Bridging the Computational Chasm

Bryan K. Clark (bkclark@illinois.edu), *University of Illinois at Urbana Champaign*
with Eli Chertkov and Garrett Vanacore and Chad Germany
Course website: https://courses.physics.illinois.edu/phys498cmp/

Motivation

Students take the plurality of their physics classes in undergraduate

Can we bridge this divide?

Some of the difference comes from graduate school being focused on involved research projects where students come to terms with the physics.

but often don't really deeply understand physics until graduate school.

Computational thinking is an important skill for both academia and industry…

but teaching computational thinking is under-developed in the physics major.

Can we improve students skills in computational thinking?

Approach

Four immersive projects

Chosen from modern, exciting areas of physics.

- Quantum Computing Simulator
- Renormalization Group on the Ising Model
- Machine Learning
- Topological Insulators

What are students given?

Project description that describe different steps they need to do and some hints on how to go about it.

No templates; students build all the code for themselves.

Minimal Lectures

Significant one-on-one help from myself and TA's.

What does class look like?

Intro lecturing (1/4 of the time)

Students work on projects as we talk to them and help them (3/4 of the time)

How are students assessed?

Oral Grading: Students must present their projects to me or a TA and explain what they've done.

Partial credit for partial results but no partial credit for incorrect results.

How much work are students putting in?

Approximately 10 hours per week

Who are the students?

Undergraduates juniors and seniors usually with an intro programming class and.sophomore level quantum and statistical mechanics.

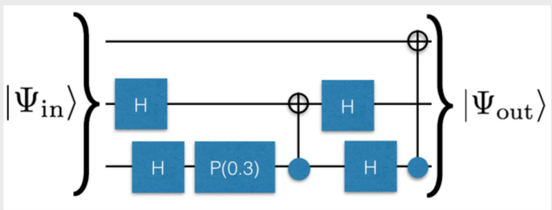
Project 1: Quantum computing simulator

Language: python

Students write code to take a quantum circuit description:

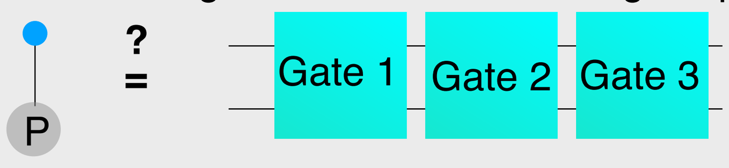
```
3
H 1
H 2
P 2 0.3
CNOT 2 1
H 1
H 2
CNOT 2 0
```

=



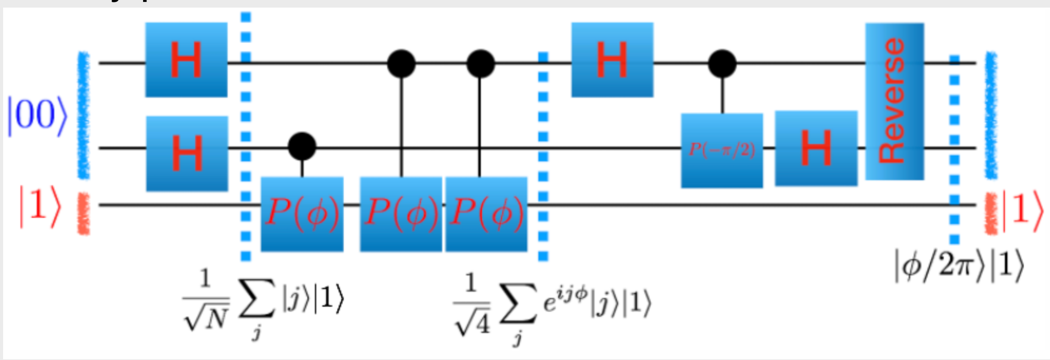
using a universal set of gates (i.e. Hadamard, Phase, CNOT) and output the quantum state produced from the circuit.

They then figure out how to make gates not in their universal set from universal gates. This is like writing simple quantum functions…



which they turn into a simple quantum compiler.

Students proceed to write code to generate quantum circuits for arbitrary phase estimation.



and use phase estimation to factor numbers.

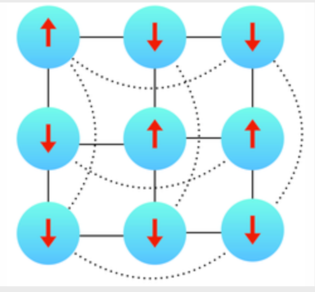
Students learn about…

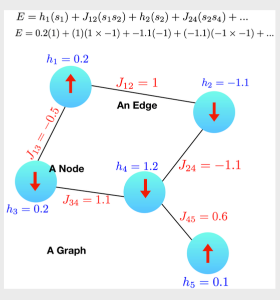
- Quantum computing
- Computational complexity and scaling of algorithms
- Incremental testing and debugging
- Optimization
- Compiling
- Numerical Precision

Project 2: Renormalization Group

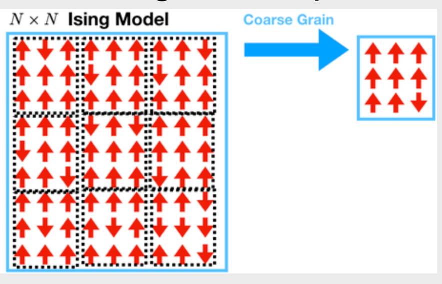
Language: C++

Students write code to simulate Ising models on arbitrary graphs via Markov Chain Monte Carlo

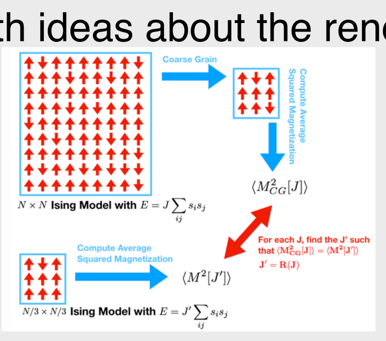




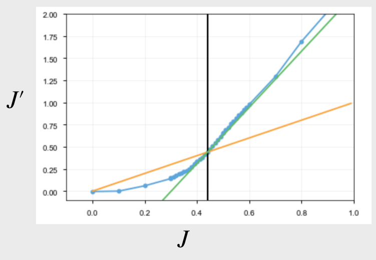
and using block-spin decimation



with ideas about the renormalization group.



computes the RG flow of the coupling constant $J \rightarrow J'$



They identify the phases and critical points by fixed points of the RG flow and compute critical exponents.

Bonus: Simulated Annealing + Parallel Tempering

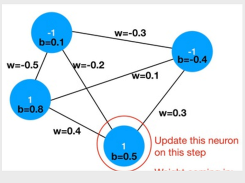
Students learn about…

- Statistical Mechanics
- Ising Models
- Renormalization Group
- MCMC
- Metropolis
- Simulated Annealing (optimization)
- Computationl Workflows
- Graphs
- Critical Exponents

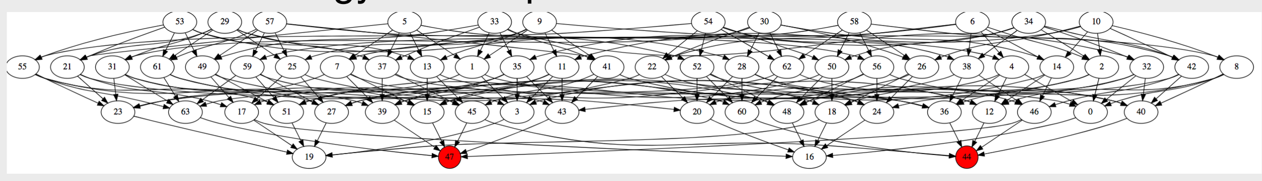
Project 3: Machine Learning

Language: C++

Students write a Hopfield network



measure the energy landscape



and empirically determined its memory capacity.

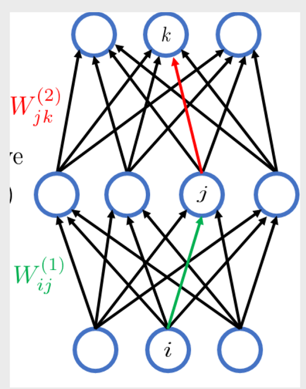
Students write a Restricted Boltzmann machine

Restricted Boltzmann Machine ⟷ Ising Model
- Neuron ⟷ Ising Spin
  - +1 = Spin Up
  - -1 = Spin Down
- $w_{ij} \longleftrightarrow J_{ij}$
- $b_i \longleftrightarrow h_i$

and discover it's just an Ising Model.

They then use it in an unsupervised capacity to learn digits.

Students write a Feed Forward Neural Network



Implement back-propagation so it can learn

And train it to identify a phase transition in an Ising model

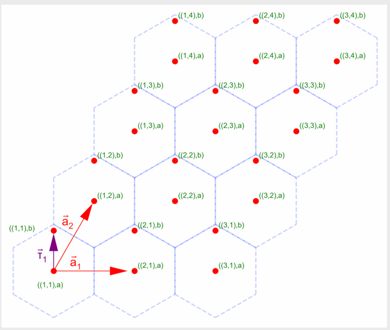
Students learn about…

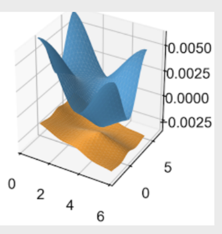
- Deep Learning
- Automatic differentiation
- Uses of Ising Models
- Spin glasses

Project 4: Topological Insulators

Language: python

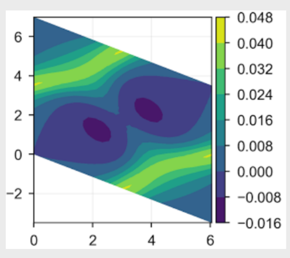
Students implement a tight-binding model of graphene,

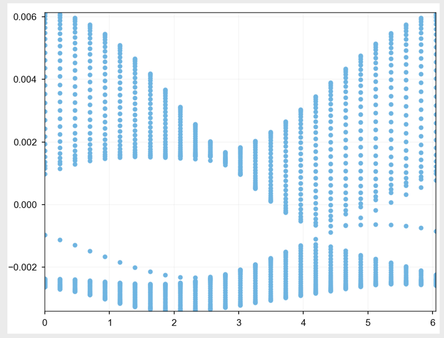




see that there are multiple ways to gap out the Dirac cones,

compute their berry curvature,





and find that some of them have a Chern number and edge modes.

Students learn about…

- Tight binding models
- Diagonalization
- Topological Insulators
- Band structures
- Edge Modes