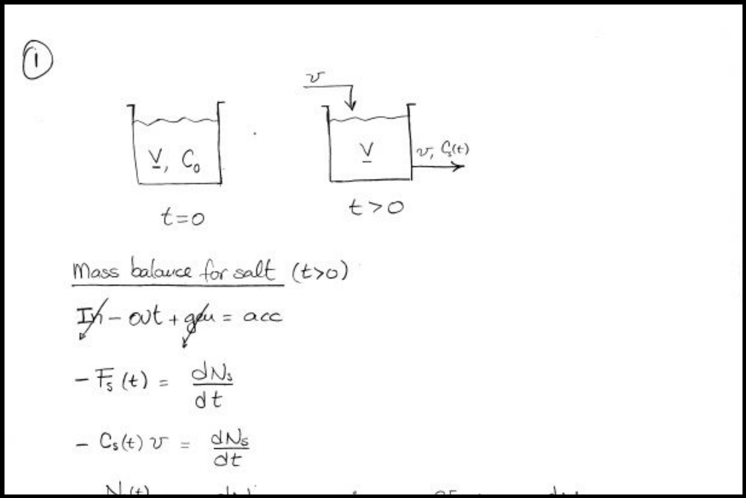
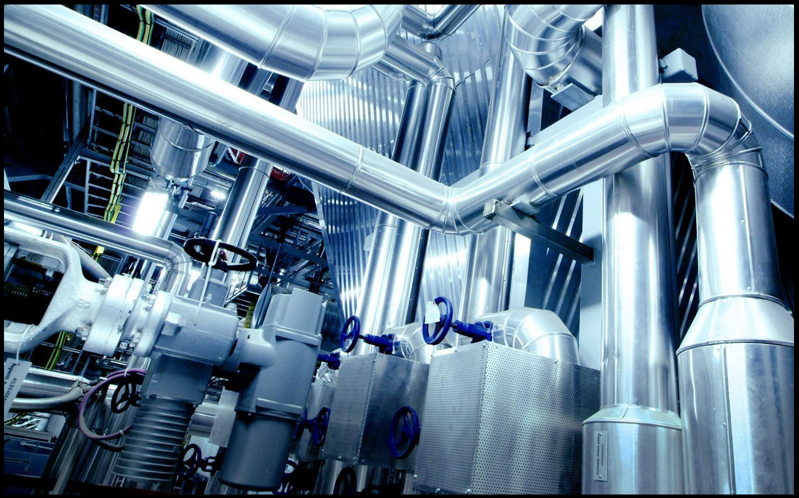


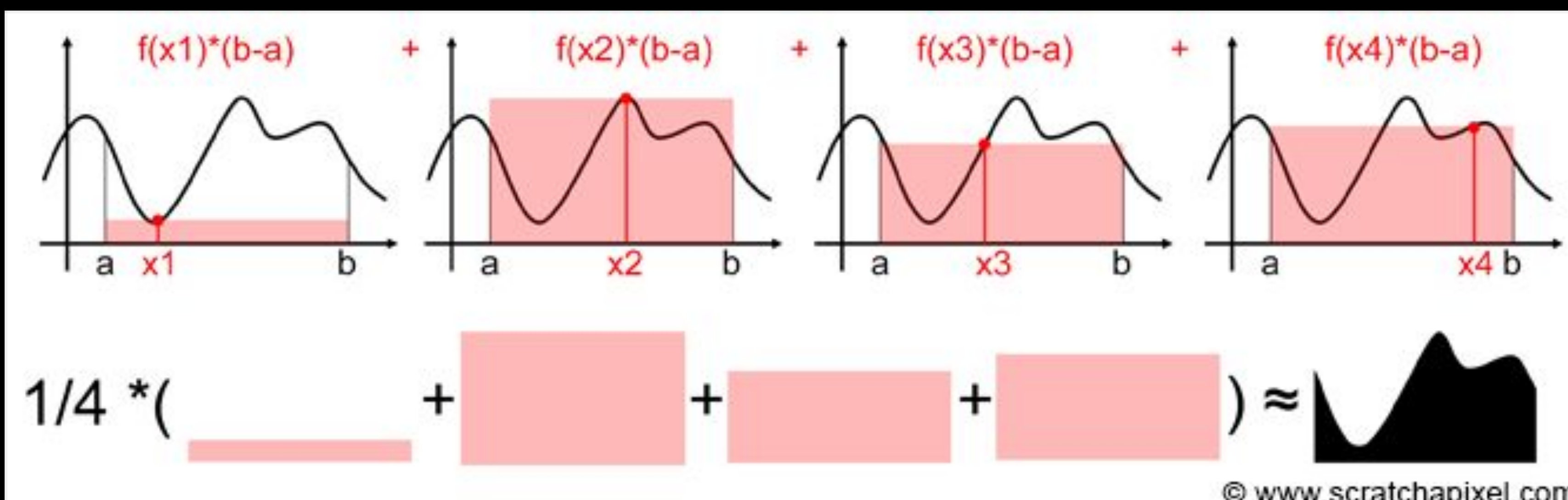


ChE352
Numerical Techniques for Chemical Engineers
Professor Stevenson

Lecture 17

Exams = Monte Carlo integration

- I can't ask you about every topic
- I can't use a predictable sequence
- Therefore, I must use a pseudo-random subset...



Midterm rules

- You will have 90 minutes.
- You may use only these resources: the two course textbooks (F&B and PNM), my slides, your own notes, your group's HWs, and Colab
- You may use laptops and/or tablets, but not phones.
- The exam will be graded based on your blue book. Show all your work clearly in your blue book and draw a box around each answer.
- If you have a question, raise your hand.

Midterm rules

- You will have 90 minutes.
- You may use only these resources: the two course textbooks (F&B and PNM), my slides, your own notes, your group's HWs, and Colab
  - You can use my graded pdfs of your group's HWs, too
  - You can be on the Wifi in order to get to these resources like PNM and Colab, but no general internet usage
  - No using the AI features in Colab

Linear algebra

5. (10 points) Define the following in one sentence (or less) each:

A. Linear operator

B. Non-singular matrix

C. Positive definite matrix

D. Dot product

E. Eigenvector

Linear algebra

5. (10 points) Define the following in one sentence (or less) each:

A. Linear operator

An operation transforming one vector into another which can be expressed as matrix multiplication by some matrix

B. Non-singular matrix

A matrix which has an inverse, aka can be solved in a linear system Ax = b. Also valid: full-rank and square, determinant != 0.

C. Positive definite matrix

A matrix for which the expression $x^T P x$ always gives a positive scalar for any nonzero column vector x of the correct length

D. Dot product

The operation of multiplying all corresponding entries of two vectors x1, x2 and summing all the results to produce a single scalar

E. Eigenvector Common mistake: not saying eigenvector depends on the matrix

A characteristic vector x of a given matrix H satisfying the expression Hx = ax for some scalar a (the corresponding eigenvalue), meaning that the vector x does not change direction when multiplied by the matrix H, it is only scaled by a constant (the eigenvalue).

Optimization formalism

$$\boxed{\left(P_1\right) \equiv \begin{array}{ll} \text{minimize:} & f\left(x\right) \\ \text{subject to:} & x \in \Omega \subset \mathbb{R}^n \end{array}}$$

Objective function

Constraint set

Where: $f:\mathbb{R}^n \to \mathbb{R},\ \Omega = \left\{x : g\left(x\right) \le 0,\quad h\left(x\right) = 0\right\}$

Or abbreviated as:

Optimal value

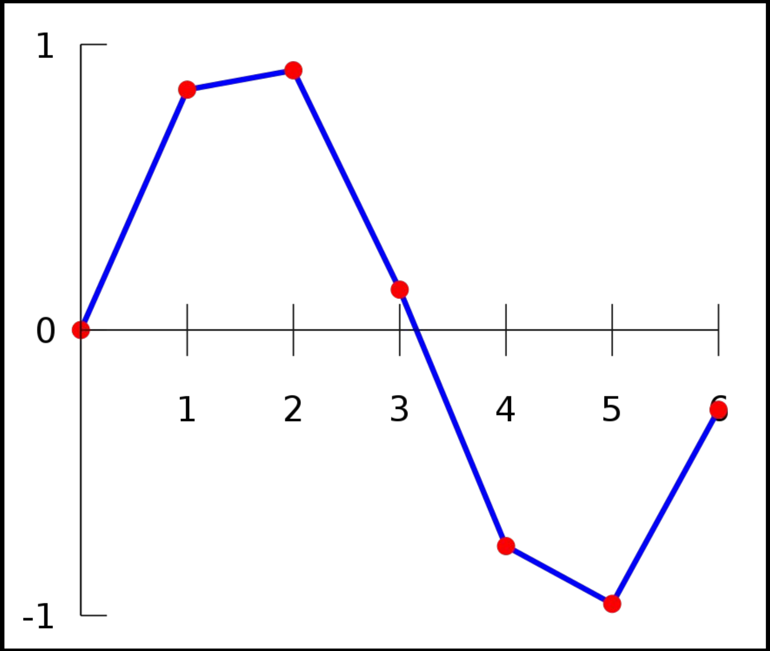
Problem name

$$\left(P_1\right) \equiv \begin{array}{ll} \min & f\left(x\right) \\ \text{s.t.} & x \in \Omega \end{array} \quad OR \quad \boxed{\min_{x\in\Omega} \quad f\left(x\right)} = v^*$$

- Which values are changing over the course of optimization? How?
- Which values are not changing?

Interpolation/regression Methods

- Linear regression
- Polynomial regression
- Lagrange polynomial interpolation
- Piecewise linear interpolation
- Cubic spline interpolation



← Which method is this?

When would you use each of these methods?

Numerical derivatives & integrals come from approximation methods

- You have learned some methods for approximating a function f(x) based on individual data points (Examples?)
- How can you use these approximation methods to get numerical derivatives and integrals?

Numerical derivatives & integrals come from approximation methods

- You have learned some methods for approximating a function f(x) based on individual data points (Examples?)
- We used functions like polynomials which have easy analytic derivatives & integrals
- We can also use these to estimate the derivative & integral of the true function f(x)

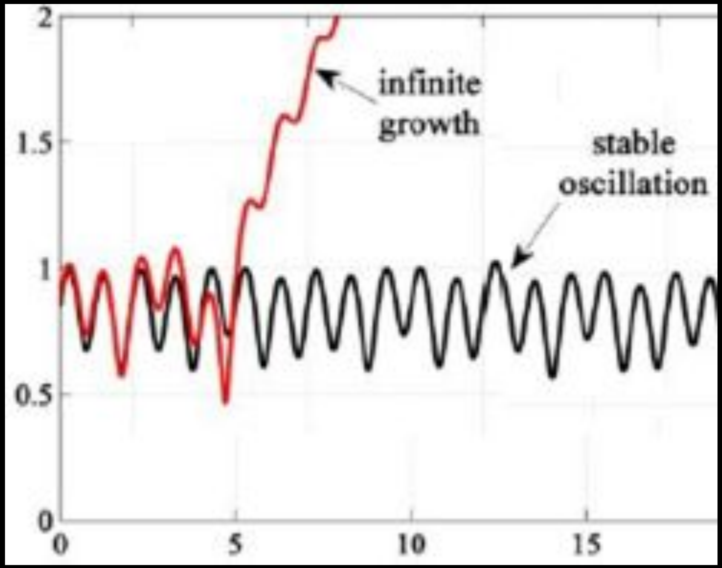
Root-finding methods: what & why?

	Bisection	Newton
Convergence?		
Always converges?		
Special conditions?		
Good when?		

Root-finding methods: what & why?

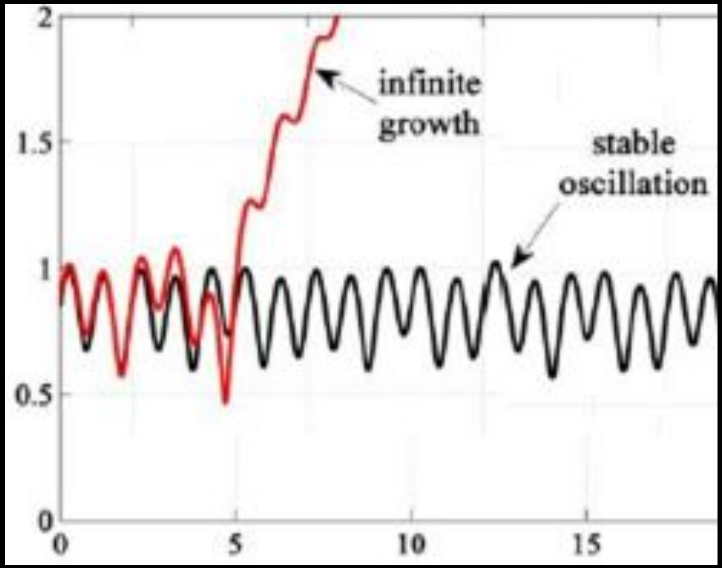
	Bisection	Newton
Convergence?	Linear	Quadratic
Always converges?	Yes	**No**, if bad p_0 or if we hit $f'(p_n) \approx 0$
Special conditions?	Need the bounds a, b	Need a guess p_0, need to have $f'(x)$
Good when?	We need stability	We need accuracy & speed

Sensitivity analysis



- How can you tell if an approximation is **stable**?

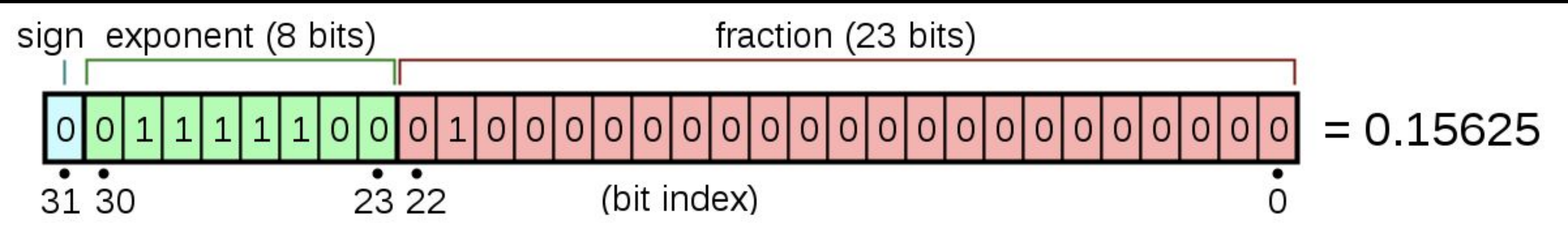
Sensitivity analysis



- How can you tell if an approximation is **stable**?
- Try **small** perturbations in input ("small" depends on the problem at hand)
- If the output changes **significantly** (as defined by the problem at hand), you have **instability**

What is floating point?

- Computer math is almost always floating point
- Like scientific notation with powers of 2 only



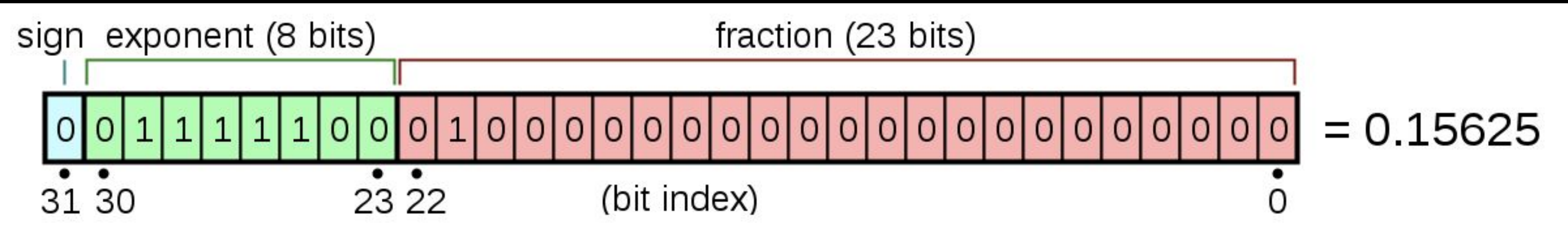
- Not every real number can be represented

How many decimal digits can we store in 23 bits?

What numbers can’t be represented?

What is floating point?

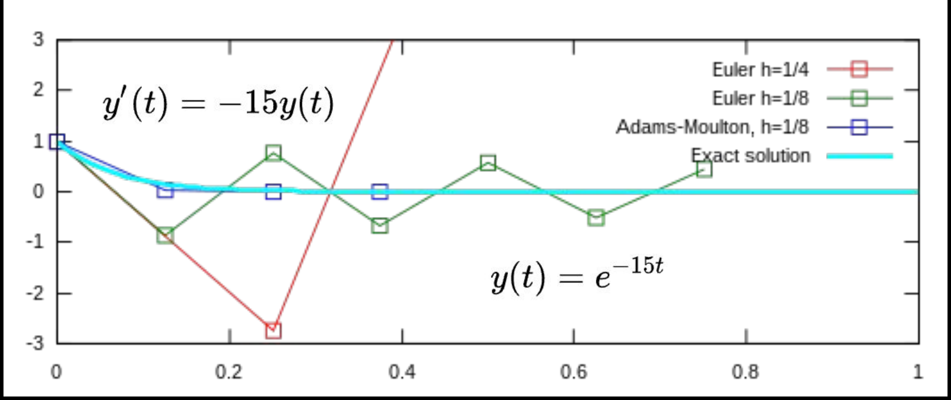
- Computer math is almost always floating point
- Like scientific notation with powers of 2 only



- np.float32 holds ~7 decimal digits
- np.float64 holds ~16 decimal digits
- Not every real number can be represented
- Too big = overflow, too small = underflow
- Only binary fractions (no exact 1/3, 1/5, etc)

Review: IVPs

- When is an IVP like an integral?
  - *Only when y' = f(t)*, no y
  - Then y = ∫ f(t), can solve with quadrature
- In general, y' = f(t, y)
  - We know t exactly, but y (after y_0) is an estimate, so y' = f(t, y) is also an estimate
  - Subject to accumulating errors, so be careful



Review: IVP systems

- Same idea, but u' = f(t, u) where u is a vector

Euler for systems (w = our estimate for u):

$$w_{i,j+1} = w_{i,j} + hf(t_j, w_{i,j})$$

Which index i, j is the timestep? What is the other index?

- Solving is similar to 1-D IVPs, but more values to keep track of
- Vector math rules apply: vector + vector = vector, scalar * vector = vector, etc

Setting up IVP systems

- Every u_i corresponds to an element $u_i' = f(t, u)$

$$
\begin{aligned}
&t_0 \le t \le t_{\max} \\
&\frac{du_1}{dt} = f_1\left(t, u_1, u_2, \ldots, u_m\right), \quad u_1\left(t = t_0\right) = a_1 \\
&\frac{du_2}{dt} = f_2\left(t, u_1, u_2, \ldots, u_m\right), \quad u_2\left(t = t_0\right) = a_2 \quad \Rightarrow \\
&\vdots \\
&\frac{du_m}{dt} = f_m\left(t, u_1, u_2, \ldots, u_m\right), \quad u_m\left(t = t_0\right) = a_m
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
&\frac{du(t)}{dt} = f\left(t, u(t)\right), \\
&u\left(t = t_0\right) = a, \\
&t_0 \le t \le t_{\max} \\
&u : \mathbb{R} \to \mathbb{R}^m, \\
&f : \mathbb{R}^{m+1} \to \mathbb{R}^m, \\
&t \in \mathbb{R}, \quad a \in \mathbb{R}^m
\end{aligned}
}
$$

Vector function

Vector function

Review: Euler for IVP systems

$$\boxed{t=z} \quad u_1(t)=C_A(z), \quad u_2(t)=C_B(z), \quad u_3(t)=C_C(z)$$

$$\boxed{u=\begin{bmatrix} C_A & C_B & C_C \end{bmatrix}^T}$$

What is the Euler step for this system?

$$f_1(t,u)=-2k_1C_A{}^2=-2k_1u_1{}^2$$

$$f_2(t,u)=-k_2C_B+k_1C_A{}^2=-k_2u_2+k_1u_1{}^2$$

$$f_3(t,u)=k_2C_B=k_2u_2$$

$$\boxed{f(t,u)=\begin{bmatrix} -2k_1u_1{}^2 & -k_2u_2+k_1u_1{}^2 & k_2u_2 \end{bmatrix}^T}$$

$$\boxed{t_0=0} \quad \boxed{a=\begin{bmatrix} C_A^o & 0 & 0 \end{bmatrix}^T}$$

Review: Euler for IVP systems

f(t, u) for this IVP: $\boxed{f(t,u)=\begin{bmatrix}-2k_1u_1^{\,2} & -k_2u_2+k_1u_1^{\,2} & k_2u_2\end{bmatrix}^T}$

Euler step definition: $w_{i,j+1} = w_{i,j} + hf(t_j, w_{i,j})$

Euler step for this IVP:

$$w_{1,j+1} = w_{1,j} + h(-2k_1 w_{1,j}^2)$$

$$w_{2,j+1} = w_{2,j} + h(-k_2 w_{2,j} + k_1 w_{1,j}^2)$$

$$w_{3,j+1} = w_{3,j} + hk_2 w_{2,j}$$

Review: higher-order IVPs

- Often we know only a higher-order derivative of our desired function y, like $y'' = f(t, y, y')$

$t_0 \le t \le t_{\max}$

- And initial conditions y_0 & y'_0

$$y^{(m)}(t) = \frac{d^m y}{dt^m} = f\left(t, y(t), y'(t), \ldots, y^{(m-1)}(t)\right)$$

- Treat every derivative of y as an element of u in an IVP system: $u_1 = y$, $u_2 = y'$
- $du_2/dt = f(t, u_1, u_2)$ $\quad$ $du_1/dt = u_2$