Computational Thinking

Honor 4471
Spring 2021

Some quick zoom etiquette

The goal is for the lecture to be interactive, but keep yourself muted unless you need to speak

Use the raise hand feature to ask questions during lectures

Having your video on promotes a better class environment

Why Computational Thinking?

goals

Solve problems computationally

Help you gain basic programming skills

Work with and process data


logistics

HONOR 4471-001 Spring 2021 Introduction to Computational

General Introduction

⚠ Please look under Modules for more information. Or look at the Zoom page under pages on how to connect for the first lecture. ⚠

This course is designed for non-Science/Engineering major students who desire a practical course for gaining basic computer programming skills. The primary objective is to teach students the basics of using an computational approach within their respective majors. The course will use the Python language to develop skills in problem-solving, debugging, acquiring real-world data, processing data, and interacting with and visualizing solutions. The course does not assume any prior experience with computers or programming of any kind. If you have prior experience with programming you should consider COMP 1010 or CS 1030/1410 instead. Although we start by not assuming any previous experience with computers, we will quickly ramp up to advanced computing topics including data analysis and machine learning. The material will be introduced in an informal, hands-on fashion with lots of examples and guided exploration of various computational concepts. The course is roughly divided into four modules,

1. Intro to Computational Thinking
2. Intro to Programming (in Python)
3. Advanced Programming Concepts and Considerations
4. Advanced Computations (Data Analysis, Machine Learning, Visualizations, etc.)

Python

Python is a programming language, and given the intended audience for this course, the focus will be on "language". We will learn to program similar to how we would learn a new language (like Spanish, French, etc.). Fairly early on we

Course assignments are not weighted.

Collapse All | View Progress

Course Resources

- Zoom Information
- Google Colab
- Notebook @CHPC
- Python tutorial (Optional)
- Reference textbook (Optional)

Module 1: Introduction to Computational Thinking

- Module 1: Introduction to Computational Thinking

Search

Activity
Chat
Teams
Assignments
Calendar
Calls
Files
...

Teams

Your teams

Paralab

Honor 4471

General
Concepts
How do I
Python

General Posts Files Wiki +

Team Meet

Let's get the conversation started

Try @mentioning a student or teacher to begin sharing ideas.

HS **Hari Sundar** 9:40 AM
Like this message once you have logged on to teams
5
Reply

HS **Hari Sundar** 10:11 AM Edited
Link for Lectures

https://utah.zoom.us/j/98541169450

Meeting ID: 985 4116 9450

See more

Join our Cloud HD Video Meeting
Zoom is the leader in modern enterprise video communications, with an easy, reliable cloud platform for video and audio conferencing, chat, and webinars across mobile, desktop, and room systems. Zo...

utah.zoom.us

Reply

Data science

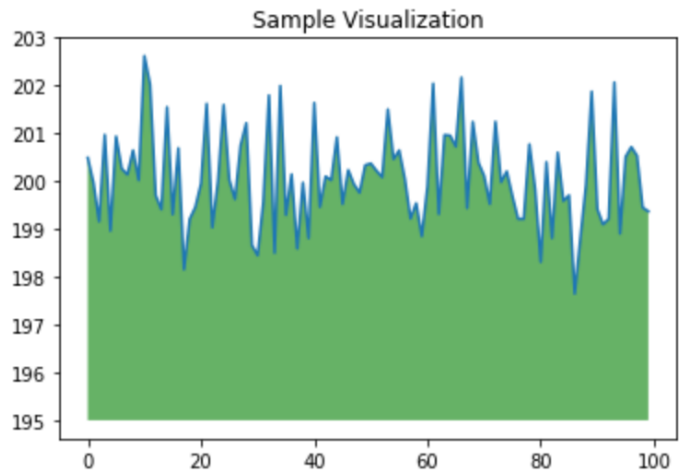
With Colab you can harness the full power of popular Python libraries to analyze and visualize data. The code cell below uses **numpy** to generate some random data, and uses **matplotlib** to visualize it. To edit the code, just click the cell and start editing.

```
[ ]  import numpy as np
     from matplotlib import pyplot as plt

     ys = 200 + np.random.randn(100)
     x = [x for x in range(len(ys))]

     plt.plot(x, ys, '-')
     plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

     plt.title("Sample Visualization")
     plt.show()
```



Homework & Projects

Simple quizzes on Canvas

- More for exploratory learning
- Bridges gap between informal and formal CS

Multiple programming assignments

- Exploratory
- Get help from Instructor/TA on Teams
- Discuss with classmates, no sharing of code

One final project

- Teams of two
- Be ambitious, but run it by TA/instructor to assess feasibility

Spring Break

INFORMAL SPRING BREAK – WEEK OF MAR 8TH

USE TO THINK ABOUT FINAL PROJECT

WE WILL BE AVAILABLE TO HELP

Getting help

The course is designed to be interactive

Get help on Teams

Ask in channels

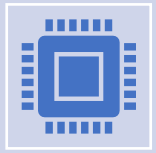
Office hours

Schedule 1-1 if needed

Poll (on Teams)

What is your major ?

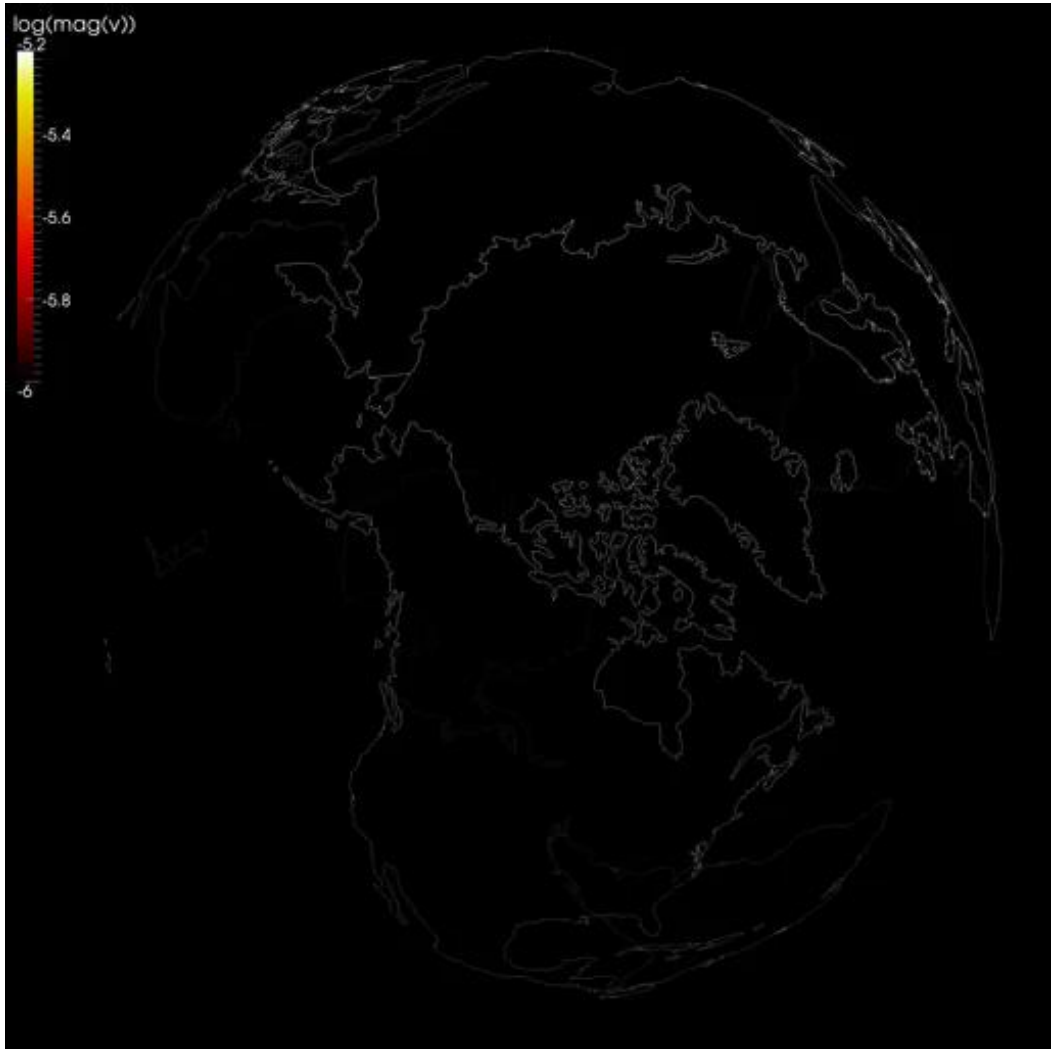
What is your academic standing ? Freshman, sophomore, …



Have you programmed before ?

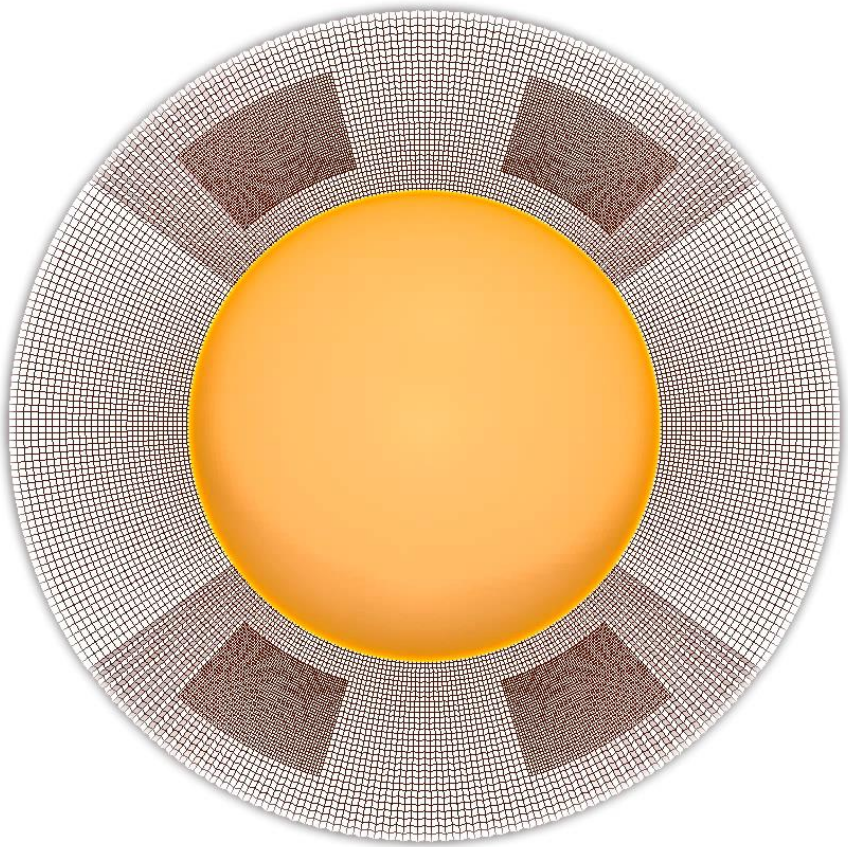
My Research

Application Domains



- Biomechanics
- Geosciences
- Computational Relativity
- CFD

Application Domains

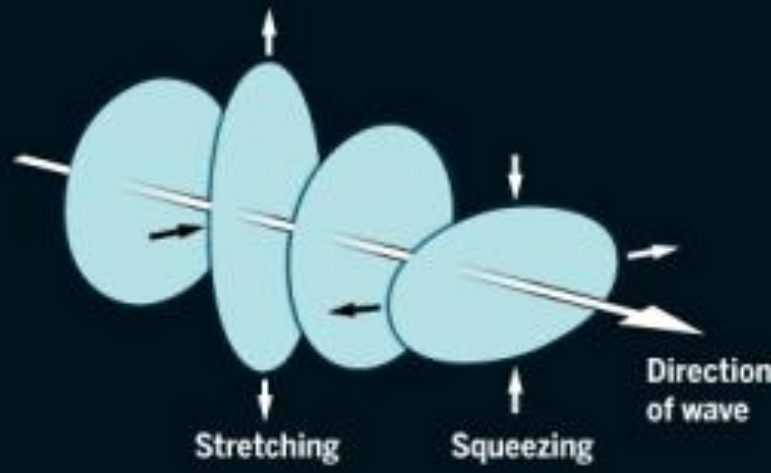


- Biomechanics
- Geosciences
- Computational Relativity
- CFD

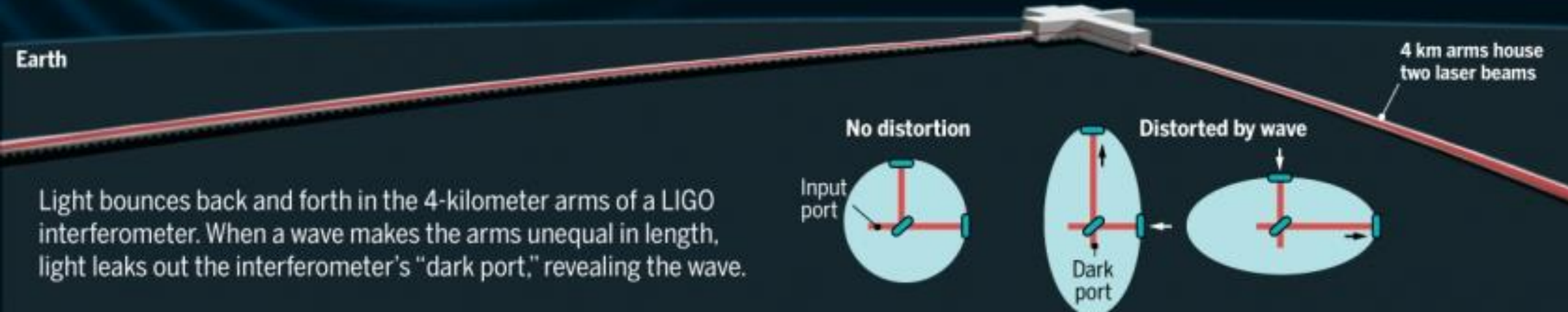
Catching a wave

As Einstein calculated, a whirling barbell-shaped mass, such as two black holes spiraling together, radiates ripples in space-time: gravitational waves.

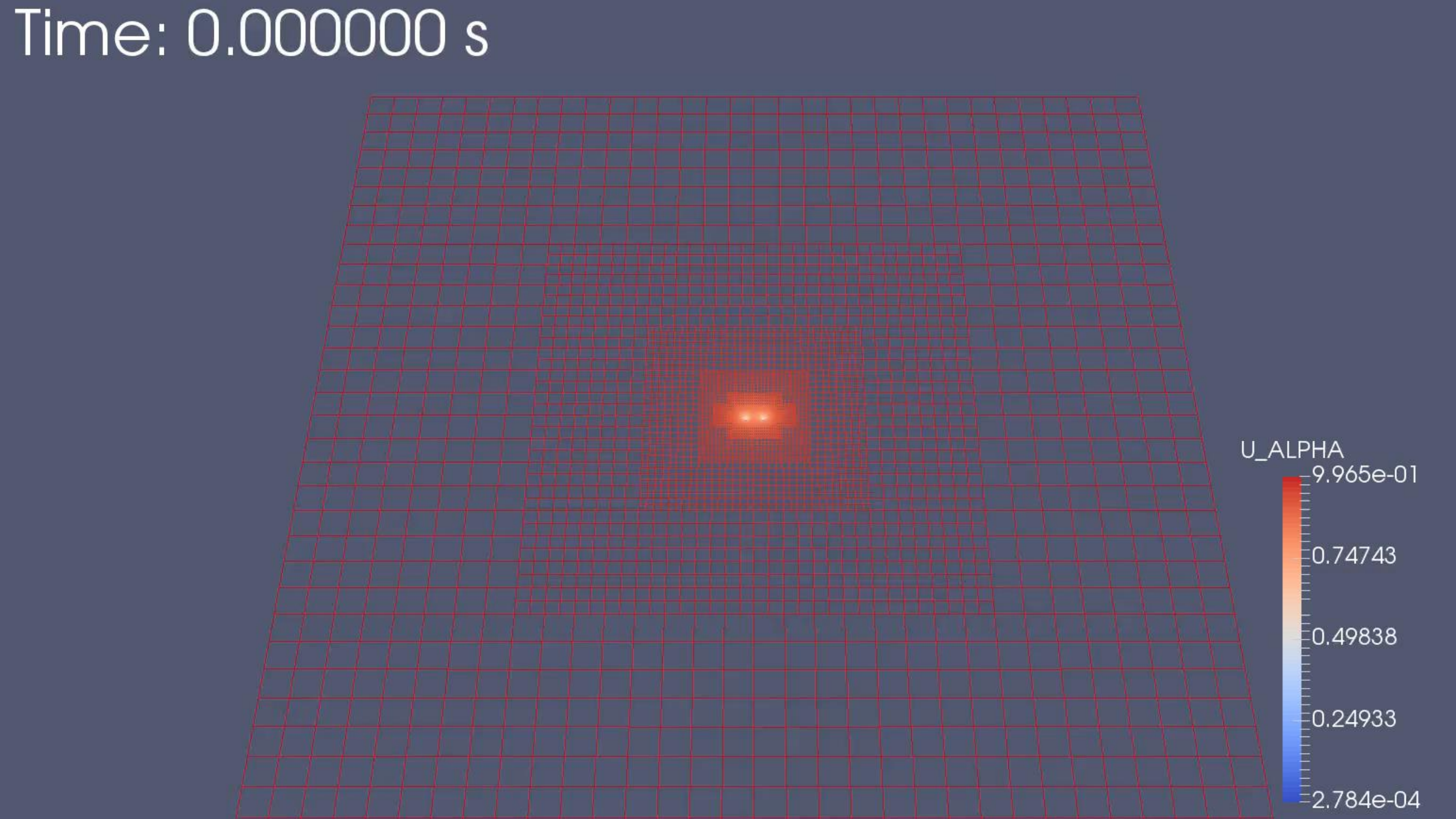
Zipping along at light speed, a wave stretches space in one direction and squeezes in the perpendicular direction, then reverses the distortions.



LIGO has detected waves of wavelength roughly equal to the distance between the detectors. The waves stretch each detector by about 1/10,000 the width of a proton.

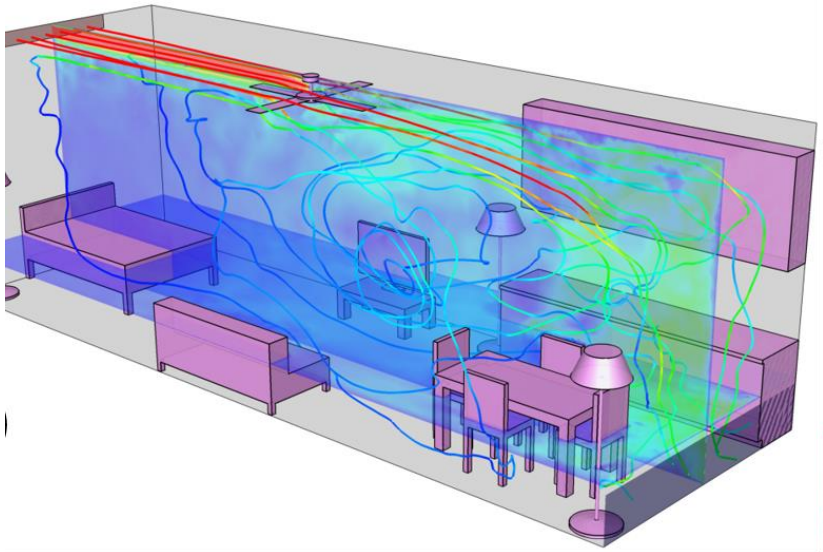


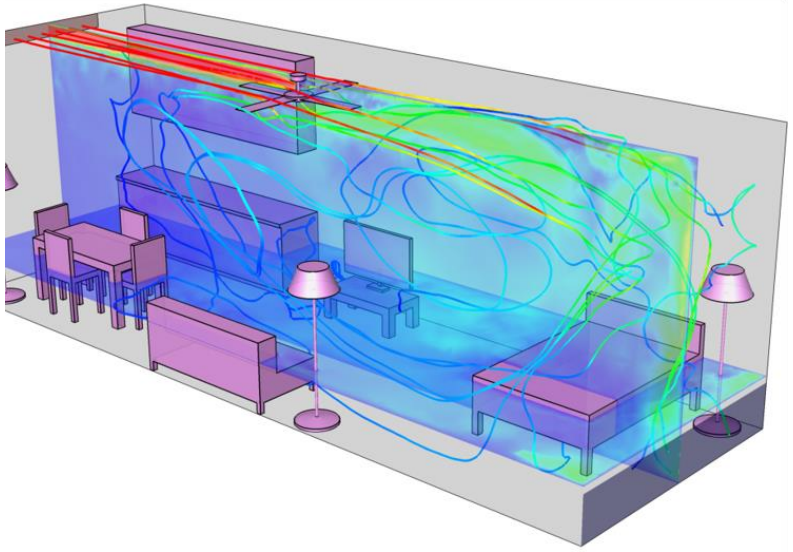
Light bounces back and forth in the 4-kilometer arms of a LIGO interferometer. When a wave makes the arms unequal in length, light leaks out the interferometer's "dark port," revealing the wave.

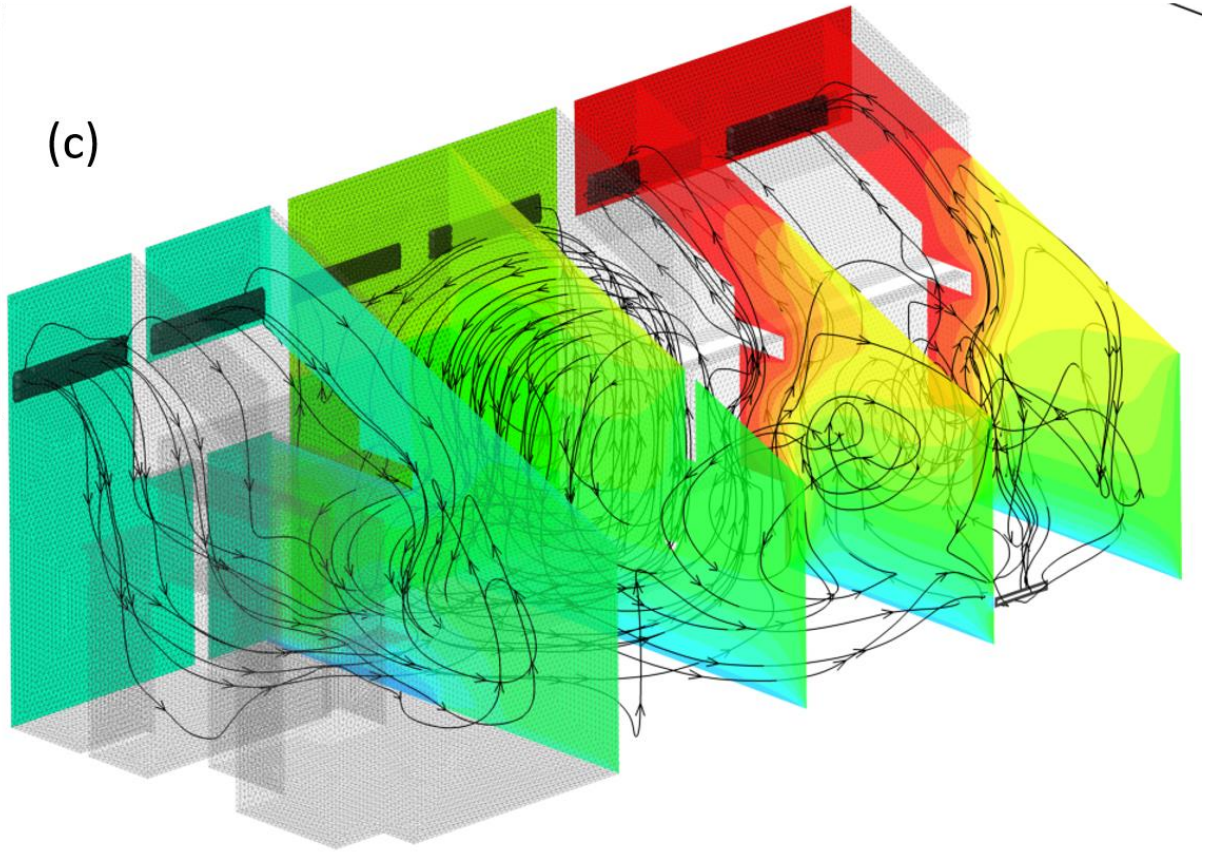

Time: 0.000000 s
U_ALPHA
9.965e-01
0.74743
0.49838
0.24933
2.784e-04

Application Domains

- Biomechanics
- Geosciences
- Computational Relativity
- CFD

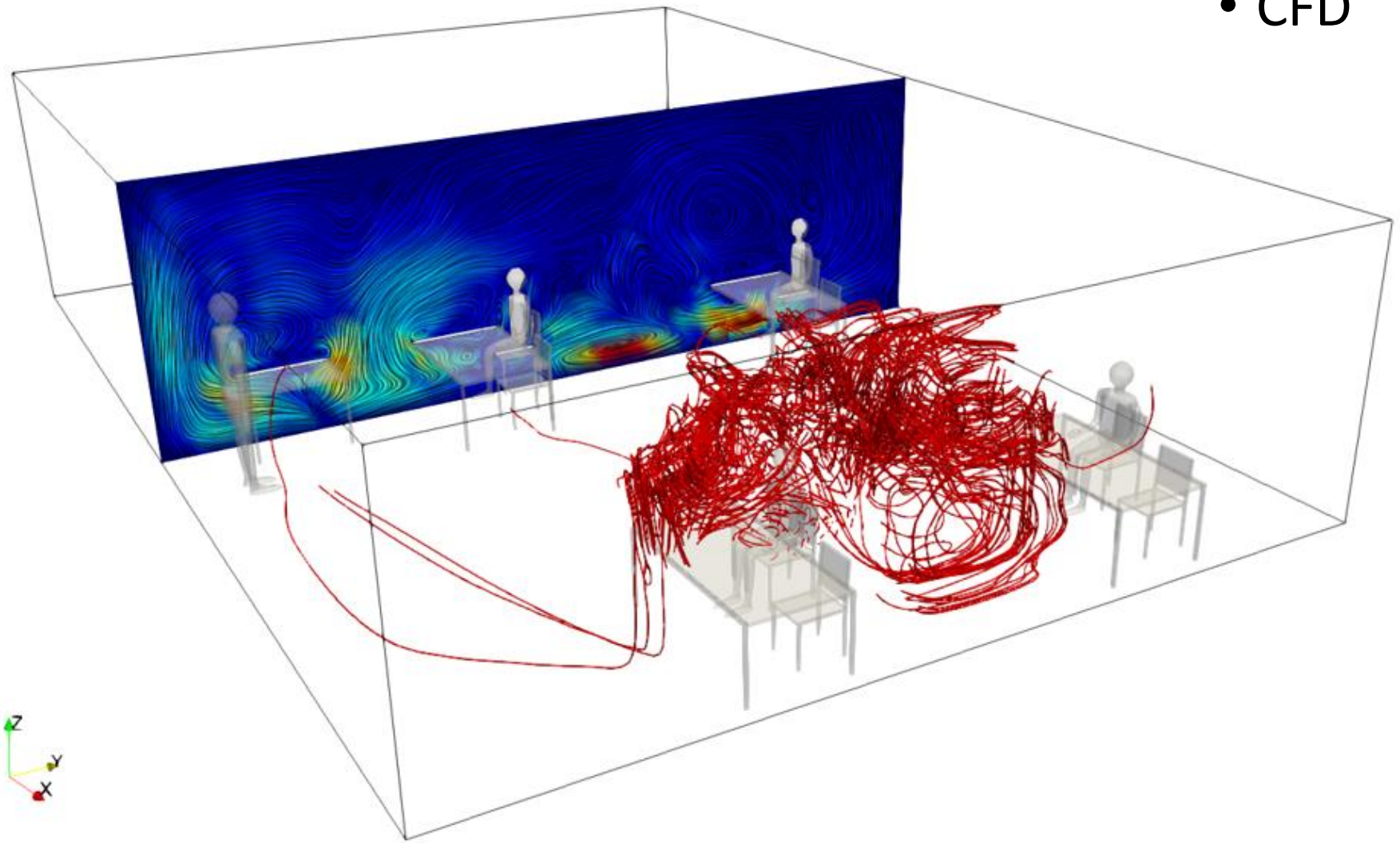






Application Domains

- Biomechanics
- Geosciences
- Computational Relativity
- CFD



Course Outline

Intro to Computational Thinking

Intro to Programming (in Python)

Advanced Programming Concepts and Considerations

Advanced Computations (Data Analysis, Machine Learning, Visualizations, etc.)

Why Computing ?

Why Computing ?

Perform tasks

- faster
- more accurately
- more reliably

Solve larger problems

Solve impractical/impossible problems

What tools do Computer Scientists use ?

Computer Systems

Hardware

Operating System

Network & Internet

Programming Language

Computational Thinking

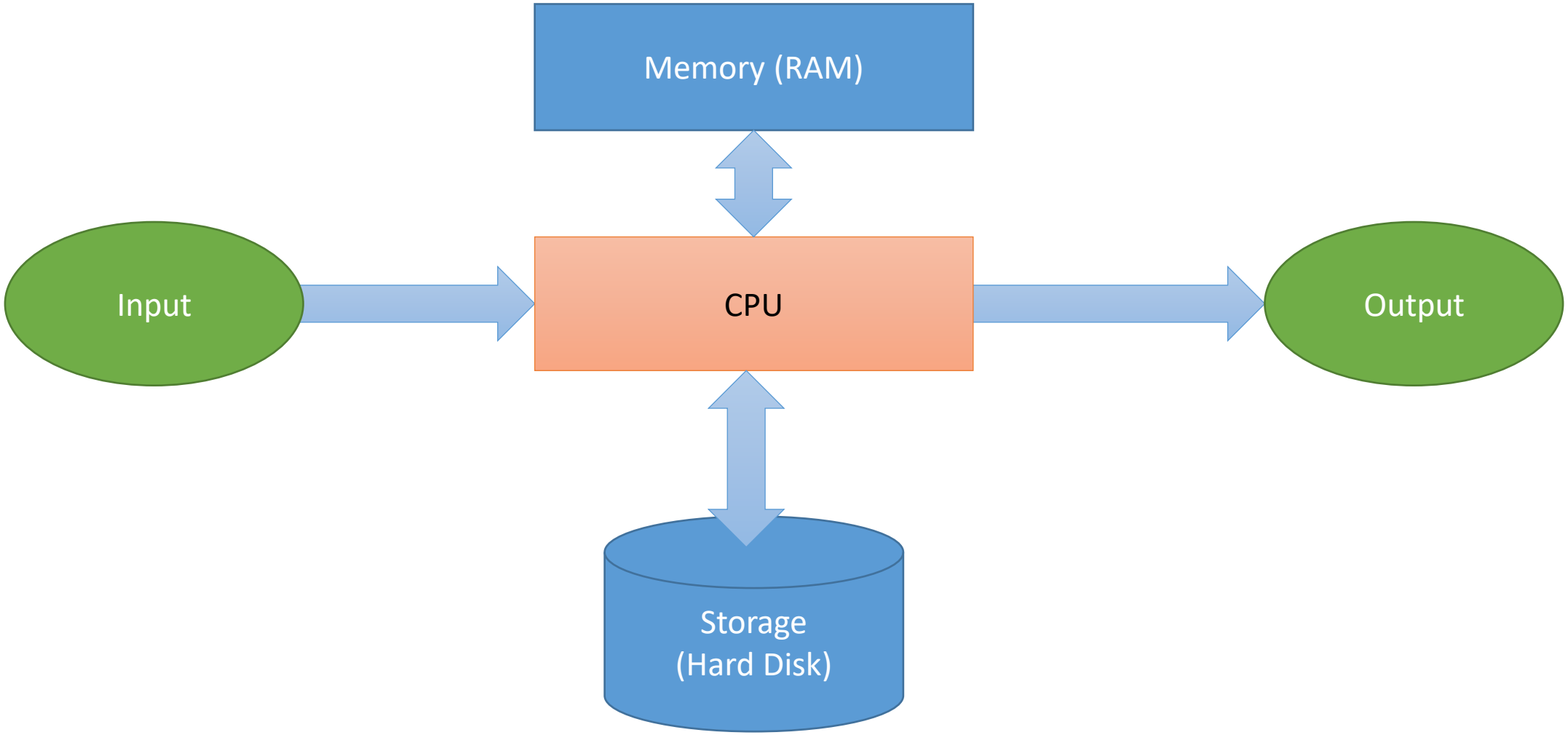
Logic

Reasoning

Data

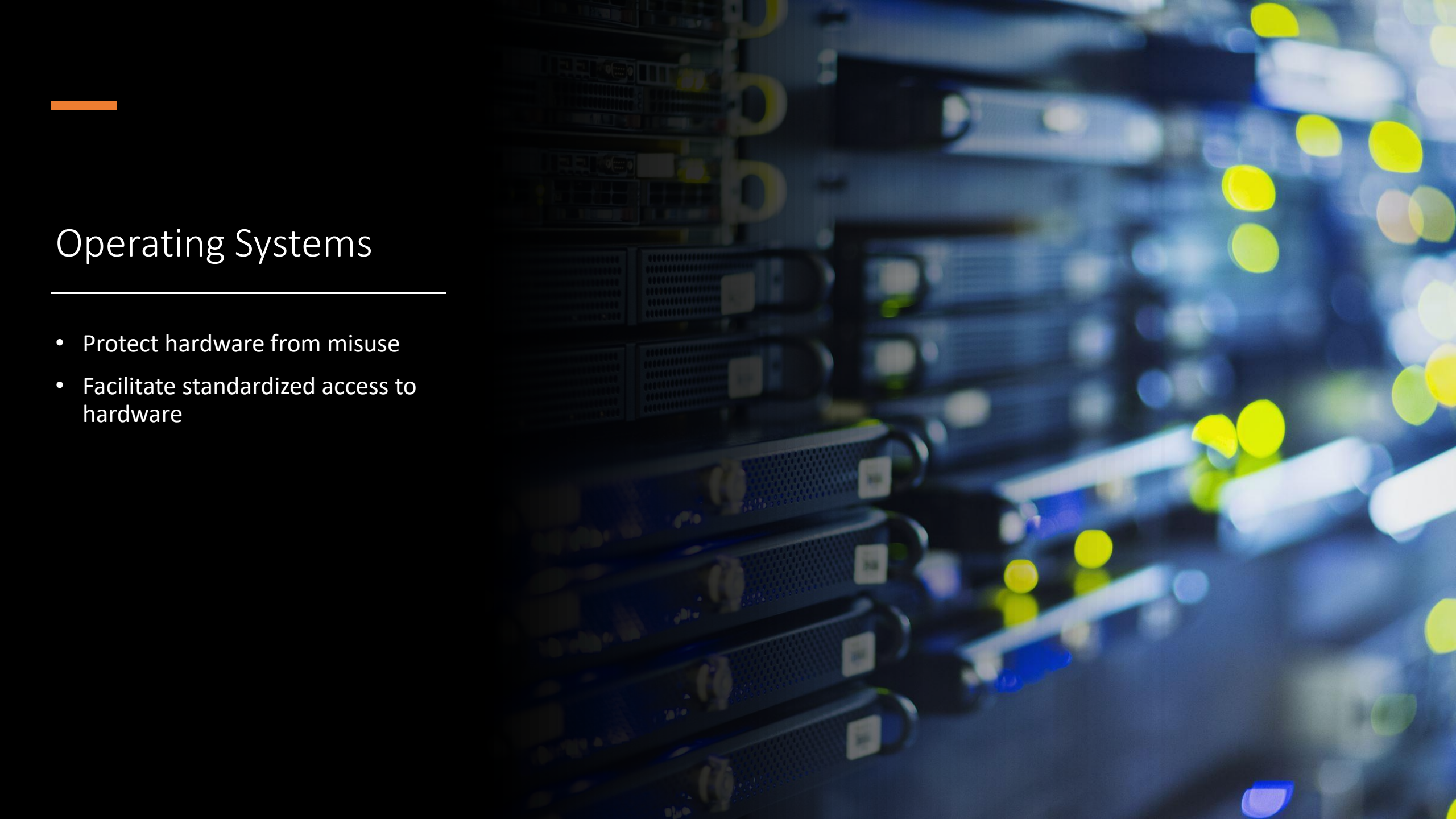
Algorithms

Computer Hardware



Operating Systems

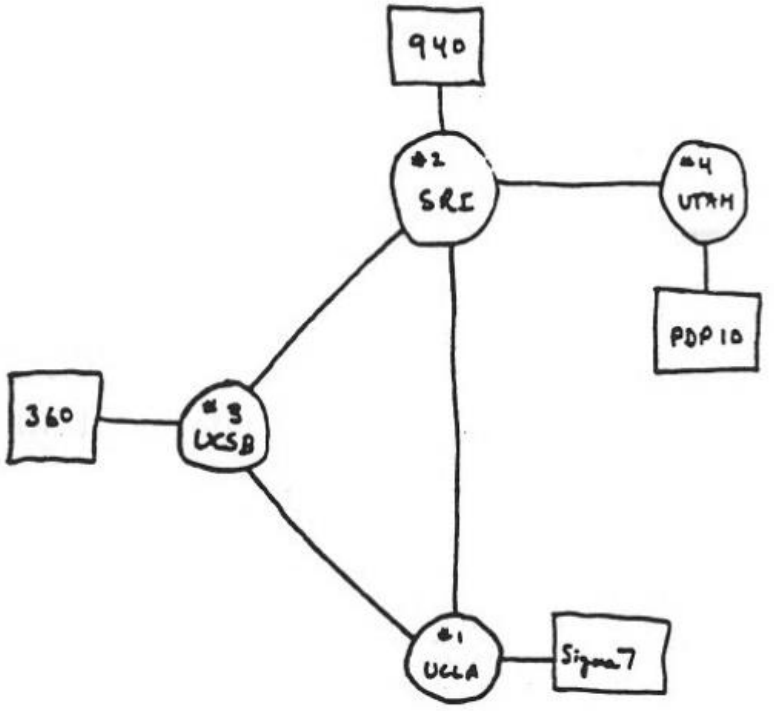
- Protect hardware from misuse
- Facilitate standardized access to hardware

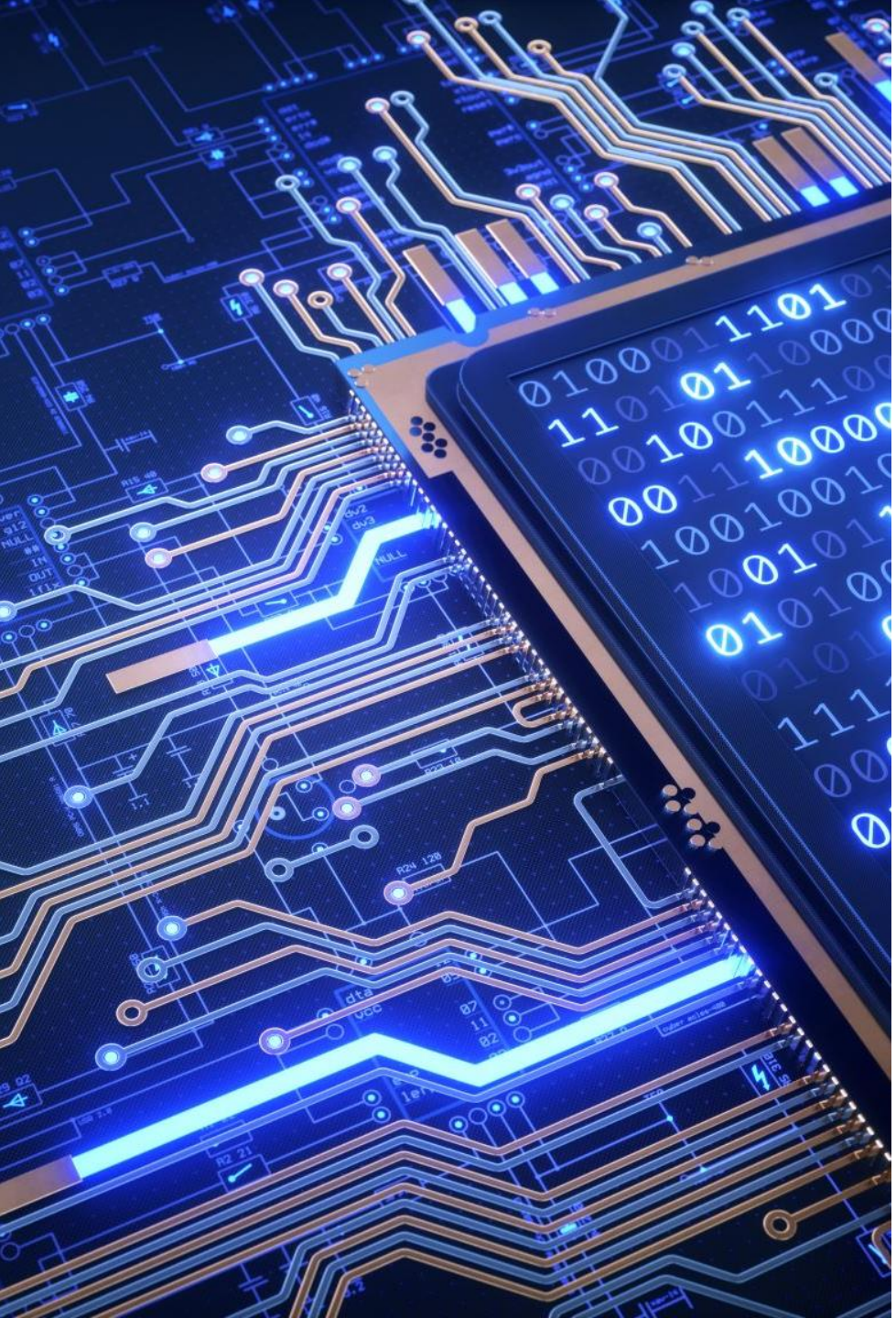


Networks and Internet

- Essential for most modern applications
  - Email
  - Lectures (during covid)
- We need to connect to other computers that are part of the internet
  - Wireless
  - Wired

Early sketch of ARPANET's first four nodes



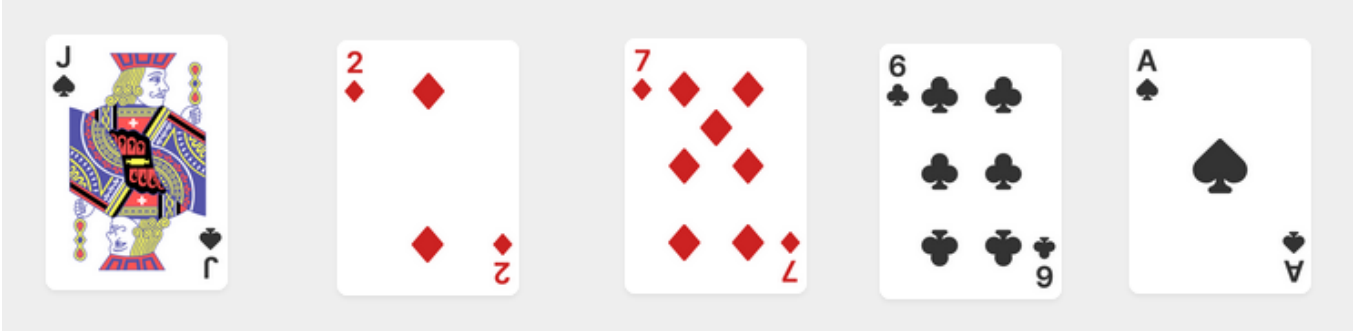


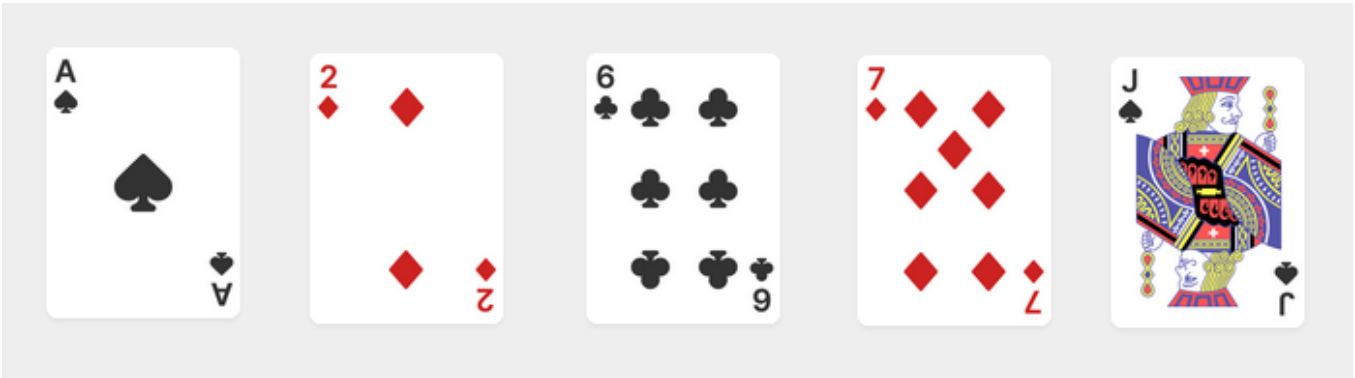
Programming Languages

- Programmability separates computers from other machines
- Needs clear and unambiguous instructions
  - Myopic view, no context
  - Humans by default use a lot of context
- Machine language vs human readable languages
  - Assemblers
  - Compilers
  - Interpreters
- Software Libraries


Algorithms

Sort some cards






Can you write precise
instructions to sort cards ?