Introduction to Numerical Analysis
18.330

Lecture 1:
Introduction
Quadrature

Laurent Demanet

January 31, 2021

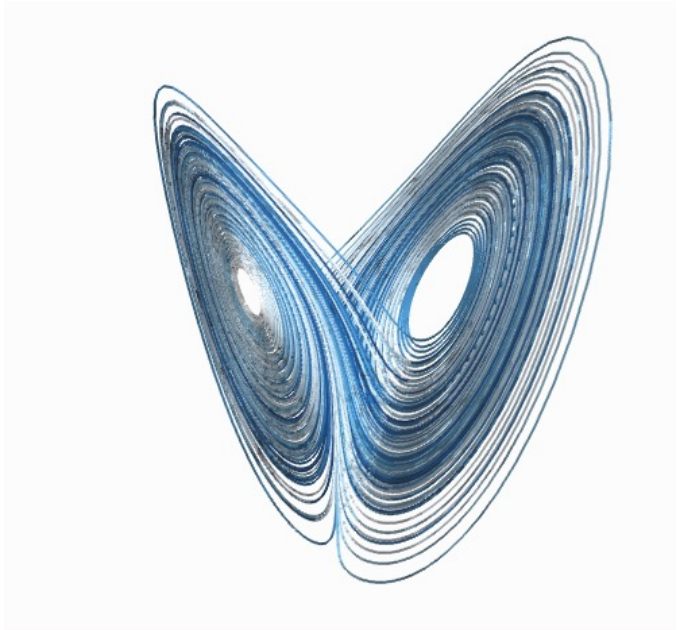
What is Numerical Analysis?

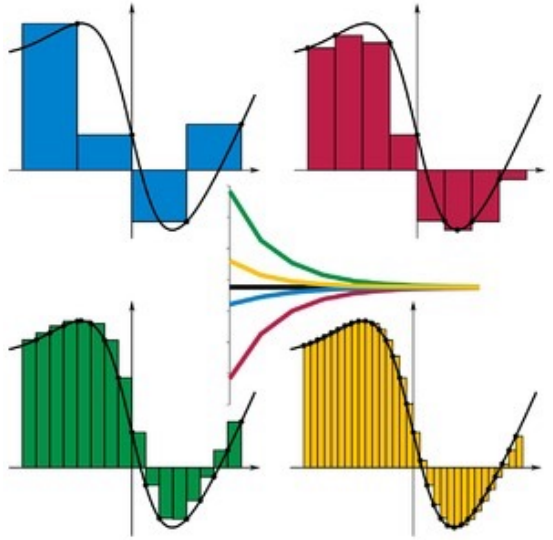
The study of algorithms that implement continuous functions as discrete objects in the computer, in order to:

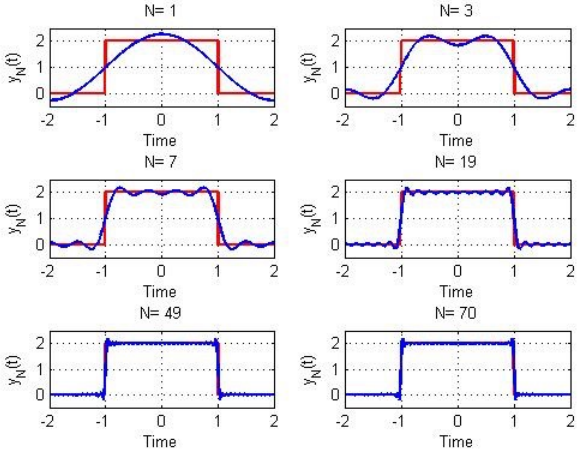
- Perform operations -- think calculus
- Solve equations (algebraic, differential)
- Solve optimization problems, including machine learning
- Infer models from observations

Math questions about those algorithms:

- Effectiveness – do they work well?
- Efficiency – are they fast?



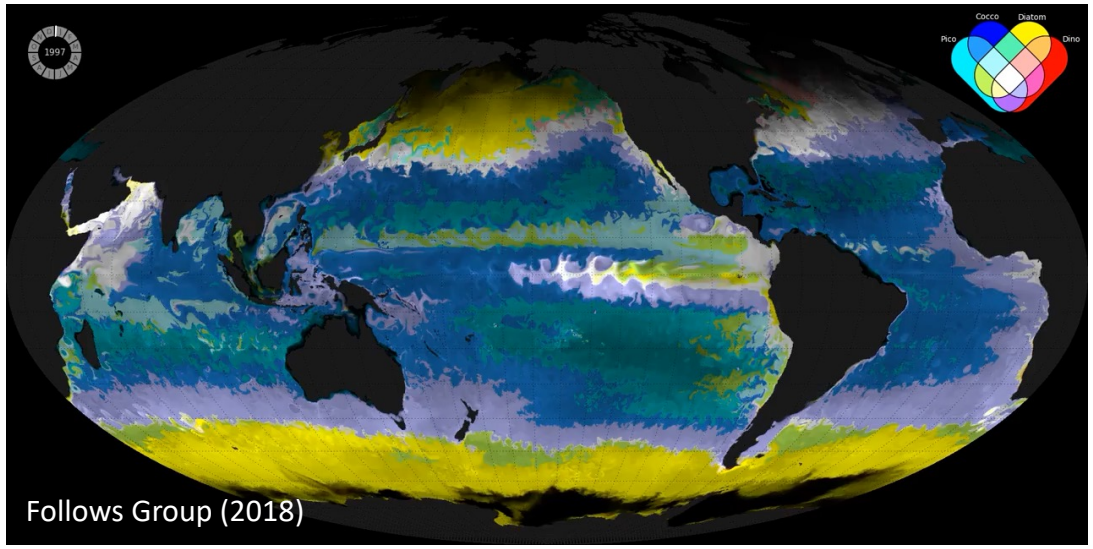




The real-life context for numerical analysis is Computational Science and Engineering (CSE).

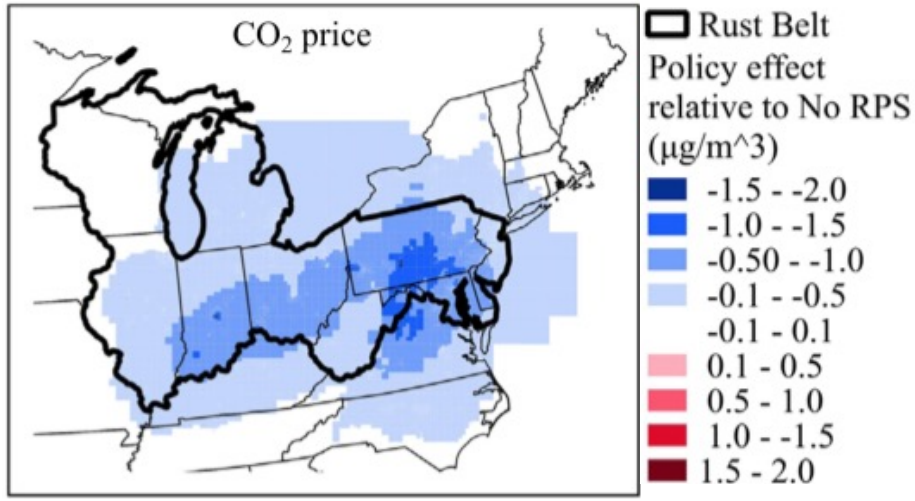
The objective of CSE is
to develop and apply
computational methods for:

Scientific discovery



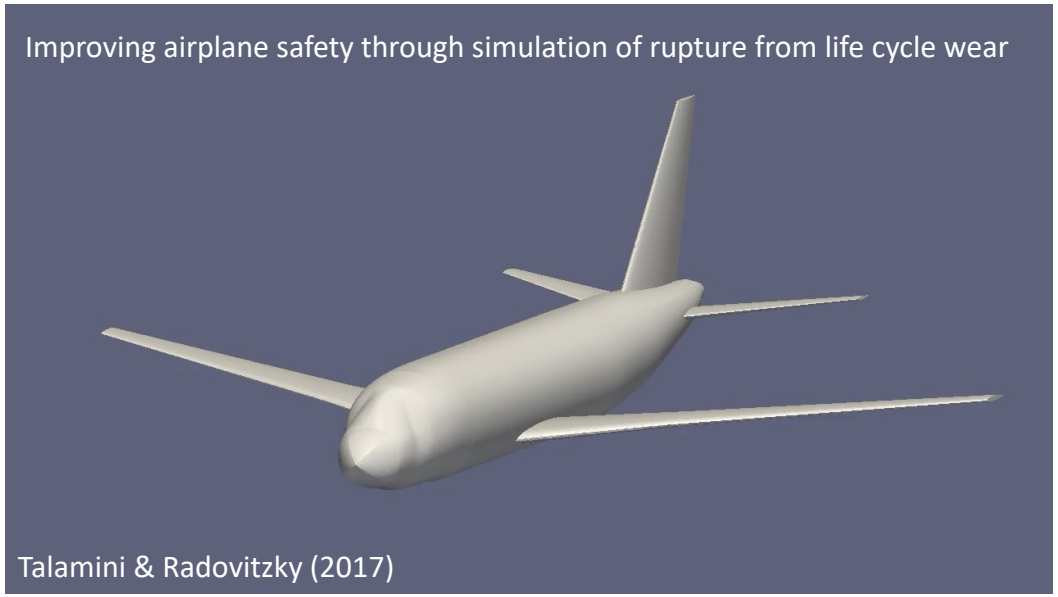
Follows Group (2018)

Decision-making for societal challenges



Selin Group (2019)

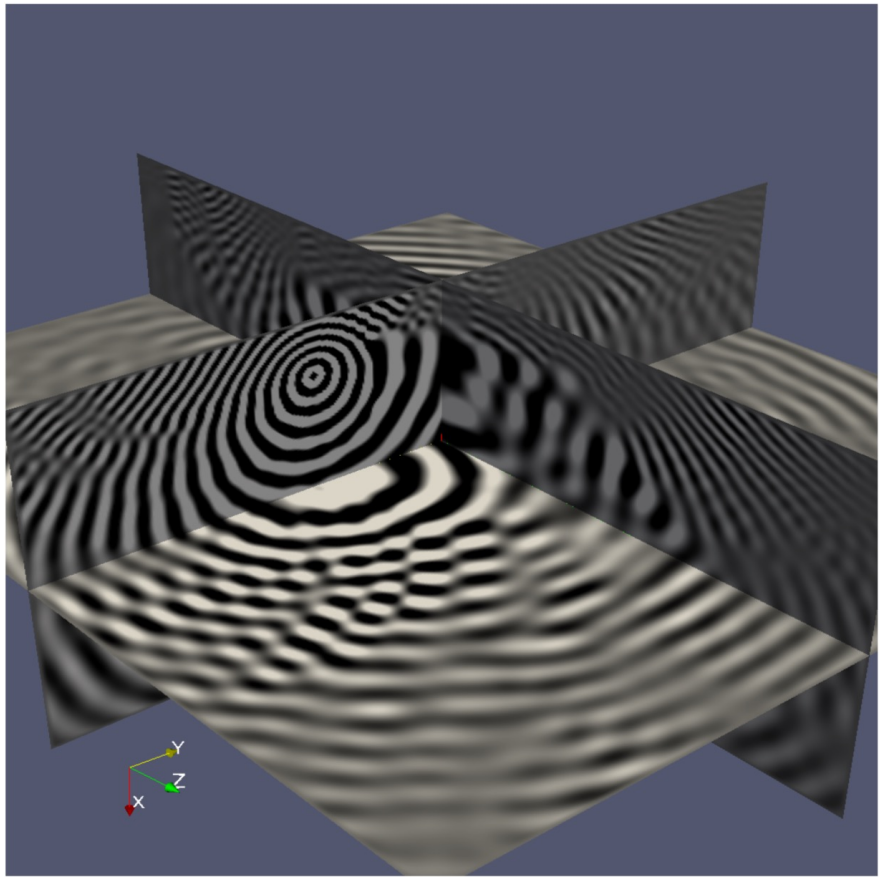
Innovation in engineering & technology

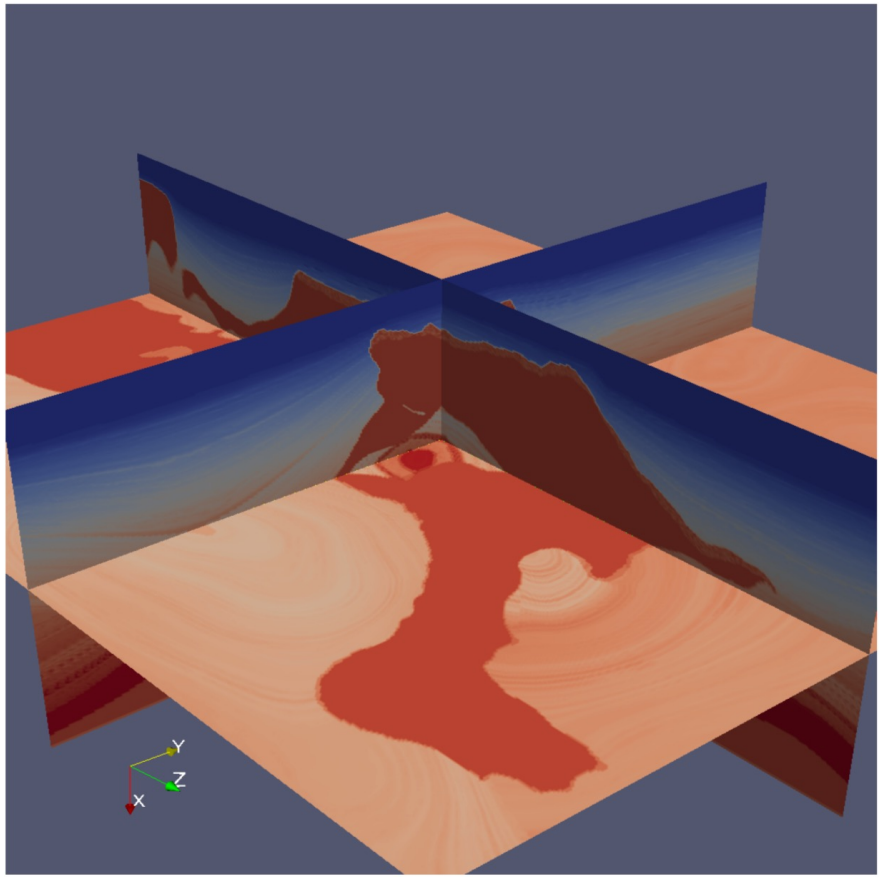


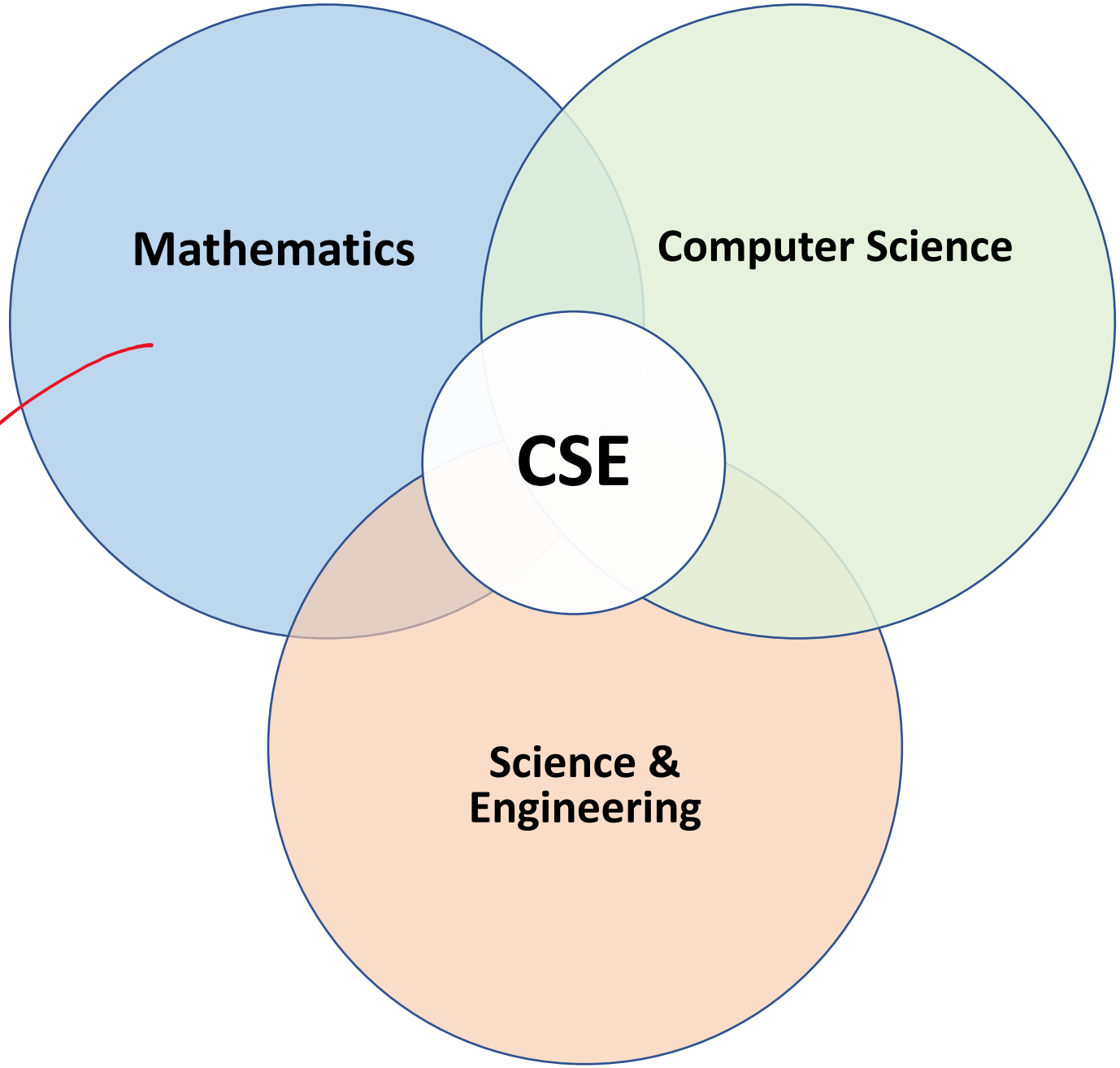
For instance, my group works on seismic imaging:

How do elastic waves propagate inside the Earth?

Can that help build a 3D map of the subsurface?







Numerical analysis is (much of) the math part of CSE

This class

Prerequisites: Calculus at the level of 18.01, 18.02, and 18.03.

Helps but not prereq: Linear algebra at the level of 18.06, programming at the level of 6.0002 or 16.0002 / 18.0002.

Live participation

Everything else on canvas, except Q&A on piazza

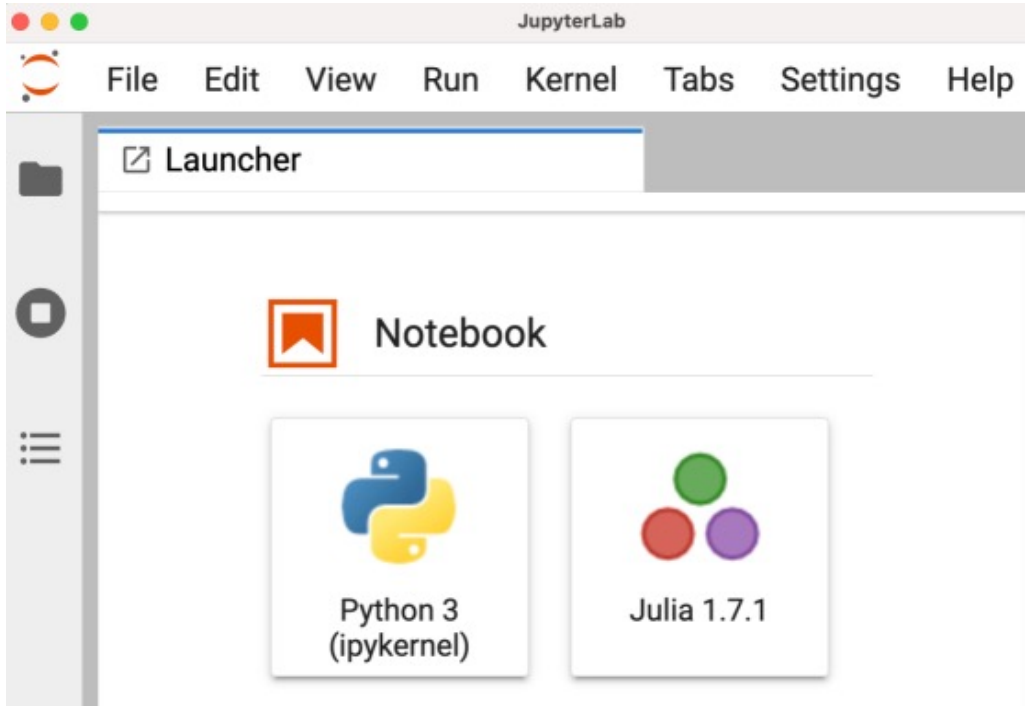
Zap me for canvas

Evaluation: 50% homework, 20% in-class midterm, 30% in-class final.

Homework:
No late copy will be allowed. The lowest problem set score will be dropped. Collaboration is allowed, but the codes and copies you turn in must be original and written by you. We encourage psetpartners.mit.edu for forming study groups.

Programming language: your choice!

- Julia with IJulia notebook (in JupyterLab)
- Python with Spyder or JupyterLab
- Matlab



Resources

Contact us: ldemanet@mit.edu or piazza (anonymous or not, PM or public)

Office hours on Tuesdays 5-6pm, 2-247

Zoom OH possible, request ahead of time.

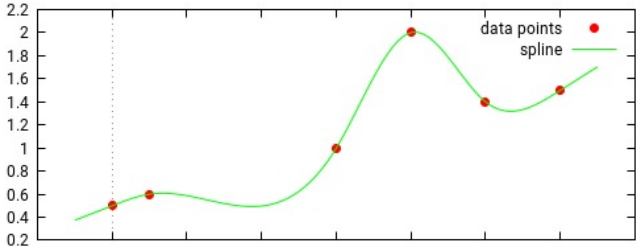
Main reference: PDF notes. Code uploaded through the term.

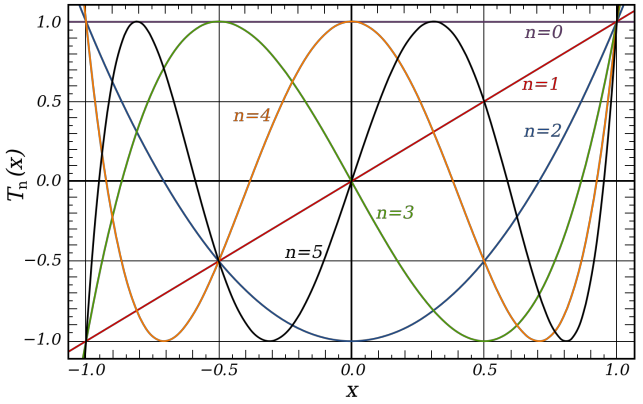
Will add links to (completely optional) books and other notes.

Zap me for canvas

Topics:

- Sum rules for integrals, difference rules for derivatives
- Interpolation, splines, and a second look at sum/difference rules
- Root finding [tentative]
- Numerical methods for initial-value problems (ODE)
- Numerical methods for boundary-value problems (still ODE)
- Fourier transform, Fourier series, Shannon sampling theory
- Bandlimited interpolation, spectral methods
- Least-squares approximation [tentative]

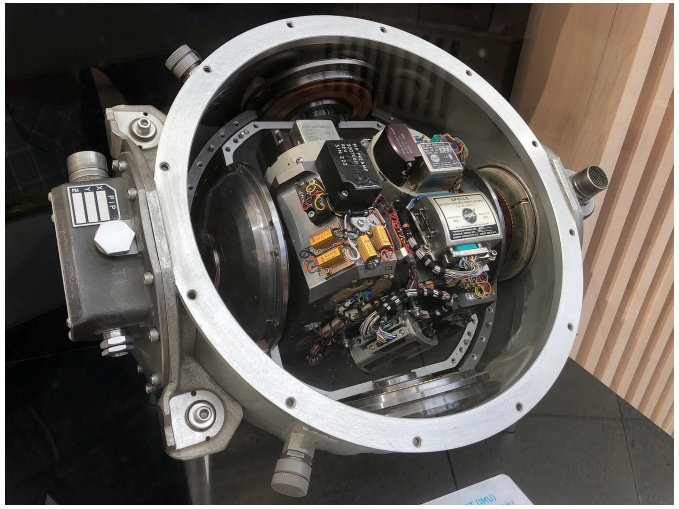


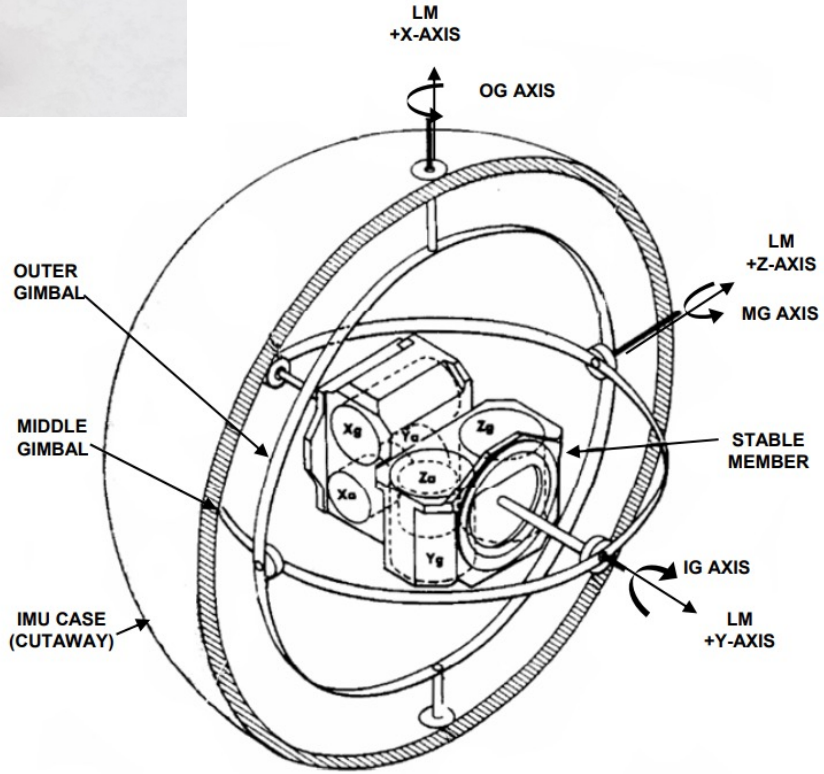


Today: Sum rules for integrals (quadrature)

Example 1: how much snow accumulated in the Boston area?

Example 2: find the position of an aircraft or spacecraft

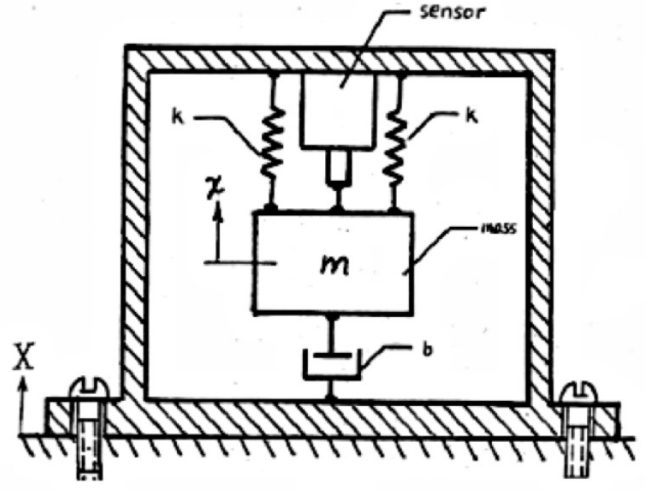




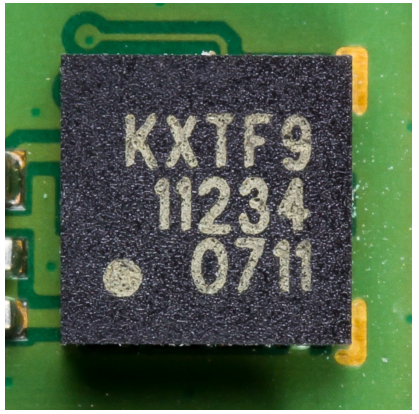
Example 2: how to find the position of an aircraft or spacecraft (without GPS)?

Inertial navigation system (INS): integrate acceleration data twice to get position

Accelerometers measure the deformations of 3 damped springs



Credit: Pocketlab



Credit: Kionix

Correct for orientation and gravity using gyroscopes/compass.

Also used in phones (tilt) and video game controllers

Three kinds of errors:

- Drift error: Accumulation of measurement errors and inaccurate orientation/gravity corrections. Ex: 50m lost over 10 sec (cheapest) to 20 mins (fancy).

- Truncation error: Use of a finite sum to approximate an integral

- Round-off error: Use of floating-point numbers to represent real numbers

Numerical analysis quantifies all three.