CE 205: Numerical Methods

2.00 Credits, 2hrs/week

Dr. Tanvir Ahmed
Assistant Professor

What are Numerical Methods?

Techniques by which mathematical problems are formulated so that they can be solved with simple arithmetic operations

↓

Addition (+), Subtraction(-), Division (/), Multiplication (*)

This may involve a large number of tedious calculations

↓

Thanks to digital computers!!

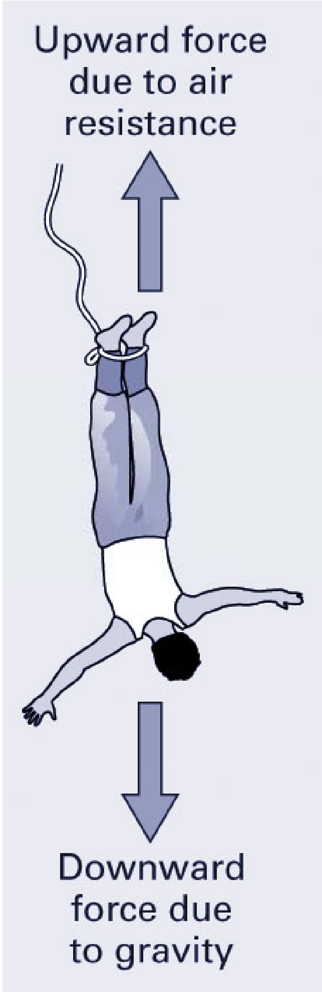
Example: Newton's 2nd law of motion

$$F = ma$$

<u>The Bungee Jumper Problem</u>
the change in velocity is determined by the gravitational forces acting on the jumper versus the drag force.

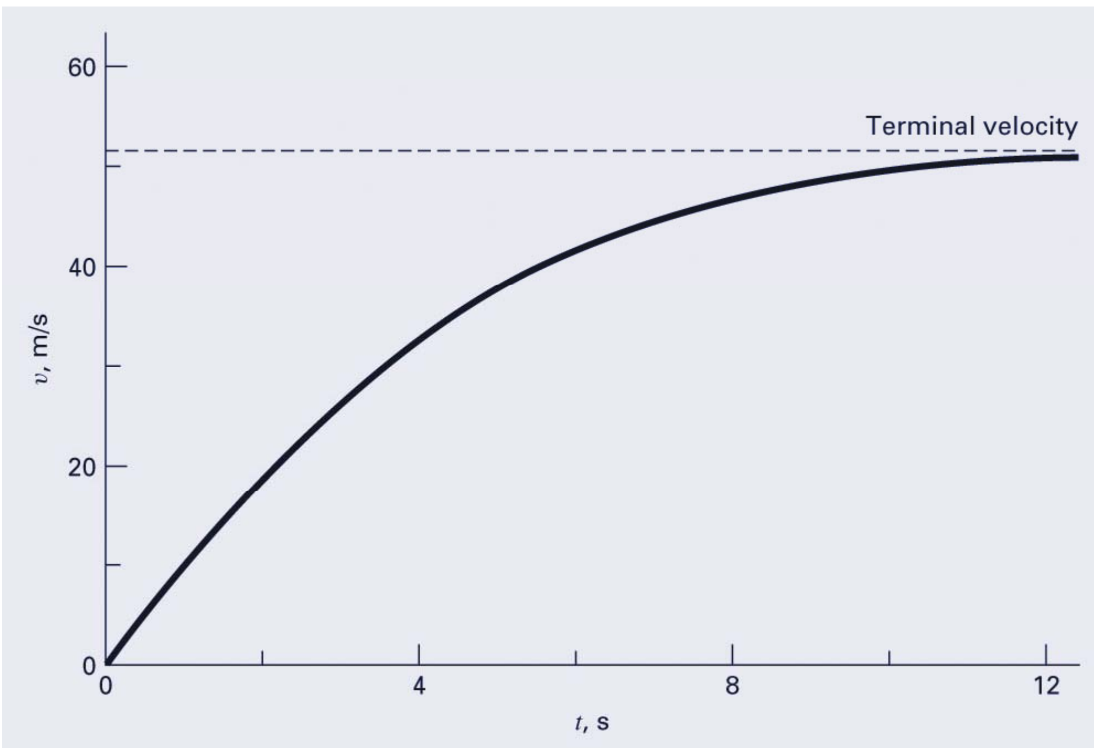
$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

c_d = drag coefficient
m = mass of the jumper
g = gravitational acceleration



Analytical solution

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2 \quad \Rightarrow \quad v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t \right)$$



Considering

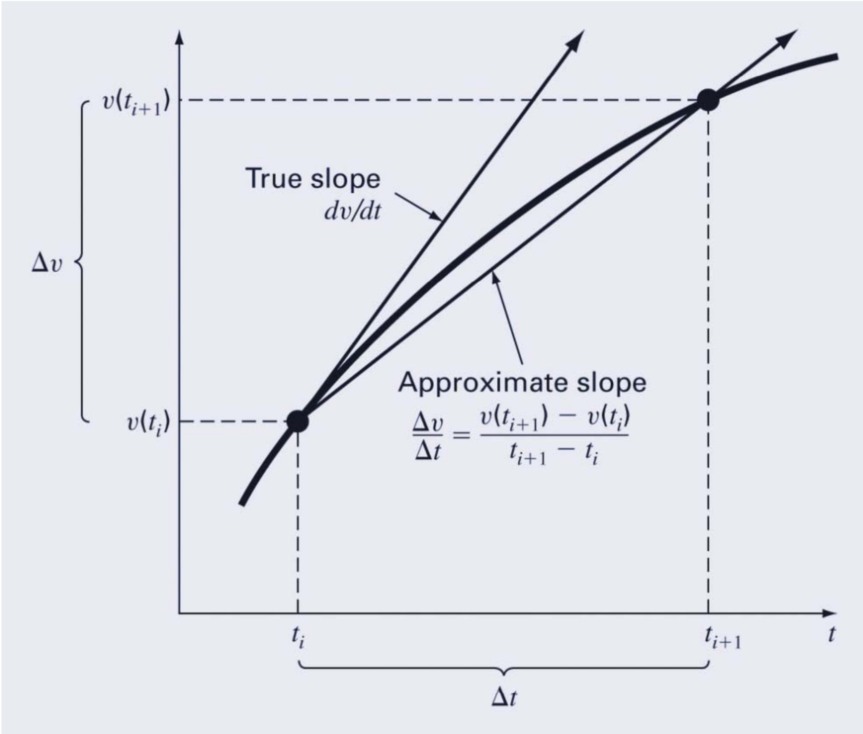
m = 68.1 kg
c_d = 0.25 kg/m

Numerical solution

Need to make some approximation regarding the time rate change of velocity

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

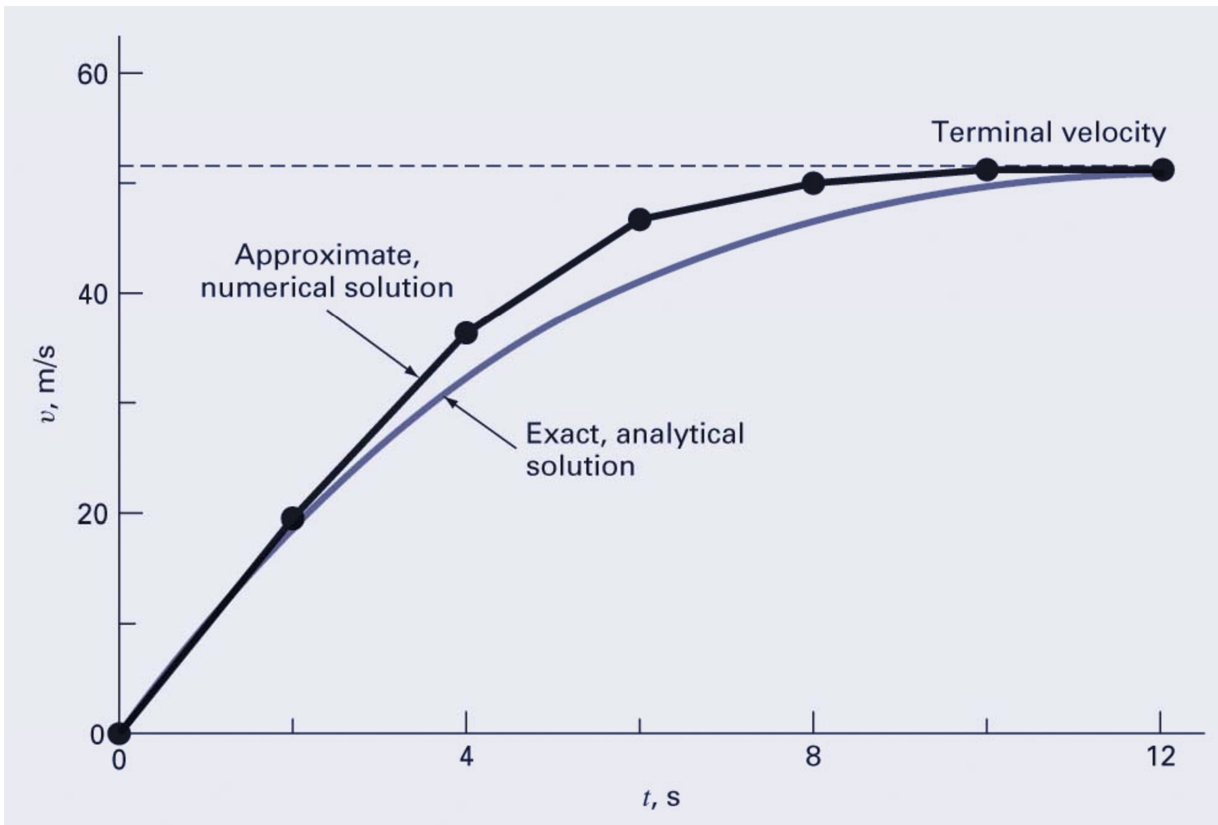
$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$



$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c_d}{m} v(t_i)^2 \right] (t_{i+1} - t_i)$$

Comparison with analytical solution

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c_d}{m} v(t_i)^2 \right] (t_{i+1} - t_i)$$



Why learn numerical methods?

•Analytical solutions can be derived for only a limited class of problems.

•A broad spectrum of problems can be solved.

- complicated, nonlinear problems
- large systems of equations

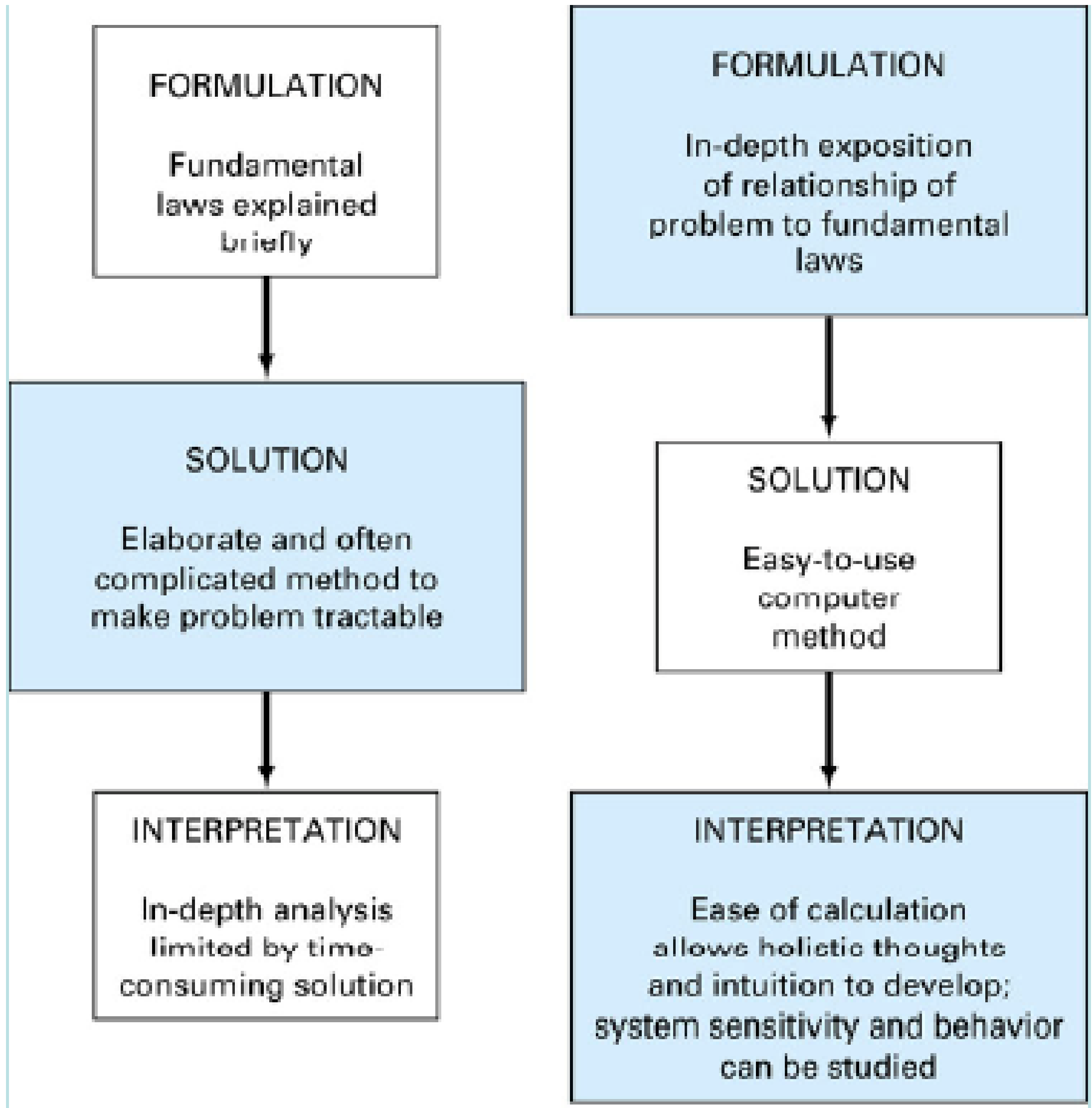
•Intelligently use commercially available computer programs ("packages")

•To make your own customized program to tackle problems which packages cannot solve

•Bolster your understanding of mathematics

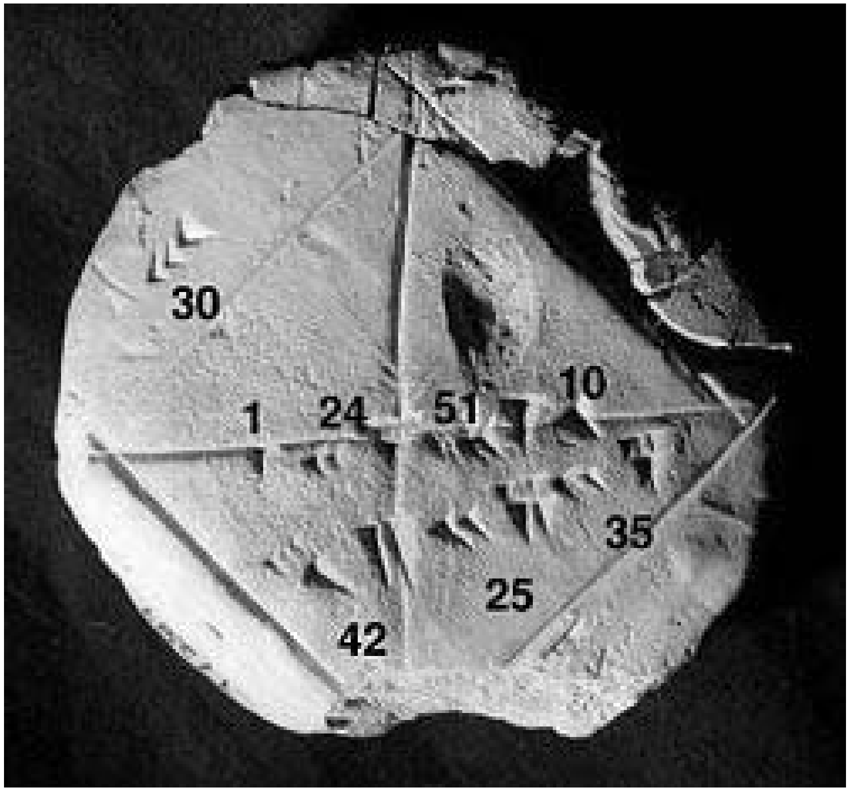
Computers and numerical methods practice

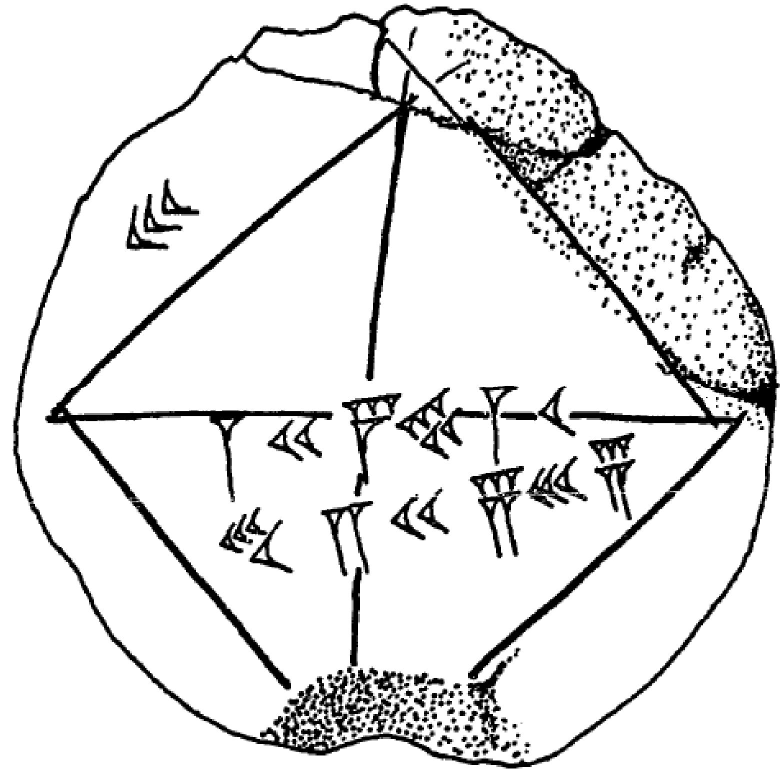
Pre computer era



Computer era

Historical evidence of numerical approximation





Babylonian tablet BC 7289, which gives a sexagesimal numerical approximation of square root of 2

$$\sqrt{2} = 1 + {}^{24}\!/_{60} + {}^{51}\!/_{60^2} + {}^{10}\!/_{60^3} = 1.41421296\ldots$$

Modern applications

- Numerical weather prediction
- Computing the trajectory of a spacecraft (requires the solution of a system of ODE)
- Simulations of car crashes by car companies (Numerical PDE solution)
- Sophisticated optimization algorithms to decide ticket prices, airplane and crew assignments and fuel needs.
- Insurance companies use numerical programs for actuarial analysis.
- And many more….

CE205 Course Outline

- Systems of linear algebraic equations
- **Interpolation and Curve-fitting**
- **Roots of Equations**
- Numerical differentiation
- Numerical integration
- **Initial value problems**
- **Two-point boundary value problems**

References

- Any standard undergraduate textbook on Numerical Methods
- Some examples:
  - -Numerical Methods for Engineers: Chapra & Canale
  - -Numerica Methods: E Balagurusamy
  - -Introductory Methods of Numerical Analysis: Sastry
  - -Numerical Analysis: Goel and Mittal

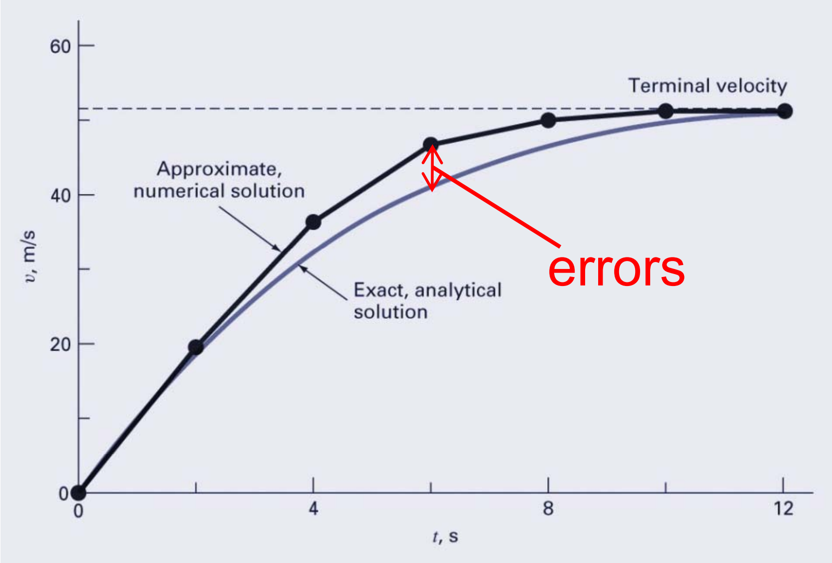
Characteristics of Numerical Computing

- Accuracy (all numerical methods introduce errors)
  - truncation errors
  - round-off errors
- Rate of convergence (how quickly we can arrive at a solution)
- Numerical stability
- Efficiency (computational burden)

Errors in Numerical Methods

In numerical methods, **approximations** are used to express exact mathematical operations

This gives rise to errors



How can we quantify it?

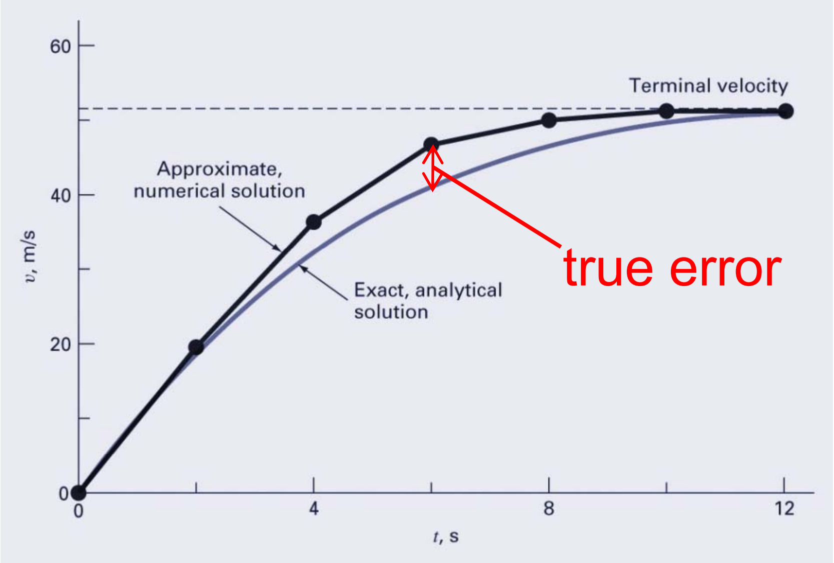
True error (E_t) *= True value - approximation*

Percent relative error (ε_t) *= (True error* (E_t) */ True value)*100*

Errors in Numerical Methods

If the problem cannot be analytically solved, the true value will not be known

We normalize the error with approximate value



Numerical methods use an iterative approach

- Present approximation is made based on previous approximation

$$\varepsilon_a = \frac{current\ approx. - previous\ approx.}{current\ approx.} \times 100\%$$

Errors in Numerical Methods

$$\varepsilon_a = \frac{current\ approx. - previous\ approx.}{current\ approx.} \times 100\%$$

- ❑Use absolute value
- ❑Continue computations until a stopping criterion is satisfied

$$|\varepsilon_a| < \varepsilon_s$$

A prespecified tolerance level

The criteria used to ensure that the result is correct upto *n* significant figures:

$$\varepsilon_s = (0.5 \times 10^{2-n})\%$$

Roundoff errors

Quantities such as π, 1/3 cannot be expressed by a fixed number of significant figures

Roundoff errors arise because digital computers cannot represent some quantities exactly.

- Digital computers have size and precision limits on their ability to represent numbers.
- Certain numerical manipulations are highly sensitive to roundoff errors

Truncation errors

Result from the use of approximations instead of exact mathematical procedures.

Example: using finite number of terms to estimate the sum of an infinite series, say

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

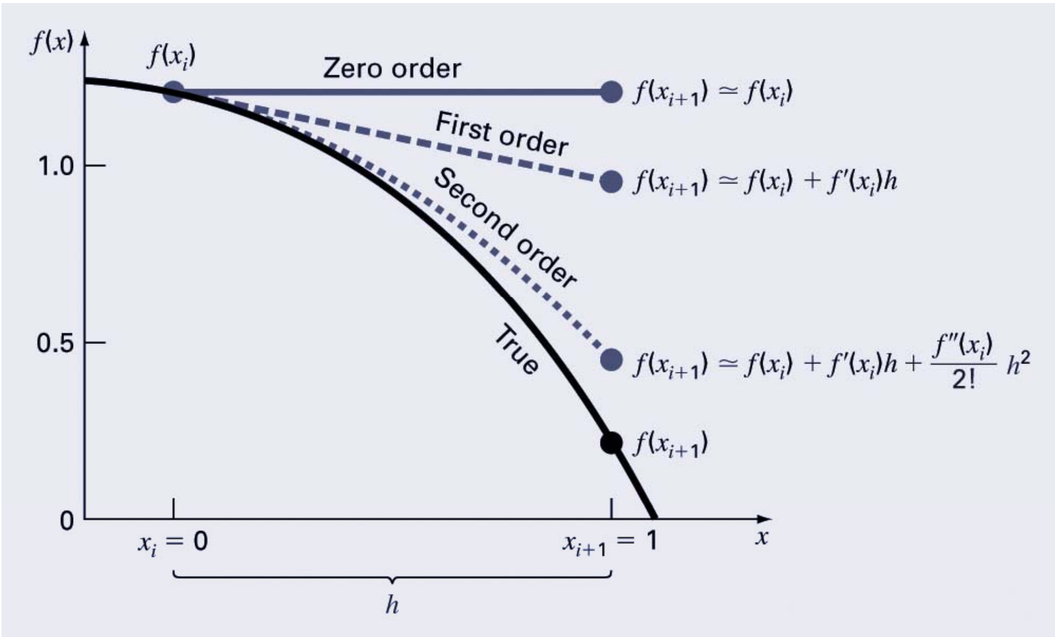
Another example: using discrete steps in the solution of a differential equation

$$\frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

How to quantify truncation errors

<u>The Taylor series:</u> any smooth function can be approximated as a polynomial

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



remainder term is of the order h^{n+1}

- The more terms are used, the smaller the error
- smaller the spacing, smaller the error for a given number of terms.

Truncation errors

Problem 1. $$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

Use the Taylor's series to approximate x_{i+1} using h = 1 and x_i = 0. Compare your results and the truncation error for 0-th, 1st and 2nd order approximations.

Problem 2.

The Mclaurin series expansion for *sinx* is the following:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

Starting from sinx = x, add terms one at a time to estimate sin(π/3). Compute the true, approximate % relative errors at each step after a new term is added.

Truncation errors

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

The first order Taylor series is used to approximate the derivative (e.g. the bungee jumper problem)

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

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \Longleftarrow \frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

This is also called <u>"Forward Difference"</u> because it uses the data at *i* and *i+1* to estimate the derivative

Numerical Differentiation

There are also backward and centered difference approximations, depending on the points used:

Forward:

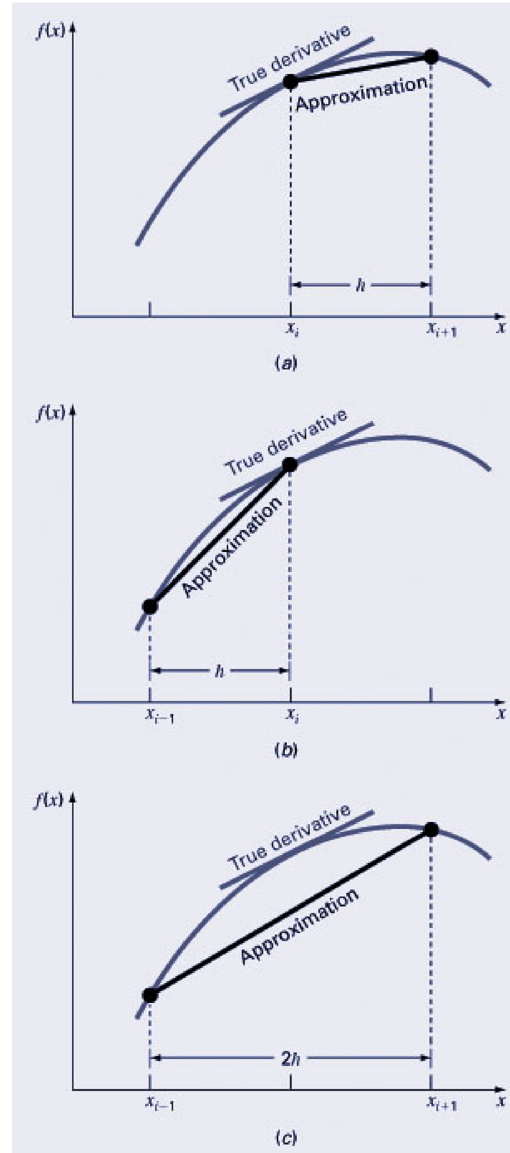
$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

Backward:

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

Centered:

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



Problem: T.E. in numerical diff.

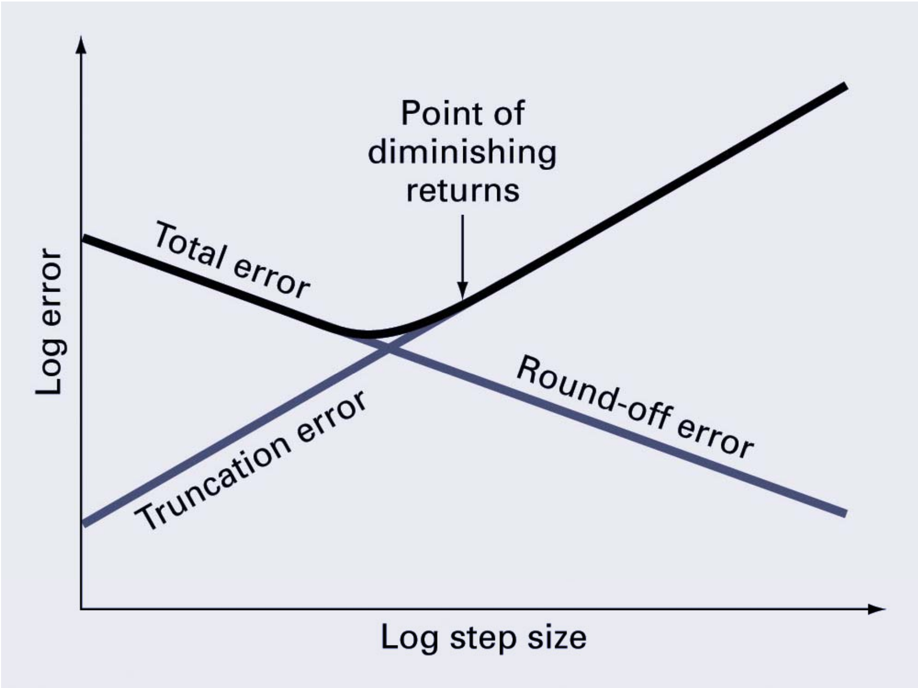
$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

Using forward, backward and centered differences, calculate the derivative of f(x) using step size of h = 0.5 and h = 0.25. Compute the truncation errors for each of the methods applied.

Total Numerical Error

The *total numerical error* is the summation of the truncation and roundoff errors.

The truncation error generally *increases* as the step size increases, while the roundoff error *decreases* as the step size increases - this leads to a point of diminishing returns for step size.



Other Errors

Blunders - errors caused by malfunctions of the computer or human imperfection.

Model errors - errors resulting from incomplete mathematical models.

Data uncertainty - errors resulting from the accuracy and/or precision of the data.