Numerical Analysis / Scientific Computing

CS450 / CSE 401 / ECE 491 / MATH 450

Andreas Kloeckner

→ 4318 Siebel

Fall 2022

Outline

What's the point of this class?

'*Scientific Computing*' describes a family of approaches to obtain approximate solutions to problems *once they've been stated mathematically*.

Name some applications:

- Numerical simulation (e.g. engineering)
  - ↳ differential equations
- machine learning/stats
  - statistical models
  - optimization
- Image and audio processing

What do we study, and how?

Problems with real numbers (i.e. *continuous* problems)

π $\rightsquigarrow$ how $\mathbb{R}$ go into a computer

What's the general approach?

- how to represent the solution
- existence / uniqueness

What makes for *good* numerics?

How good of an answer can we expect to our problem?

- how to even measure sol. quality
- given that we can't even represent numbers (!)

How fast can we expect the computation to complete?

- math. statement (discrete)
- method
- algorithm

- cost
  - $O(n^3)$ asymptotic
  - efficiency
- efficient ?

Implementation concerns

How do numerical methods *get implemented*?

- like anything in computing: lies
- tools / languages: "tower of abstractions"
  - ↳ methods
  - ↳ robustness / guarantees

Class web page

https://bit.ly/cs450-f22

- Assignments
  - HW1!
  - Pre-lecture quizzes
  - In-lecture interactive content (bring computer or phone if possible)
- Textbook
- Exams
- Class outline (with links to notes/demos/activities/quizzes)
- Discussion forum
- Policies
- Video

Programming Language: Python/numpy

- Reasonably readable
- Reasonably beginner-friendly
- Mainstream (top 5 in 'TIOBE Index')
- Free, open-source
- Great tools and libraries (not just) for scientific computing
- Python 2/3? 3!
- `numpy`: Provides an array datatype
  Will use this and `matplotlib` all the time.
- See class web page for learning materials

Demo: Sum the squares of the integers from 0 to 100. First without numpy, then with numpy.

Supplementary Material

- Numpy (from the SciPy Lectures)
- 100 Numpy Exercises
- Dive into Python3

Sources for these Notes

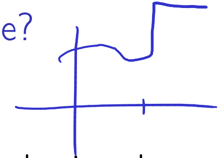
- M.T. Heath, Scientific Computing: An Introductory Survey, Revised Second Edition. Society for Industrial and Applied Mathematics, Philadelphia, PA. 2018.
- CS 450 Notes by Edgar Solomonik
- Various bits of prior material by Luke Olson

Open Source <3

These notes (and the accompanying demos) are open-source!

Bug reports and pull requests welcome:
https://github.com/inducer/numerics-notes

What problems *can* we study in the first place?



To be able to compute a solution (through a process that introduces errors), the problem...

- existence
- uniqueness
- depend continuously on the inputs

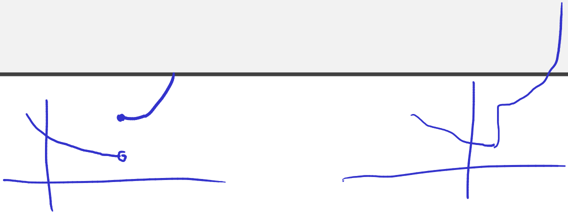
If it satisfies these criteria, the problem is called *well-posed*. Otherwise, *ill-posed*.

→ otherwise trouble if inputs are perturbed (say by rounding error)

Dependency on Inputs

We excluded discontinuous problems–because we don't stand much chance for those.
. . . what if the problem's input dependency is just *close to discontinuous*?

- sensitive output to input
- absence of error amplification



Approximation

$$f(x+h) \approx \sum_{i=0}^{n} \frac{f^{i}(x)}{i!} h^{i}$$

When does approximation happen?

Before computation:
- modeling
- measurement

During computation:
- rounding
- truncation / discretization

Demo: Truncation vs Rounding [cleared]

Example: Surface Area of the Earth

Compute the surface area of the earth.
What parts of your computation are approximate?

$4\pi r^2$

Measuring Error

How do we measure error?
Idea: Consider all error as being *added onto* the result.

$$\text{Relative error} = \frac{\text{abs. error}}{\text{mag. of true value}} = \frac{|x - \tilde{x}|}{|x|}$$

$$\text{abs. error} = \text{mag. of true answer} - \text{approx. answer} = |x - \tilde{x}|$$

Recap: Norms

$$\begin{pmatrix}1\\3\\5\end{pmatrix}$$

↑ vector

What's a norm?

$$f: \mathbb{R}^n \longrightarrow \mathbb{R}_0^+$$

$$\|\vec{x}\|$$

Define *norm*.

$\|\cdot\|: \mathbb{R}^n \to \mathbb{R}_0^+$ is called a norm iff

$$\|\vec{x}\| > 0 \Leftrightarrow \vec{x} \neq \vec{0}$$

$$\|\gamma \vec{x}\| = |\gamma| \, \|\vec{x}\|$$

$$\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$$

Norms: Examples

Examples of norms?

$$\left\| \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \right\|_2 = \sqrt[2]{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Demo: Vector Norms [cleared]

Norms: Which one?

Does the choice of norm really matter much?