

# Quantum Computing: Programs and Systems

Wednesday, September 7, 2022

Rutgers University

Yipeng Huang

How does a computer work?

What are the parts of a computer?

What are our fundamental assumptions about how computers work / what computers are made of?

What is quantum computing?

Quantum chemistry  
& high energy physics

General purpose  
computation

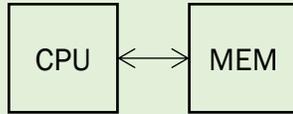
High dimensional  
nonlinear optimization

Computational neuroscience  
& pattern recognition

Fluid dynamics  
& plasma physics

Quantum chemistry  
& high energy physics

General purpose  
computation



Classical digital  
von Neuman architectures

High dimensional  
nonlinear optimization

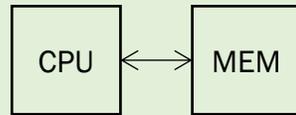
Computational neuroscience  
& pattern recognition

Fluid dynamics  
& plasma physics

Quantum chemistry  
& high energy physics

Digital  
(discrete variables)

General purpose  
computation



Classical digital  
von Neuman architectures

High dimensional  
nonlinear optimization

Computational neuroscience  
& pattern recognition

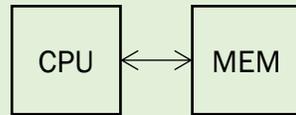
Fluid dynamics  
& plasma physics

Analog  
(continuous variables)

Quantum chemistry  
& high energy physics

Digital  
(discrete variables)

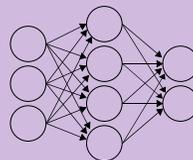
General purpose  
computation



Classical digital  
von Neuman architectures

High dimensional  
nonlinear optimization

Computational neuroscience  
& pattern recognition



Analog neuromorphic  
networks

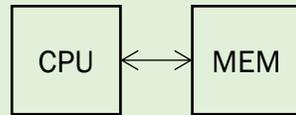
Fluid dynamics  
& plasma physics

Analog  
(continuous variables)

Quantum chemistry  
& high energy physics

Digital  
(discrete variables)

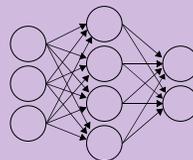
General purpose  
computation



Classical digital  
von Neuman architectures

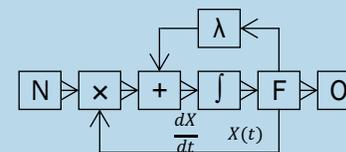
High dimensional  
nonlinear optimization

Computational neuroscience  
& pattern recognition



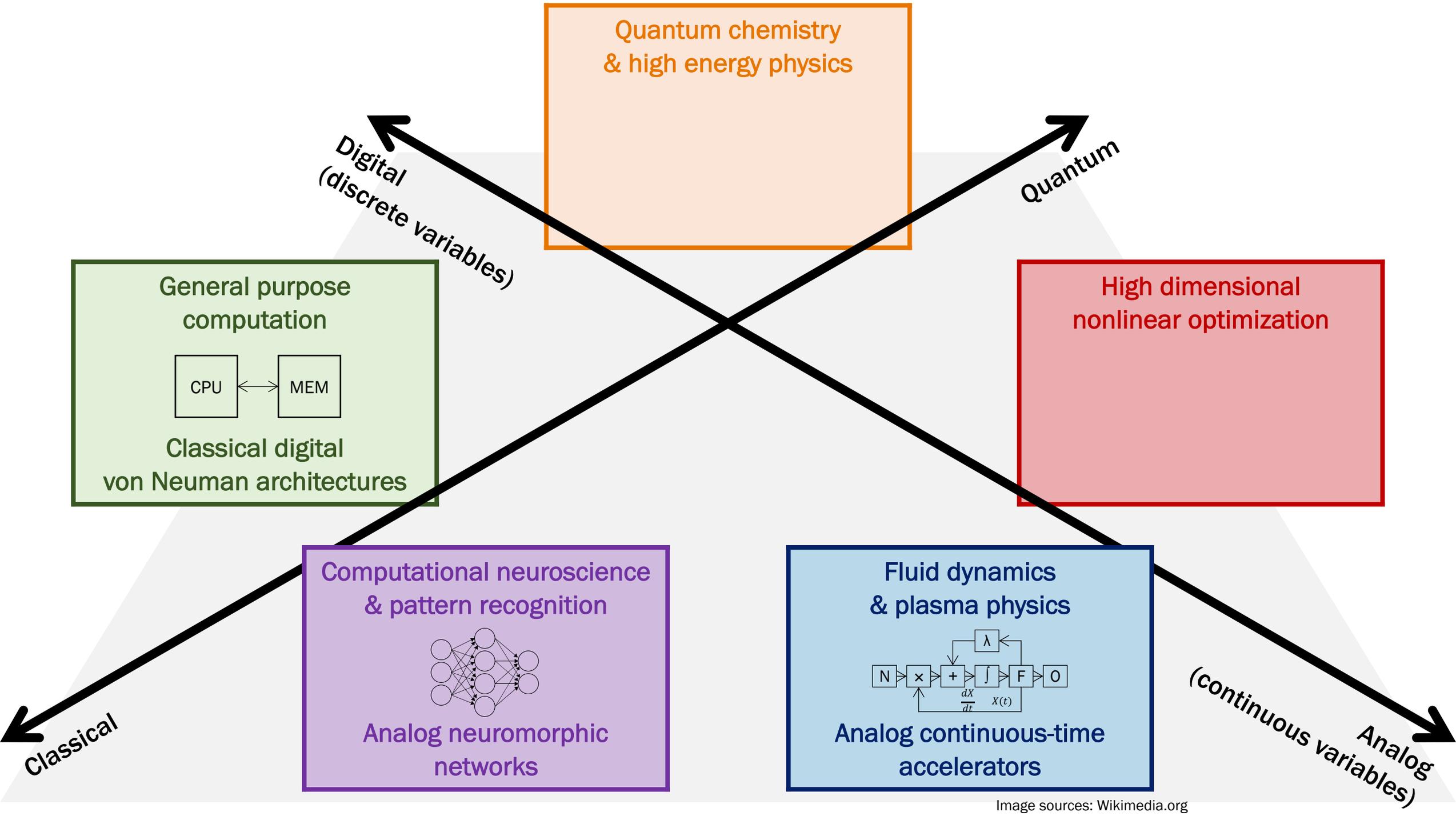
Analog neuromorphic  
networks

Fluid dynamics  
& plasma physics



Analog continuous-time  
accelerators

Analog  
(continuous variables)



Quantum chemistry  
& high energy physics

Digital  
(discrete variables)

Quantum

High dimensional  
nonlinear optimization

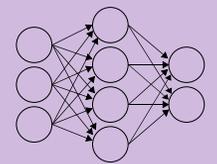
General purpose  
computation

CPU ↔ MEM

Classical digital  
von Neuman architectures

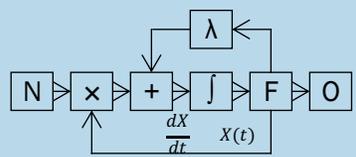
Classical

Computational neuroscience  
& pattern recognition



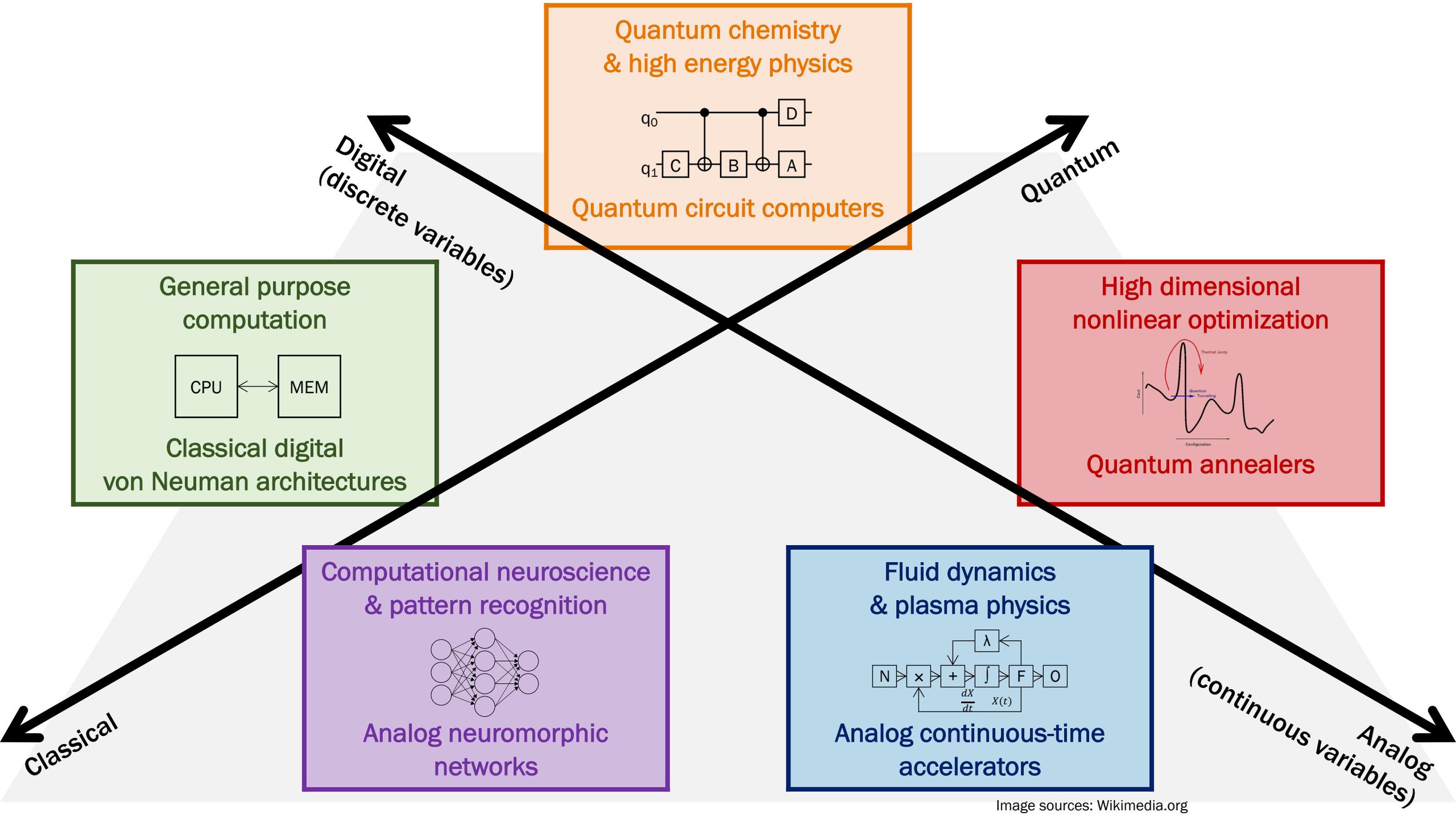
Analog neuromorphic  
networks

Fluid dynamics  
& plasma physics



Analog continuous-time  
accelerators

Analog  
(continuous variables)



Digital  
(discrete variables)

Quantum

Quantum chemistry  
& high energy physics

Quantum circuit computers

$q_0$  ———●———●——— D  
 $q_1$  [C] ⊕ [B] ⊕ [A]

General purpose  
computation

CPU ↔ MEM

Classical digital  
von Neuman architectures

High dimensional  
nonlinear optimization

Quantum annealers

Classical

Analog  
(continuous variables)

Computational neuroscience  
& pattern recognition

Analog neuromorphic  
networks

Fluid dynamics  
& plasma physics

Analog continuous-time  
accelerators

	1940s	1950s	1960s	1970s	1980s	1990s	2000s	2010s
<b>Analog continuous-time computing</b>	Analog computers for rocket and artillery controllers.	Analog computers for field problems. 	1 <sup>st</sup> transistorized analog computer. 	Analog-digital hybrid computers.	...			

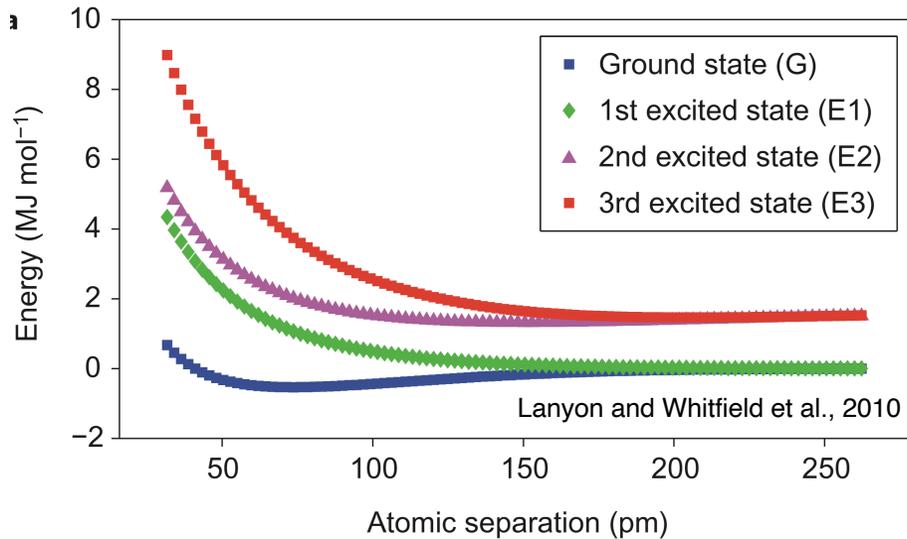
<b>Digital discrete-time computing</b>	Turing's <i>Bomba</i> .  Stored program computer.	1 <sup>st</sup> transistorized digital computer.  Microprogramming.	Moore's law projection for transistor scaling.  Instruction set architecture.	Dennard's scaling for transistor power density.  Reduced instruction set computers.	VLSI democratized.	FPGAs introduced.  GPUs introduced.	End of Dennard's scaling. CPUs go multicore.  Nvidia introduces CUDA.	Cloud FPGAs: Microsoft Catapult. Amazon F1.  ASICs: Google TPUs. DE Shaw Research Anton.
--	---	---	---	---	--------------------	---	--	--

**Heterogenous architectures**

Transistor scaling and architectural abstractions drive digital revolution, make analog alternatives irrelevant

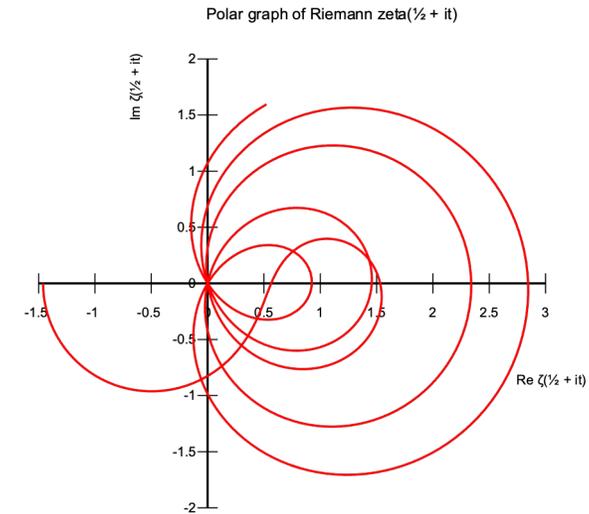
Scaling challenges drive heterogenous architectures

# Motivation: Race to practical quantum computation



## Quantum algorithms for chemical simulations

- Calculate properties of molecules directly from governing equations
- Use quantum mechanical computer to simulate quantum mechanics!



## Shor's quantum algorithm for factoring integers

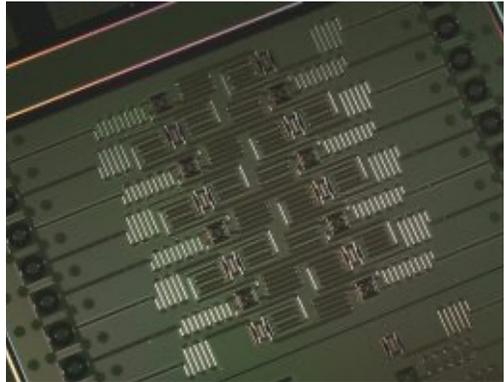
- Factor large integers to primes in polynomial time complexity
- Surpasses any known classical algorithm taking exponential time complexity

Hundreds of algorithms @ [QuantumAlgorithmZoo.org](http://QuantumAlgorithmZoo.org)

# Motivation: Race to practical quantum computation

## Superconducting qubits

## Trapped ion qubits



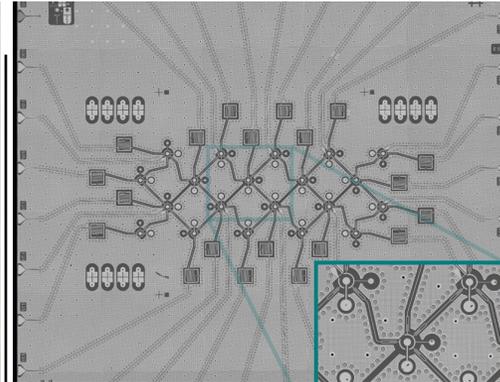
**IBM**



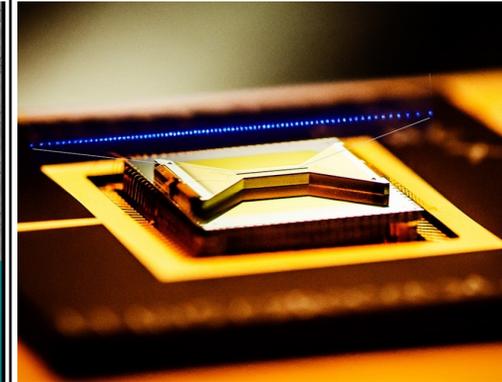
**Google**



**Intel**



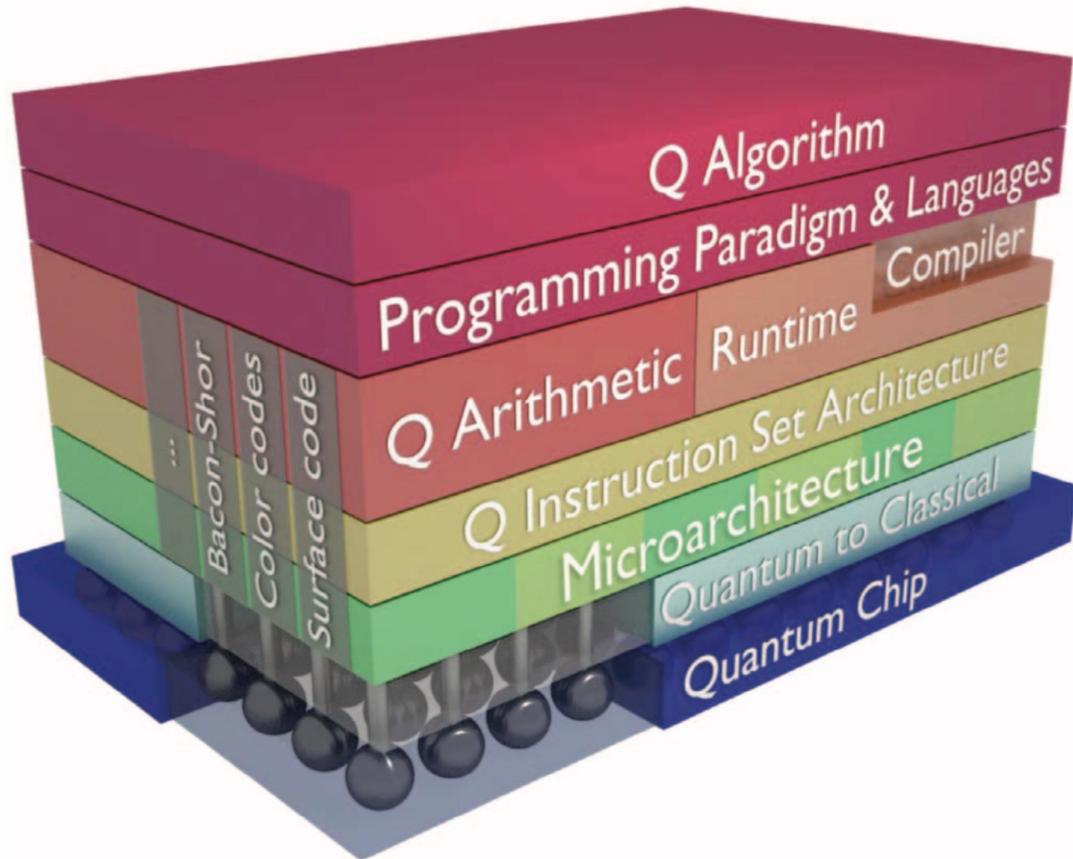
**Rigetti**



**University of  
Maryland /  
IonQ**

**Many research teams now competing towards more reliable and more numerous qubits.**

# Broad view of open challenges in quantum computer engineering



- A complete view of full-stack quantum computing.
- In short, challenges are in finding and building abstractions.
- In each layer, why we don't or can't have good abstractions right now.
- Recent and rapidly developing field of research.

Figure 1. Overview of the quantum computer system stack.

A Microarchitecture for a Superconducting Quantum Processor. Fu et al.

# What this class is about

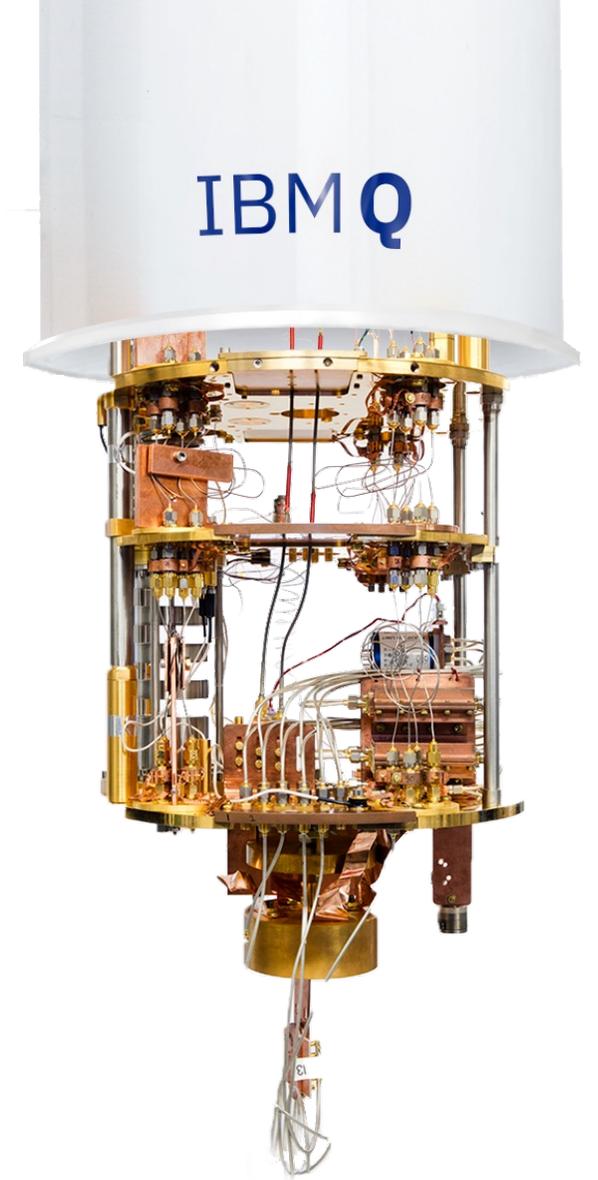
Course on latest developments in quantum computer engineering

What is quantum computer engineering??

- realizing quantum algorithms
  - on prototype quantum computers
- a rapidly growing field!!

Goals of the course:

- build foundations for understanding quantum algorithms
- explore open-source tools for using quantum computers
- read and discuss recent developments
- build foundation for you to pursue research or to be experts in industry



# Outline

- **Curiosity:** *digital, discrete time abstractions; unconventional computing*
- **Community:** welcome to class; prerequisites; introductions
- **Learning:** preview of the syllabus
- **Expectations:** reading; problem sets; programming; final project

# Outline

- **Curiosity:** digital, discrete time abstractions; unconventional computing
- **Community: welcome to class; prerequisites; introductions**
- **Learning:** preview of the syllabus
- **Expectations:** reading; problem sets; programming; final project

# Prerequisites

**Python programming:** working with Git, extending open source projects, Jupyter notebooks

**Algorithms:** need to have good understanding of classical algorithms before tackling quantum ones

**Probability**

**Linear algebra:** vector, matrix notation and multiplication. Matrix properties.

**Complex numbers**

**Architecture:** need to have solid understanding of how classical computers work before tinkering with quantum ones

**Access to iLab CS computing resources:** <https://resources.cs.rutgers.edu/>

# Useful, but not strictly required

## **Quantum information science course**

- Bra-ket, gates, circuits, measurement, superposition, entanglement
- 2021 Fall: ECE 493. Soljanin. Quantum Computing Algorithms.
- 2022 Spring: Physics 421. Schnetzer. An Introduction to Quantum Comp

## **Quantum mechanics**

- Problems and methods for quantum chemistry

# Introductions via show of hands

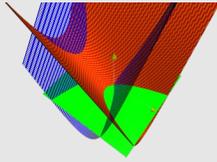
- Majors / minors: computer science, ECE, math, physics, other
- Degree progress: MS, seniors, juniors
- Who is in 583 section? Who is in 443 section?
- Experience in algorithms: CS 344 / CS 206
- Experience in architecture: CS 211 / ECE major
- Experience in Python programming

# Personal introductions

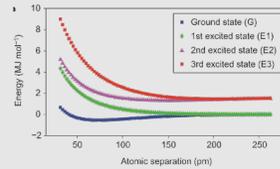
Yipeng

- Assistant professor, Rutgers, 2020 -
- Postdoc, Princeton, 2018 - 2020
- PhD, Columbia, - 2018

Nonlinear  
scientific  
computation



Quantum  
simulation &  
optimization



**New and extreme  
workload challenges**

**Multicore CPUs, GPUs,  
FPGAs, ASICs,  
analog, quantum,  
etc.**

**Limitations in  
transistor scaling**

Dennard's  
scaling  
already  
ended

Moore's law  
increasingly  
costly to  
sustain

## Open challenges in emerging architectures:

### Problem abstractions

- How do you accurately solve big problems?

### Programming abstractions

- Can you borrow ideas from conventional computing?

### Architecture abstractions

- How to interface with the unconventional hardware?

# My work in problem and programming abstractions for emerging architectures

<b>Continuous-time analog scientific computation</b>	<b>Accelerator chip prototype</b>	<b>Support for solving differential equations</b>	<b>Support for solving linear algebra</b>	<b>Support for solving nonlinear equations</b>	<b>Fluid dynamics application feasibility study</b>
	Successful hand-off to MIT, Ulm University, and two companies for further research.	JSSC 2016 (co-authored).	ISCA 2016. One of twelve Micro Top Picks best architecture papers of 2016.	MICRO 2017. Micro Top Picks honorable mention.	PI for DARPA STTR phase 1 grant. Thesis nominated for ACM dissertation award.
<b>Quantum algorithm debugging &amp; simulation</b>		<b>Assertions for quantum program patterns and bugs</b>	<b>Graphical model inference for quantum program simulation and analysis</b>	<b>Analog computing support for quantum control &amp; measurement</b>	
		ISCA 2019. mentees placed at MICRO SRC. IBM Qiskit open-source contribution.			

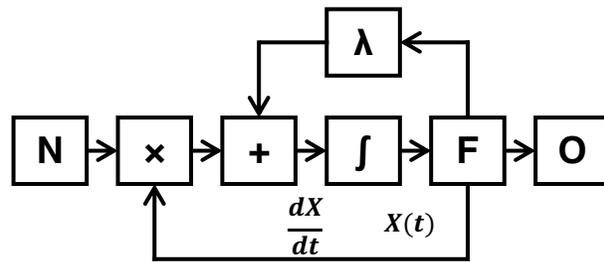
Requirements for supporting workloads

- How to do problems?
- How to get high accuracy solutions?
- How to handle large problem sizes?

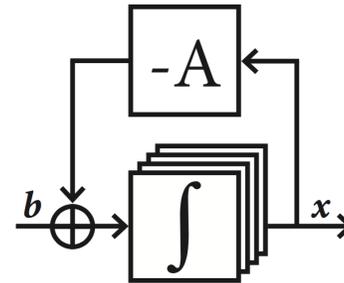
$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} - \frac{1}{\text{Re}} \nabla^2 \vec{u} = \text{RHS}$$

Numerical primitives as architectural abstractions

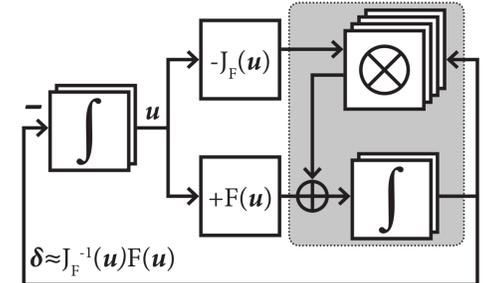
Analog-digital support for **differential equations**



Analog-digital support for **linear algebra**

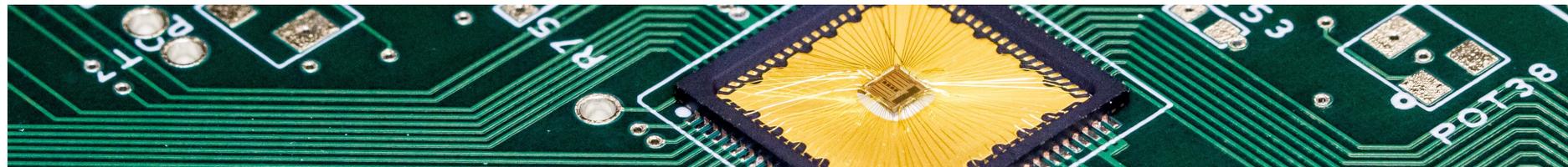


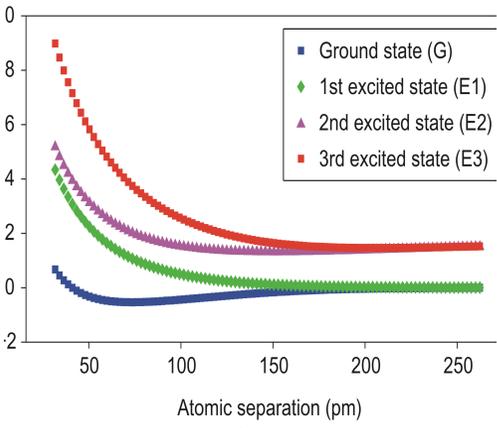
Analog-digital support for **nonlinear equations**



Unconventional architecture hardware prototyping

**Prototype continuous-time analog accelerator**





# Awe-inspiring quantum algorithms

## Chemistry simulations from governing equations

Quantum computers as quantum mechanics simulator

## Shor's algorithm for factoring integers

Surpasses any known classical algorithm

## Hundreds more near-term and far-future algorithms

QuantumAlgorithmZoo.org

# My work in bridging quantum software-hardware gap

## Assertions for quantum program patterns and bugs

ISCA 2019. IBM Qiskit open-source contribution.

## Graphical model inference for quantum program simulation and analysis

ASPLOS 2021. Google Cirq simulation backend publicly available.

## Analog computing support for quantum control & measurement

# Now-viable quantum prototypes

## Superconducting qubits

IBM, Google, Rigetti, ...

## Trapped ion qubits

IonQ, UMD, ...

## Dozens of candidate qubit technologies

May yet surpass current leaders in capacity and reliability

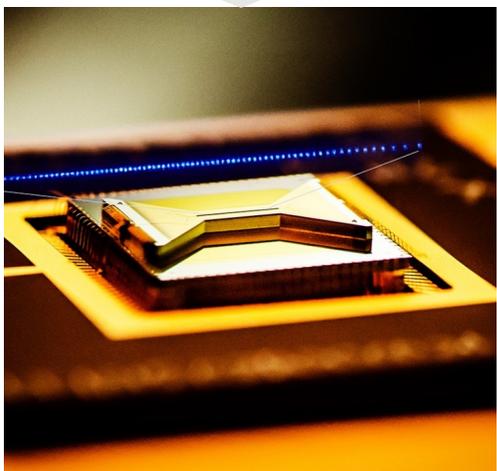


Image sources: Lanyon and Whitfield et al., 2010, IonQ

# Research here at Rutgers

- Prof. Mario Szegedy, quantum algorithms, complexity theory
- Prof. Yipeng Huang, quantum program simulation and analysis
- Prof. Zheng Zhang, quantum circuit compilation
- Prof. Emina Soljanin, quantum communications
- Prof. Steve Schnetzer, high energy physics and quantum computing
- Prof. Vatsan Chakram, solid state superconducting quantum devices

# Welcome all to class

**We welcome in this class diverse backgrounds and viewpoints spanning various dimensions:**

- race, national origin, gender, sexuality, disability status, class, religious beliefs

**We will treat each other with respect and strive to create a safe environment to exchange questions and ideas.**

# Outline

- **Curiosity:** digital, discrete time abstractions; unconventional computing
- **Community:** welcome to class; prerequisites; introductions
- ***Learning: preview of the syllabus***
- **Expectations:** reading; problem sets; programming; final project

# Preview of the syllabus

- Quantum computing fundamentals: qubits, gates, circuits
- Quantum computing algorithms: simple algorithms, QFT, factoring
- Near-term intermediate-scale quantum algorithms
- A systems view of quantum computer engineering
- Programming frameworks
- Emerging languages and representations
- Extracting success
- Prototypes

# Outline

- **Curiosity:** digital, discrete time abstractions; unconventional computing
- **Community:** welcome to class; prerequisites; introductions
- **Learning:** preview of the syllabus
- **Expectations: reading; problem sets; programming; final project**

# Logistics

<https://yipenghuang.com/teaching/2022-fall/>

<https://rutgers.instructure.com/courses/194734>

**Attend live, in-person**

**Office hours TBA**

## **One of the few uppermost division classes you might take**

- Very different expectations from any other class

### **Components**

- Reading discussions. (12% for 583 section; 20% for 443 section).
- Graded problem sets. (24% for 583 section; 40% for 443 section).
- Programming assignments; including: QAOA implementation in Google Cirq and VQE implementation in Qiskit. (24% for 583 section; 40% for 443 section).
- Final open-ended project. (40% for 583 section; 0% for 443 section).