

CS 357 – Numerical Methods 1

Prof. Mariana Silva

SPRING 2020

Course overview

- **Course website:**
 - Schedule
 - Calendar
 - Syllabus
 - Office hours queue
 - Python tutorial
- **PrairieLearn demo:**
 - HW
 - Quiz

Introduction and “Big Idea”

Prof. Mariana Silva

SPRING 2020

What are...

Numerical Methods ?

Numbers in a computer
(and how computer understands these numbers)



- Mathematical model
 - “algorithms” derived from math ideas to solve equations numerically
- Complexity of the problem
 - Slow vs fast
- Accuracy
 - Accurate vs inaccurate

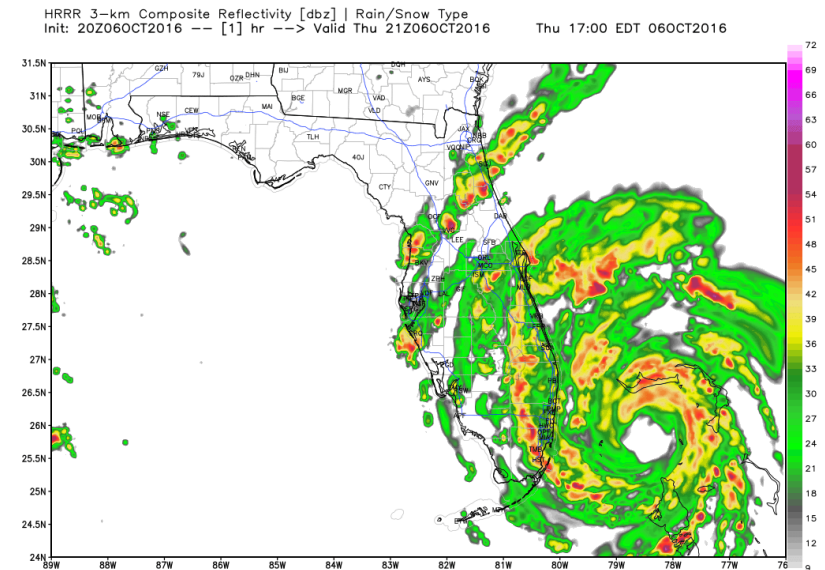
Method = Math + Complexity + Accuracy

Why is this course important?

1. Understanding and reconstruction of known problems
 - Natural disasters
 - Catastrophic failures
2. Prediction of unknown situations
 - Weather conditions
 - Behavior of new materials
3. Optimization of existing problems
 - Image recognition
 - Reduce fabrication costs



**Explosion of
Ariane 5 in 1996**



Goals for this course

- Understand how numbers are represented in the computer.
- When developing code, you will likely run into numerical errors. What are the sources of these errors?
- How can you avoid numerical errors?
- How can you choose a suitable algorithm for a given application?
- Use existing libraries to solve real applications.

(Numerical) **Method** = **Math** + Complexity + Accuracy

Mathematical model:

What equations can we use to represent our problem?

Accuracy:

Are we getting accurate results?

Why is the method not giving me the correct solution?

Complexity:

How long does it take to solve this problem?

Is it cost-effective?

Your entire CS 357
semester in a few
slides!

Are you ready?

Accuracy

- Why a numerical method might not give the right answer?
 - Computers have finite representation of numbers
 - Sometimes the “right answer” cannot be represented in a finite way
 - Example:

$$\pi = 3.1415926535897932384626433832795028841971\dots$$

Demo: Waiting for the number 1

```
from time import sleep

x = 0.0

while x != 1.0:
    x += 0.1
    print(repr(x))

    sleep(0.1)
```

What is going to happen when we run this code?

- A. Code will stop after printing 11 values for x
- B. Code will stop after printing 10 values for x
- C. Code will not stop
- D. Code will not start

Monte Carlo Methods

Texas Holdem Game: we would like to determine the probability of winning of a given starting hand

Physical experiment
vs
Numerical experiment

Ultimate Hold'em
DEALER QUALIFIES WITH PAIR OR BETTER

5♣ 4♦

4♣ 9♥ J♥ K♠ 8♥

7♦ K♣

PLAY

TRIPS ANTE = BLIND

BLIND BET
ONLY HIGHEST WIN AWARDED
WHEN DEALER IS BEATEN

Royal Flush	500:1
Straight Flush	50:1
Four of a Kind	10:1
Full House	3:1
Flush	5:2
Straight	1:1
All Other	Push

TRIPS BET
ONLY HIGHEST WIN AWARDED
BET PAID EVEN IF YOU FOLD

Royal Flush	50:1
Straight Flush	40:1
Four of a Kind	30:1
Full House	8:1
Flush	7:1
Straight	4:1
Three of a Kind	3:1

Numerical Experiments

- What do we want to know about a numerical experiment?
 1. What questions are we attempting to answer?
 2. What is the outcome of the experiment?
 3. Is it repeatable?
 4. Is the answer accurate?
 5. How long will it take?

Time vs accuracy trade-off

Question: Is running this method (with a certain accuracy) a good use of our time and/or computer resources?

Complexity

How long does it take to solve a problem?

Given A, B matrices of size $m \times m$, the matrix-matrix multiplication $A \cdot B$ takes τ seconds.

How long does it take to perform $C \cdot D$, matrices of size $2m \times 2m$?

```
from time import process_time
import numpy as np
from time import process_time
```

```
n = 2000
A = np.random.randn(n,n)
B = np.random.randn(n,n)

t = process_time() # store the time
C = A @ B
t = process_time() - t
print(t)
```

```
A = np.random.randn(2*n,2*n)
B = np.random.randn(2*n,2*n)

t2 = process_time() # store the time
C = A @ B
t2 = process_time() - t2
print(t2)
```

Linear system of equations: Image processing

How can we use linear operators to create blurred images? How can we do the inverse process?

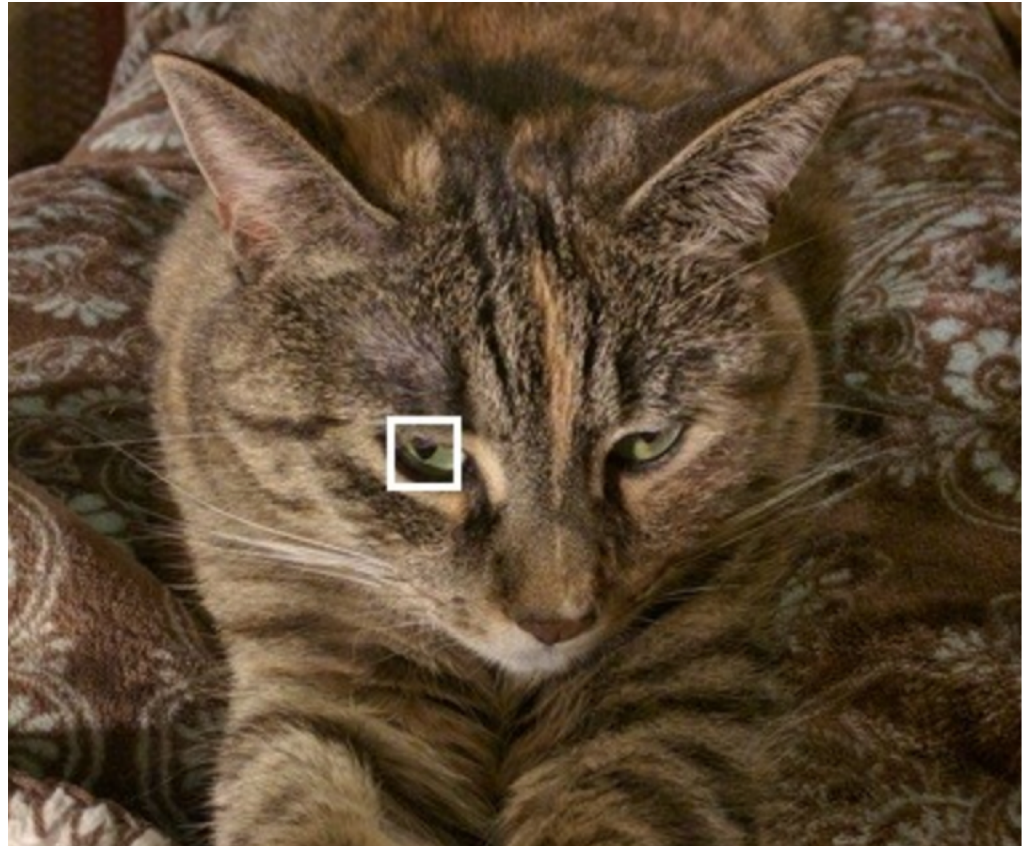
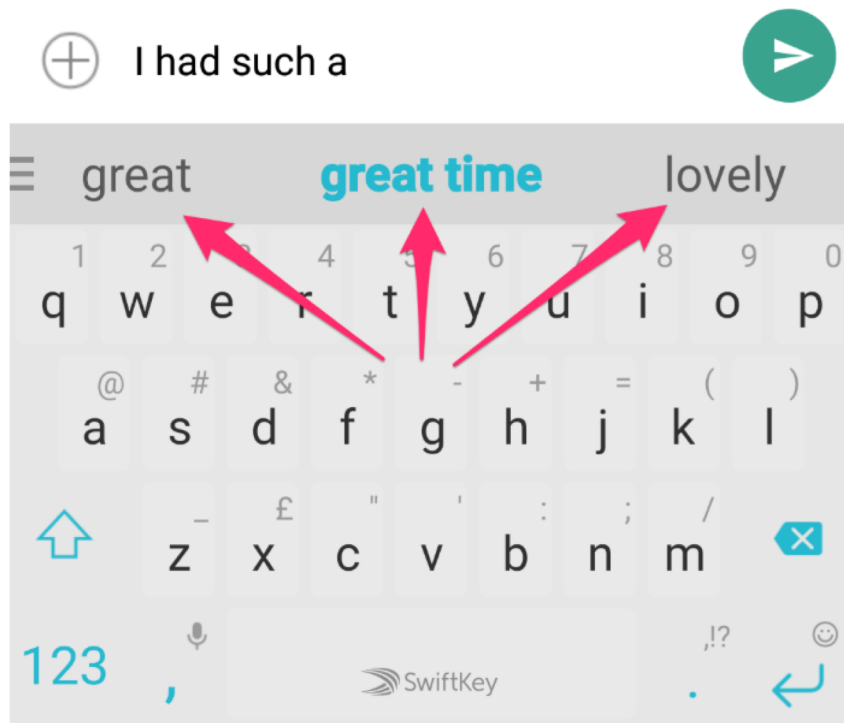


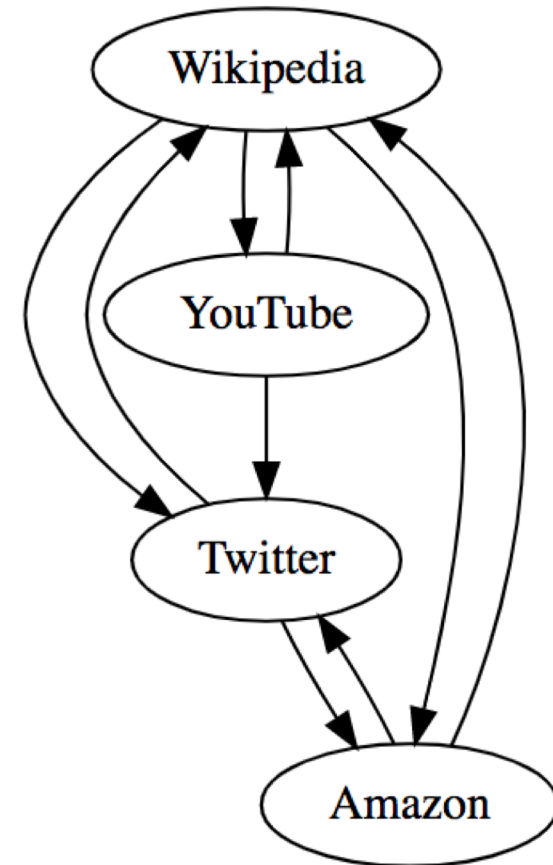
Image credit: <https://datacarpentry.org/image-processing/>

Markov chain

Word prediction

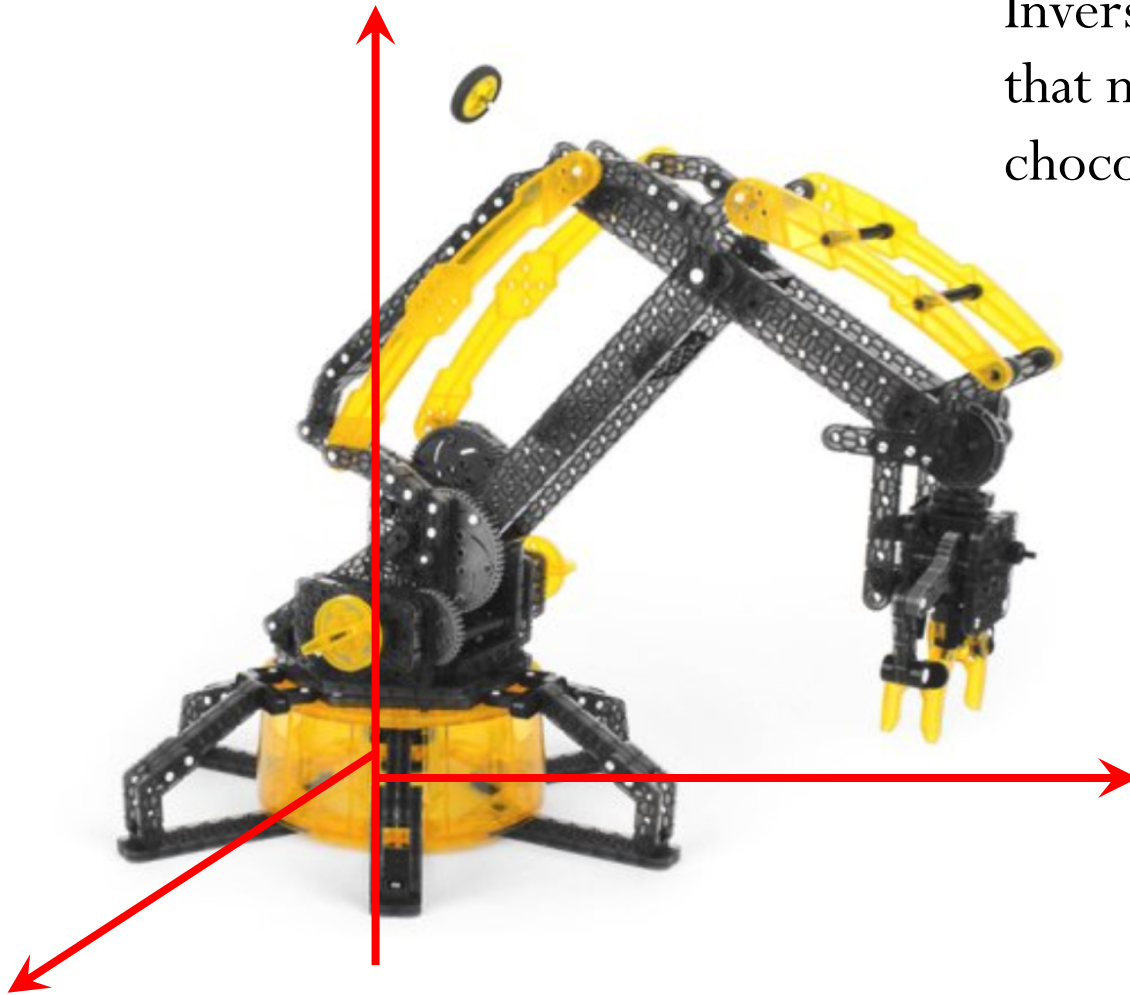


Page Rank



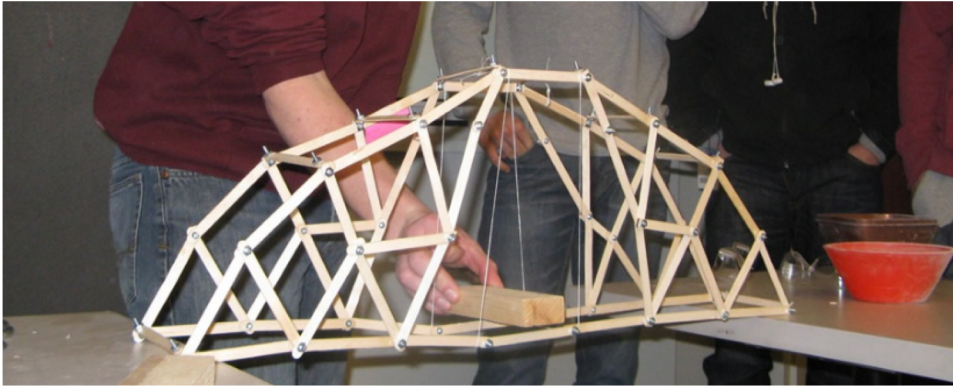
Nonlinear system of equations

Inverse kinematics: find the angles that make the robotic hand grab a chocolate candy!

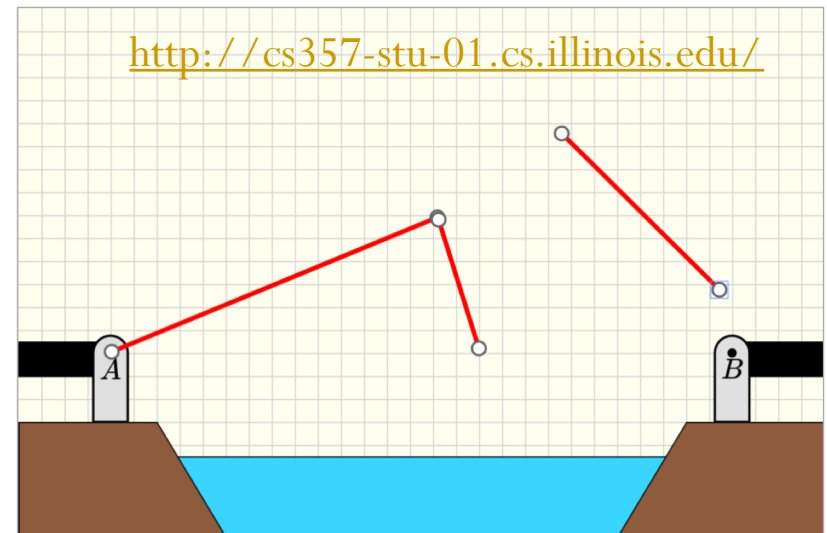
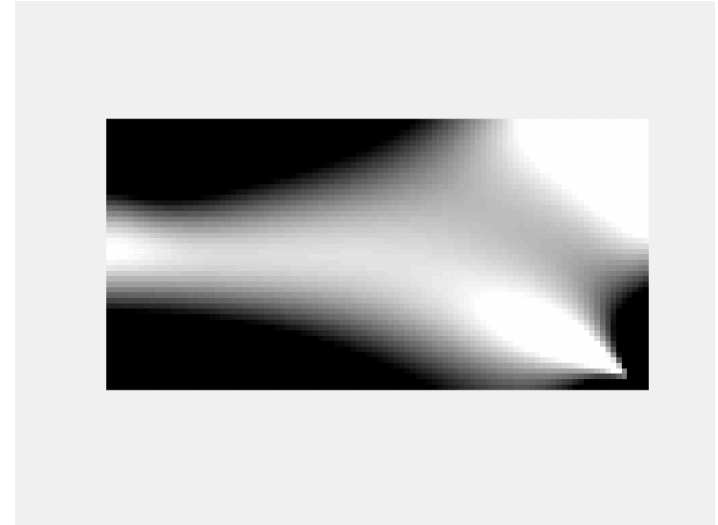


Optimization

Bridge design (high school projects)



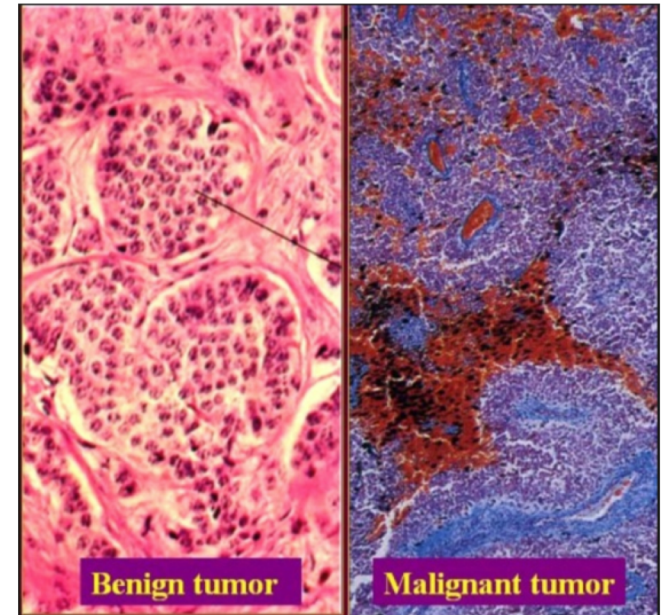
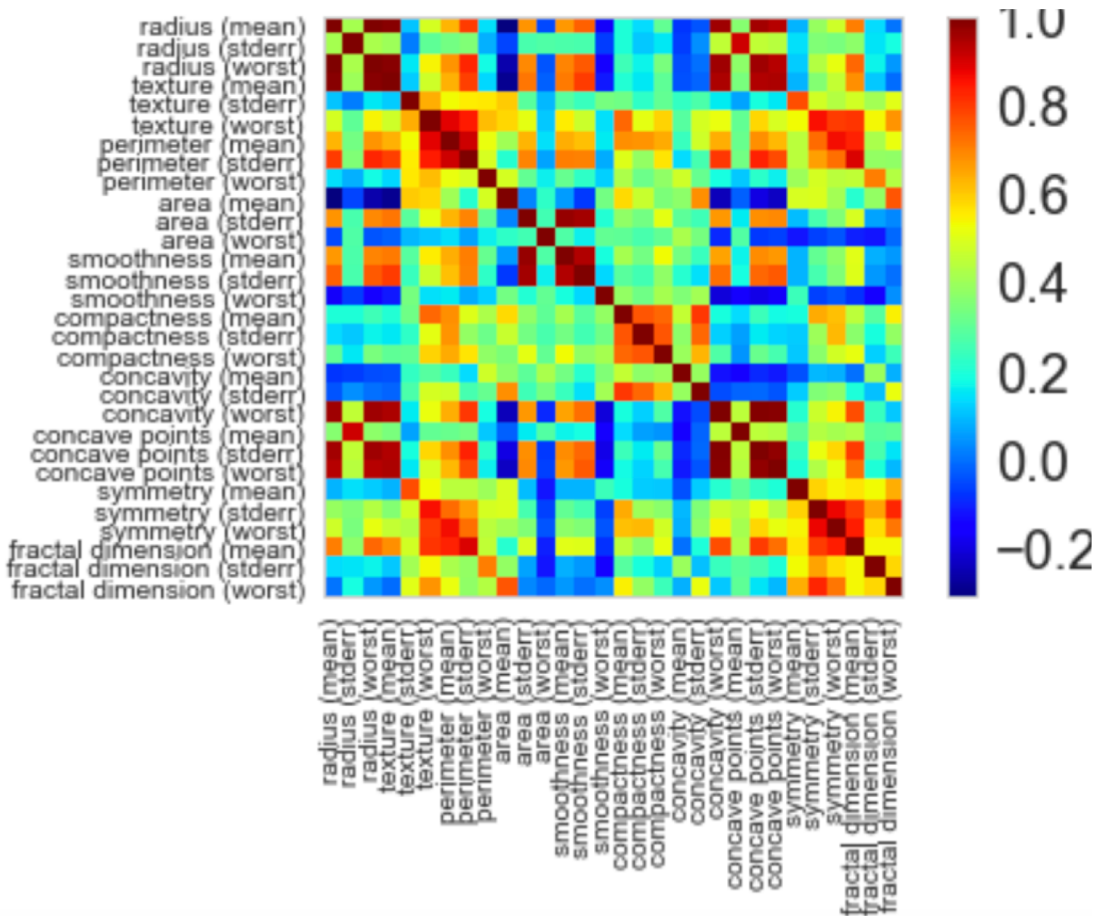
Numerical simulations to find optimized bridge designs



(Tolerance for members to be considered connected is 1/2 square grid.)

Linear Least Squares

Dataset containing the characteristics of cells for several patients. Can we make predictions if cells are benign or malignant?



Principal component analysis

Sometimes our dataset has too many features? How can we reduce the feature space and still keep the most important information?

