Numerical Methods

CS 357 - Fall 2016

Introduction

Numerical Methods: What?

- ‘Numerical’?
- ‘Method’?

Accuracy

- Why might a numerical method **not give the right answer?** (i.e. be inaccurate)
  - Because (unlike in the special cases that math has taught you), mostly we *can't write down the answer*. Not in a finite amount of space anyway. And a computer *is* finite.

Demo: Waiting for 1

Numerical Experiments

Model:

- Small-scale behavior easy to describe
- Large-scale behavior desired, but hard to understand

Demo: Brownian Motion

Numerical Experiments

- What are we going to want to know about a numerical experiment?

Class web page

`bit.ly/cs357-f16`

- Assignments
  - HW0!
  - Pre-lecture quizzes
  - In-lecture interactive content (bring computer or phone if possible)
- Exams
- Class outline q(with links to notes/demos/activities/quizzes)
- Scribbles
- Virtual Machine Image
- Piazza

- Policies
- Video
- Interactive Questions
- Calendar
    - Office Hours

In-class activity: Complexity of Matrix-Matrix Multiplication

Recap: Understanding Asymptotic Behavior, *O*(·) *Notation*

Demo: Cost of Matrix-Matrix Multiplication

- ○ Can we say anything exact about our results?
- ○ How do we say something exact without having to predict individual values exactly?

$\text{Time}(n) = C \cdot n^3$ ✗

measurement noise:
not true

$O(\cdot)$ notation

$$\text{Time}(n) = O(g(n))$$

$\Leftrightarrow$ There is a constant C so that

$$\text{Time}(n) \leq C \cdot g(n)$$

For mat-mat:

$$\text{Time}(n) = O(n^3)$$

Making Predictions with $O(\cdot)$-Notation

- Suppose you know that $\text{Time}(n) = O(n^2)$. And you know that for $n_1 = 1000$, the time taken was 5 seconds. Estimate how much time would be taken for $n_2 = 2000$.

$$\text{Time}(n) \leq C \cdot n^2$$

$$\text{Time}(n_1 = 1000) = 5s = C \cdot n_1^2$$

$$\text{Time}(n_2 = 2000) = ? = C \cdot n_2^2$$

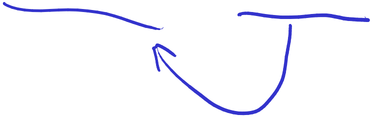
$$= C \cdot \left(\frac{n_2}{n_1} \cdot n_1\right)^2$$

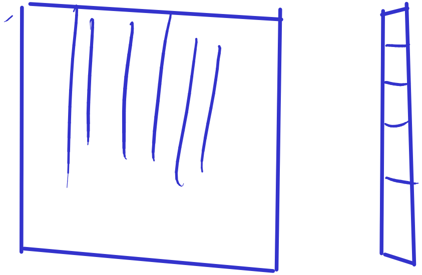
$$= \underbrace{C \cdot n_1^2}_{} \cdot \left(\frac{n_2}{n_1}\right)^2$$

$$= \text{Time}(n_1) \cdot \left(\frac{n_2}{n_1}\right)^2 = 5s \cdot 2^2 = 20s$$

Part 1:
Models, Errors, and Numbers

1 Python, Numpy, and Matplotlib





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:** Python
- **Demo:** numpy