Honors Numerical Analysis

Jan 24

Overview / Logistics:

contact info, office hours, recitation, etc.

- check syllabus
- cims.nyu.edu/~oneil/na22

- Compare with Math 252 (more programming, more mature mathematics)

- Grading: (overall numerical average ⟹ Letter grade)
  - 30% homework (6 assignments, 5% each)
  - 30% Midterm
  - 40% Final

  Midterm and Final: Written, in class.

  Discussion on HW OK, not collaboration.

- Programming: There will be programming!

  Language: <u>Julia</u> only via NYU JupyterHub (more details, config to come)

  Programming advice/tips will be given throughout course.

- Motivating Examples:
  - evaluate cosine
  - evaluate the square root

Demo compute $\cos(x)$, $x = 2\pi, 4\pi, \ldots 2^n\pi$

What is going on?

Your computer doesn't know how to evaluate "cos", it can only add/multiply/subtract/divide.

$\Rightarrow$ Need an <u>algorithm</u> for computing $\cos(x)$.

Taylor expand:

$$\cos(x) = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$$

If x is <u>small</u>, only a few terms are needed for very good approximation.

If x is <u>big</u>, first map it to $[0, 2\pi)$ since

$$\cos(x) = \cos(x + 2\pi n)$$

In this mapping, for large n, information is <u>lost</u>:

$$x = 12{,}345{,}678{,}901{,}234{,}5\ldots \approx 10^{14}$$

(computer stores "16 significant digits")

$$y = x - 2 \cdot \pi \cdot \underbrace{1964875822956 7}_{n}$$

$$= 5.640625 \in [0, 2\pi)$$

$$\cos(x) = 0.8010052709\ldots$$

$$\cos(y) = 0.8005641268\ldots$$

The problem is that computing $y = x - 2\pi n$ cancels most of the "significant digits"

$\Rightarrow$ catastrophic cancellation

We must understand how computers store numbers and do arithmetic.

Demo 2   Given $c > 0$, how do we compute $x = \sqrt{c}$?

Again, your computer can only do $+ - \times \div$! We need an algorithm...

Ancient Babylonian Method: (See Wikipedia)

Equivalent to solving the equation

$$\underbrace{x^2 - c}_{f(x)} = 0$$

$f(x) = 0$ ← nonlinear equation solving - big topic in NA, very important

Babylonian Method: Guess $x_0 \approx \sqrt{c}$.

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

Obviously, if $x_k = \sqrt{c}$, then

$$x_{k+1} = \frac{1}{2}\left(\sqrt{c} + c/\sqrt{c}\right)$$

$$= \frac{1}{2}\left(\sqrt{c} + \sqrt{c}\right)$$

$$= \sqrt{c}$$

, so $\sqrt{c}$ is a fixed point of this iteration.

Questions

This is numerical analysis.
- Why does this iteration converge?
- If it converges, does it give the right answer?
- How fast?
- Where does it come from?

Implement Demo in Julia

Demo 3 : Finite difference ?

Pontificate on Programming for Science

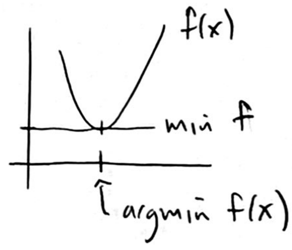
- text files
- editors : Emacs/Vim , Vscode , etc.
- terminal (basic commands)
- Mac / Linux / Windows (subsystem Linux)
- Languages : C/Fortran vs. Matlab/Julia/Python
- JupyterHub
  - ↳ re program code in JupyterHub.

Additional Math Areas that we will address in this course :

- Nonlinear systems ⟶ $\vec{f}(\vec{x}) = \vec{0}$
- Linear algebra
  - solve $A\vec{x} = \vec{b}$
  - compute $A = QR$, SVD, LU
  - Find $A\vec{v} = \lambda\vec{v}$
  - $\underset{\vec{x}}{\operatorname{argmin}} \| A\vec{x} - \vec{b} \|_2$ Least squares
- Optimization

$$y = \underset{x}{\operatorname{argmin}} \ f(\vec{x})$$

↗



- Interpolation, Function approximation
  - Given data, find function that interpolates it, or well approximates it

- Integration

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{N} f(x_i)\, w_i$$

how do we find "the best" nodes and weights

- A "Fast Algorithm": Fast Fourier Transform

Compute $\hat{f}_k = \sum_{j=0}^{n-1} f_j e^{2\pi i jk/n}$ for $k = 0, \dots, n-1$

  - Has ubiquitous uses in EE, numerical analysis, everywhere

- Monte Carlo methods
  - Main idea: approximate $\int_a^b f(x)\,dx$ as $\frac{1}{N} \sum f(x_i)$

"random" number on $[a,b]$.

- Other topics? Suggestions?

Numerical differential equations?