Computation to Support Understanding of Discrete and Continuous Quantum Systems

Christian D. Solorio (he/him)
Elizabeth Gire
David Roundy

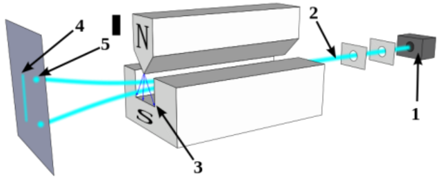
Oregon State University



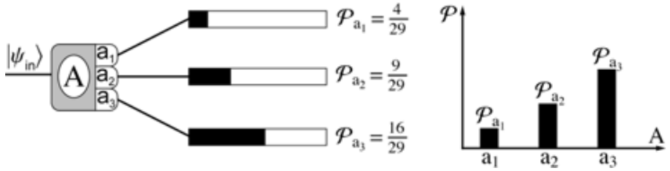
DUE 1836604

A spins-first Approach at OSU

Junior-level introductory QM course starts with spin systems

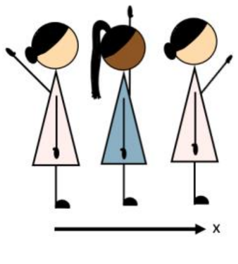


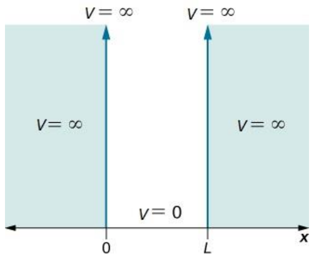
$$\begin{pmatrix}1\\0\end{pmatrix} |+\rangle$$



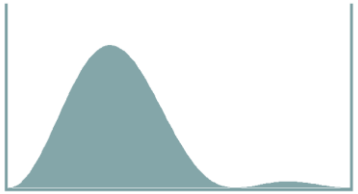
Discrete systems

The course ends with the particle in a box

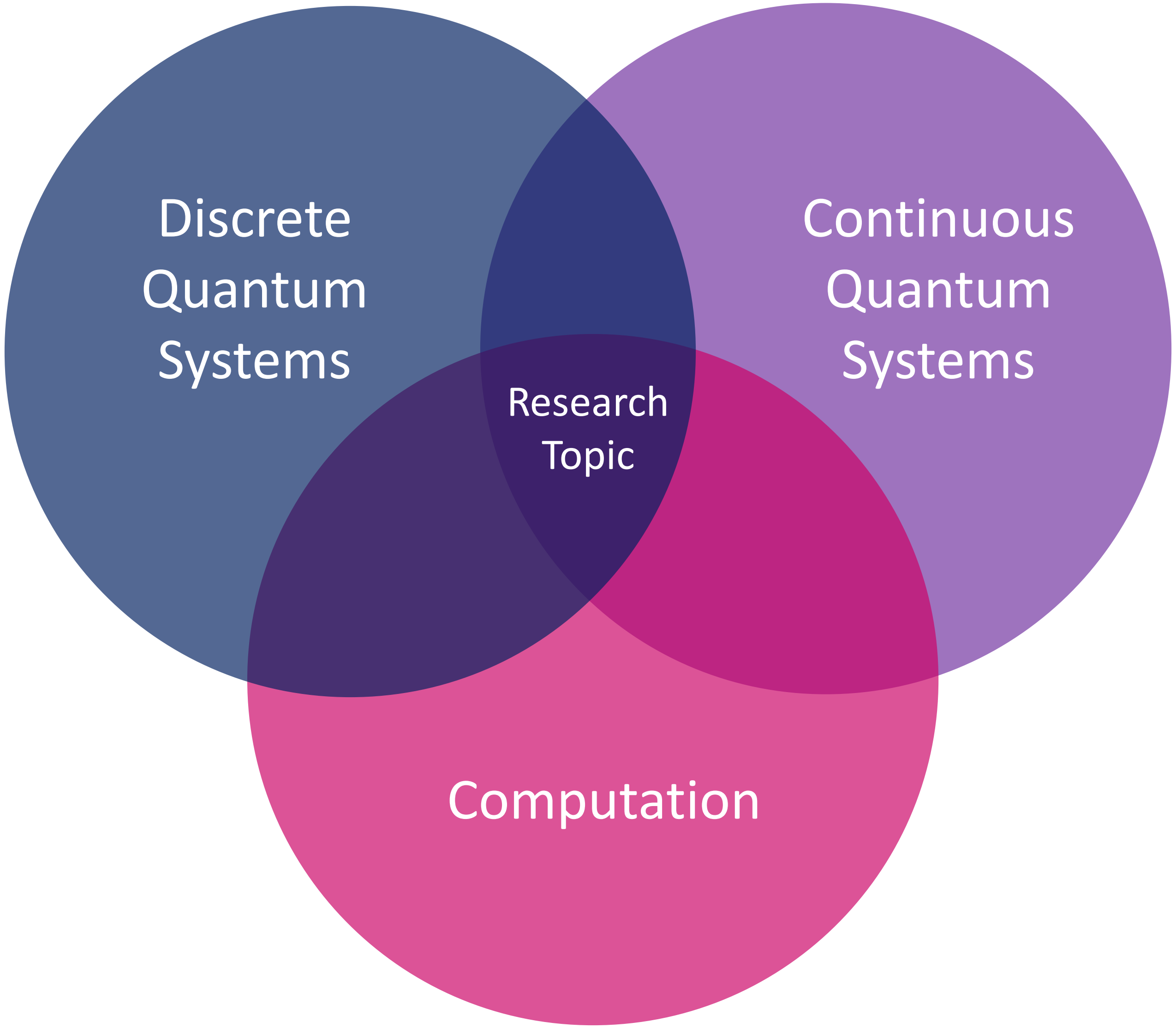




$$\psi(x) = \sum_{n=1}^{\infty} c_n \phi_n(x)$$



Continuous system



The connection between discrete and continuous quantum systems gets lost

QM Calculations in Code

- Wavefunctions are discretized for computational operations

$$\psi(x) = Ax(x - L)$$

$$\Downarrow$$

$$\psi(x) \to \psi(0), \psi(\Delta x), \psi(2\Delta x), \ldots \psi(L)$$

Computational Lab Course

Pair-programming in Python

Wavefunctions-first approach

- Mean position
- Expansion in a sinusoidal basis set
- Position operator
- Kinetic energy operator

```python
import numpy as np
L=1
n_max = 10
dx = .25

def psi(x):
    return np.sqrt(495)/L**5/np.sqrt(L)*x*(x-L)**4

def phi_n(n, x):
    return np.sqrt(2/L)*np.sin(n*np.pi*x/L)

def c_n(n):
    sum = 0
    for x in np.arange(0,L,dx):
        sum += np.conjugate(phi_n(n,x))*psi(x)*dx
    return sum

def f_nmax(x):
    f=0
    for n in list(range(1, n_max+1,1)):
        f += c_n(n)*phi_n(n,x)
    return(f)

import matplotlib.pyplot as plt

plt.figure()
for n in list(range(1, n_max+1, 1)):
    print("c_"+str(n)+" = "+str(c_n(n)))
    plt.scatter(n,c_n(n), color='black')
plt.xlabel("n")
plt.ylabel("c_n")
plt.title("c_n vs n")
plt.xticks(list(range(1, n_max+1, 1)))

plt.figure()

xvals = np.arange(0,L,0.001)

plt.plot(xvals, psi(xvals), label='Ψ(x)')
plt.plot(xvals, f_nmax(xvals), label="Approximation")
plt.plot(xvals, c_n(9)*phi_n(9,xvals), label="c9*φ_n(x)")
plt.xlabel("x")
```

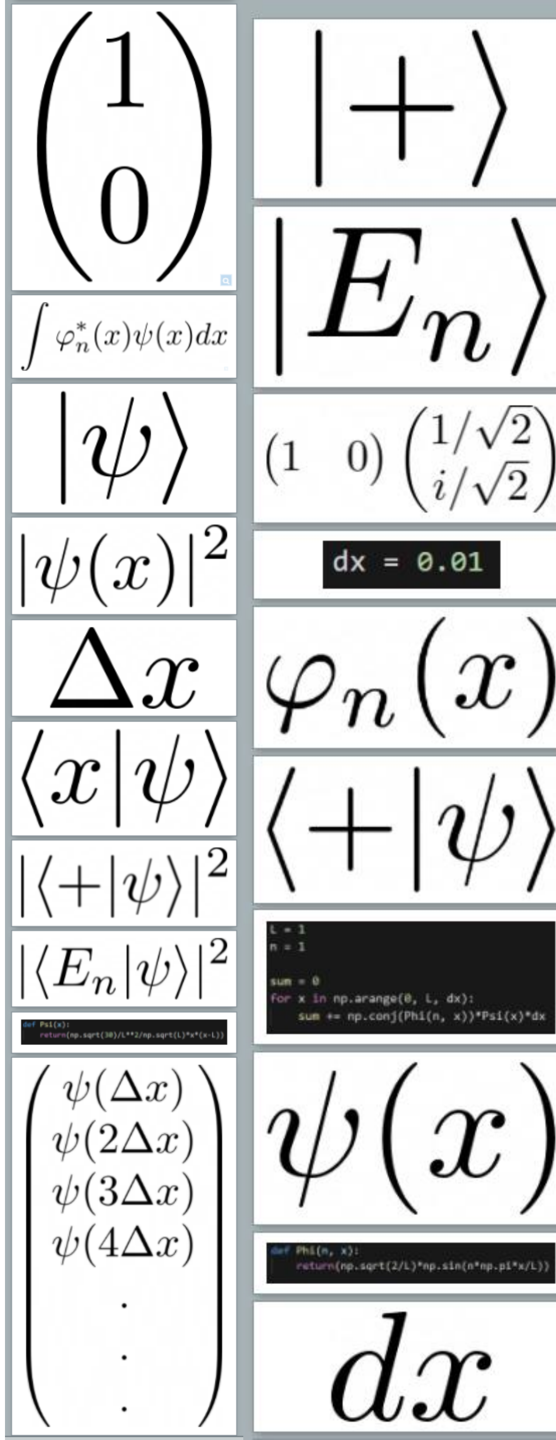
Research Questions

1 How do students think of a code representation of the particle in a box in terms of discrete and continuous?

2 How do students characterize “discrete” and “continuous” in the context of quantum mechanics?
(see my PERC poster!)

Card Sorting Task

- Conducted semi-structured interviews with 6 participants
- Used Optimal Workshop's OptimalSort
- Presented with 20 cards



Interview Structure

- Identified each card
- Sorted the cards in a way that made sense to them
- Resorted the cards in another way
- Sorted the cards into a “discrete” group and a “continuous” group

Interview Structure

- Identified each card
- Sorted the cards in a way that made sense to them
- Resorted the cards in another way
- **Sorted the cards into a "discrete" group and a "continuous" group**

Code Snippet Sorting Examples

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
dx = 0.01
```

Code Snippet Sorting Examples

A All Discrete

"I don't know if there's a way in code to do things that aren't technically discrete, but that's because I don't know a lot about code."

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
dx = 0.01
```

Code Snippet Sorting Examples

A — All Discrete

"I don't know if there's a way in code to do things that aren't technically discrete, but that's because I don't know a lot about code."

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
dx = 0.01
```

B — All Continuous

"So yeah, maybe that would be better [in the continuous group] because it is a function of x rather than a bra-ket notation or a matrix."

Code Snippet Sorting Examples

A

All Discrete

Limitations of code

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
dx = 0.01
```

B

All Continuous

What the code represents

Code Snippet Sorting Examples

E

"The way this is being coded, you're not really making a continuous function in actuality, … but I think it is trying to mimic a continuous thing, it's just not that."

Discrete

```
dx = 0.01
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

Continuous

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

Code Snippet Sorting Examples

E

"The way this is being coded, you're not really making a continuous function in actuality, … but I think it is trying to mimic a continuous thing, it's just not that."

"Energy eigenstates are like packets of energy, so I'll throw all of the energies in [the discrete group]"

Discrete

```
dx = 0.01
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

Continuous

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

Code Snippet Sorting Examples

E

- Relationship to position
- “So, wavefunctions. I’m also putting that in continuous”
- “I’m putting this position [group] into continuous”

Discrete

```
dx = 0.01
```

```
def Phi(n, x):
    return(np.sqrt(2/L)*np.sin(n*np.pi*x/L))
```

Continuous

```
def Psi(x):
    return(np.sqrt(30)/L**2/np.sqrt(L)*x*(x-L))
```

```
L = 1
n = 1

sum = 0
for x in np.arange(0, L, dx):
    sum += np.conj(Phi(n, x))*Psi(x)*dx
```

Sorting Overview

- **All Discrete:** 1 participant
- **All Continuous:** 2 participants
- **Only dx is discrete:** 2 participants
- **Half and half:** 1 participant
- System the code represented
- Limitations of computation

Takeaways

Breadth of ways that students describe discrete and continuous

Muddiness of “discrete” and “continuous”

Variety in how students view computation

Thank you for watching!

Many thanks to the OSUPER team, our participants, and our funding supporters!

Find the slides: beav.es/???

Additional Questions: solorich@oregonstate.edu

Find resources & research:
- paradigms.oregonstate.edu
- osuper.science.oregonstate.edu

See PERC poster:

8595

DUE 1836604