h

derivation of forward, backward, and central differences

- first let's assume that the function is continuous and differentiable and that $x_i = x_{i-1} + h$
- taylor expansions about x_i:

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(\xi_+)$$

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(\xi_-)$$

  with $\xi_+ \in (x_i, x_{i+1})$, $\xi_- \in (x_{i-1}, x_i)$.
- forward divided difference:

$$\frac{f(x_{i+1}) - f(x_i)}{h} = f'(x_i) + O(h)$$

- backward divided difference:

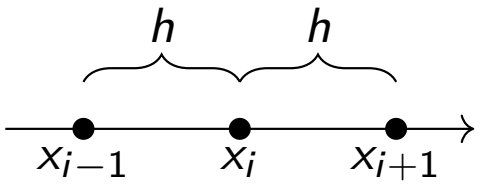
$$\frac{f(x_i) - f(x_{i-1})}{h} = f'(x_i) - O(h)$$

- central divided difference (subtract expansions):

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

summary and examples

finite differences:



summary table:

method	formula	order
forward	$\frac{f_{i+1}-f_i}{h}$	$O(h)$
backward	$\frac{f_i-f_{i-1}}{h}$	$O(h)$
central	$\frac{f_{i+1}-f_{i-1}}{2h}$	$O(h^2)$

- example: consider the function $f(x)=\sin x$ so $f'(x)=\cos x$
- we approximate $f'(x_0)$ at $x_0=1 \quad \Rightarrow \quad f'(1)=\cos(1)$ and we compare:
  - forward difference: $d_f(h)=\frac{f(1+h)-f(1)}{h}$
  - backward difference: $d_b(h)=\frac{f(1)-f(1-h)}{h}$
  - central difference: $d_c(h)=\frac{f(1+h)-f(1-h)}{2h}$
- the exact error is

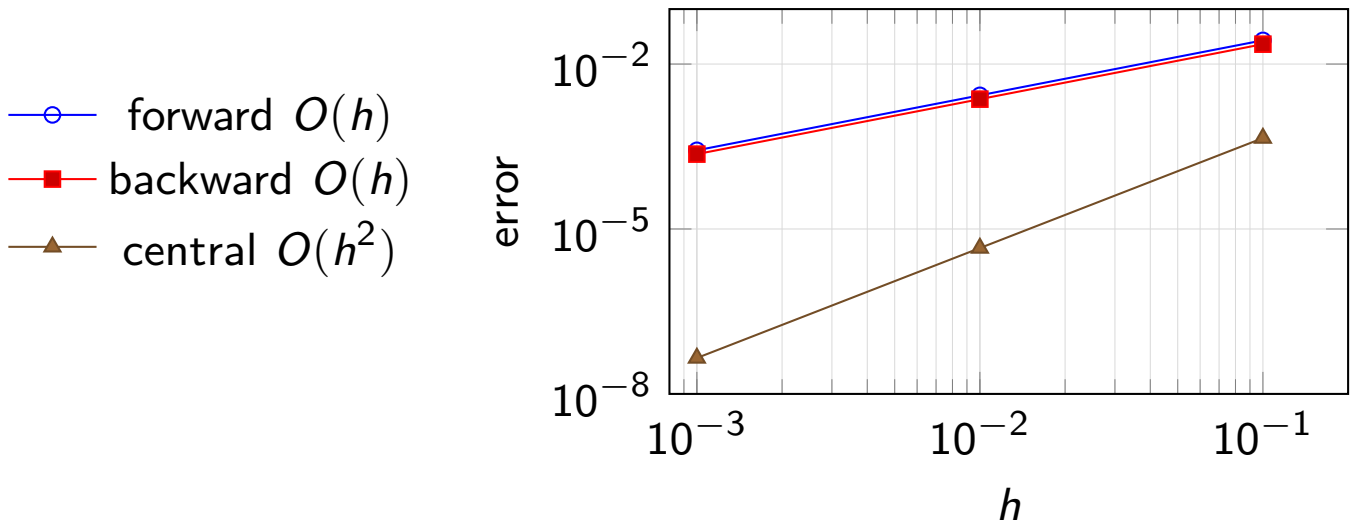
$$\text{error} = |\text{approximation} - \cos(1)|$$

numerical accuracy comparison

- exact value: $\cos(1) \approx 0.540302306$
- results:

h	forward error	backward error	central error
10^{-1}	2.7×10^{-2}	2.3×10^{-2}	4.5×10^{-4}
10^{-2}	2.7×10^{-3}	2.3×10^{-3}	4.5×10^{-6}
10^{-3}	2.7×10^{-4}	2.3×10^{-4}	4.5×10^{-8}

- observations:
  - forward and backward errors scale like $O(h)$
  - central difference error scales like $O(h^2)$
  - halving h reduces central error by ≈ 4



example (another one)

- we wish to approximate function

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.25$$

  at $x = 0.5$ with two different step-sizes of $h = 0.5$ and $h = 0.25$
- here the derivative can be computed analytically via $f'(x) = -0.4x^3 - 0.45x^2 - x - 0.25$ and hence $f'(x = 0.5) = -0.9125$
- to compute the approximations, set $x_i = 0.5, x_{i+1} = 1, x_{i-1} = 0$ for $h = 0.5$ then plug in the formulae for $f'(x_i)$ for the three divided differences
- central difference approx is more accurate than forward or backward differences
- as predicted by Taylor series analysis, halving h approximately halves the error of the backward and forward differences and
- quarters the error of the centered difference

example (solution)

** for $h = 0.5$, grid values:

$$x_{i-1} = 0, \quad f(x_{i-1}) = 1.2, \qquad x_i = 0.5, \quad f(x_i) = 0.925, x_{i+1} = 1.0, \quad f(x_{i+1}) = 0.2$$

- forward divided difference and backward divided difference:

$$f'(0.5) \approx \frac{0.2 - 0.925}{0.5} = -1.45, \quad |\text{re}| = 58.9, \quad f'(0.5) \approx \frac{0.925 - 1.2}{0.5} = -0.55, |\text{re}| = 39.7\%$$

- centered divided difference:

$$f'(0.5) \approx \frac{0.2 - 1.2}{1.0} = -1.0, \qquad |\text{re}| = 9.6\%$$

** for $h = 0.25$, grid values:

$$x_{i-1} = 0.25, \quad f(x_{i-1}) = 1.10351, \quad x_i = 0.5, \quad f(x_i) = 0.925, x_{i+1} = 0.75, f(x_{i+1}) = 0.6363$$

- forward divided difference:

$$f'(0.5) \approx \frac{0.6363 - 0.925}{0.25} = -1.155, \qquad |\text{re}| = 26.5\%$$

- backward divided difference:

$$f'(0.5) \approx \frac{0.925 - 1.10353}{0.25} = -0.714, \qquad |\text{re}| = 21.7\%$$

- centered divided difference:

$$f'(0.5) \approx \frac{0.6363 - 1.1035}{0.5} = -0.934, \qquad |\text{re}| = 2.4\%$$

error propagation

- suppose we have a function $f(x)$ that is dependent on one variable x
- assume that $\hat{x}$ is an approximation of x
- **objective:** assess the effect of discrepancy between x and $\hat{x}$ on funciton value:

$$\Delta f(\hat{x}) = |f(x) - f(\hat{x})|$$

- issue is when the ground truth x is not known, but can use taylor series approximation

$$f(x) = f(\hat{x}) + f'(\hat{x})(x - \hat{x}) + 1/2 f''(\hat{x})(x - \hat{x})^2 + \dots$$

- dropping the second- and higher-order terms yields

$$f(x) - f(\hat{x}) \approx f'(\hat{x})(x - \hat{x}) \quad \Rightarrow \Delta f(\hat{x}) \approx |f'(\hat{x})| \Delta \hat{x}$$

- **example:** given $\hat{x} = 2.5$ and an error $\Delta \hat{x} = 0.01$, estimate the resulting error in function $f(x) = x^3$
- **solution:** well, using the above equation

$$\Delta f(\hat{x}) \approx |f'(\hat{x})| \Delta \hat{x} = 3 \cdot 2.5^2 \cdot 0.01 = 0.1875$$

and since $f(\hat{x} = 2.5) = 15.625$, then we can say that the real function value will be $f(x) = 15.625 \pm 0.1875$

- how is this useful?

extending uncle taylor to more than one variable

- let's be real: the world is not a function of one variable
- most problems we solve, most ODEs, integrals, are functions of many variables
- how does the taylor theorem extend to higher dimensions where $x \in \mathbb{R}^m$ not just a scalar $x \in \mathbb{R}$ and the function $f(x) : \mathbb{R}^n \to \mathbb{R}$ at a point $a \in \mathbb{R}^n$
- do you really wanna know? be honest with me

$$f(\boldsymbol{x}) = f(\boldsymbol{a}) + \sum_{1 \leq |\alpha| \leq k} \frac{1}{\alpha!} (D^\alpha f)(\boldsymbol{a})(\boldsymbol{x} - \boldsymbol{a})^\alpha$$

$$+ \sum_{|\alpha| = k+1} \frac{k+1}{\alpha!} (\boldsymbol{x} - \boldsymbol{a})^\alpha \int_0^1 (1-t)^k (D^\alpha f)(\boldsymbol{a} + t(\boldsymbol{x} - \boldsymbol{a}))\, dt.$$

- looks absolutely disgusting
- explanation of the theorem + comparison with the single dimension version

$$f(x) = \sum_{k=0}^{n} f^{(k)}(a) \frac{(x-a)^k}{k!} + \underbrace{\int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)\, dt}_{R_n}$$

let's chill now and go back to fotsa

- fotsa (first order taylor series approximation) in higher dimensions can be written as

$$f(x) \approx f_{\text{lin}}(x) = f(a) + \nabla^\top f(a)(x-a)$$

- sotsa (second order...or quadratic) in higher dimensions can be written as

$$f(x) \approx f_{\text{quad}}(x) = f_{\text{lin}}(x) + \frac{1}{2}(x-a)^\top \nabla^2 f(a)(x-a)$$

- **example:** let $f(x) = f(x_1, x_2) = x_1 e^{x_2} + \cos(x_1 x_2)$. Compute fotsa and sotsa of $f(x)$ around $a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- **fotsa and sotsa:**

$$f_{\text{lin}} = f(a) + \nabla^\top f(a)(x-a) = 2 + \begin{bmatrix} e^0 - 0\sin(1\cdot 0) \\ 0 \cdot e^{a_2} - 0\sin(0\cdot 1) \end{bmatrix}^\top \begin{bmatrix} x_1 - 1 \\ x_2 - 0 \end{bmatrix} = 1 + x_1 + x_2$$

$$f_{\text{quad}} = f_{\text{lin}} + (x_1 - 1)x_2 = 1 + x_1 + x_2 + (x_1 - 1)x_2$$

detailed solutions of previous example

- function value: $f(1,0) = 1 \cdot e^0 + \cos(0) = 2$
- gradient: $\nabla f(1,0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- linear approximation:
  $f_{\text{lin}}(x) = f(a) + \nabla f(a)^\top (x-a) = 2 + (x_1 - 1) + x_2 = \boxed{1 + x_1 + x_2}$
- compute second derivatives:

$$\frac{\partial^2 f}{\partial x_1^2} = -x_2^2 \cos(x_1 x_2), \quad \frac{\partial^2 f}{\partial x_2^2} = x_1 e^{x_2} - x_1^2 \cos(x_1 x_2),$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = e^{x_2} - \sin(x_1 x_2) - x_1 x_2 \cos(x_1 x_2)$$

- $\nabla^2 f(x)$ at $(1,0)$:

$$\nabla^2 f(1,0) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- quadratic approximation:

$$f_{\text{quad}}(x) = f(a) + \nabla f(a)^\top (x-a) + \tfrac{1}{2}(x-a)^\top \nabla^2 f(a)(x-a)$$

- since $(x-a)^\top \nabla^2 f(a)(x-a) = 2(x_1 - 1)x_2$:

$$\boxed{f_{\text{quad}}(x_1, x_2) = 2 + (x_1 - 1) + x_2 + (x_1 - 1)x_2}$$

what about error propagation in higher dimensions?

- well, similar to the previous slide on error propagation, we can extend the formula to higher dimensions
- we can write

$$\Delta f(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \approx |\frac{\partial f(x)}{\partial x_1}\Delta\hat{x}_1| + |\frac{\partial f(x)}{\partial x_2}\Delta\hat{x}_2| + \ldots + |\frac{\partial f(x)}{\partial x_n}\Delta\hat{x}_n|$$

  where $\Delta\hat{x}_j = x_j - \hat{x}_j$
- **example:** sensitivity analysis. deflection y of the top of a sailboat mast is given by this nonlinear equation

$$y = f(F, L, E, I) = \frac{FL^4}{8EI}$$

  where where F is the uniform side loading (N/m), L is the height (m), E defines modulus of elasticity (N/m2), I depicts moment of inertia
- estimate the error in y given the following data:

$$\begin{aligned} \tilde{F} &= 750 \text{ N/m} & \Delta\tilde{F} &= 30 \text{ N/m} \\ \tilde{L} &= 9 \text{ m} & \Delta\tilde{L} &= 0.03 \text{ m} \\ \tilde{E} &= 7.5 \times 10^9 \text{ N/m}^2 & \Delta\tilde{E} &= 5 \times 10^7 \text{ N/m}^2 \\ \tilde{I} &= 0.0005 \text{ m}^4 & \Delta\tilde{I} &= 0.000005 \text{ m}^4 \end{aligned}$$

$$\Delta y \approx \frac{L^4}{8EI}\Delta\hat{F} + \frac{FL^3}{2EI}\Delta\hat{L} + \frac{FL^4}{8E^2I}\Delta\hat{E} + \frac{FL^4}{8EI^2}\Delta\hat{I} = 0.0065 + 0.0021 + 0.0011 + 0.0016 = 0.0114$$

- hence $y = 0.164 \pm 0.0114$ or y is between 0.152 & 0.175

module summary

Error Definitions

True error: $E_t = \text{true value} - \text{approximation}$

True percent relative error: $\varepsilon_t = \dfrac{\text{true value} - \text{approximation}}{\text{true value}} 100\%$

Approximate percent relative error: $\varepsilon_a = \dfrac{\text{present approximation} - \text{previous approximation}}{\text{present approximation}} 100\%$

Stopping criterion: Terminate computation when

$$\varepsilon_a < \varepsilon_s$$

where ε_s is the desired percent relative error

Taylor Series

Taylor series expansion:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \cdots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

where

Remainder:

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1}$$

or

$$R_n = O(h^{n+1})$$

Numerical Differentiation

First forward finite divided difference:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

(Other divided differences are summarized in Chaps. 4 and 23.)

Error Propagation

For n independent variables $x_1, x_2, \ldots, x_n$ having errors $\Delta\tilde{x}_1, \Delta\tilde{x}_2, \ldots, \Delta\tilde{x}_n$, the error in the function f can be estimated via

$$\Delta f = \left|\frac{\partial f}{\partial x_1}\right| \Delta\tilde{x}_1 + \left|\frac{\partial f}{\partial x_2}\right| \Delta\tilde{x}_2 + \cdots + \left|\frac{\partial f}{\partial x_n}\right| \Delta\tilde{x}_n$$