Blending Computational Exercises into a One-Semester Quantum Mechanics Course

Dr. Michael Olson – St. Norbert College Physics

Tyler Bennett (SNC Physics 2021)

Cullen Voss (SNC Physics 2021)

PICUP Virtual Conference (Session 3), July 1st 2020

St. Norbert College – small, liberal-arts college located in De Pere, WI

Small Physics program (Three faculty, 20-24 majors)

- Upper-level courses taught on an alternate-year rotation (sophomores-seniors)
- No space within a crowded curriculum for a stand-alone computational physics course

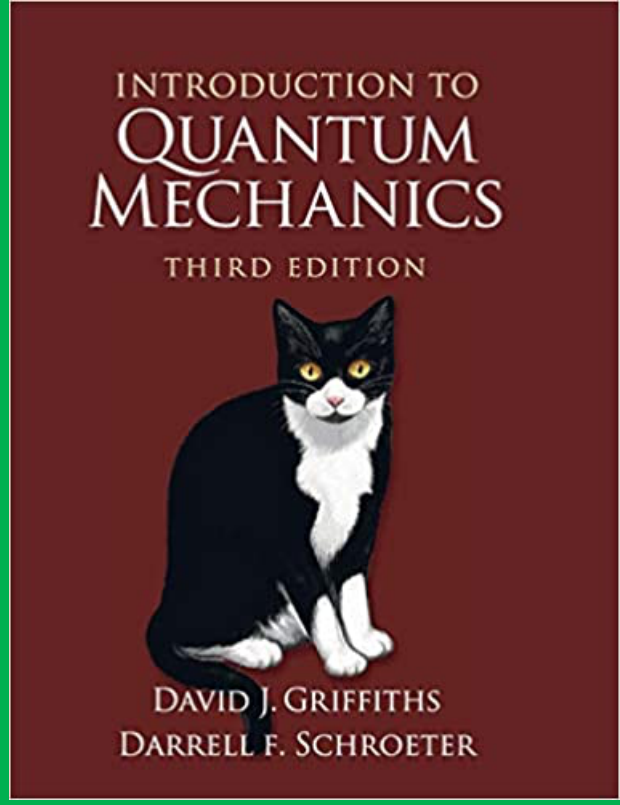
Department-wide initiative to integrate computation throughout physics curriculum

- Faculty attendance at PICUP and PICUP/Alpha conferences
- Limited implementation in introductory courses (simple calculations and plotting in Excel)
- Repeated implementation in upper-level courses (Classical, Thermo/Stat-Mech, Electronics)

Spring 2020 – Quantum Mechanics (Junior/Senior)

- Historically no (or very limited) computational activities within the course
- First implementation of significant, integrated computational segment
- **No instructor experience with Python prior to PICUP/Alpha Immersion (July 2019)**

Classic Text, Now in it's Third Edition ➔



Includes a significant number of computational exercises

- Locate problems that mirrored existing PICUP QM exercises
- Identify theoretical "on-ramp" ➔ **Eigenvalue Problem**

- **Began with a classical system ➔ masses and springs on air track**
- Solved two- and three-mass coupled systems by hand (review)
- Compared directly to corresponding physical systems

PICUP Exercise: *"Using the Finite-Difference Approximation and Hamiltonians to Solve 1D Quantum Mechanics Problems",* by Marie Lopez del Puerto

Mirrored in Griffiths (Problem 2.61) ➔ Provided two different "looks" at the same thing...

Blending of Text Problems and PICUP Exercises

Solve Time-Independent Schrödinger Equation numerically , point by point, across 1D ISW

- Creates an (N x N) eigenvalue problem
- Eigenvalues ➔ stationary-state energies
- Eigenvectors ➔ wavefunctions.

Find energies and wavefunctions for Infinite Square Well by finite difference methods

N = 3 ➔ theoretical solutions by hand (Griffiths Problem 2.61) ➔ PICUP Exercises 1-3

N = 5 ➔ entered values by hand into Wolfram Alpha

N = 10 and N =100 ➔ performed in Mathematica (Griffiths Problem 2.61)

Repeat the N = 100 case in Python (Spyder) ➔ PICUP Exercise #4

- Repeat with sinusoidal potential energy ➔ Griffiths Problem 2.62
- Repeat with simple harmonic oscillator potential energy ➔ PICUP Exercise #7

Extensions to Existing Exercises

***"Quantum Dynamics in 1D with a Series Solution"*, by Gardner Marshall (in Python)**

- Perform a Fourier decomposition of composite wave-function
- Determine Fourier constants and recreate wave-function as a series of ISW solutions
- Generate time-evolution of wave function in Python
- **Create animation of time evolution → *Tyler Bennett***

Extension to the Finite-Difference Exercise Set → introduce "delta-function" perturbation at center of Infinite Square Well (Classic QM Problem)

- Solve problem theoretically (Griffiths Prob. 7.1) using **non-degenerate perturbation theory**
- Adapt Python code to generate finite difference solutions (N = 1→ 5) for delta function
- Code results of theoretical exercise into Python → corrected energies and wave-functions
- **Compare corrected (perturbation) results with finite-difference solutions → Cullen Voss**

Some Conclusions: The Good, The Bad, & The Ugly

The Good:

- **Dedicated lab sessions** allowed for focus on computational activities
- Taking theory to computation ➔ **enhanced student engagement** with core material
- Fostered a sense among students that they were “creating something"
- Opportunity for students to exercise their creativity
- Allowed students to further develop critical programming skills

The Bad:

- Despite previous computational experience, **inequities in programming experience** evident
- Alternate year course rotation ➔ **difficult to develop uniform programming background**
- **Instructor inexperience** with Python ➔ some inefficient programming and guidance

The Ugly:

- **Lack of interest/willingness (of some) to engage the computational aspects of the course.**
- Also observed in electronics courses (Arduino IDE programming) ➔ **“resistance to coding”**