Quantum Computing: Programs and Systems

Wednesday, September 2, 2020

Rutgers University

What this class is about

<u>Graduate seminar</u> on latest developments in <u>quantum computer engineering</u>

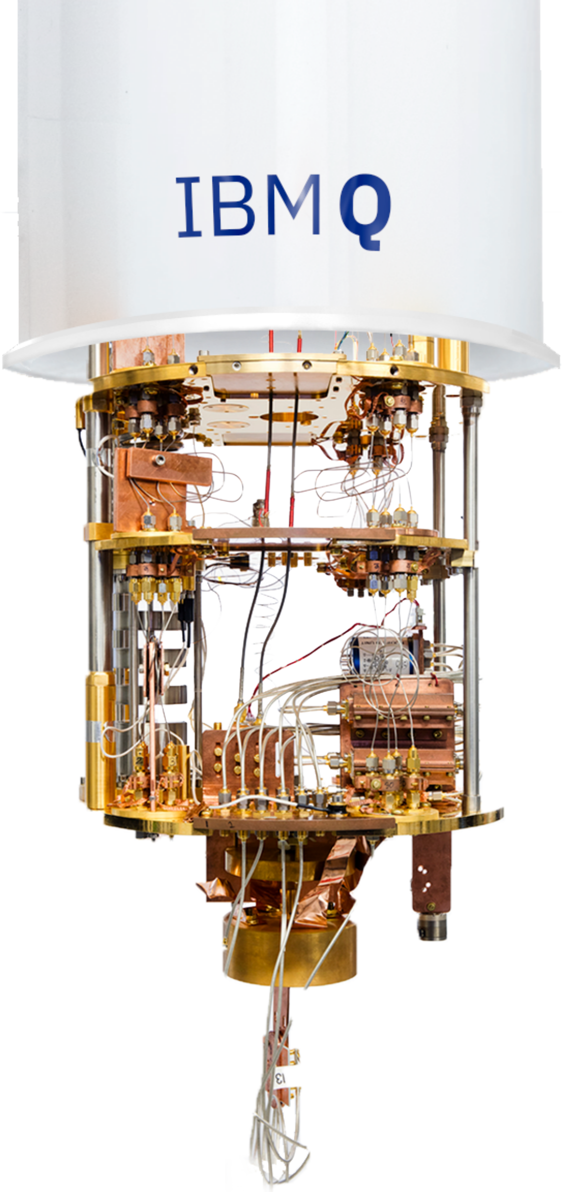
What is quantum computer engineering??

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

—a rapidly growing field!!

Goals of the course:

- read and discuss recent developments
- build foundation for you to pursue research or to be experts in industry



Prerequisites

Algorithms: time and space complexity of algorithms

Complex numbers

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

Probability and statistics

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

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

Useful, but not strictly required

Quantum information science course

- Introduction to Quantum Information Science
- Quantum Computing Systems
- Prof. Emina Soljanin, Rutgers Electrical and Computer Engineering
- Bra-ket, gates, circuits, measurement, superposition, entanglement

Quantum mechanics

- Problems and methods for quantum chemistry

Logistics

I will record videos Mondays and Wednesdays 6:40-8pm Eastern Time

Strongly recommend that you attend live

- Interact with fellow students
- Pause me to ask questions

The university is striving to accommodate in this challenging time

- Time zones
- Videos are recorded and posted

Office hours are 11 am Friday or by appointment

Survey already posted on Canvas

- Time zone?
- Can you access reading materials?
- Can you access live/recorded videos?

Outline

- Course overview
- ***<u>Personal introductions</u>***
- Why now is a good time to study quantum computing
- Preview of the syllabus
- Course objectives & activities

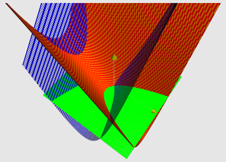
Personal introductions

Personal introductions

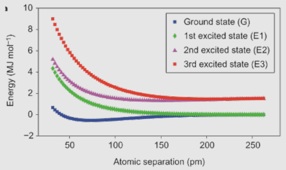
Yipeng

- 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

Continuous-time analog scientific computation	**Accelerator chip prototype**	**Support for solving differential equations**	**Support for solving linear algebra**	**Support for solving nonlinear equations**	**Fluid dynamics application feasibility study**
	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.
Quantum algorithm debugging & simulation		**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. mentees placed at MICRO SRC. IBM Qiskit open-source contribution.			

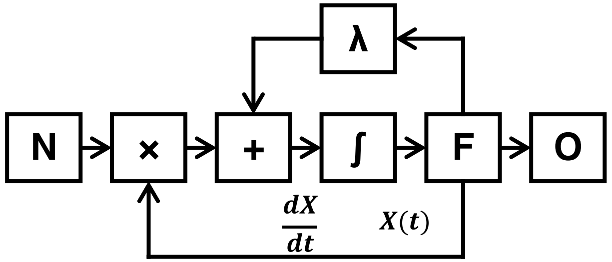
Requirements for supporting workloads

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

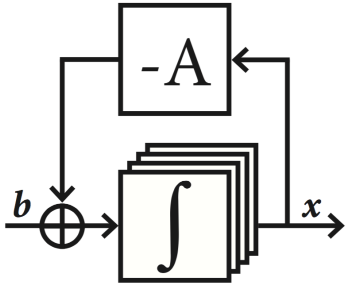
$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} - \frac{1}{\mathrm{Re}}\nabla^2 \vec{u} = RHS$$

Numerical primitives as architectural abstractions

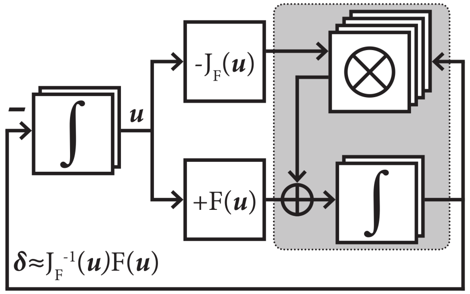
Analog-digital support for **differential equations**



Analog-digital support for **linear algebra**

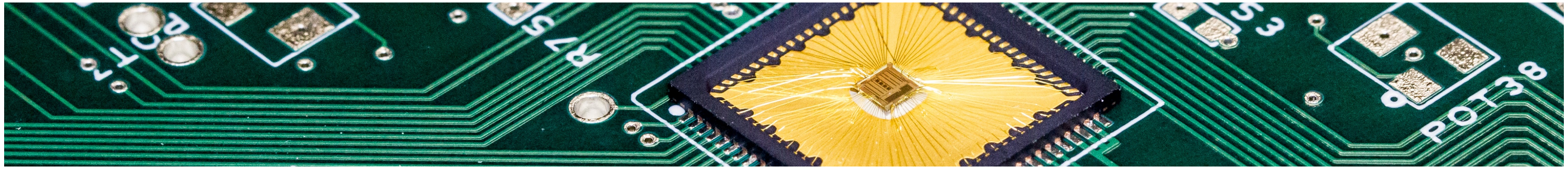


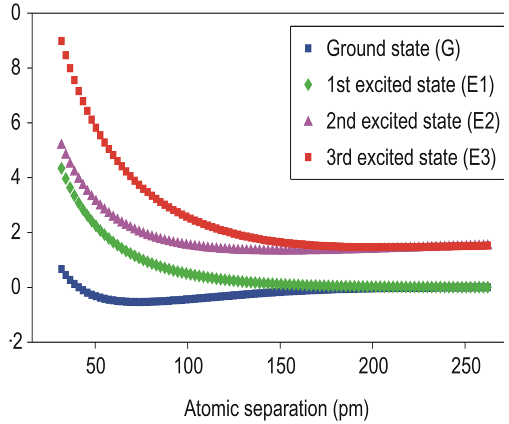
Analog-digital support for **nonlinear equations**



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.		

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

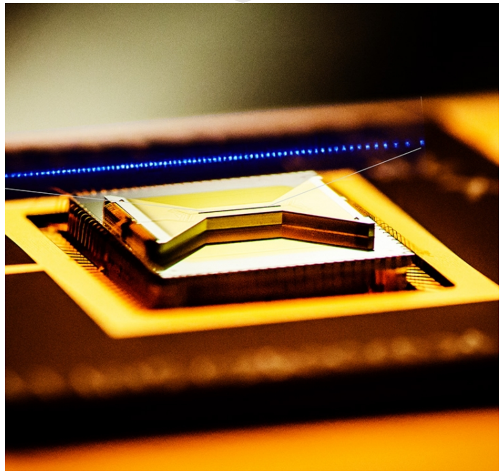


Image sources: Lanyon and Whitfield et al., 2010, IonQ

Personal introductions

Why I am excited about quantum computing:

- Broad field, rapidly changing, many new topics

Something I am interested in computer science / engineering broadly:

- New paradigms for computing

Personal introductions

My name is:

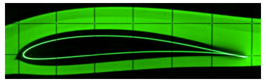
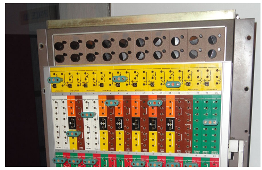
How far along I am in my studies:

Why I am interested in quantum computing:

Something I am interested in computer science / engineering broadly:

Outline

- Course overview
- Personal introductions
- ***Why now is a good time to study quantum computing***
- Preview of the syllabus
- Course objectives & activities

	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.
	Stored program computer.	Microprogramming.	Instruction set architecture.	Reduced instruction set computers.		GPUs introduced.	Nvidia introduces CUDA.	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

Image sources: AnalogMuseum.org, ComputerHistory.org

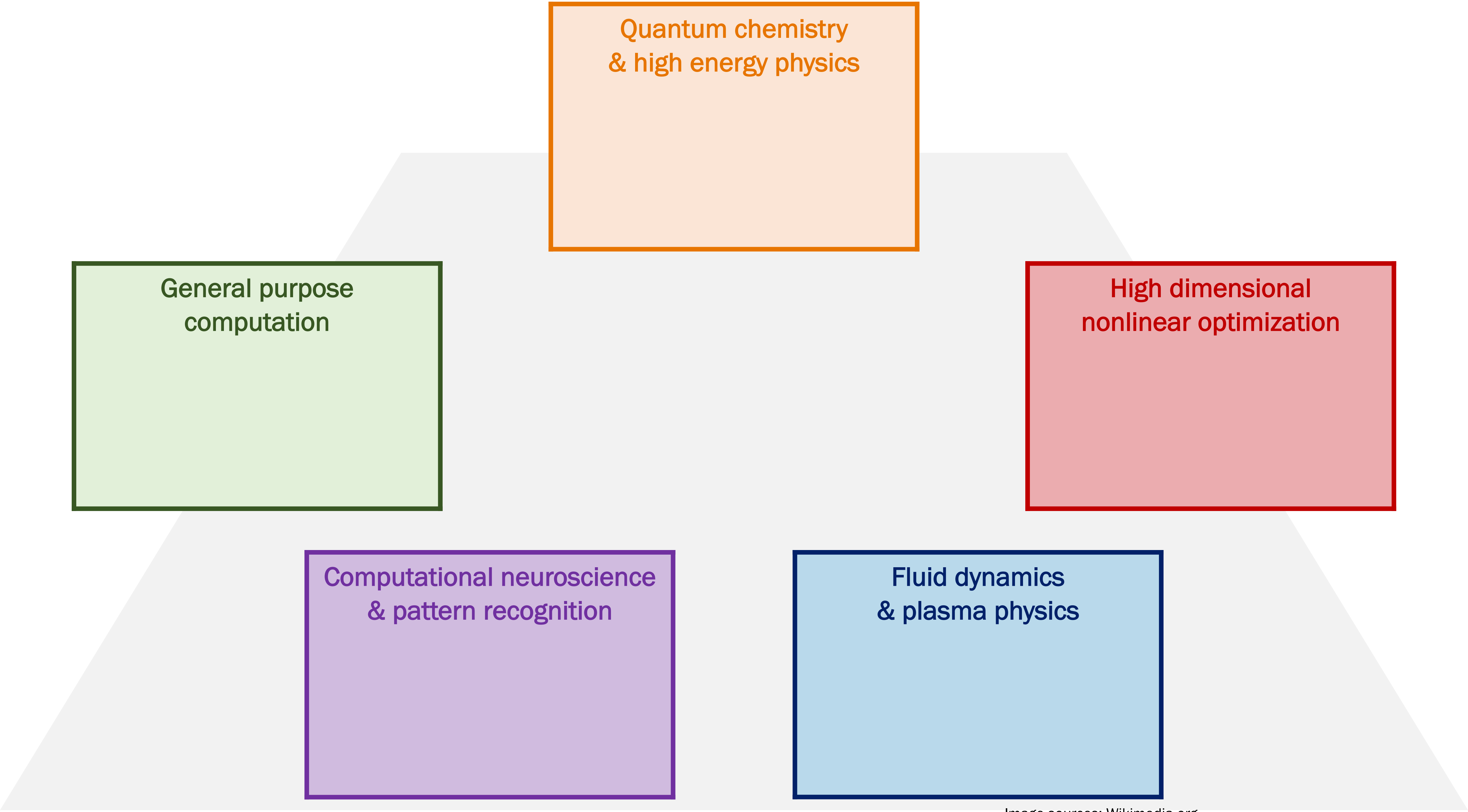


Image sources: Wikimedia.org

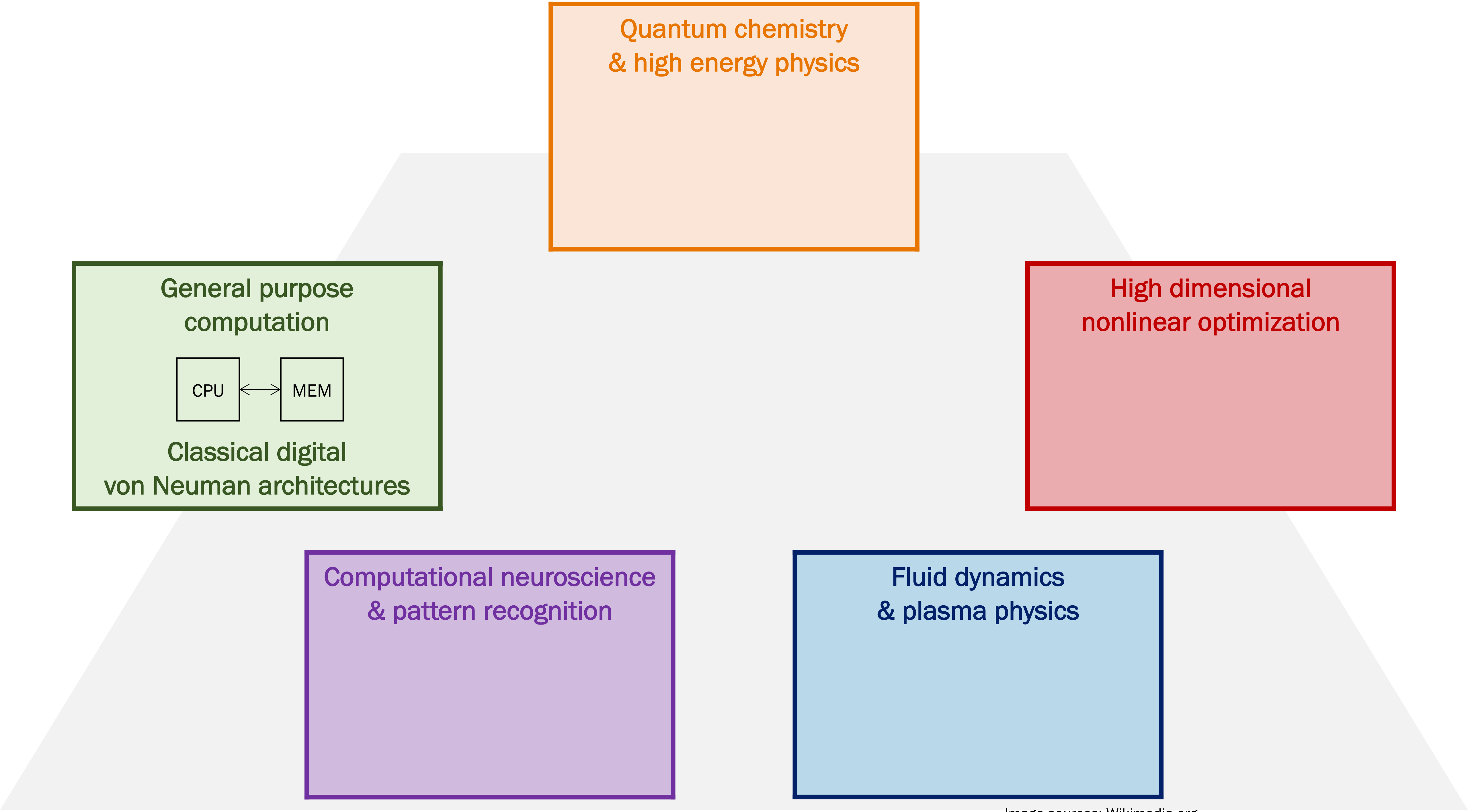
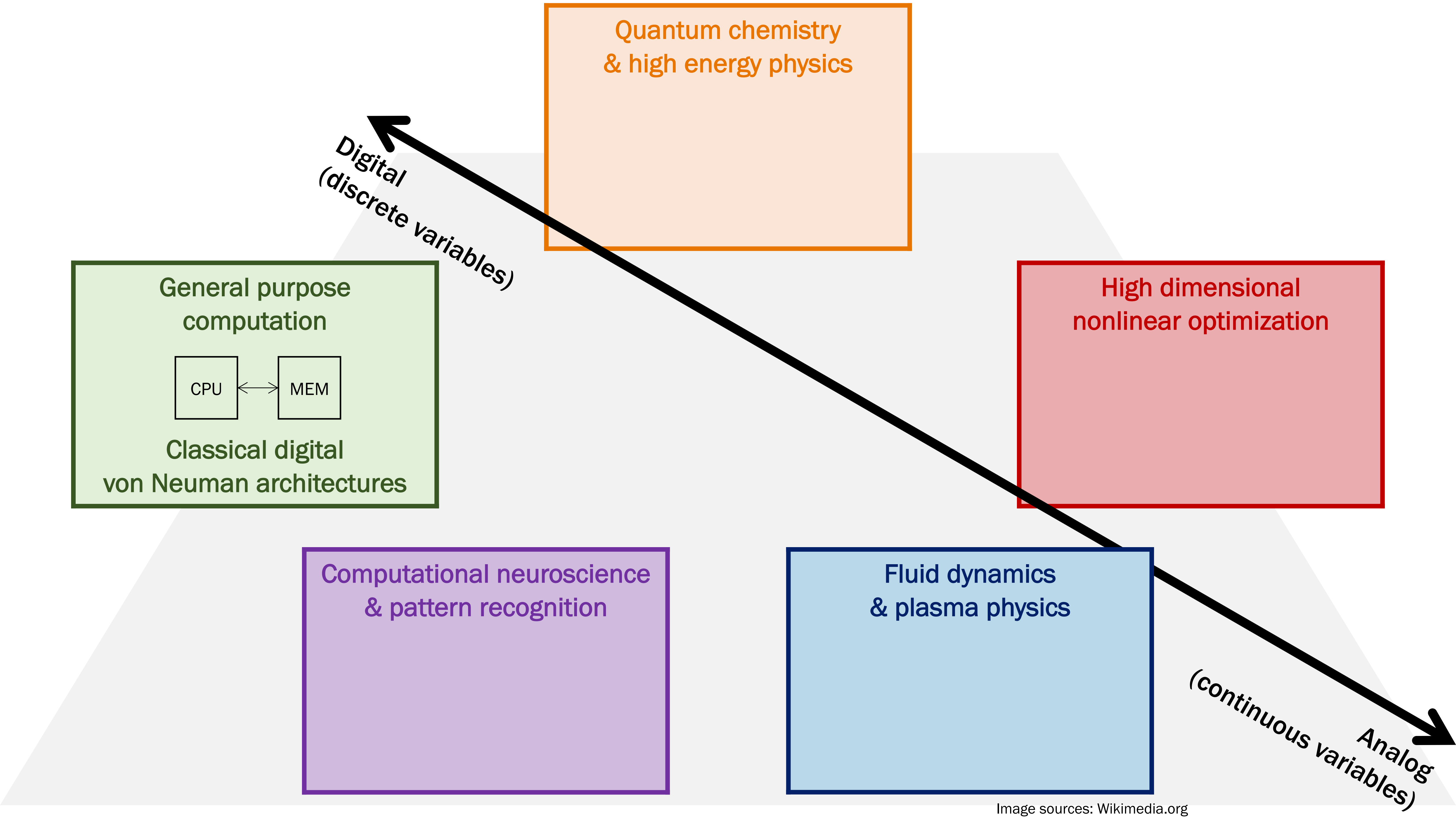
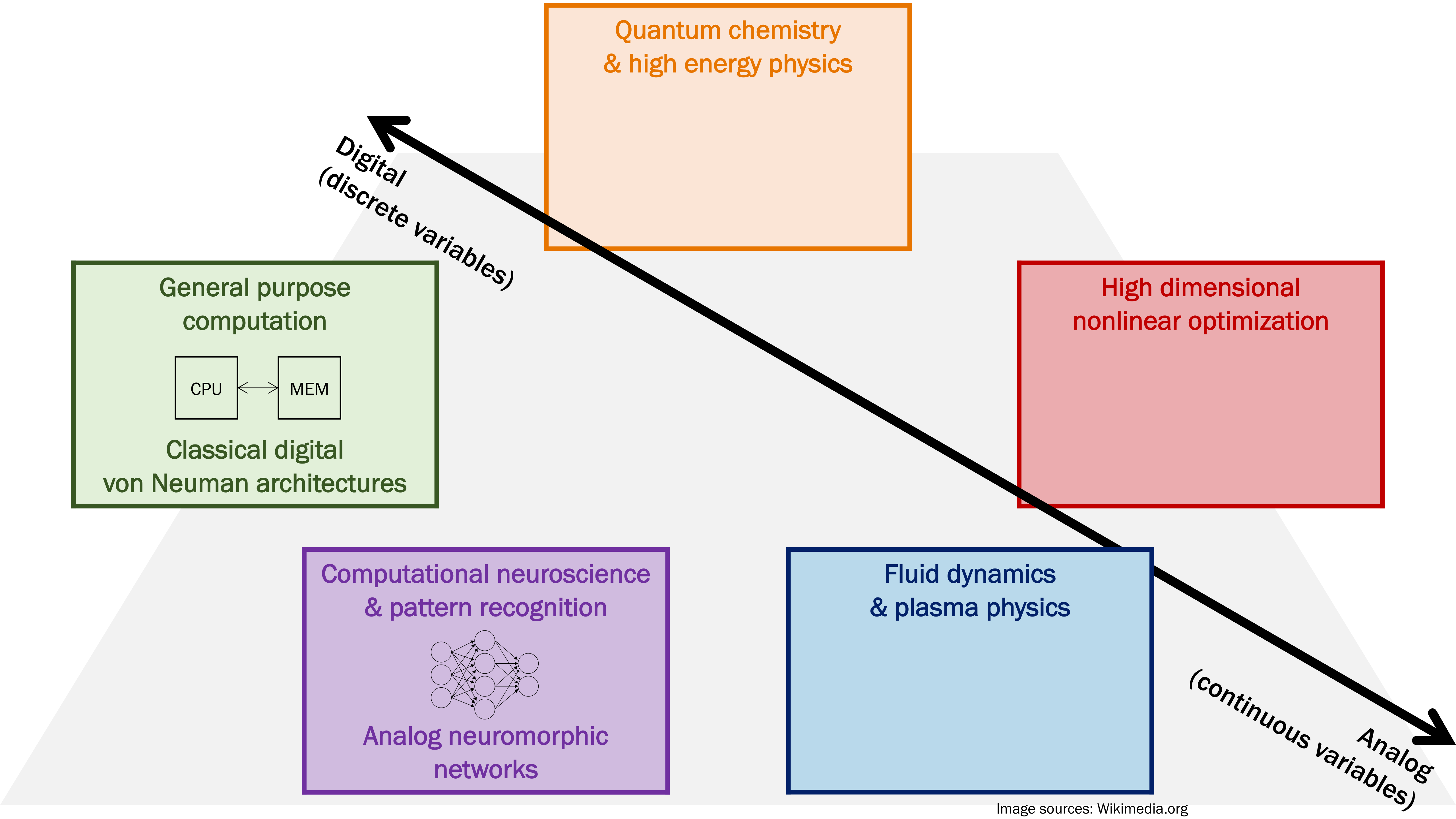
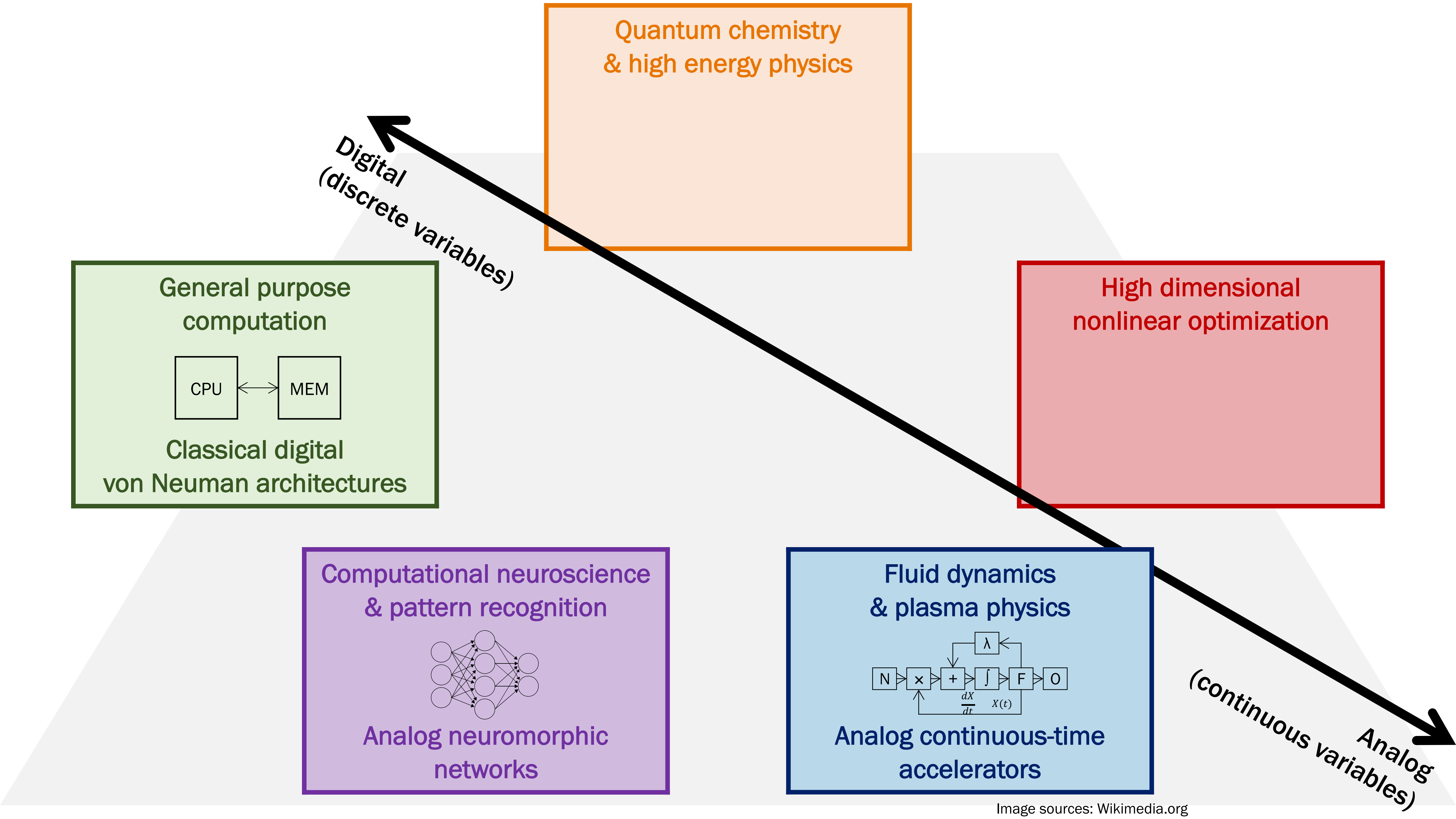
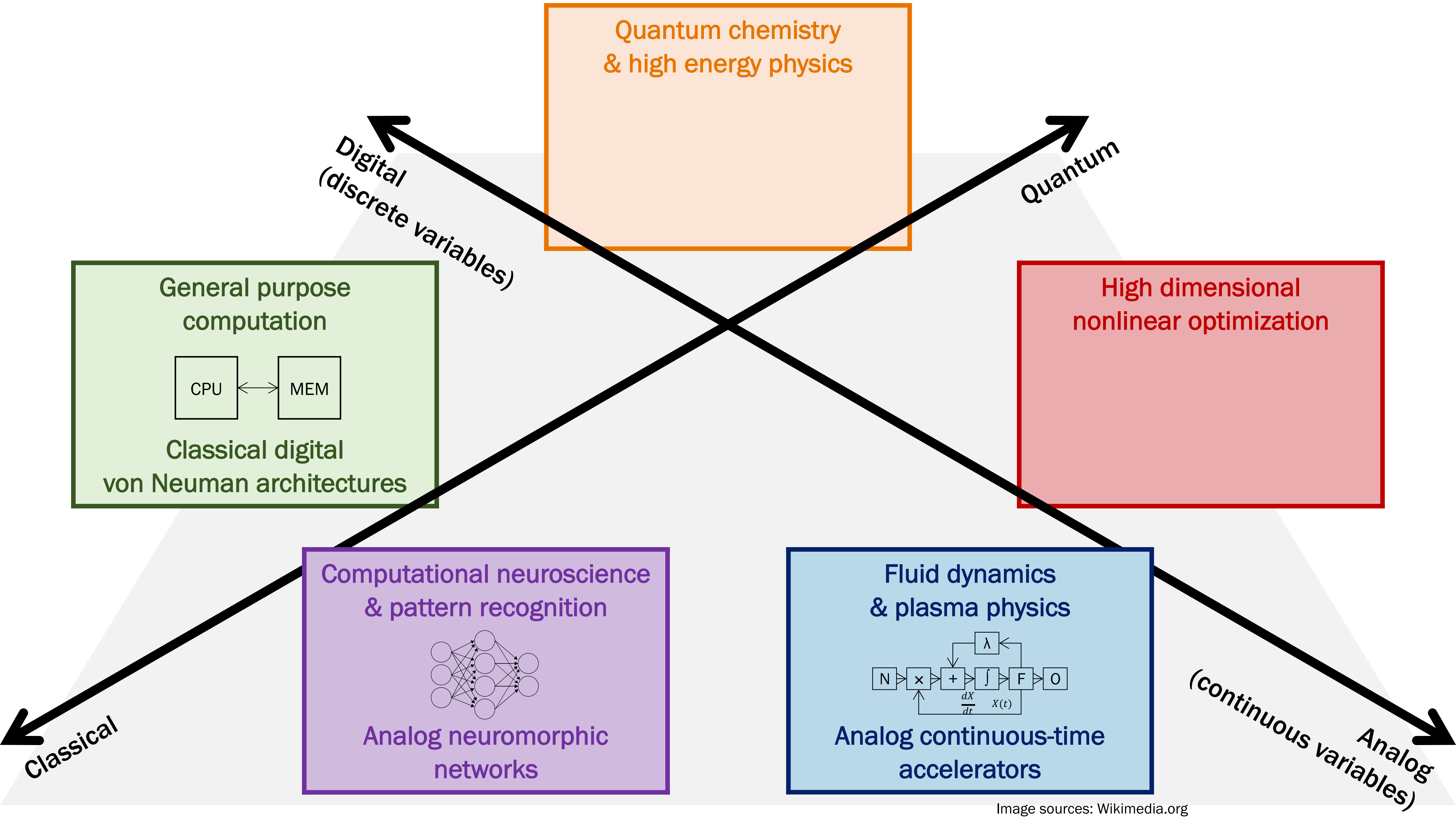


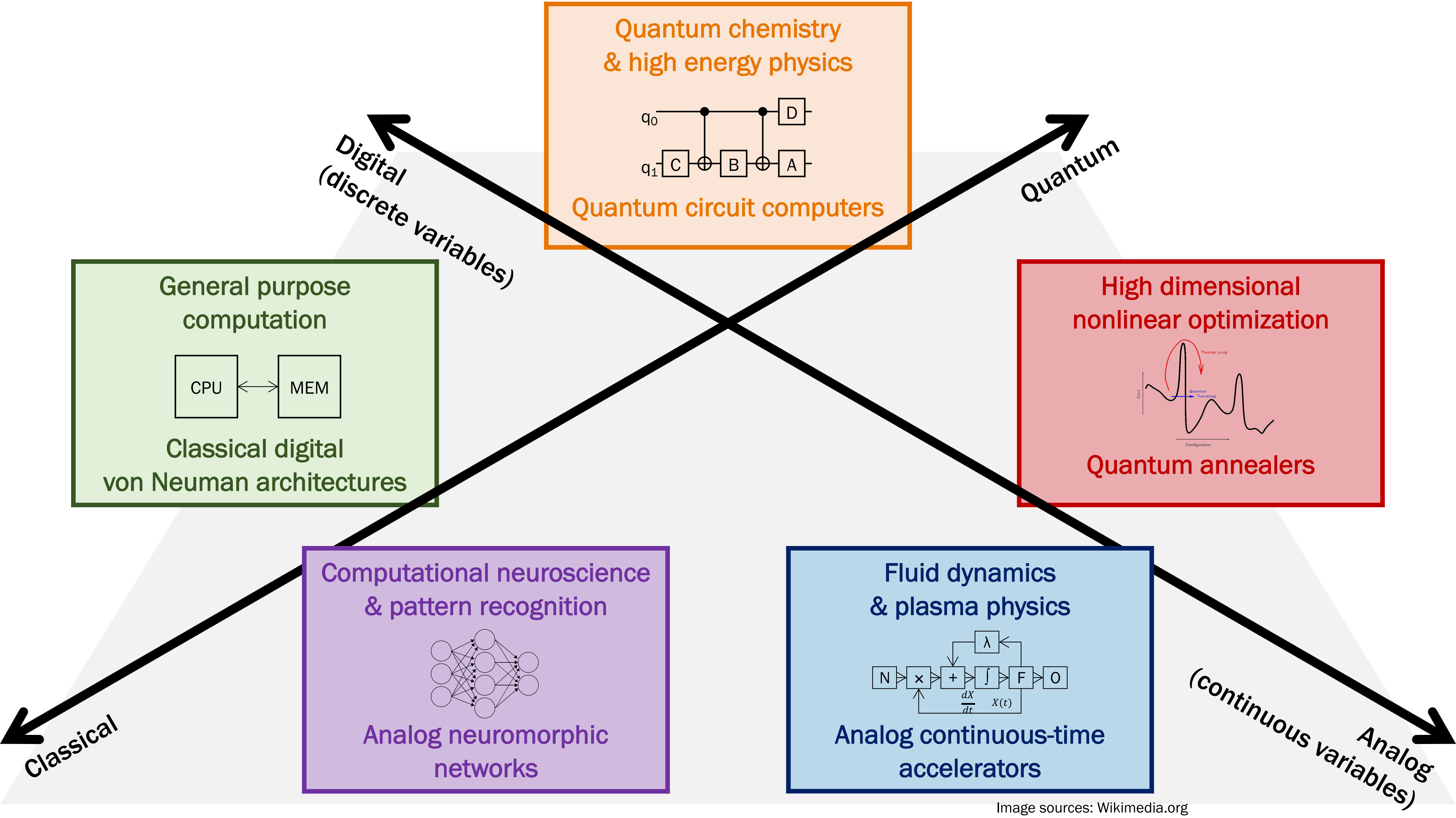
Image sources: Wikimedia.org


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

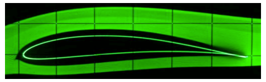
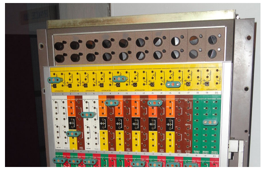
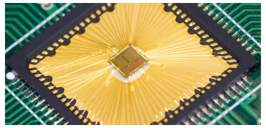

Quantum chemistry
& high energy physics
Digital
(discrete variables)
General purpose
computation
CPU
MEM
Classical digital
von Neuman architectures
High dimensional
nonlinear optimization
Computational neuroscience
& pattern recognition
Analog neuromorphic
networks
Fluid dynamics
& plasma physics
Analog
(continuous variables)
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
$\frac{dX}{dt}$
$X(t)$
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



Quantum chemistry
& high energy physics
q0
q1
C
B
D
A
Quantum circuit computers
Digital
(discrete variables)
Quantum
General purpose
computation
CPU
MEM
Classical digital
von Neuman architectures
High dimensional
nonlinear optimization
Quantum annealers
Computational neuroscience
& pattern recognition
Analog neuromorphic
networks
Fluid dynamics
& plasma physics
λ
N
×
+
∫
F
O
X(t)
Analog continuous-time
accelerators
Classical
Analog
(continuous variables)
Image sources: Wikimedia.org

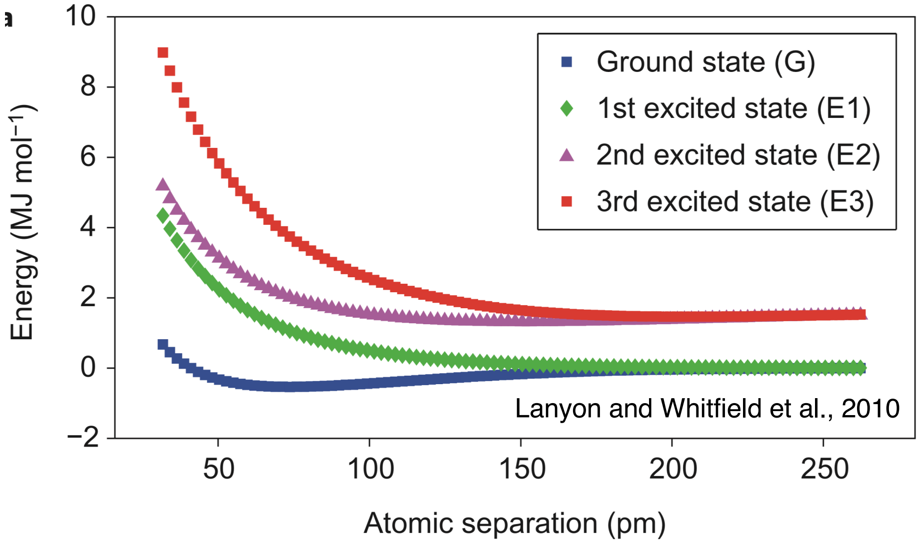
	1940s	1950s	1960s	1970s	1980s	1990s	2000s	2010s
Analog continuous-time computing	Analog computers for rocket and artillery controllers.	Analog computers for field problems. 	1st transistorized analog computer. 	Analog-digital hybrid computers.	…	Analog neural networks proposed.	VLSI analog computers proposed.	Columbia University prototype analog accelerators. 
Digital discrete-time computing	Turing's *Bomba*. Stored program computer.	1st 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.			CPUs go multicore.	
Quantum computing					Feynman. "Simulating Physics with Computers."	Shor's algorithm. Demo of ion trap quantum computation.	Demo of super-conductor quantum computation.	IBM quantum cloud. Google quantum supremacy.

Challenges in digital scaling motivate revisiting analog alternative

Fundamental limits in digital model of computing motivate exploring quantum possibilities

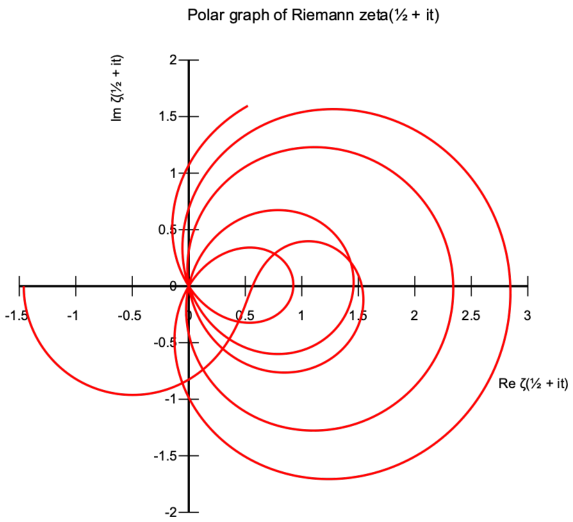
Image sources: AnalogMuseum.org, ComputerHistory.org

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!



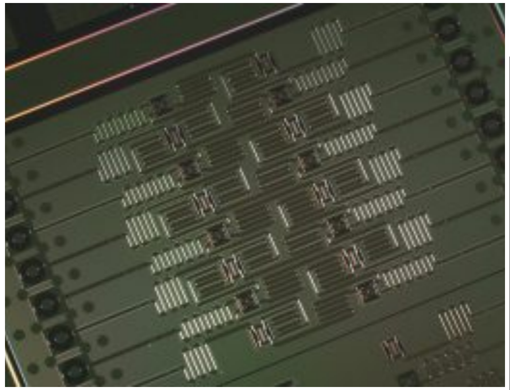
Shor's quantum algorithm for factoring integers

- 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

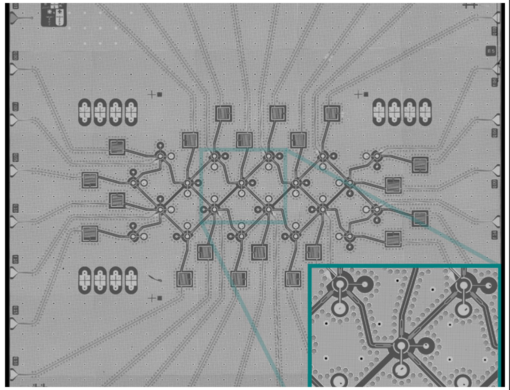
Superconducting qubits



IBM

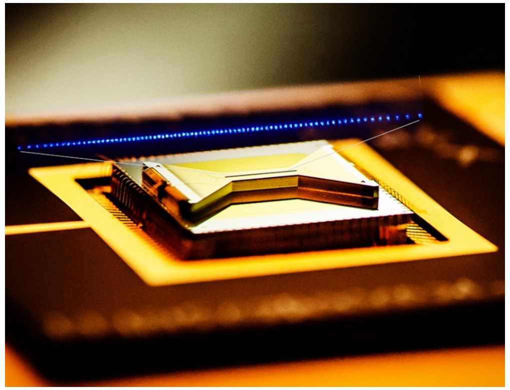
Google

Intel



Rigetti

Trapped ion qubits



University of Maryland / IonQ

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

Research here at Rutgers

- Prof. Emina Souljanin, quantum communications
- Prof. Zheng Zhang, quantum circuit compilation
- Prof. Yipeng Huang, quantum program simulation and analysis
- Prof. Mario Szegedy, quantum algorithms, complexity theory

Outline

- Course overview
- Personal introductions
- Why now is a good time to study quantum computing
- ***Preview of the syllabus***
- Course objectives & activities

Preview of the syllabus

- A systems view of quantum computer engineering
- Near-term intermediate-scale quantum algorithms
- Programming frameworks
- Emerging languages and representations
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes

Preview of the syllabus

- ***<u>A systems view of quantum computer engineering</u>***
- Near-term intermediate-scale quantum algorithms
- Programming frameworks
- Emerging languages and representations
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes

Semantic gap

- Need languages, abstractions…

Tools gap

- Need optimizing compilers, simulators, debuggers…

Infrastructure gap

- Need more abundant, more reliable qubits…

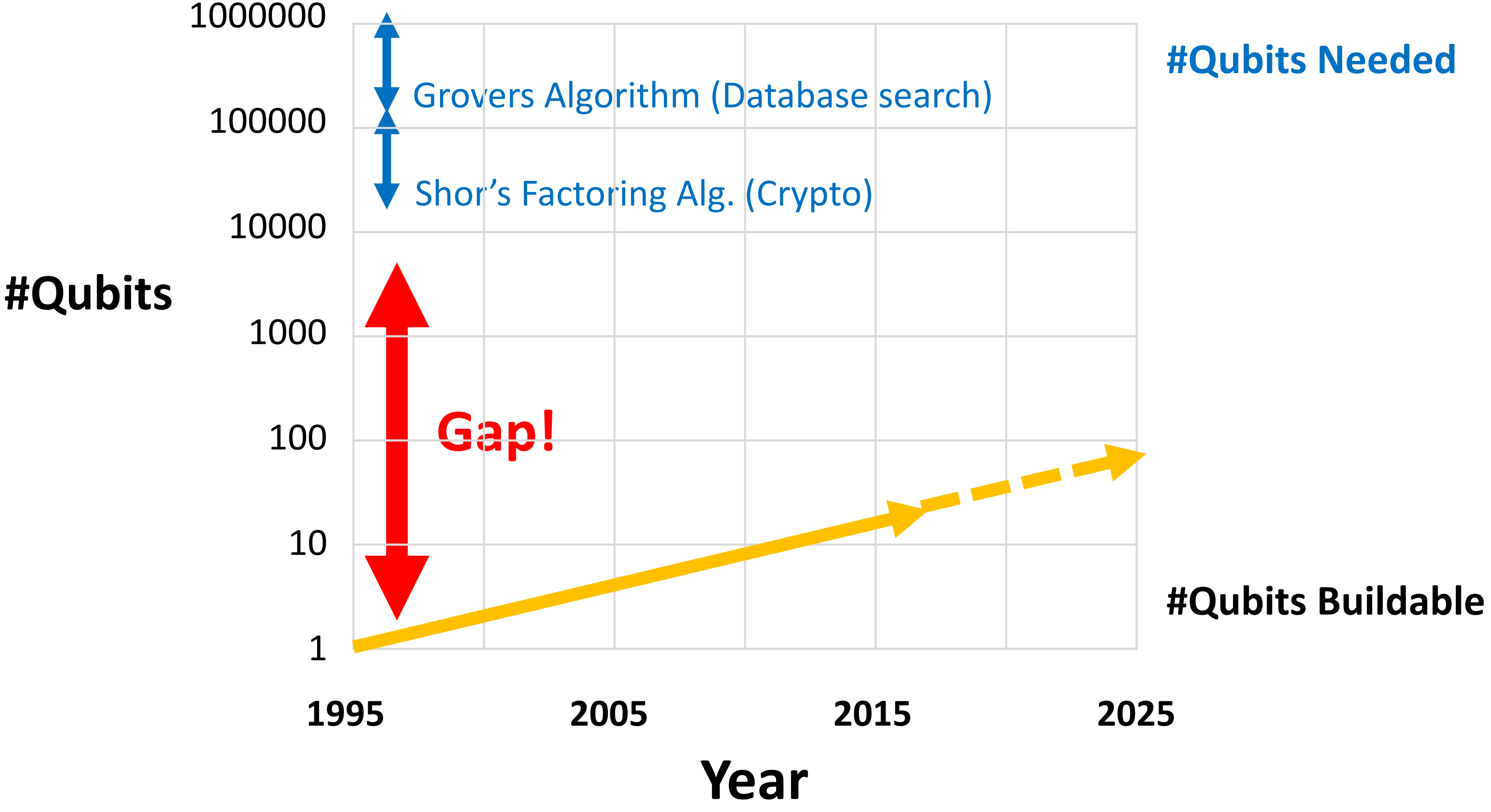
Educational gap

- Need researchers, students…

Preview of the syllabus

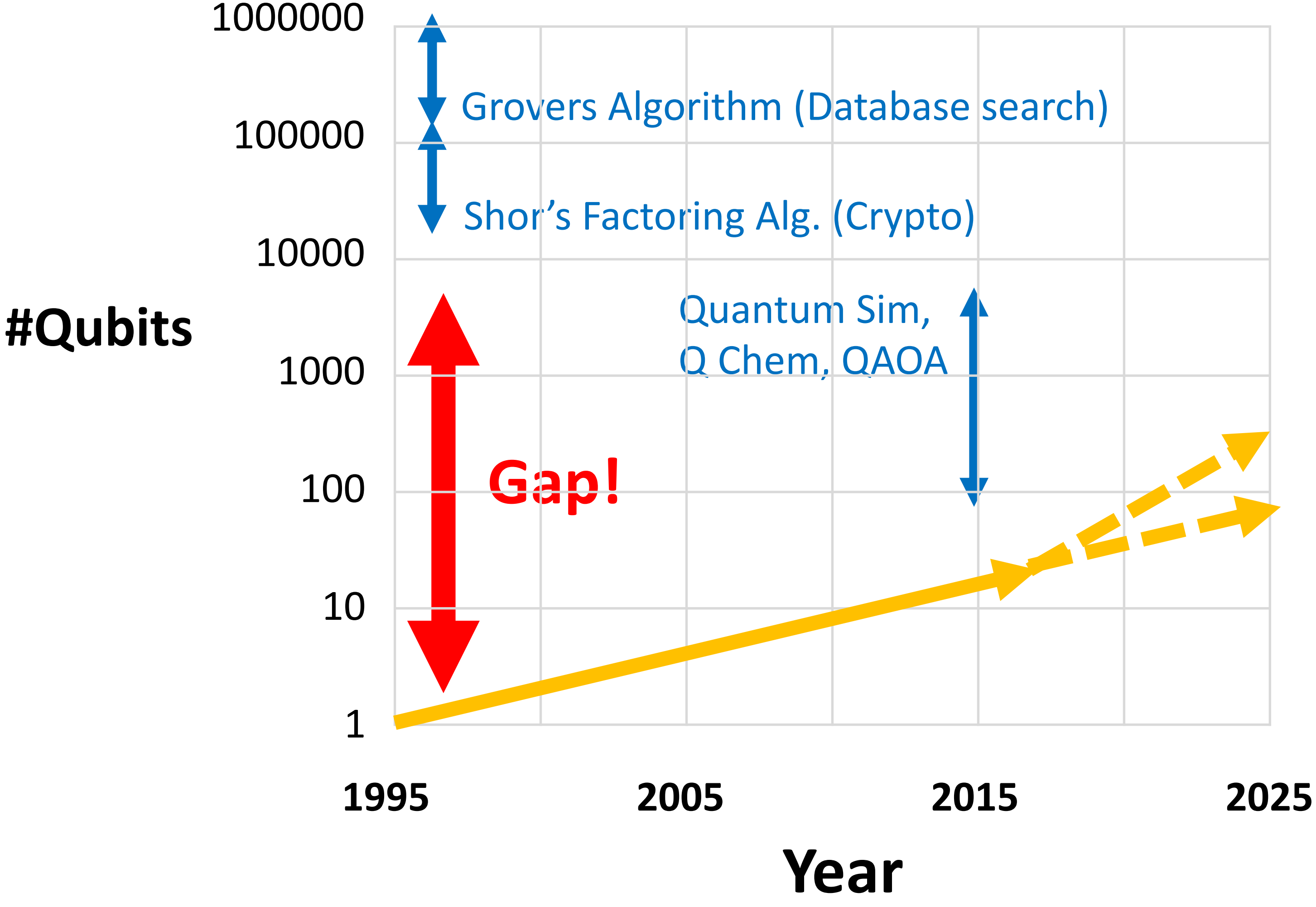
- A systems view of quantum computer engineering
- ***<u>Near-term intermediate-scale quantum algorithms</u>***
- Programming frameworks
- Emerging languages and representations
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes

Algorithms to Machines Gap



Source: Prof. Margaret Martonosi

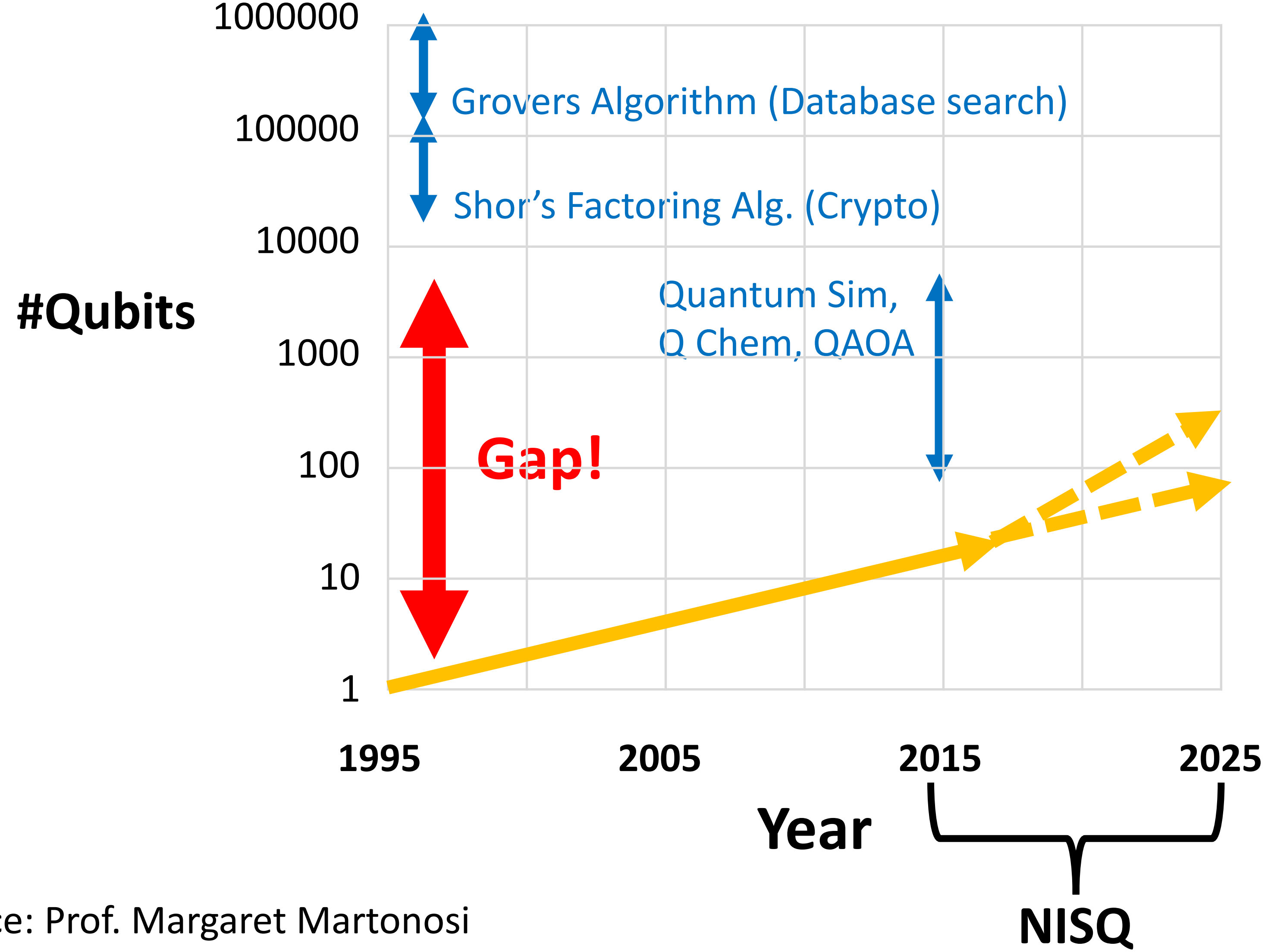
Algorithms to Machines Gap: Algorithm Progress



- New breed of QC algorithm:
  - Lower qubit needs
  - Iterative with classical phases
  - Not exponential speedup, but promising demonstrations
- Hundreds of QC Algorithms in Quantum Zoo
- https://math.nist.gov/quantum/zoo/

Source: Prof. Margaret Martonosi

Algorithms to Machines Gap: Next Ten Years = NISQ Era



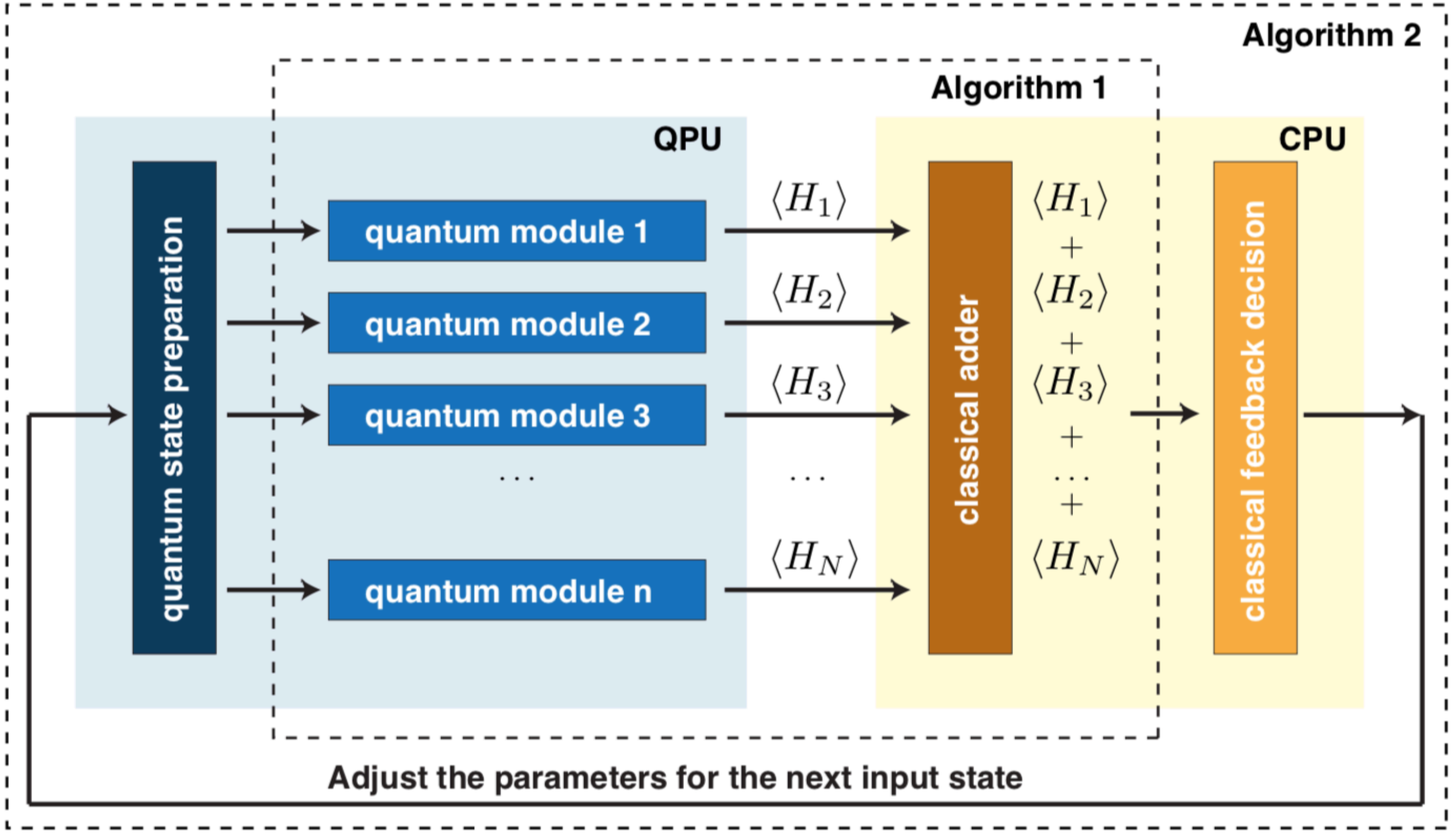
- Noisy Intermediate-Scale Quantum (NISQ)
  - Preskill, Jan 2018
  - 10-1000 qubits
- Large enough to support interesting experiments
- Too small for known algorithms with exponential speedup
- Too small for ECC

Source: Prof. Margaret Martonosi

What are variational algorithms and why are they important?

- It’s like using a classical computer to train a quantum neural network.

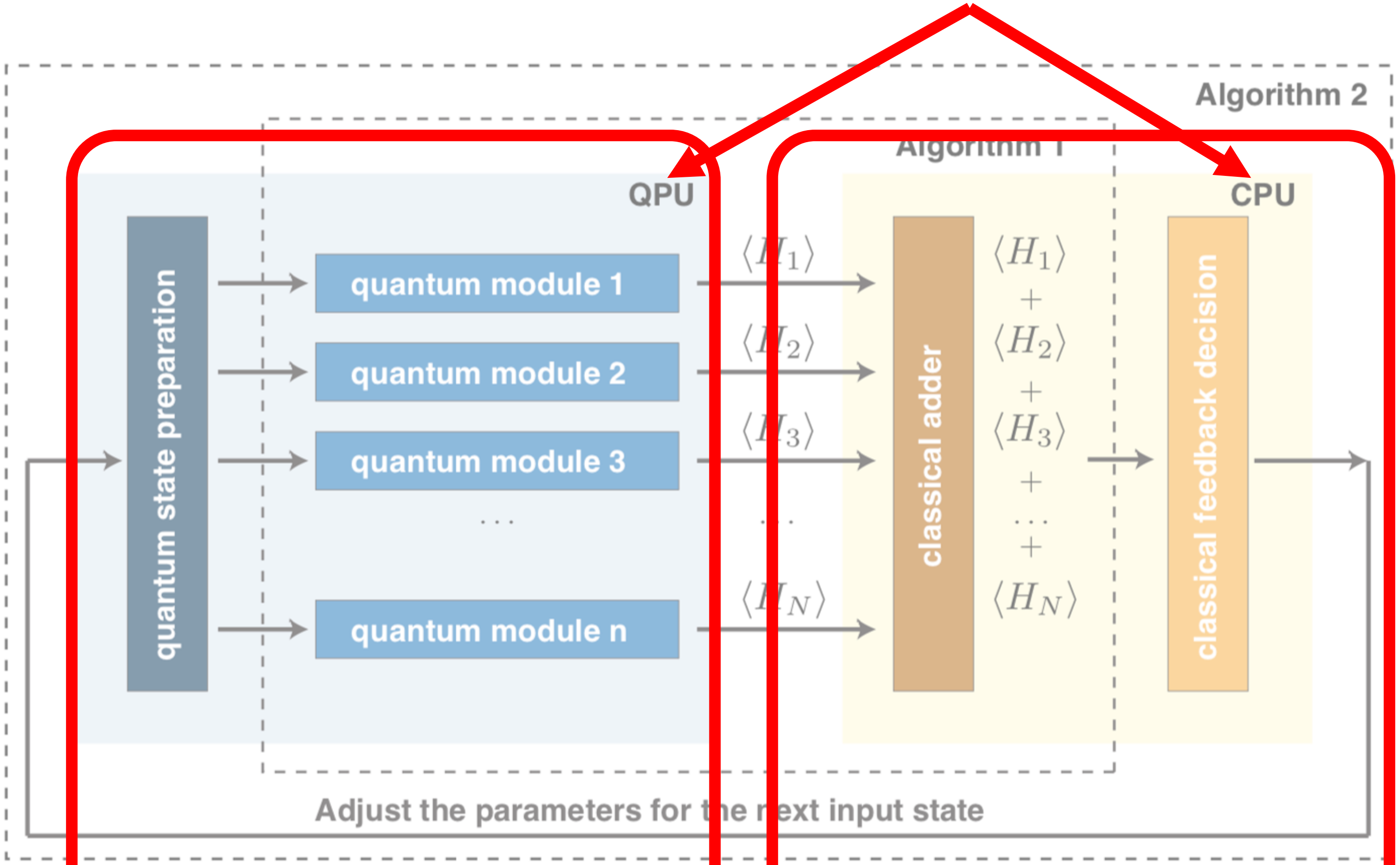
What are variational algorithms and why are they important?



Source: Peruzzo et al., 2013

What are variational algorithms and why are they important?

Use quantum & classical computation



Quantum evaluates an objective function

Classical optimizes for better parameters

Source: Peruzzo et al., 2013

What are variational algorithms and why are they important?

- It’s like using a classical computer to train a quantum neural network.
- Quantum computer can be unreliable, needs 50-100 qubits, realizable in the near future.
- Most likely candidates for first demonstrations of useful quantum.
- Major examples include quantum approximate optimization algorithm (QAOA), variational quantum eigensolver (VQE).

Preview of the syllabus

- A systems view of quantum computer engineering
- Near-term intermediate-scale quantum algorithms
- ***Programming frameworks***
- ***Emerging languages and representations***
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes

Programming assignments

- Google Cirq and IBM Qiskit
- Experiments with QAOA and VQE algorithms
- Execute on prototype quantum computers

Preview of the syllabus

- A systems view of quantum computer engineering
- Near-term intermediate-scale quantum algorithms
- Programming frameworks
- Emerging languages and representations
- ***Claims and counter claims for quantum advantage***
- ***Prototypes***

Debates & presentations

Contentious topics in quantum computer engineering

Quantum programming: Verification vs. Debugging

Quantum/classical boundary: Prototypes vs. Simulation (Google vs. IBM)

Quantum device candidates: Superconductors vs. Ion traps

In this class, you will present the competing viewpoints in debate format

Preview of the syllabus

- A systems view of quantum computer engineering
- Near-term intermediate-scale quantum algorithms
- Programming frameworks
- Emerging languages and representations
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes

Outline

- Course overview
- Personal introductions
- Why now is a good time to study quantum computing
- Preview of the syllabus
- ***Course objectives & activities***

An online, semi-synchronous, graduate seminar

One of the few uppermost division classes you might take

- Very different expectations from any other class

Components

- Reading discussions
- Debates & presentations
- Programming assignments

Reading discussions

“Reading List” tab in Canvas and on webpage

- One required reading and required summary a week.

Learn from each other's reading summaries

- Everyone is welcome in this class
- Learn from each other's prior experiences throughout computer science

Learn & improve our research reading skills

How to read and summarize papers efficiently:

- Who cares, so what?
- Why prior work is insufficient.
- Key insight.
- Methodology.
- Findings.

Deeper evaluation of the reading:

- What are limitations of the science in the paper?
- Is the paper effectively communicated?
- What background knowledge is preventing me from fully understanding this paper?