Digital Twin for ^{40}Ca Trapped Ion Experiments
Lab Notebook: Jewoo Park
Supervised by: Koray Mentesoglu

Background Reading + Concepts
Concepts:

- Trapped ion control
- Programming in Julia

Resources

1. IonSim: https://examples.ionsim.org/tutorial-notebooks/getting_started.html

Tentative Timeline

Goal: Build a digital twin of our trapped ion experiment to simulate expected results

Week 1: Join lab, do paperwork/training
Weeks 2: Install Julia and IonSim, go through example notebooks, learn about trapped ion dynamics
Week 3,4: Implement qubit-only dynamics, noiseless.
Week 5,6: Add shot-to-shot noise
Week 7,8: Implement experiments with 1 ion and motional mode control
Week 9-13: Experimentally relevant exploration (to be decided)

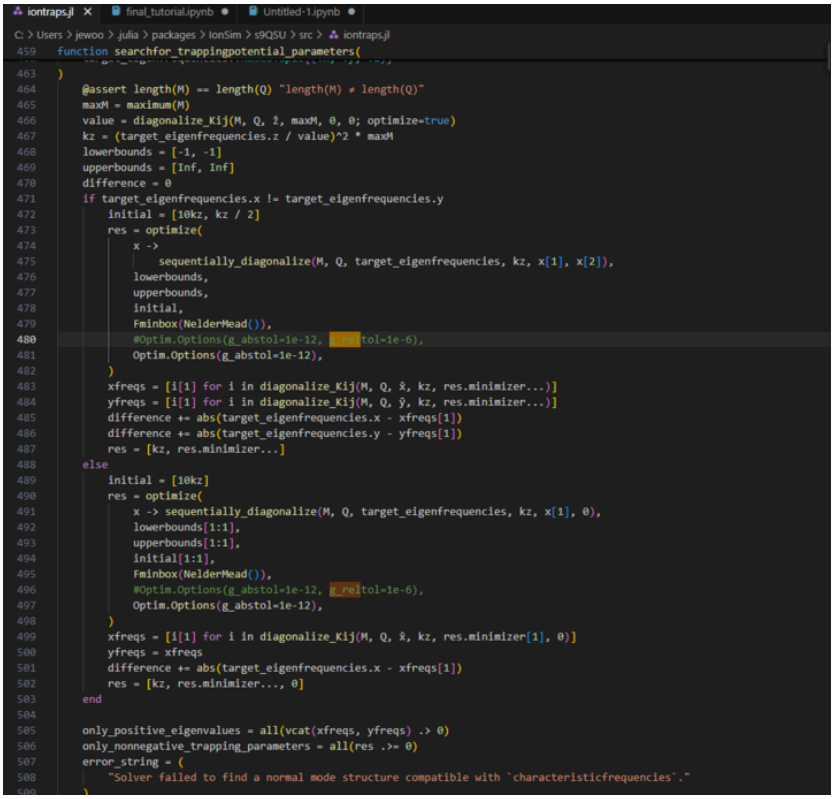
Timeline

Week 1 (July 8 ~ July 15)

- Reading Hempel thesis and understanding the basic concept of Ion trap
- Reference https://examples.ionsim.org/intro.html
- Familiarize with IonSim and Julia language, initial background setup

Week 2 (July 16 ~ July 22)

- Identified a critical version conflict between the `IonSim.jl` package and its dependency, `Optim.jl`
- Manually edited the package source file (`iontraps.jl`) to correct the behavior and proceed with simulations.
- Focused mainly on implementing Rabi oscillations and applying laser pulses on a single Ca+ ion.



Jewoo Park 6:03 PM

Hey everyone,

As I mentioned to some of you earlier, I'm traveling this week and unfortunately won't be able to join our meetings. I just wanted to give a quick update on what I've been working on!

Over the past few days, I've been reviewing the basic syntax of Julia and exploring IonSim in general. While experimenting with some of the functions IonSim provides, I ran into a strange issue with the linearchain function. After some digging, I discovered that the problem was due to a conflict between the Optim and IonSim packages.

To work around it, I manually edited the IonSim/iontraps.jl source file—commenting out the original function and modifying the parameters to achieve the desired behavior.

Since Caleb and I both experienced this issue, and I've tried installing IonSim in various environments, I suspect this might be a common problem for new users. I've included a screenshot of the modified code below for reference.

For the rest of this week, I'll be focusing on implementing and optimizing the Pulse and Rabi functions we discussed last week.

I'll aim to share some results by next week. I'll also look into identifying compatible versions of IonSim, Optim, and Julia so we can avoid having to patch the source manually in the future.

Apologies again for missing the meetings—if there are any updates or things I should work on during the week, I'd really appreciate it if someone could let me know. I'll do my best to catch up over the weekend once I return.

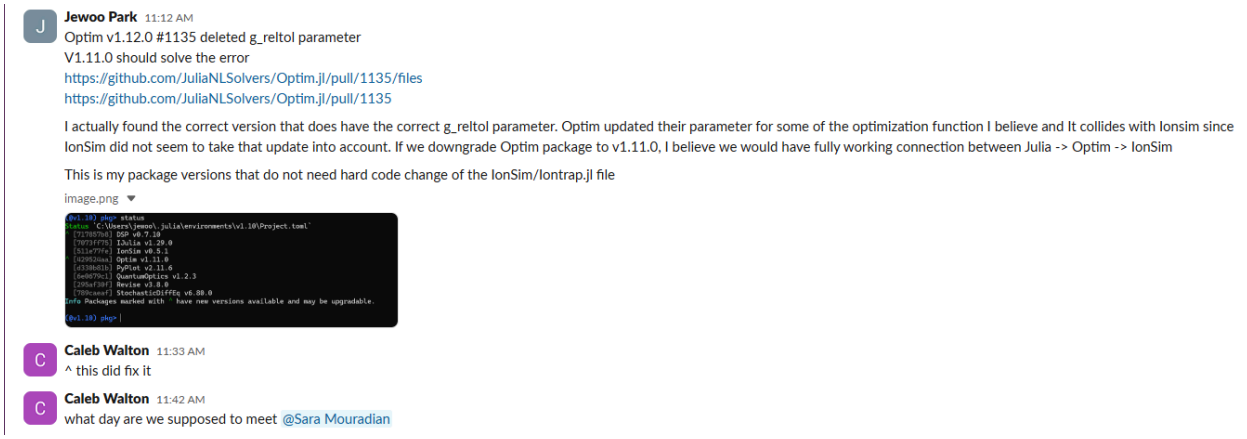
Thanks again!

image.png

Week 3 (July 23 ~ July 29)

- Investigated and found the correct version combination of all required packages: IonSim, Optim, QuantumOptics, and Julia.
- Dependency: Julia ⇒ Optim ⇒ QuantumOptics ⇒ IonSim
- Identified that Optim v1.12.0 removed a parameter required by IonSim. Downgrading Optim to v1.11.0 resolved the conflict.
- Shared the stable package environment with the team, providing a solution that avoids manual source code changes.
- Optim v1.12.0 #1135 deleted g_reltol parameter
- https://github.com/JuliaNLSolvers/Optim.jl/pull/1135/files
- https://github.com/JuliaNLSolvers/Optim.jl/pull/1135

Correct version of all the dependency package

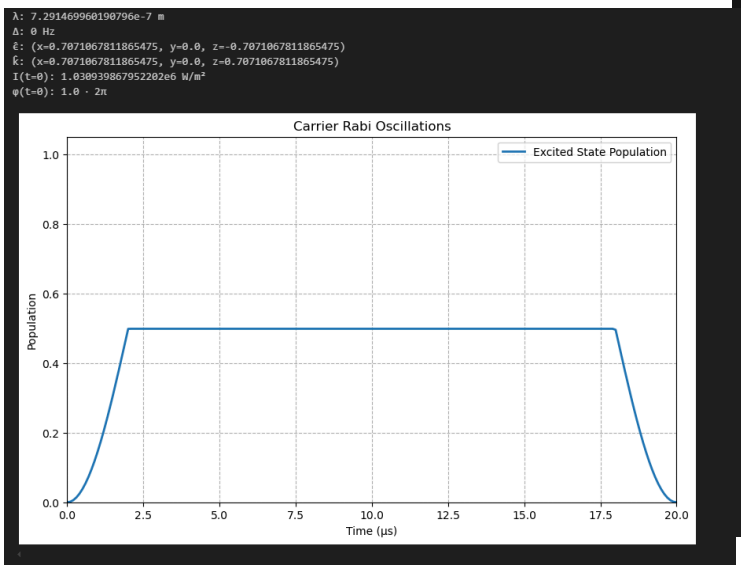


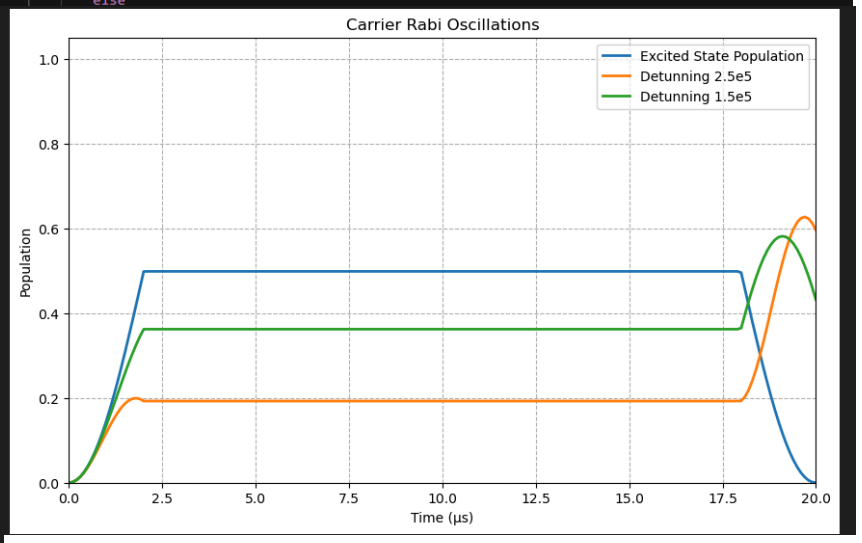
Our Team's slack channel after the version share

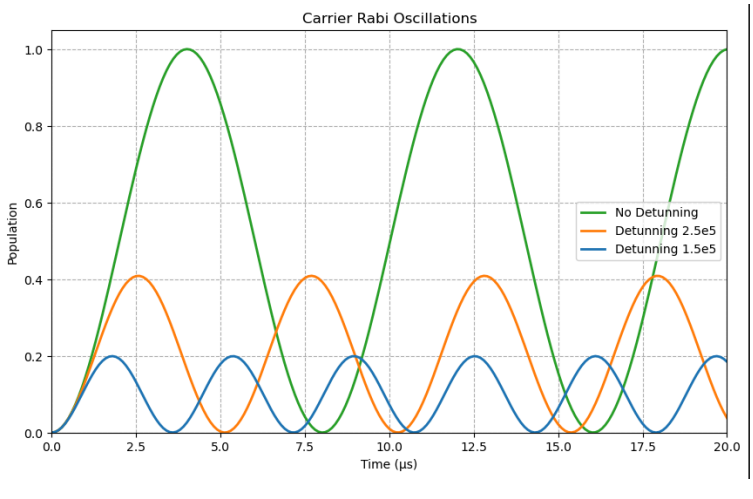
Week 4 (July 30 ~ August 5)

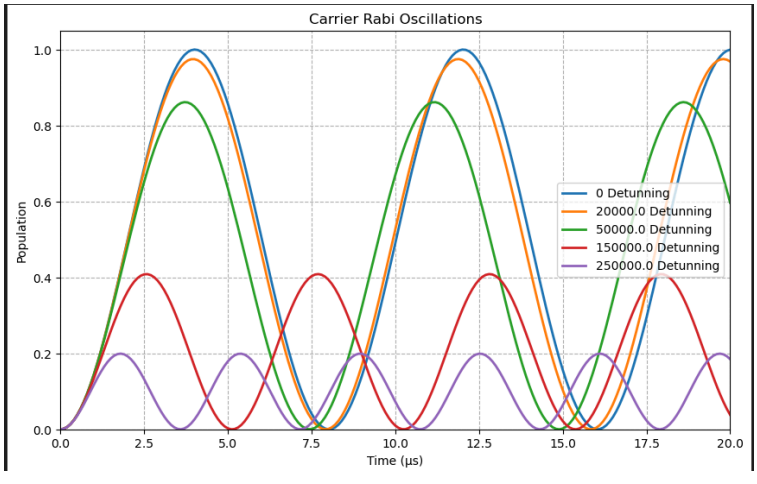
- Implemented a laser intensity function to simulate phase shifts.
- Created a generalized `pulse` function to apply desired laser pulses to the ion, ensuring easy manipulation during simulations
- Observed and plotted the desired phase shift of the ion with various laser detunings.

```julia
function pulse(T:: Chamber , tspan, pitime)
    # Define the laser that will drive the transition
    L = T.lasers[1]

    # Combine all components into a single Trap object, which represents the full experiment
    # This is the main object that holds the entire state of our physical system.

    pi2_time = pitime*1e6/2

    res_intensity = intensity_from_pitime(L, pitime, T.iontrap.ions[1], ("g", "e"), T)

    function intensity_funtion(t)
    if(t<=pi2_time)
        return res_intensity
    elseif(t>=tspan[end] - pi2_time)
        return res_intensity
    else
        return 0.0
    end

    end
    intensity!(L, intensity_funtion)

    function phase_funtion(t)
        if(t<=pi2_time)
            return 2*pi
        elseif(t>=tspan[end] - pi2_time)
            return pi
        else
```









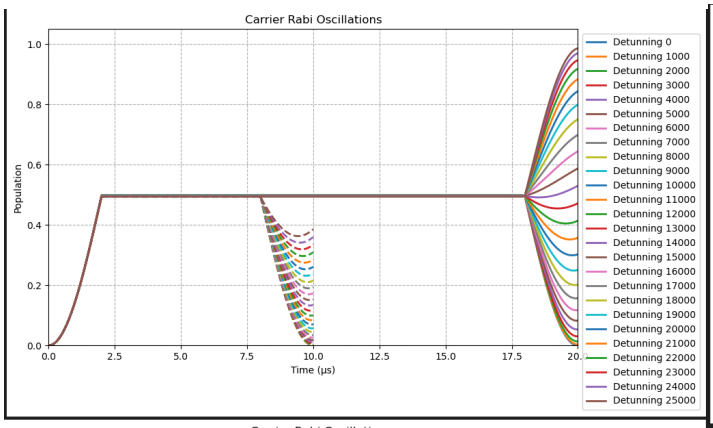
Week 5 (August 6 ~ August 12)

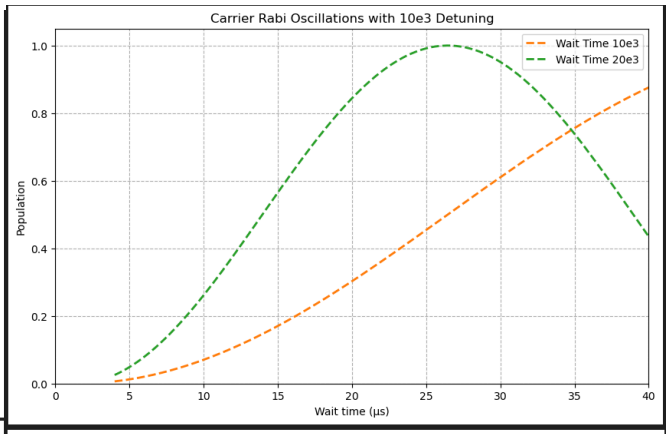
- Changed the observing parameter to "wait time" rather than a continuous time interval on the x-axis in order to better mimic the discrete, sequential nature of a real-world experiment.

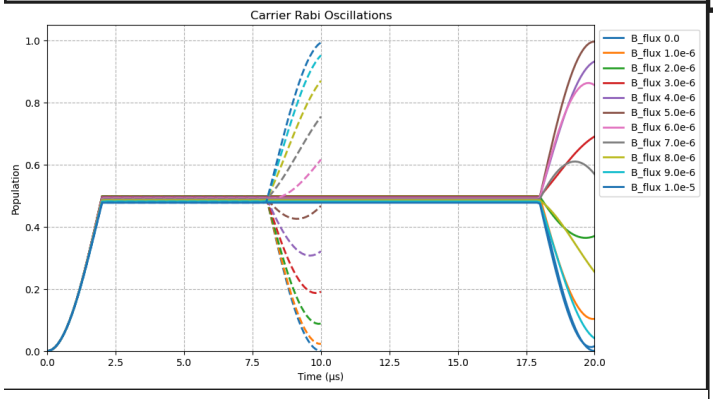
- Plotted the population change with respect to wait time to simulate a Ramsey experiment.

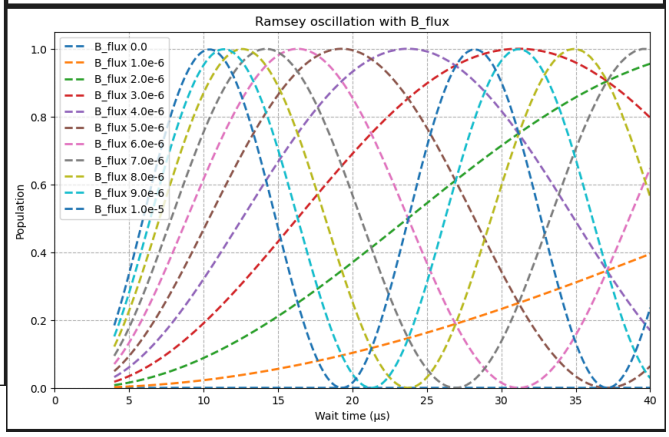
- Pivoted focus to include magnetic flux shifts to observe similar phase shift behavior.

- Successfully observed Ramsey oscillations with given detunings in the magnetic field.





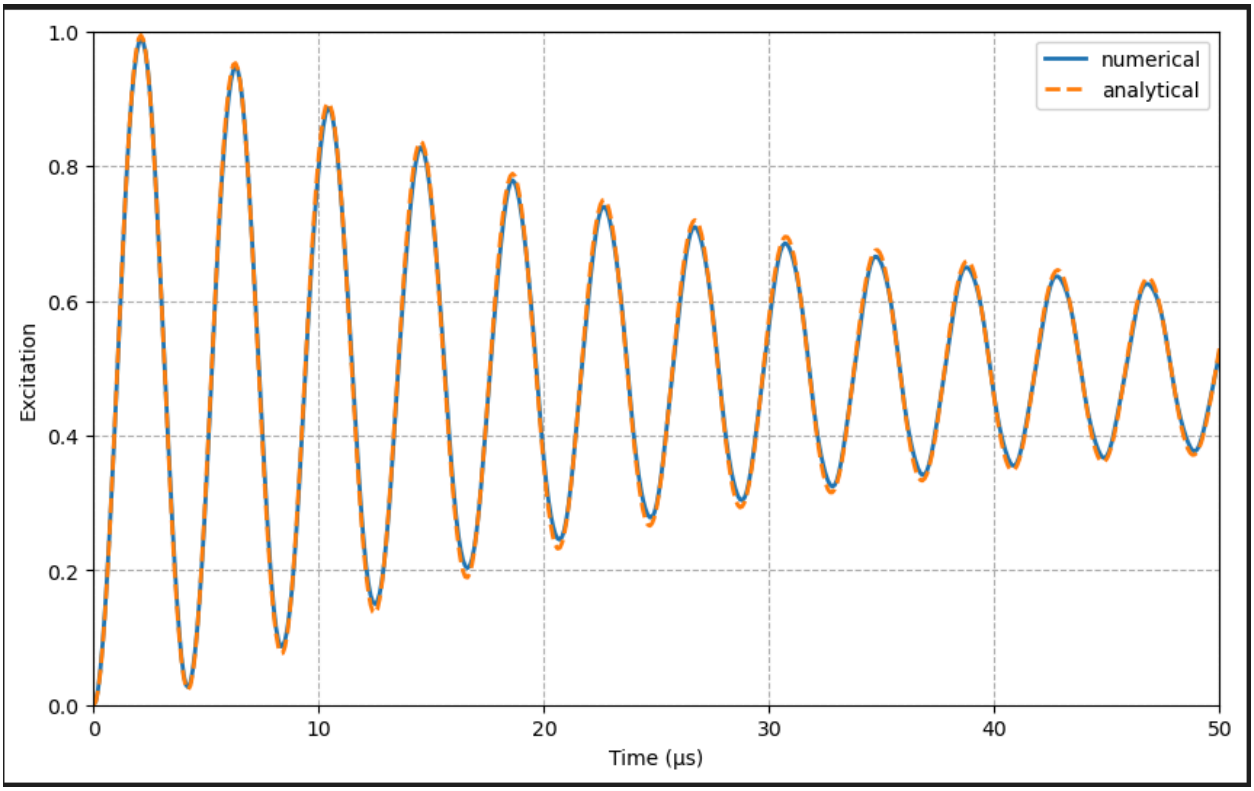


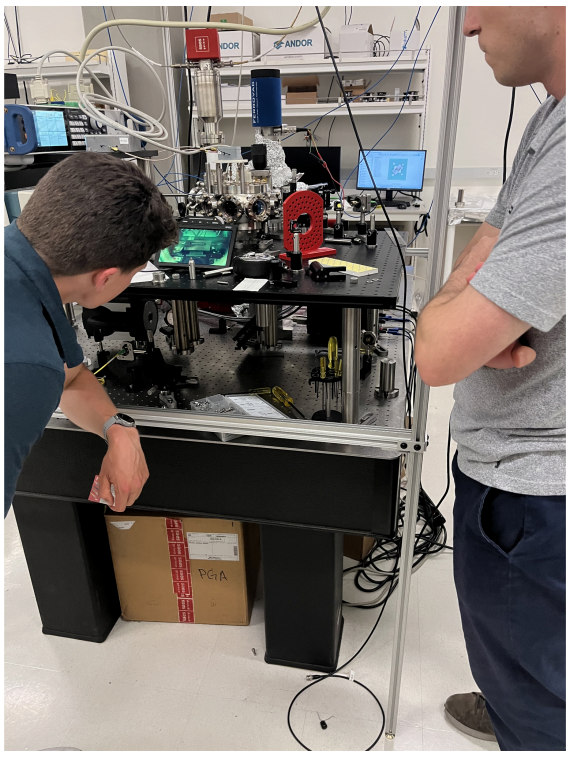


Week 6 (August 13 ~ August 19)

- Introduced a new degree of freedom in the single ion state by modeling it with a thermal state instead of a single Fock state.

- Observed the characteristic decay in Rabi oscillation amplitude due to the incoherent sum of oscillations from different initial Fock states.





Assisted with the alignment of laser and optical pathways for the fast chamber system

Summary & Future Work

Summary: Over the past six weeks, significant progress was made in developing a digital twin for the $^{40}Ca^{+}$ experiment. Key accomplishments include establishing a stable and reproducible simulation environment for the team by resolving critical package dependency issues in Julia. Core single-qubit operations were successfully implemented and verified, including Rabi oscillations, Ramsey interferometry, and the application of custom laser pulses. The model was further advanced by incorporating more realistic experimental factors, such as magnetic field effects, laser detuning, and initial thermal states.

Future Work: The immediate next steps will focus on expanding the simulation's complexity and realism. This includes implementing shot-to-shot noise to better model experimental uncertainty. The goal is to progress towards simulating specific experimental protocols relevant to the lab's current research.