Assignment Mamo 2100

Per Erik Grønvik - pegro6469@oslomet.no

A report about numerically solving a Shrodingers equation on a quantum ring

Introduction

Thanks and Credits:

In this project i got help from Maryam and also discussed with fellow students. I also utilized GPT in this project. GPT and Maryam was used to verify the calculations and to find mistakes. The file 07_Finite-square-well-1D.ipynb was modified in order to fit to the project.

Other thanks is directed towards Andre L. my professor and Ankasha my TA from India for helping me understand the theory of this class better. @PhysicalChemistry on youtube has been a good source and also the second book in the corriculum "CQP".

Software and decisions:

The language and presentation has a pedagogical use in mind. The Problem statement is "How one can solve the Schördinger equation numerically and find the ground state, potentials and the first and second exciteted states." as I hope this could help someone else understand this better. That someone could also be myself later.

In this project I have chosen to write the report i Flexil, a program that simulates pen and paper. This allowed me to practice for the exam while also doing the project. Some disadvantages is formatting support and no support for Latex. I have learned Latex by creating a formula sheet for this class, so this part was not important for me. The code was written in Visual Studio Code.

Theory

SOLVING $\hat{H}\Psi = E\Psi$ WHERE $\hat{H}(U,A) = \frac{1}{2}\left(-\frac{i}{R}\frac{d}{d\theta} + A(\theta)\right)^2 + U\theta$ NUMERICALLY

① Simplify the hamiltonian

$\hat{H}(U,A) = \frac{1}{2}\left(-\frac{i}{R}\frac{\hat{d}}{d\theta} + A(\theta)\right)^2 + U\theta$ ∧ unsquaring minding that $\frac{\hat{d}}{d\theta}$ is non-commutative $(a+b)^2 = a^2 + ab + ba + b^2$

$\frac{1}{2}\left(\frac{(-1)^2(i)^2}{R^2}\frac{\hat{d}^2}{d\theta^2} - \frac{iA(\theta)}{R}\frac{\hat{d}}{d\theta} + \frac{i}{2R}\frac{dA(\theta)}{d\theta} + A(\theta)^2\right) + U\theta$ ∧ MULTIPLY

$\frac{-1}{2R^2}\frac{\hat{d}^2}{d\theta^2} - \frac{iA(\theta)}{2R}\frac{\hat{d}}{d\theta} + \frac{i}{4R}\frac{\hat{d}}{d\theta}A(\theta) + \frac{A(\theta)^2}{2} + U\theta$ ∧ Knowing that $A(\theta)$ is a constant in this specific case, $\frac{\hat{d}}{d\theta}A(\theta) = 0$.

$\frac{-1}{2R^2}\frac{\hat{d}^2}{d\theta^2} - \frac{iA(\theta)}{2R}\frac{\hat{d}}{d\theta} + \frac{A(\theta)^2}{2} + U\theta$

② Insert into the Shrodingers equation

$\hat{H}(U,A)\Psi = \frac{-1}{2R^2}\frac{\hat{d}^2}{d\theta^2}\Psi - \frac{iA(\theta)}{2R}\frac{\hat{d}}{d\theta}\Psi + \frac{A(\theta)^2}{2}\Psi + U\theta = E\Psi$

③ Creating a numerical eqution

Minding that numerically:

$f'(x_n) \approx \frac{-f(x_{n-1}) + f(x_{n+1})}{2\Delta x}$ $f''(x_n) \approx \frac{f(x_{n-1}) - 2f(x_n) + f(x_{n+1})}{(\Delta x)^2}$

- I use Δ instead of d to visualize that this is the step difference, distinct from any d.
- I switch the order of the parts to standardize with the direction of the matrix.

We get

$\frac{-1}{2R^2}\left(\frac{\Psi(\theta_{n-1}) - 2\Psi(\theta_n) + \Psi(\theta_{n+1})}{(\Delta\theta)^2}\right) - \frac{iA(\theta)}{2R}\left(\frac{-\Psi(\theta_{n-1}) + \Psi(\theta_{n+1})}{2\Delta\theta}\right) + \frac{A(\theta)^2}{2}\Psi + U(\theta)\Psi$

$\frac{-1\Psi}{2R^2(\Delta\theta)^2}\left((\theta_{n-1}) - 2(\theta_n) + (\theta_{n+1})\right) + \frac{\Psi iA(\theta)}{4R\Delta\theta}\left((\theta_{n-1}) - (\theta_{n+1})\right) + \frac{A(\theta)^2}{2}\Psi + U(\theta)\Psi$

4) Setting up the hamiltonian matrix

$$\left(\frac{-1}{2R^2(\Delta\theta)^2}\left(\psi_{(\theta_{n-1})}-2\,\psi_{(\theta_n)}+\psi_{(\theta_{n+1})}\right)+\frac{iA(\theta)}{4R\Delta\theta}\left(\psi_{(\theta_{n-1})}-\psi_{(\theta_{n+1})}\right)+\frac{A(\theta)^2}{2}+U(\theta)\right)\psi\, I_{(i\times j)\,\hat{n}=j}$$

$$\Delta\theta=\frac{2\pi}{n}=\frac{\text{circumference}}{\text{steps}}$$

A visual way to show how the hamiltonian is discretizied, is by a I matrix I with cordinates ixj. n=j got an operator symbol to show that when the expression is inserted by multiplication by 1, you have to do an operation, by substituting the n, thus moving the grid point of the object. Showing the step of creating the column vector in a more visual manner. Mind that the coordinates i and j start on 0 according to programming convention and that the axis are continius so that 0-1=n and n+1=0.

$$\hat{H}_{matrix}=\hat{H}+\begin{pmatrix}1_{\hat{n}=0} & & & \\ & 1_{\hat{n}=1} & & \\ & & 1_{\hat{n}=2} & \\ & & & \ddots \\ & & & & 1_{\hat{n}=n}\end{pmatrix}$$

$\hat{n}=j$ does nothing as n does not exist.

$$\frac{-1}{2R^2(\Delta\theta)^2}\begin{pmatrix}-2 & 1 & 0 & \cdots & 1\\ 1 & -2 & 1 & \cdots & 0\\ 0 & 1 & -2 & 1 & \vdots\\ \vdots & 0 & \ddots & \ddots & 1\\ 0 & \cdots & 1 & 1 & -2\end{pmatrix}+\frac{iA(\theta)}{4R\Delta\theta}\begin{pmatrix}0 & -1 & 0 & 0 & \cdots & 0 & 1\\ 1 & 0 & -1 & 0 & \cdots & 0 & \\ 0 & 1 & 0 & -1 & \ddots & & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & & & \\ -1 & 0 & 0 & 0 & \cdots & 1 & 0\end{pmatrix}+\left(\frac{A(\theta)^2}{2}+U(\theta)\right)\begin{pmatrix}1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1\end{pmatrix}$$

For numerical methods you can consider:

- $\psi=\psi_{(\theta)}=$ The wave function, which can be considered an eigenfunction. A upgraded eigenvector.
- $\hat{H}=$ Hamiltonian, a matrix representation of the Hamiltonian as it is discretized.
- $E=$ Energy the eigenvalue of a particular wave.

$$\hat{H}_{matrix}\psi=E\psi \Rightarrow \hat{H}_{matrix}\psi-E\psi=0 \Rightarrow \left(\hat{H}_{matrix}-E\right)\psi=0$$

Recognize!

$$\det(A-\lambda I)x=0 \Rightarrow \det\left(\hat{H}_{matrix}-EI\right)\psi=0 \qquad \psi\neq 0$$

then

$$(H-E_x I)\psi=0$$

- We find the eigenvalues E for psi, by solving det (H - EI) psi = 0. This returns possible values of E that satisfy the scrodingers equation. These are the possible allowed energy levels of the system.
- To find the wavefuntion(Eigenfuction) psi we solve (H - EI) psi = 0 for each E.

(5) This is the calcutations done by the code

$\Psi(n,1)$

$$\frac{-1}{2R^2(\Delta\theta)^2}\cdot 1 \simeq \frac{-1}{2R^2(\Delta\theta)^2} + 1\cdot\frac{iA(\theta)}{4R\Delta\theta} = \frac{iA(\theta)}{4R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta) \simeq \frac{-1}{2R^2(\Delta\theta)^2} + \frac{iA(\theta)}{R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta)$$

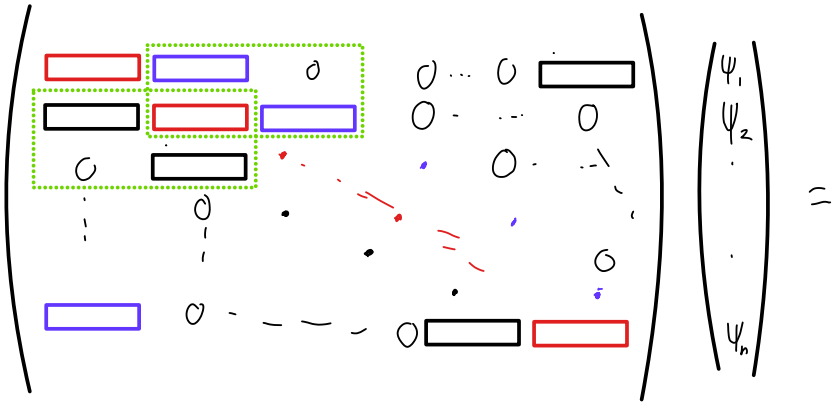
$\Psi(1,1)$

$$\frac{-1}{2R^2(\Delta\theta)^2}\cdot -2 \simeq \frac{2}{2R^2(\Delta\theta)^2} + 0\cdot\frac{iA(\theta)}{4R\Delta\theta} = 0 + \frac{A(\theta)^2}{2} + U(\theta) \simeq \frac{2}{2R^2(\Delta\theta)^2} + \frac{A(\theta)^2}{2} + U(\theta)$$

$\Psi(1,2)$

$$\frac{-1}{2R^2(\Delta\theta)^2}\cdot 1 \simeq \frac{-1}{2R^2(\Delta\theta)^2} + -1\cdot\frac{iA(\theta)}{4R\Delta\theta} = \frac{-iA(\theta)}{4R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta) \simeq \frac{-1}{2R^2(\Delta\theta)^2} - \frac{iA(\theta)}{R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta)$$

VISUALIZATION OF MATRIX

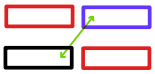


(6) The imaginary part

We se that for each line, every imaginary product is added to the negative of itself. Knowing that A(theta) is a constant, they should always cancel out in all the equation sets.

$$\left(\frac{2}{2R^2(\Delta\theta)^2} + \frac{A(\theta)^2}{2} + U(\theta)\right) + \left(\frac{-1}{2R^2(\Delta\theta)^2} - \frac{iA(\theta)}{R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta)\right) + 0 + 0 + \ldots + 0 + \left(\frac{-1}{2R^2(\Delta\theta)^2} + \frac{iA(\theta)}{R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta)\right)$$

When we look at the determinant, we can se that it is often multiplied with its negative part, known as the complex conjugate, this is causing only the real part to have a contribution. Since the system is a ring, we also have the effect that each part with a complex number also get an equivocal negative, leading them to get canceled out.

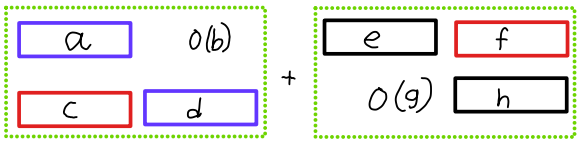


$$\left(\frac{-1}{2R^2(\Delta\theta)^2} - \frac{iA(\theta)}{R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta)\right)\cdot\left(\frac{-1}{2R^2(\Delta\theta)^2} + \frac{iA(\theta)}{R\Delta\theta} + \frac{A(\theta)^2}{2} + U(\theta)\right)$$

Variable $\cdot \frac{iA(\theta)}{R\Delta\theta}$ + same Variable $\cdot \frac{-iA(\theta)}{R\Delta\theta}$ = No imaginary part

$\frac{iA(\theta)}{R\Delta\theta}\cdot\frac{-iA(\theta)}{R\Delta\theta} = \frac{A(\theta)^2}{(R\Delta\theta)^2}$ = No imaginary part

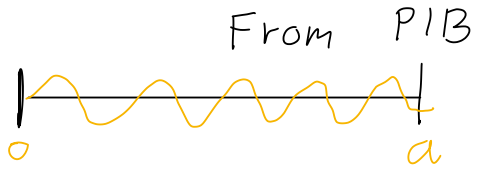
$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \quad \text{DETERMINANTEN}$$



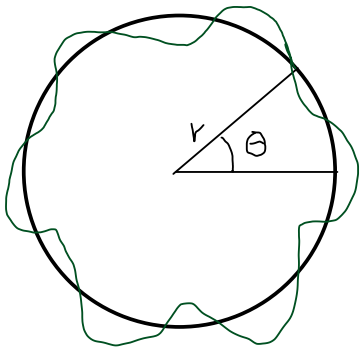
$$\left(\frac{iA(\theta)}{R\Delta\theta}\right)^2 - \left(\frac{iA(\theta)}{R\Delta\theta}\right)^2 = 0$$

(7) Normalization of a particle in a ring

From PIB



Like the wavefunction on a line, the ring has wavefunction, but in a ring the wave function occilates around the circle and joins its starting point. It has multiple functions with different energies. This wavefunction depends on the coordinates of the particle. It is a function of theta.



$$\Psi(\theta+2\pi)=\Psi(\theta)$$

Psi is given to us as:

$$\Psi=\cos(\theta)$$

But, it could very well have other forms:

$$A\cos(\theta k) \text{ or } A\sin(\theta k) \text{ or } Ae^{ik\theta}$$

For a particle on a ring, normalize the function over a period. Hardcoding the solution is also possible to save computational power. Or even better yet, coding a function that hard codes it whenever possible, is best.

$$1=\int_0^{2\pi}\bar{\Psi}\Psi\,d\theta=\int_0^{2\pi}|\Psi(\theta)|^2 d\theta \text{ and } \Psi(\theta)=Af(\theta) \Rightarrow |A|^2\int_0^{2\pi}|U(\theta)|^2\,d\theta$$

$$\text{Normalization factor } A=\frac{1}{\sqrt{\int_0^{2\pi}U(\theta)^2\,\Delta\theta}}$$

$$\int_0^{2\pi}\cos^2(\theta)\,d\theta=\int_0^{2\pi}\frac{1+\cos(2\theta)}{2}\,d\theta$$

$$\frac{1}{2}\left(\int_0^{2\pi}1\,d\theta+\int_0^{2\pi}\frac{\cos+(2\theta)}{2}\,d\theta\right)=\pi$$

The code

The function ring_potential takes in value theta and returns its angle.

```python
def ring_potential(theta):  # U-ring potential
    return np.cos(theta)
```

The discretizition is explained in the analog part 4. The function returns theta and the Hamiltonian matrix.

```python
def discretize_hamiltonian(N, R, A):
    dtheta = 2 * np.pi / N
    theta = np.linspace(0, 2 * np.pi, N, endpoint=False)

    # Create the Hamiltonian matrice to add the forces into it.
    main_diag = -2 * np.ones(N, dtype=complex)
    off_diag = np.ones(N - 1, dtype=complex)
    H = -(np.diag(main_diag) + np.diag(off_diag, 1) + np.diag(off_diag, -1)) / (R**2 * dtheta**2)

    # Periodic boundry conditions for the kinetic part
    H[0, -1] = H[-1, 0] = -1 / (R**2 * dtheta**2)

    # Add the iA-termen to H
    iA_term = -1j * A / (2 * R * dtheta)
    H += np.diag(iA_term * np.ones(N - 1), 1) - np.diag(iA_term * np.ones(N - 1), -1)

    # Periodic boundry conditions for imaginary term
    H[0, -1] += -iA_term
    H[-1, 0] += iA_term

    # Foinally add the potential to the Hamiltonian
    U = ring_potential(theta)
    H += np.diag(U)

    return theta, H
```

To solve the ring problem, we use the eigh function to return eigenvalues and eigenvectors from the hamiltonian matrix. Rename them to energies and wavefunctions, which is what we found. We normalize the wavefunctions explained in theory part 7.

```python
def solve_ring_problem(N, R, A, num_states=3):
    theta, H = discretize_hamiltonian(N, R, A)
    eigenvalues, eigenvectors = eigh(H)

    energies = eigenvalues[:num_states]
    wavefunctions = eigenvectors[:, :num_states]

    dtheta = theta[1] - theta[0]
    for i in range(num_states):
        # Compute the normalization factor for each wavefunction
        integral = np.sum(np.abs(wavefunctions[:, i])**2) * dtheta # This is Pi
        norm_factor = 1 / np.sqrt(integral)
        wavefunctions[:, i] /= norm_factor
    return theta, energies, wavefunctions
```

This is the start of the Main method. It starts with initializing all known variables.

```
# Parametere
N = 1000                    # Number of steps
R = 1.0                     # Radius
A = 0.6                     # A(theta) constant
# U is defined later due to theta not being constructed yet.
num_states = 3              # Number of states
```

Runs the code to solve the problem.

```
# Løsning
theta, energies, wavefunctions = solve_ring_problem(N, R, A, num_states)
```

Prints the energy to the run windows, for comparing scripts.

```
# Energier
print("Energy eigenvalues (in arbitrary units):")
for i in range(num_states):
    print(f"E_{i+1} = {energies[i]:.4f}")
```

Calculates the ground state probability density and ring potential and plots this on a graph.

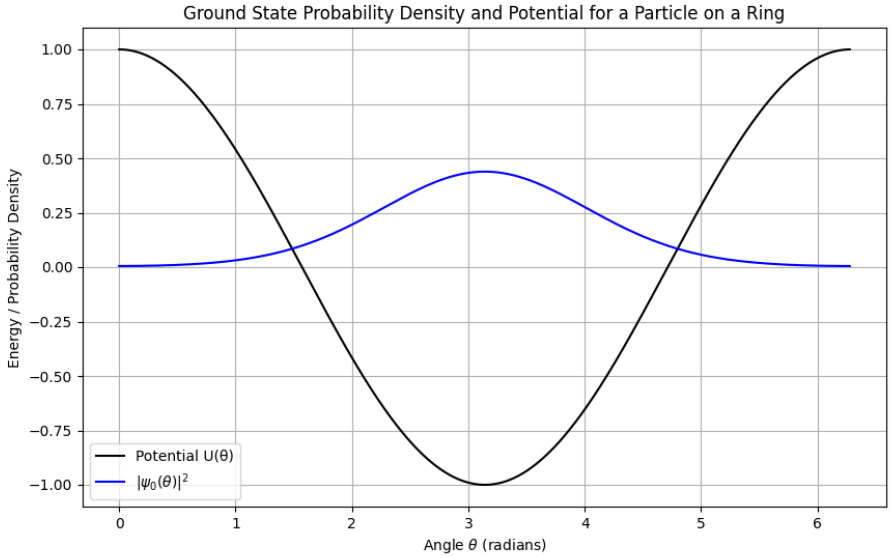
```
ground_state_probability_density = np.abs(wavefunctions[:, 0])**2
U = ring_potential(theta)
plt.figure(figsize=(10, 6))
plt.plot(theta, U, label='Potential U(θ)', color='black')
plt.plot(theta, ground_state_probability_density, label=r'$|\psi_0(\theta)|^2$', color='blue')
plt.xlabel(r'Angle $\theta$ (radians)')
plt.ylabel('Energy / Probability Density')
plt.title('Ground State Probability Density and Potential for a Particle on a Ring')
plt.legend()
plt.grid(True)
plt.show()
```

Plots the first three eigenstates

```
plt.figure(figsize=(10, 6))
for i in range(num_states):
    plt.plot(theta, np.abs(wavefunctions[:, i])**2, label=f'$|\psi_{i+1}(\theta)|^2$ (E={energies[i]:.2f})')
plt.xlabel(r'Angle $\theta$ (radians)')
plt.ylabel(r'$|\psi_n(\theta)|^2$')
plt.title('Probability Density of the forst few Eigenstates on a Ring')
plt.legend()
plt.grid(True)
plt.show()
```

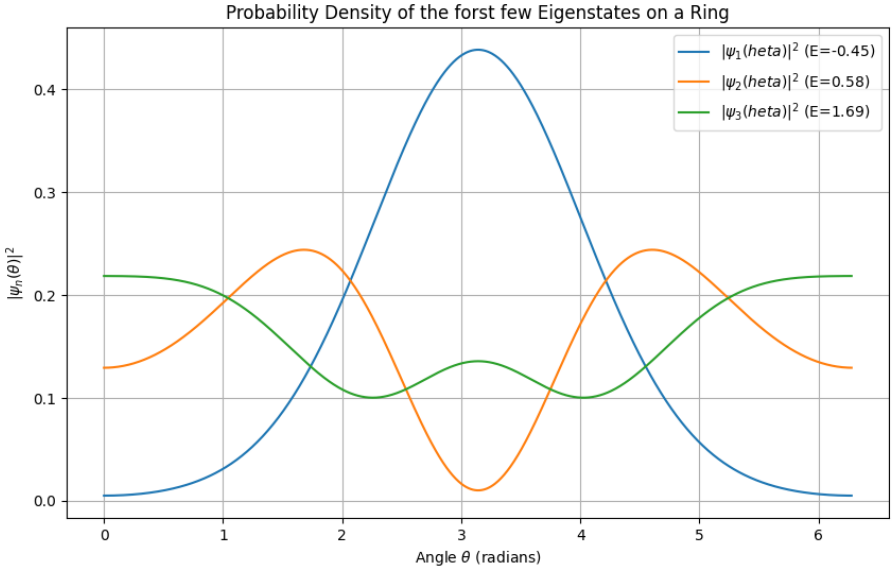
Results and discussion

① Ground state and ring potential



If the energy of the particle known as E is higher than the potential this means that the particle has access to that area and is most likely found in that location. As the ground state has a higher potential near approximate 3.2 , it means that the particle is most likely to be found here. The probability of finding the particle where it's lower than the potential is very low, but not zero.

② First and second excited state



The energy of the first exited state absolute of psi 2 squared, has two humps on its curve signifying that it is most likely to be found in two distinct regions 1,7 and 4,6. The distribution is split between two regions. The second exited state absolute of psi 3 squared is oscillating even faster. It has two positive humps and one less distinct one. Increased energy looks like it has a correlation to how much it is oscillating, in general increasing with one with each higher eigenstate. The probability never reaches zero, signifying that the particles can be found anywhere on the ring.

What I learned by doing this project

First I made the mistake of caluculating the problem as commutative and no classmates, GPT ot Maryam was able to see my mistake. I used (a*b)ˆ2 = aˆ2 + 2ab + bˆ2 insted of (a*b)ˆ2 = aˆ2 + ab + ba +bˆ2. The script with the original squared term and calculated term, gave different answers and finding out why took a long time. To find out why I calculated the matrix with N3. This did not give the same answer, this led to the discovery that calculating it commutative vs non-commutative gave nearly identical answers except for a R factor by 2 in one of the devisors. To avoid doing similar mistakes forward I decided that adding an operator on the d/dt operator was the best approach, at least while learning.

The scripts still did not match each other after calculating and the code was running without any errors. I was clueless, because now my handwritten notes showed that the matrices should be equal. I tried GPT and it insisted that the errors was due to numerical differences and rounding errors. Wrong answer but i learned something, it makes sense that having more calculations in the matrice could cause rounding errors. I put the codes N to 3 on the codes and the difference was still present. That is when GPT discovered the most subtle and scary coding mistake. = should have been += and the first kinetic energy was over written. This was very useful.

When discretizing the matrix I was a bit confused, because i did not excactly remember and could not understand how to set it up. It was not well explained in the notes, the book or online where I could find. I realized how it was done and understood that a step was missing in the explanation. I tried to fix this myself by creating the I ixj where j=n operator. I belive this will shave of the time for anyone to understand how to set up the matrix.

When it comes to efficiency and accuracy i belive that solving the calculation before creating the matrix, could remove unecessary links and thus create more efficiency and accuracy per 'run time unit' on calculating the eigenstates and eigenvalues. This could lead to more accurate calculations in complex systems. A function that could simplify the equation would be optimal.

A program like word was deeply missed during this project, the formatting was a hasle. But the flow of the document in the end was a big upside. Reading the handwritten notation has it up and downsides. It is closer to what we have been taught, but still understanding my handwriting is easiest done by me. As i have learned by using Overleaf for my formula sheets, this would probably be the best application for this, but that is yet to be tested.