Quantum Computing: Programs and Systems

Wednesday, January 17, 2024

Rutgers University

Yipeng Huang

How does a computer work?

electrical signals

decisions

data encoded as bits 1/0

gates to work on bits

What are the parts of a computer?

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

info has to be stored.

decisions are binary I/O

gates made of transistors

clock, discrete time

discrete time digital.

What is quantum computing?

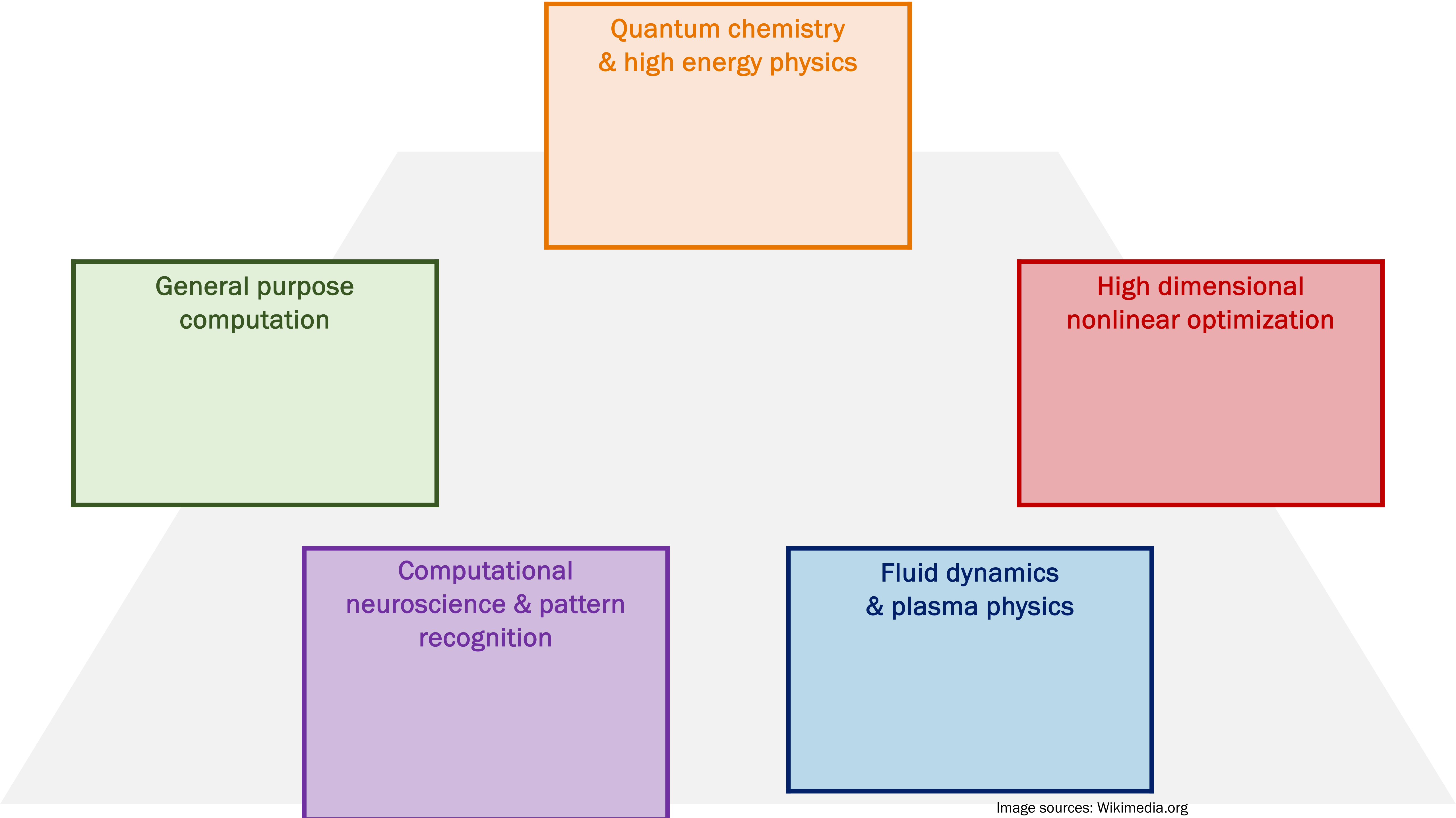
"qubits" vs. bits

able to represent more than two states.

also has gates.

new problem
difficult

difficult to construct..


Quantum chemistry & high energy physics
General purpose computation
High dimensional nonlinear optimization
Computational neuroscience & pattern recognition
Fluid dynamics & plasma physics
Image sources: Wikimedia.org

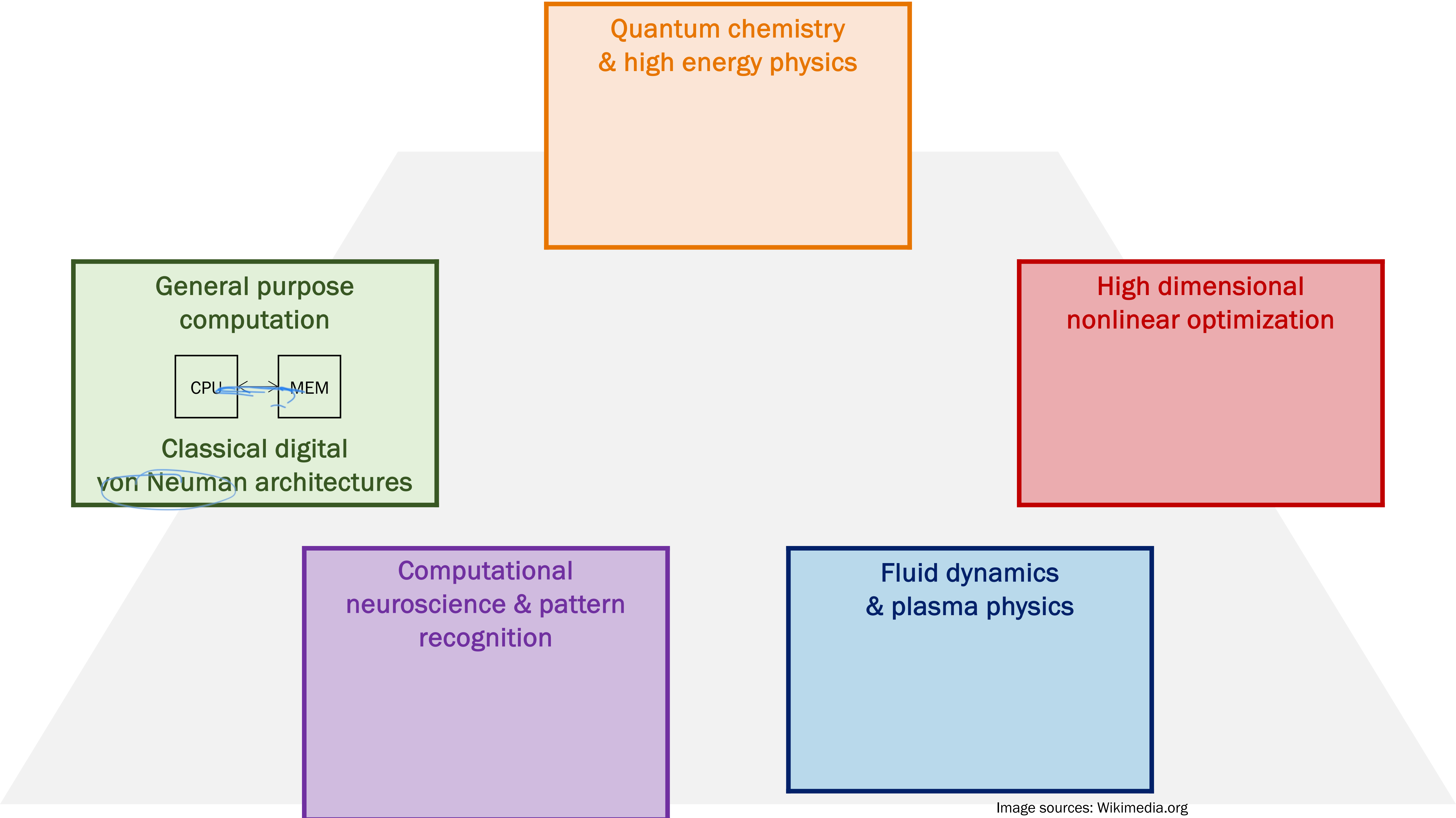
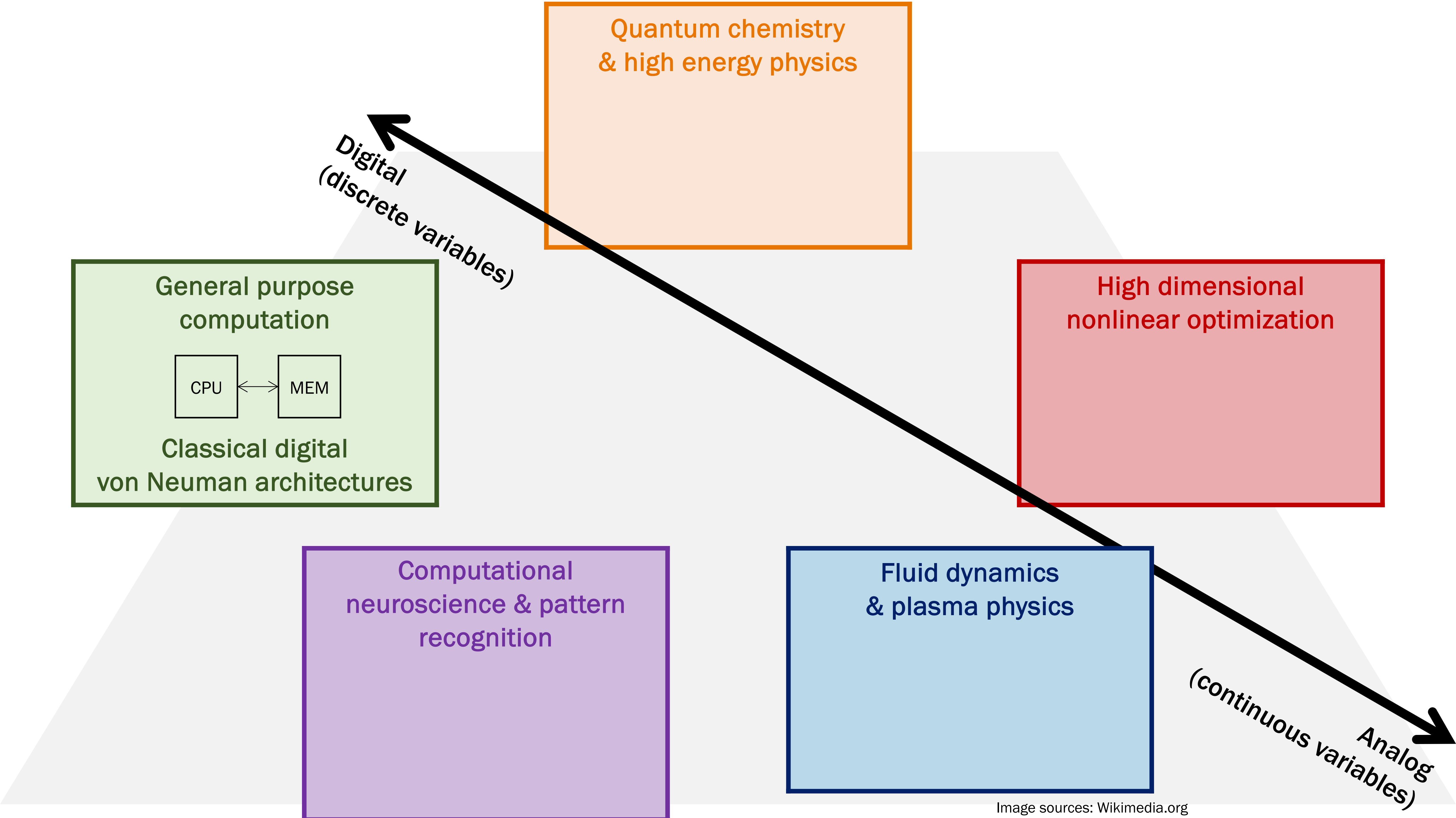


Image sources: Wikimedia.org


Quantum chemistry
& high energy physics
Digital
(discrete variables)
General purpose
computation
CPU
MEM
Classical digital
von Neuman architectures
High dimensional
nonlinear optimization
Computational
neuroscience & pattern
recognition
Fluid dynamics
& plasma physics
Analog
(continuous variables)
Image sources: Wikimedia.org

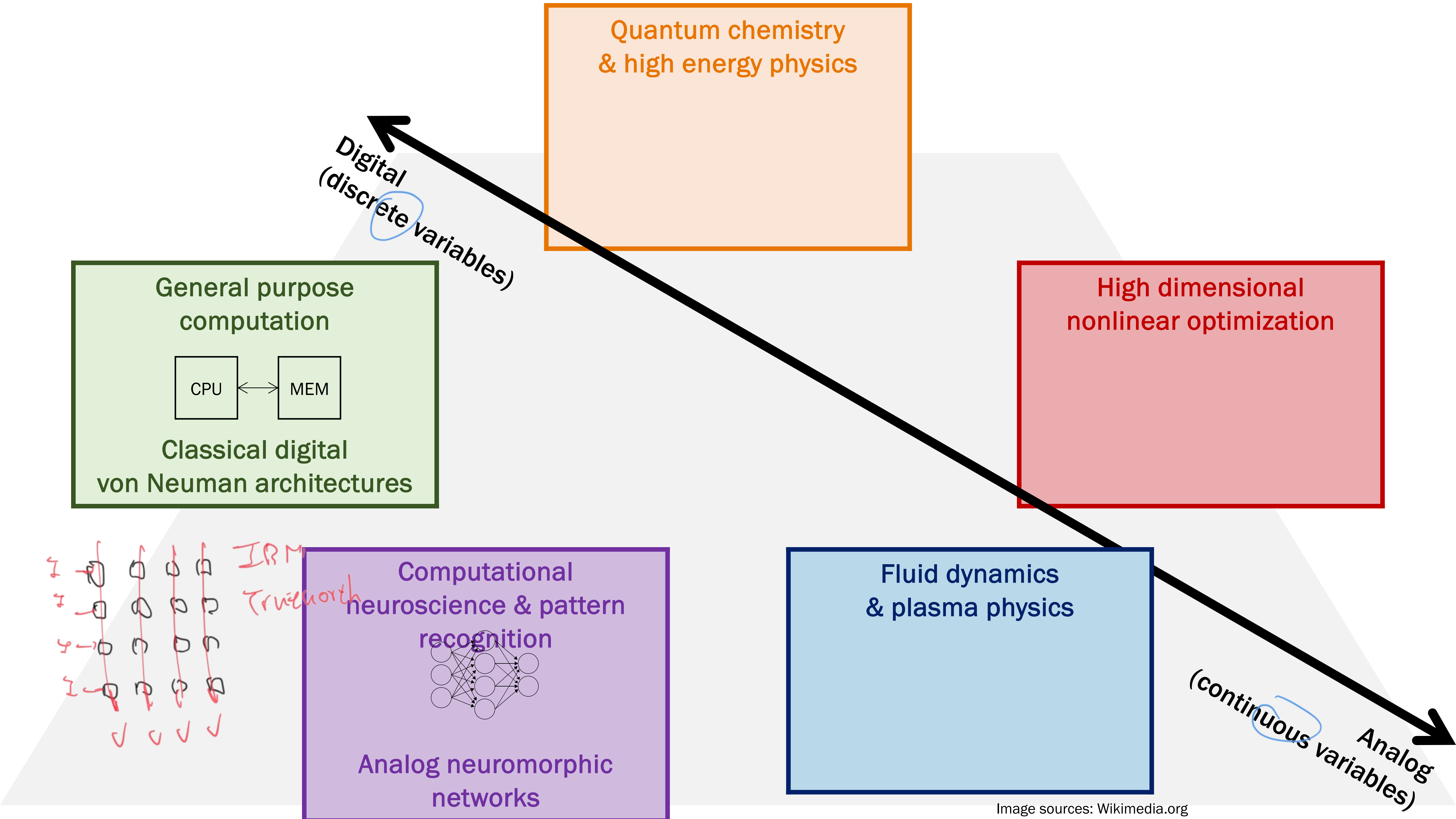
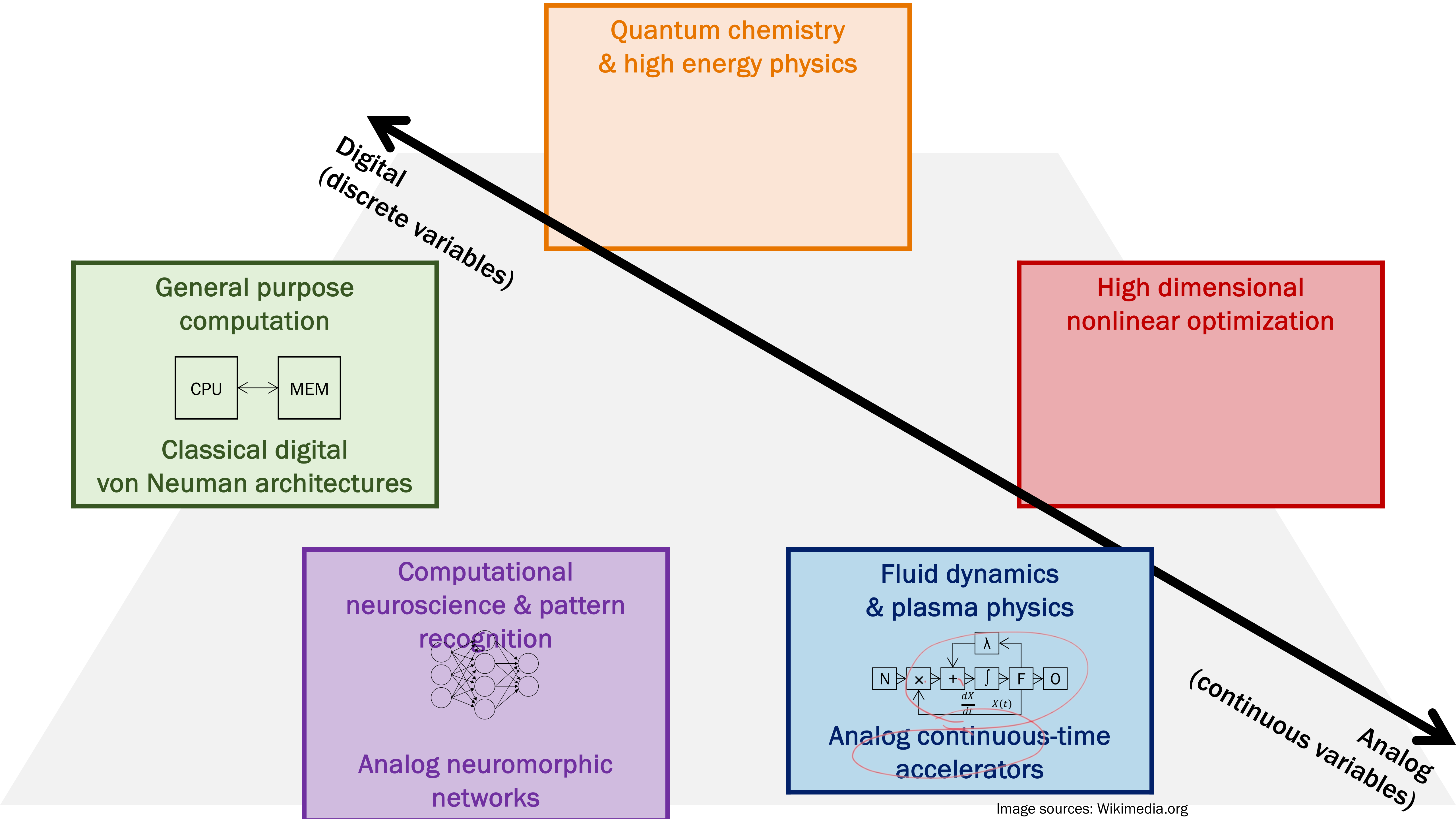
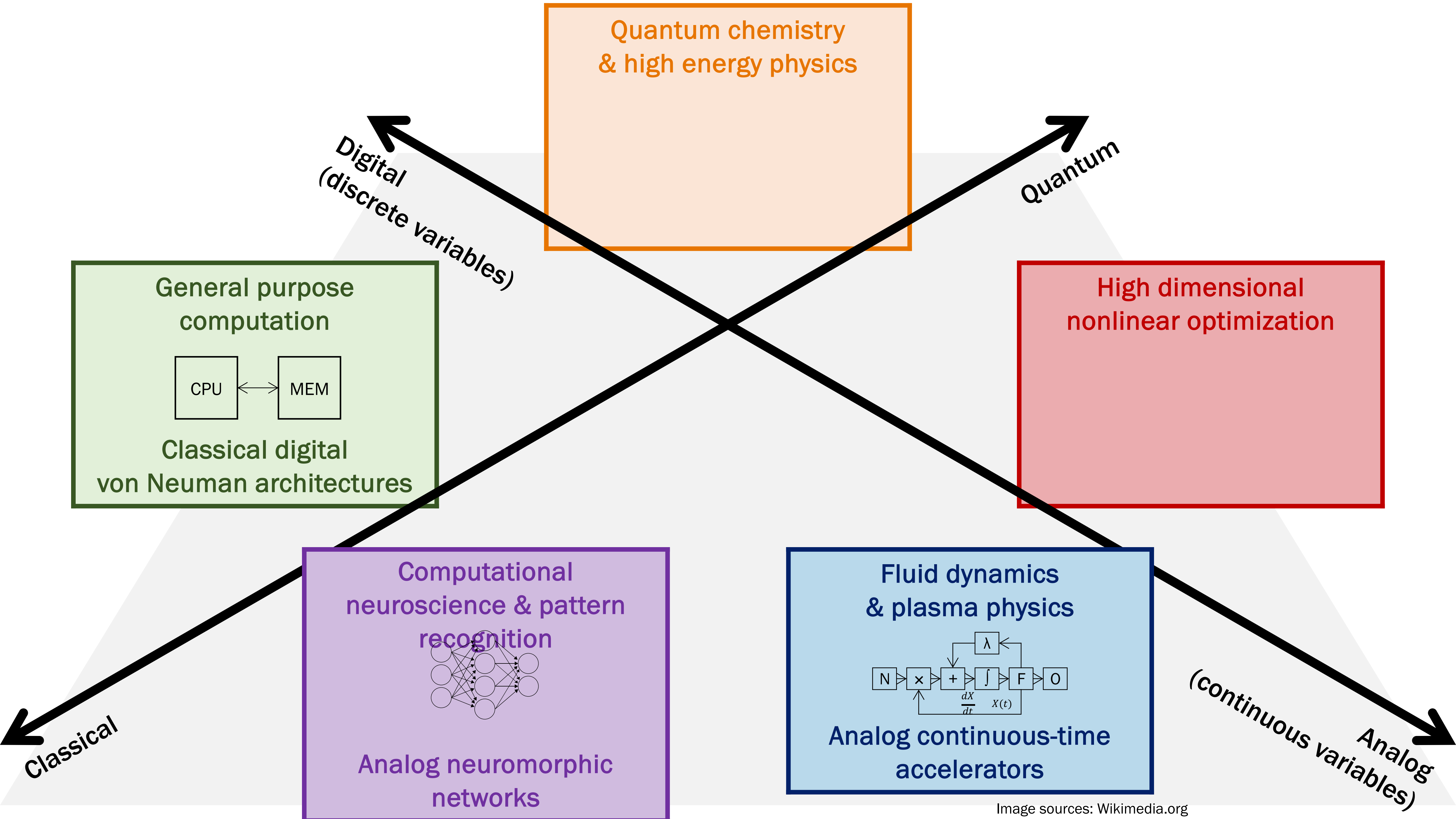


Image sources: Wikimedia.org


Quantum chemistry
& high energy physics
Digital
(discrete variables)
General purpose
computation
CPU
MEM
Classical digital
von Neuman architectures
High dimensional
nonlinear optimization
Computational
neuroscience & pattern
recognition
Analog neuromorphic
networks
Fluid dynamics
& plasma physics
λ
N
×
+
∫
F
O
X(t)
Analog continuous-time
accelerators
(continuous variables)
Analog
Image sources: Wikimedia.org


Quantum chemistry
& high energy physics
Digital
(discrete variables)
Quantum
General purpose
computation
CPU
MEM
Classical digital
von Neuman architectures
High dimensional
nonlinear optimization
Computational
neuroscience & pattern
recognition
Analog neuromorphic
networks
Fluid dynamics
& plasma physics
λ
N
×
+
∫
F
O
dX/dt
X(t)
Analog continuous-time
accelerators
Classical
(continuous variables)
Analog
Image sources: Wikimedia.org

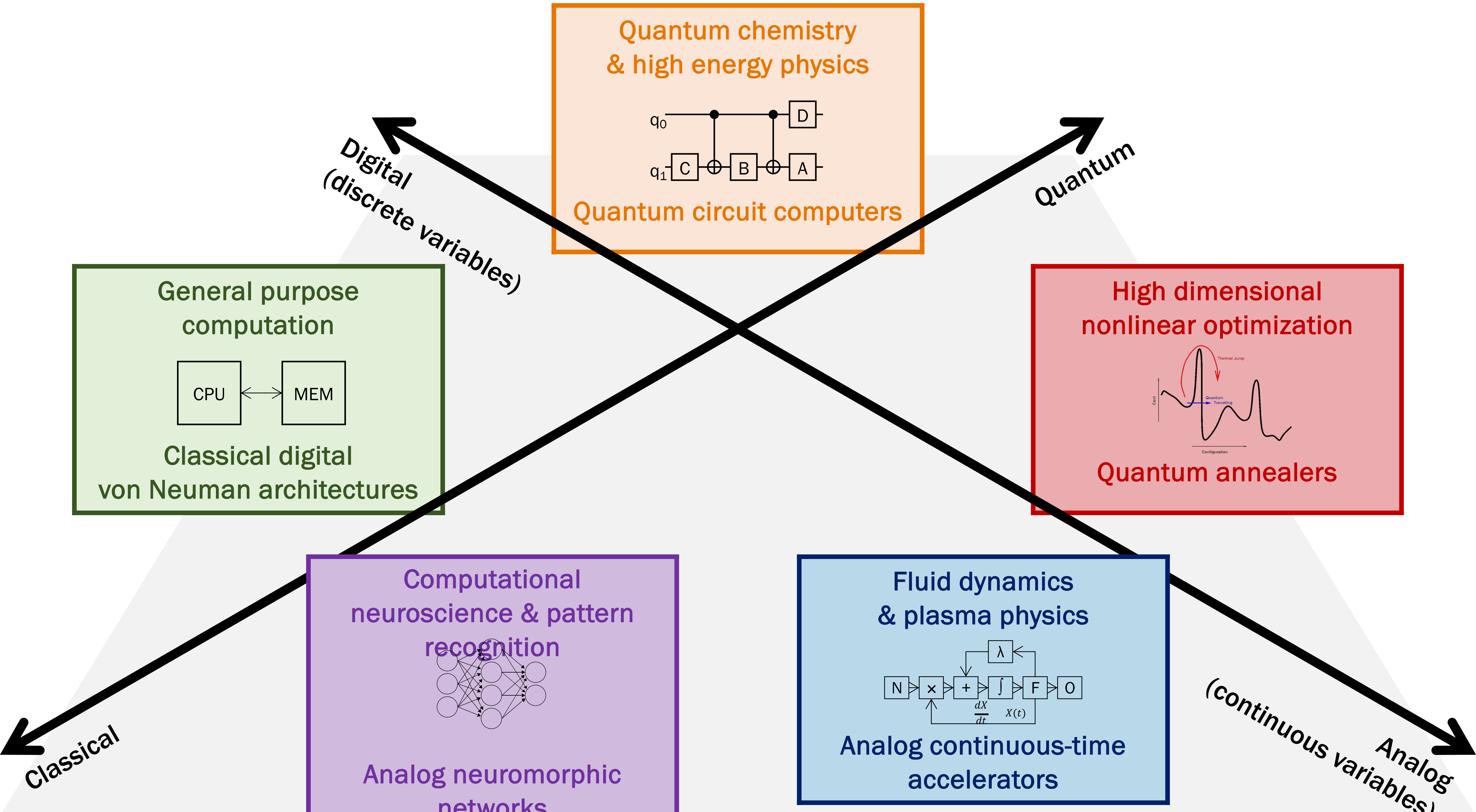
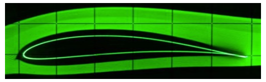
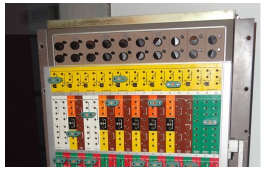


Image sources: Wikimedia.org

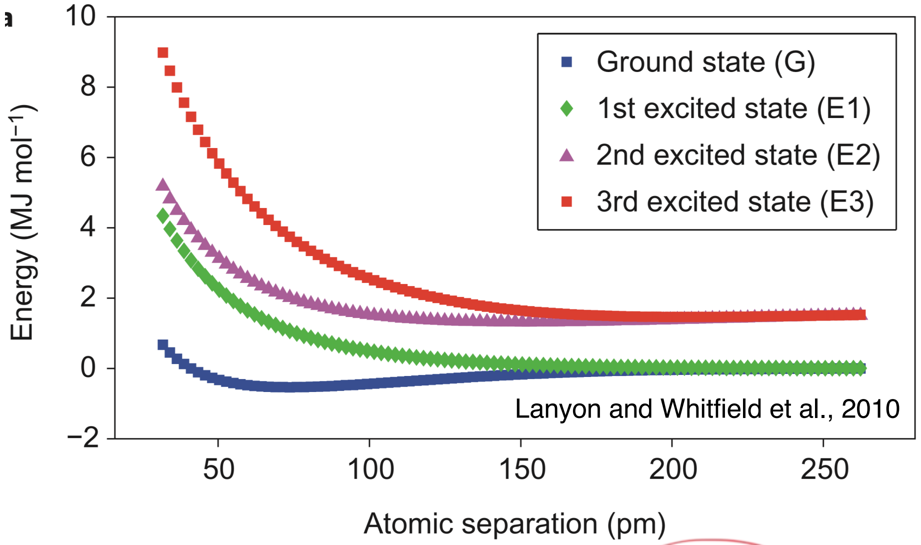
	1940s	1950s	1960s	1970s	1980s	1990s	2000s	2010s
Analog continuous-time computing	Analog computers for rocket and artillery controllers.	Analog computers for field problems. 	1st transistorized analog computer. 	Analog-digital hybrid computers.	…			
Digital discrete-time computing	Turing's *Bomba*.	1st transistorized digital computer.	Moore's law projection for transistor scaling.	Dennard's scaling for transistor power density.	VLSI democratized.	FPGAs introduced.	End of Dennard's scaling. CPUs go multicore.	Cloud FPGAs: Microsoft Catapult. Amazon F1.
						Heterogenous architectures		
	Stored program computer.	Microprogramming.	Instruction set architecture.	Reduced instruction set computers.		GPUs introduced.	Nvidia introduces CUDA.	ASICs: Google TPUs. DE Shaw Research Anton.

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

Scaling challenges drive heterogenous architectures

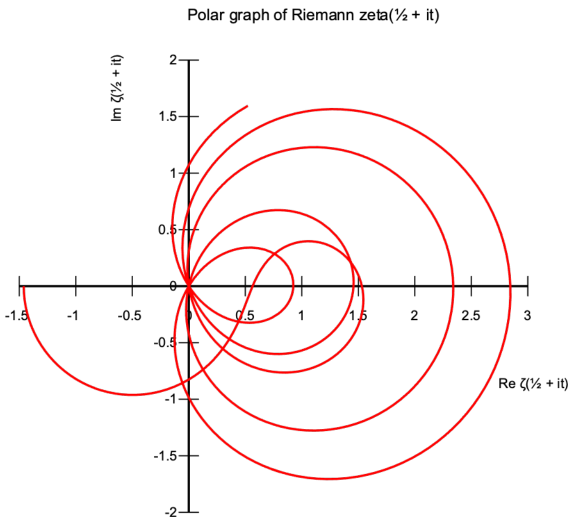
Image sources: AnalogMuseum.org, ComputerHistory.org

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

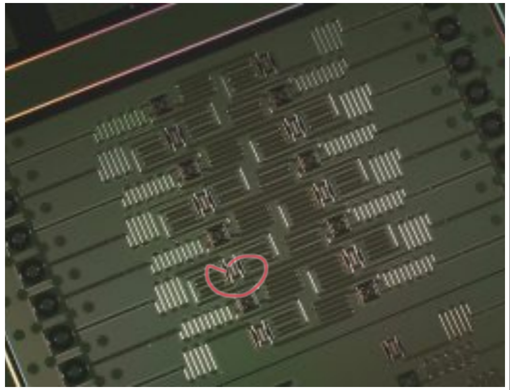
- Factor large integers to primes in polynomial time complexity
- Surpasses any known classical algorithm taking exponential time complexity

Hundreds of algorithms @ QuantumAlgorithmZoo.org

Motivation: Race to practical quantum computation

Superconducting qubits

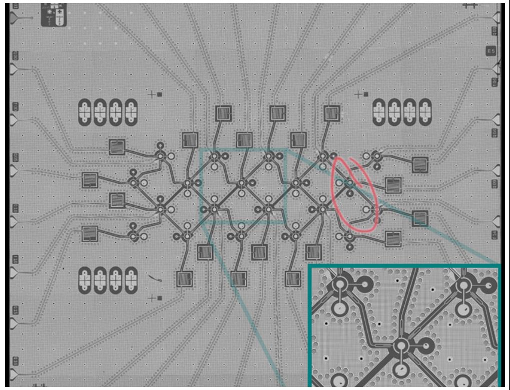
Trapped ion qubits



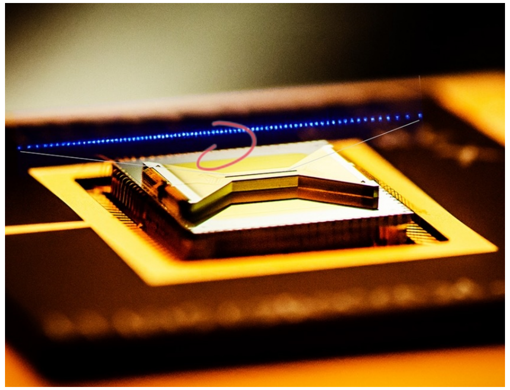
IBM

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

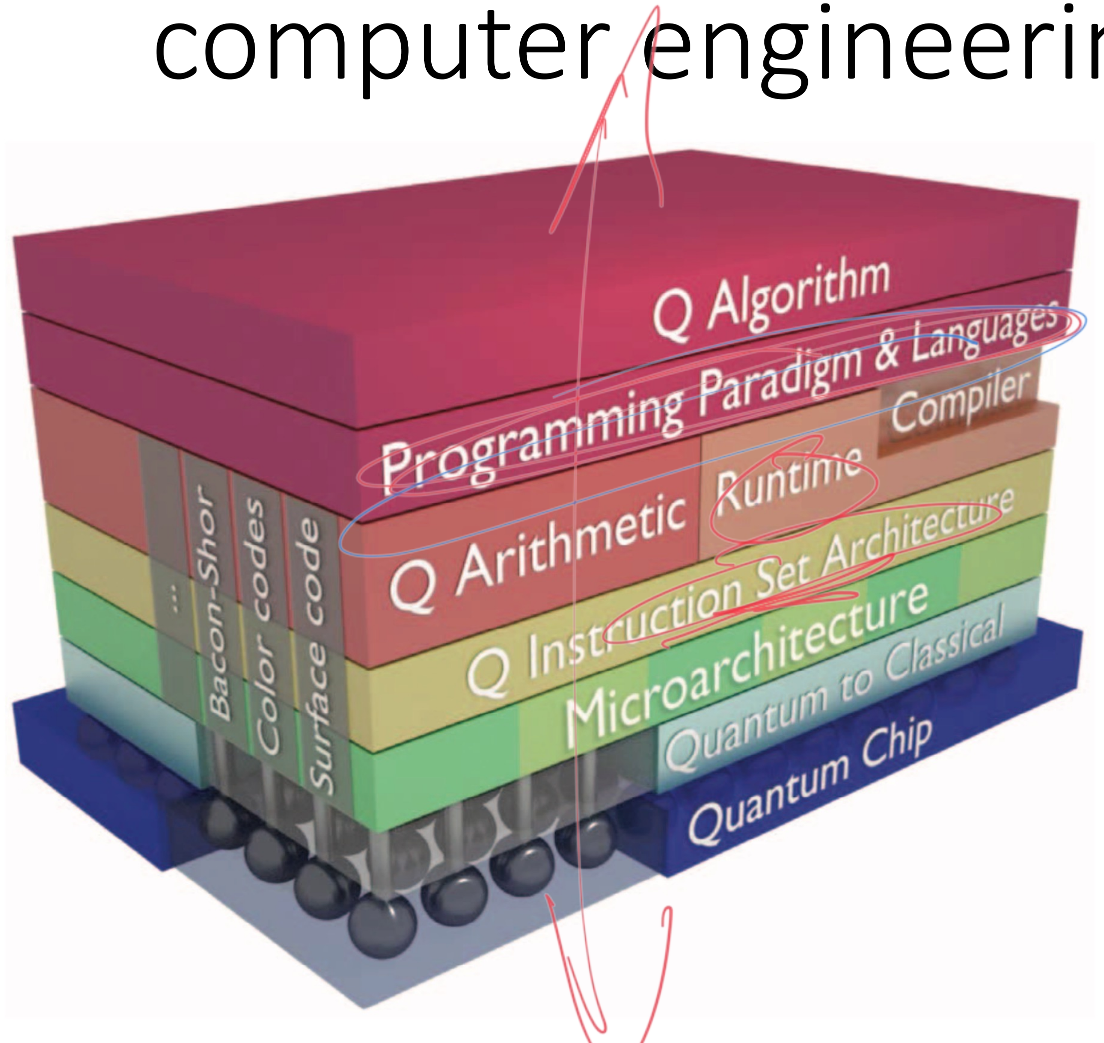


Figure 1. Overview of the quantum computer system stack.

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

- 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.

What this class is about

Course on latest developments in
<u>quantum computer engineering</u>

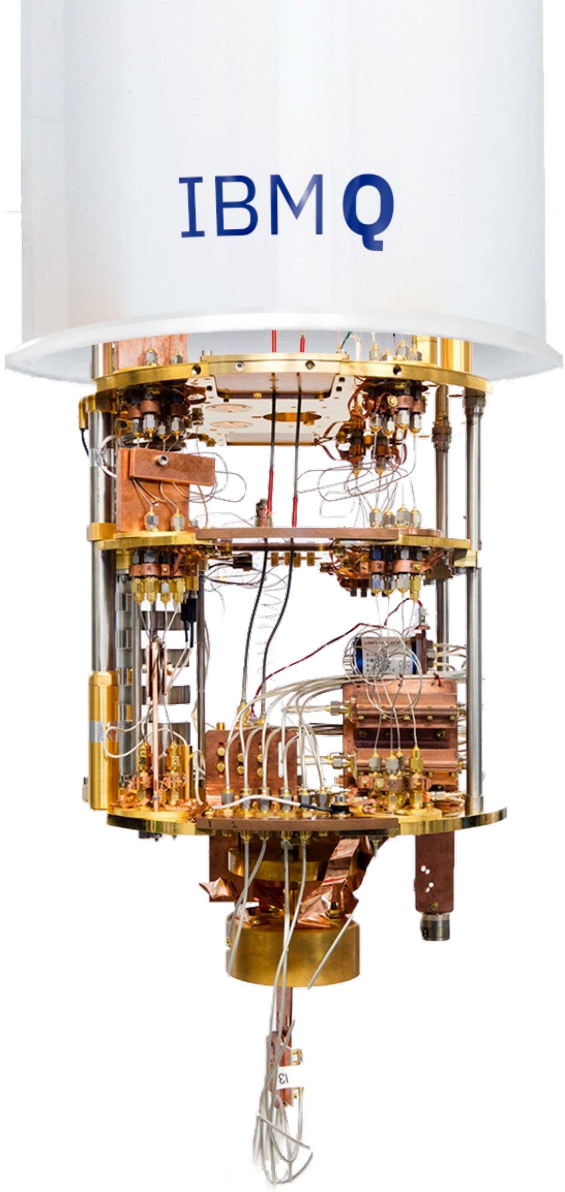
What is quantum computer engineering??

- realizing <u>quantum algorithms</u>
- on <u>prototype quantum computers</u>

—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

(Taking advice from https://www.chronicle.com/article/how-to-teach-a-good-first-day-of-class/)

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

(Taking advice from https://www.chronicle.com/article/how-to-teach-a-good-first-day-of-class/)

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

CS 206 205 CS 344

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

CS 211

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
- 2023 Fall: ECE 493/557. Soljanin. Intro to Quantum Info. Science
- 2024 Spring: Physics 421. Roy. Introduction to Quantum Computing

Quantum mechanics

- Problems and methods for quantum chemistry

Introductions via show of hands

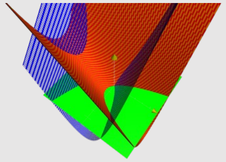
- Majors / minors: computer science, ECE, math, physics, other
- Degree progress: MS, seniors, juniors
- Who is in 558 section? Who is in 443 section?
- Experience in algorithms: CS 344 / CS 206
- Experience in architecture: CS 211 / ECE major
- Experience in Python programming

Personal introductions

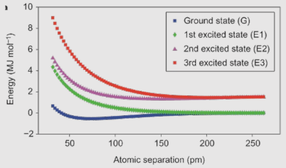
Yipeng

- 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?

Image sources: Lanyon and Whitfield et al., 2010

My work in problem and programming abstractions for emerging architectures

	Accelerator chip prototype	Support for solving differential equations	Support for solving linear algebra	Support for solving nonlinear equations	Fluid dynamics application feasibility study
Continuous-time analog scientific computation	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.

		Assertions for quantum program patterns and bugs	Graphical model inference for quantum program simulation and analysis	Analog computing support for quantum control & measurement	
Quantum algorithm debugging & simulation		ISCA 2019. mentees placed at MICRO SRC. IBM Qiskit open-source contribution.			

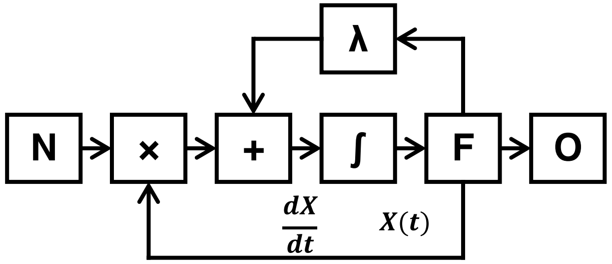
Requirements for supporting workloads

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

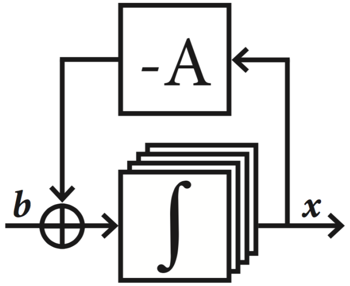
$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} - \frac{1}{\mathrm{Re}} \nabla^2 \vec{u} = RHS$$

Numerical primitives as architectural abstractions

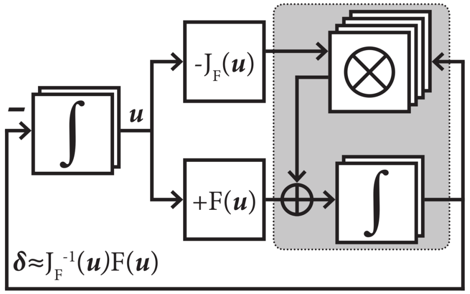
Analog-digital support for **differential equations**



Analog-digital support for **linear algebra**

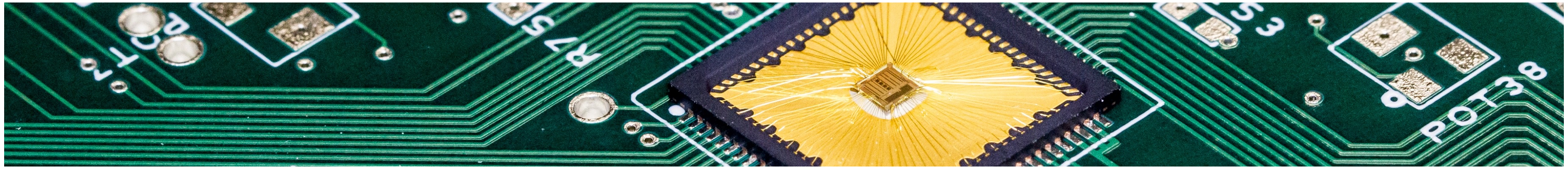


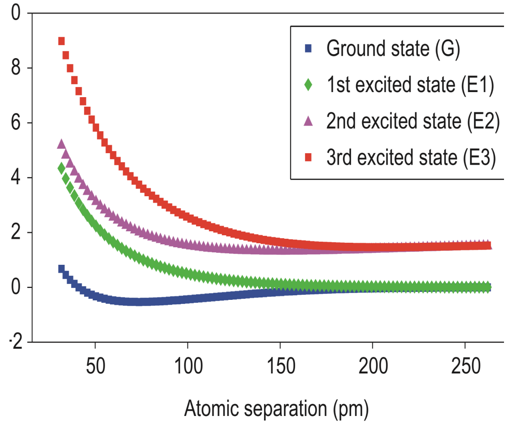
Analog-digital support for **nonlinear equations**



Unconventional architecture hardware prototyping

Prototype continuous-time analog accelerator



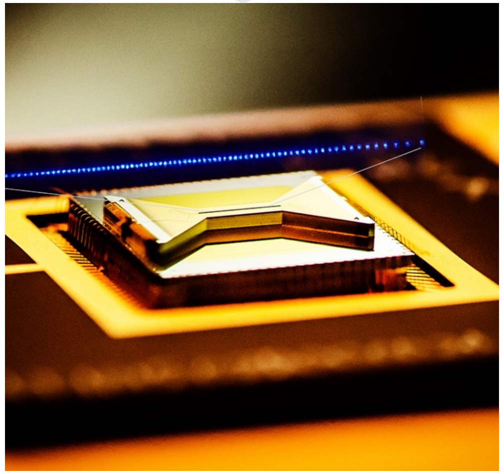


Awe-inspiring quantum algorithms

Chemistry simulations from governing equations	Shor's algorithm for factoring integers	Hundreds more near-term and far-future algorithms
Quantum computers as quantum mechanics simulator	Surpasses any known classical algorithm	QuantumAlgorithmZoo.org

My work in bridging quantum software-hardware gap

Assertions for quantum program patterns and bugs	Graphical model inference for quantum program simulation and analysis	Analog computing support for quantum control & measurement
ISCA 2019. IBM Qiskit open-source contribution.	ASPLOS 2021. Google Cirq simulation backend publicly available.	



Now-viable quantum prototypes

Superconducting qubits	Trapped ion qubits	Dozens of candidate qubit technologies
IBM, Google, Rigetti, ...	IonQ, UMD, ...	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

(Taking advice from https://www.chronicle.com/article/how-to-teach-a-good-first-day-of-class/)

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

Shor's.
Grover's

Nielsen & Chuang

Simulation.
optimization

Simulation
compilation
architecture
microarchitecture

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*

(Taking advice from https://www.chronicle.com/article/how-to-teach-a-good-first-day-of-class/)

Logistics

https://yipenghuang.com/teaching/2024-spring-558/

https://rutgers.instructure.com/courses/268800

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 558 section; 20% for 443 section).
- Graded problem sets. (24% for 558 section; 40% for 443 section).
- Programming assignments; including: QAOA implementation in Google Cirq and VQE implementation in Qiskit. (24% for 558 section; 40% for 443 section).
- Final open-ended project. (40% for 558 section; 0% for 443 section).

Proof Gradescope

Deutsch/ Deutsch-Jozsa.

Proposal Final Report. 2-3 students