ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Towards a simulation of phonon states in quantum crystalline beams

Semester Thesis

Florian Wepfer

June 2024

Advisors: Dr. Andreas Adelmann, Sebastian Heinekamp

ETH Zürich

Abstract

Circular accelerators have been proposed as storage devices for qubits. Building on previous work, this thesis works towards the goal of numerically simulating such a system. In this context, three tasks have been carried out: The choice of a basis suitable for simulating the inner degrees of freedom of the ion used as qubit; a discussion of work done by another author on a similar system with the goal of using their findings to verify the results from our simulation; and an investigation into an error in the current version of the simulation where, in constant magnetic fields the ground state energy of the ion changes over time, depending on the length of the time step used to simulate the time evolution.

I propose to use correlation consistent valence only basis sets, and identify two possible numerical experiments. Finally, I am able to reduce, but not completely eliminate the energy drift.

Contents

Chapter 1

Introduction

Though the advantage of quantum computers over classical machines has been demonstrated numerous times, several challenges remain before quantum computers can become practical. One of those challenges is that of storing and effectively manipulating the ions to be used as qubits. At the moment, this is done in linear ion traps, where ions are caught and stored next to each other, at rest. Brown and Roser suggested changing the linear ion trap for a circular storage ring, where they would travel in circles along an accelerator[1]. In this project, we consider one possible implementation of this system, using Be^+-ions and storing the information in two hyperfine states of the valence electron. In a time-varying magnetic field, this electron is subject to the Hamiltonian:

$$H = \frac{1}{2m}p^2 + V(r) + \frac{\mu_B}{\hbar}(\vec{L} + g_s\vec{S} + g_I\vec{I})\vec{B}(t) + \eta\vec{L}\cdot\vec{S} + A\vec{I}\cdot\vec{J} \tag{1.1}$$

Where $\vec{p}$ is the momentum of the electron, $V(r)$ is its potential energy in the field of the core, $\vec{L}$ is the electrons orbital angular momentum, $\vec{S}$ its spin and $\vec{J}$ its total angular momentum. Additionally, $\vec{I}$ is the spin of the core. μ_B, g_S and g_I are constants describing how these angular momenta individually interact with the magnetic field, and η and A describe the way they couple to each other, the orbital angular momentum to the spin of the electron by the Zeemann effect, and the total angular momentum of the electron to the spin of the core by the hyperfine splitting.

Previous work has been carried out to simulate such a system numerically by R. Husistein [2], and later by M. Solinas[3]. They used pySCF, a quantum chemistry python package [4] to simulate the internal state of the electron and real-time time dependent Hartree-Fock to simulate its evolution with time. To complete this project, however, some tasks remain and for this semester thesis, three of those have been carried out:

Basis sets: The simulation of the internal degrees of freedom of the Beryllium ion used in our simulation requires the use of a basis to describe the state of the electron. I compare some of the available sets and choose one suited for our purposes. I calculate the interaction of the orbital angular momentum operator with these basis functions, and implement a function that calculates this interaction numerically.

Test of the Simulation: I summarize a thesis that considers systems similar to ours and evaluate whether the theoretical considerations as well as the experiments carried out there can be used to verify the results of our simulation. In particular, we could try to simulate stimulated Raman transitions between some hyperfine states of the ion, or try to replicate some measurements of the decoherence time of the qubit.

Energy Drift: The majority of my time during this semester was devoted to fixing an error in the present version of the simulation: Under the influence of a constant magnetic field, the energy of the ion drifts during the time evolution, and it does so more with smaller time steps, which is the opposite of what one might expect.

Chapter 2

Methods

2.1 Hartree-Fock

Finding the precise wavefunction of systems of more than one electron is, in most cases, impossible. Thus we require the help of approximate methods solve the Schrödinger equation in those cases. One such method, the one used for this project, is the Hartree-Fock method, which will be presented here: The goal of the Hartree-Fock method is to find the wave function satisfying Paulis exclusion principle such that the expectation value of the Hamiltonian is minimized, that is to say $E = \langle \Psi | H | \Psi \rangle$ should be minimal, where for an atom of K nuclei and N electrons, the Hamiltonian reads:

$$H = \sum_{i=1}^{N} h(i) + \frac{1}{2} \sum_{i \neq j=1}^{N} g(i,j), \text{ with} \tag{2.1a}$$

$$h(i) = -\frac{1}{2}\Delta_i - \sum_{n=1}^{K} \frac{Z_n}{|\vec{r_i} - \vec{R_n}|} \text{ and} \tag{2.1b}$$

$$g(i,j) = \frac{1}{|\vec{r_i} - \vec{r_j}|} \tag{2.1c}$$

By the variational principle this is then the wave function assumed by the system in the ground state. To satisfy Paulis exclusion principle, we make an ansatz for the n-particle wave function using a Slater determinant of single particle wave functions:

$$\Psi(\vec{x_1}, ..., \vec{x_N}) = \frac{1}{\sqrt{N!}} det(\psi_j(\vec{x_i}))_{i,j=1,...,N} \tag{2.2}$$

This then gives an energy expectation value of

$$E = \sum_k \langle \psi_k | h | \psi_k \rangle + \frac{1}{2} \sum_{k,l} (\langle \psi_k \psi_l | g | \psi_k \psi_l \rangle - \langle \psi_k \psi_l | g | \psi_l \psi_k \rangle) \tag{2.3}$$

Based on this expression we can define the operators

$$J_k(\vec{x})\psi(\vec{x}) = \int \psi_k^*(\vec{x'})\frac{1}{|\vec{x'}-\vec{x}|}\psi_k(\vec{x'})\psi(\vec{x})d^3x' \text{ and} \tag{2.4a}$$

$$K_k(\vec{x})\psi(\vec{x}) = \int \psi_k^*(\vec{x'})\frac{1}{|\vec{x'}-\vec{x}|}\psi(\vec{x'})\psi_k(\vec{x})d^3x' \tag{2.4b}$$

as well as

$$J = \sum_k J_k \text{ and } K = \sum_k K_k. \tag{2.5}$$

In the interest of brevity, a detailed variation of the functional shall be omitted here, see [5] for more details, but note that it leads to the eigenvalue equation

$$F\psi_k := (h + J - K)\psi_k = \varepsilon_k\psi_k. \tag{2.6}$$

After inserting a basis expansion $\psi_k = \sum_p C_{pk}\chi_p$, this equation, in matrix form, yields the Roothaan equation:

$$\mathbf{F'P'} = \mathbf{SP'}\varepsilon \tag{2.7}$$

With the Fock matrix $\mathbf{F'}_{\mu\nu} = \langle\chi_\mu|F|\chi_\nu\rangle$, the overlap matrix $\mathbf{S}_{\mu\nu} = \langle\chi_\mu|\chi_\nu\rangle$, and the density matrix $\mathbf{P'}_{\mu\nu} = \sum_i C^*_{\mu,i}C_{\nu,i}$.

This equation can be solved iteratively, by calculating the Fock matrix for some initial guess of the wave function, then solving the eigenvalue equation and repeating the process with the newly found eigenvectors of the Fock operator until the process converges.

2.2 Real-time time dependent Hartree-Fock

The Hartree-Fock method gives the ground state of the system for a Hamiltonian at one point in time. We, however are interested in how the state changes with time. Therefore, a method is needed by which the time evolution of a state under a time-varying Hamiltonian can be found. Such a method is the real-time time dependent Hartree-Fock method.

In general, the basis chosen above is not orthogonal, and so the overlap matrix $\mathbf{S}$ not the identity. However, by applying a suitable orthogonalisation scheme such as the Löwdin orthogonalisation, we can change that. By setting $\mathbf{V} = \mathbf{S}^{1/2}$, and then $\mathbf{P} = \mathbf{VP'V}^T$ and $\mathbf{F} = \mathbf{V}^{-T}\mathbf{F'V}^{-1}$, Equation 2.7 transforms into a regular eigenvalue equation.

In this orthonormal basis, the time evolution of the density matrix is given by the Liouville-von Neumann equation:

$$i\frac{d\mathbf{P}(t_i)}{dt} = [\mathbf{F}(t_i), \mathbf{P}(t_i)] \tag{2.8}$$

Which can be solved numerically by a number of methods, but the ones used in this thesis, proposed in [6] are the modified midpoint unitary transformation and the explicit second order Magnus propagator. Using the former method the density matrix at time step t_{k+1} is given by

$$\mathbf{P}(t_{k+1}) = \mathbf{U}(t_k)\mathbf{P}(t_{k-1})\mathbf{U}^\dagger(t_k), \text{ where} \tag{2.9a}$$
$$\mathbf{U}(t_k) = exp(2i\Delta t\mathbf{F}(t_k)), \tag{2.9b}$$

and using the latter by

$$\mathbf{P}(t_{i+1}) = \mathbf{U'}\mathbf{P}(t_{i+1/2})\mathbf{U'}^\dagger, \text{ where } \mathbf{U'} = exp(i\Delta t\mathbf{F}(t_i)/2), \text{ and} \tag{2.10a}$$
$$\mathbf{P}(t_{i+1/2}) = \mathbf{U}\mathbf{P}(t_{i-1/2})\mathbf{U}^\dagger; \mathbf{U} = exp(i\Delta t\mathbf{F}(t_i)). \tag{2.10b}$$

This time evolution was implemented by R. Husistein [2], but the current implementation has an error in its calculation, which I will analyze and propose some improvements for in section 4.3.

2.3 Basis sets

The Hartree-Fock method as presented above yields expansion coefficients of the wave function in some basis of the relevant Hilbert space. This requires that the basis is chosen before the start of the calculations. Choosing a basis however, is in itself not a trivial task. Historically, inspired by the solution to the one-electron problem, so-called Slater-type orbitals have been used:

$$\chi(r,\theta,\varphi) = \tilde{N}Y_{l,m}(\theta,\varphi)r^l e^{-\zeta r} \tag{2.11}$$

Where (r,θ,φ) are the spherical coordinates relative to the core of the atom, $Y_{l,m}$ is the spherical harmonic of index (l,m), $\tilde{N}$ is a normalisation constant and ζ is the effective core potential. For a one electron problem, the effective core potential is just the number of positive charges in the core, however in multi-electron problems the core potential is modified to reflect the shielding of the positively charged core by the negatively charged electrons. To approximate the atomic orbitals using these functions to the desired accuracy, one finds a number of effective core potentials such that a superposition of Slater type orbitals using these core potentials approximates the atomic orbital well. The higher the number of these functions, the better the approximation, hence the number of effective core potentials is usually reflected in the name of the basis, containing the word "single" if one ζ is used per atomic orbital, "double" if two are used and so on. The spherical

harmonics on the other hand are given by:

$$Y_l^m(\theta,\varphi) = (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(cos\theta) e^{im\varphi}, \text{ with} \tag{2.12}$$

$$P_l^m(x) = \frac{(-1)^m}{2^l l!}(1-x^2)^{m/2}\frac{d^{l+m}}{dx^{l+m}}(x^2-l)^l. \tag{2.13}$$

There is a problem that arises if one uses Slater type orbitals, however: They are not easily integrable. To fix this issue, Gauss-type orbitals can be used. These are given by

$$\chi(r,\theta,\varphi) = \bar{N} Y_{l,m}(\theta,\varphi) r^l e^{-\zeta r^2} \tag{2.14}$$

or, in cartesian coordinates and using different indexing

$$\chi(x,y,z) = N x^k y^p z^q e^{-\zeta r^2} \tag{2.15}$$

Here, analogous to above, $\bar{N}$ and N are normalisation constants found by requiring that each basis function integrates to unity over all of space. Usually, and in particular in the present case, multiple of these so-called primitive Gauss-type orbitals are superimposed to form a contracted Gauss-type orbital, which is then used as a basis function. Which primitive orbitals are contracted and how is, in general, a complicated process that involves fitting the effective core potentials and the contraction coefficients to the problem at hand.

A large number of basis sets, optimized for different elements in different settings have been published and are at our disposal. They can be downloaded from basissetexchange, an online resource [7] or are already implemented in pySCF. From basissetexchange we get a representation of the basis in the NWChem format, in which a basis set of one particular angular momentum, eg. 'S' is expressed as follows:

'Atom' 'Symbol for angular momentum'

ζ_1 $c_{1,1}$ $c_{1,2}$ ...

ζ_2 $c_{2,1}$ $c_{2,2}$ ...

...

where a basis function is then given by

$$\varphi_{i;l,m}(r,\theta,\varphi) = \sum_j c_{j,i} \tilde{N} Y_{l,m}(\theta,\varphi) r^l e^{-\zeta_j r^2}, \tag{2.16}$$

and the basis contains a function for each possible configurations of l and m.

Depending on the basis set, there might be a higher number of effective core potentials used to approximate one atomic orbital, or basis functions with higher angular momentum numbers might be included. Additionally, a basis set might include basis functions for all electrons or only for the valence shell. A few of the most popular basis sets were considered for our task. The findings are summarized here:

STO-NG: In the STO-NG basis set, N Gauss-type-orbitals are used to approximate one Slater-type-orbital[8]. By the method of least squares, each Slater type orbital is replaced by the best-fitting linear combination of N gaussian orbitals. These sets are not valence only, meaning they consider all electrons of the atom and not just the valence shells, which are most relevant for our calculations.

Pople basis sets: Denoted by X-YZg in the case of a double zeta basis, where X is the number of primitive Gauss-type-orbitals used to approximate each core atomic orbital, and X and Y are the number of Gauss-type-orbitals for the two zetas of the valence orbitals. These basis sets improve upon the STO-NG sets, by keeping the accuracy available for the core electrons the same, but increasing it for the valence electron by including two functions of different ζ.

Jensen basis sets: These sets are denoted pc-N and are similar to the cc-pvNz sets, but are optimized for density functional theory, so of little use in our case[9].

Karlsruhe basis sets: Karlsruhe basis set are a large family of basis sets that are geared towards computing properties of larger molecules, not, as in our case, one single ion[10].

cc-pvNz: Correlation consistent polarized valence N zeta basis sets are the current state of the art. They are usually valence only, but variants that include core-valence correlations exist.[11] In the development of this set, care was taken to describe electron correlations well, and the inclusion of functions of higher angular momentum makes it possible to better represent polarized atoms.

2.4 Angular momenta of Gaussian orbitals

We want to implement a function that evaluates the angular momentum operator, $\vec{L}$ on a basis of cartesian Gauss-type orbitals, given in Equation 2.15, i.e. we would like to evaluate the matrix $\langle g_{k',p',q'}|L_i|g_{k,p,q}\rangle$ for some integers k, p, q, k', p', q' and $i \in \{1, 2, 3\}$. For $i = 1$, the angular momentum operator

acts as follows on a Gauss-type orbital:

$$\begin{aligned} L_1 g_{k,p,q} &= -i\hbar \left(y\frac{\partial}{\partial z} - z\frac{\partial}{\partial y} \right) g_{k,p,q} \\ &= -i\hbar \left(q\frac{N_{k,p,q}}{N_{k,p+1,q-1}} g_{k,p+1,q-1} - p\frac{N_{k,p,q}}{N_{k,q+1,p-1}} g_{k,p-1,q+1} \right) \end{aligned} \tag{2.17}$$

Since the normalisation constant is given as

$$N_{k,p,q}^2 = \left(\frac{2\xi}{\pi} \right)^{3/2} \frac{(8\xi)^{k+p+q} k!p!q!}{(2k)!(2p)!(2q)!} \tag{2.18}$$

We can simplify Equation 2.17 to

$$L_1 g_{k,p,q} = -i\hbar \left(q\sqrt{\frac{2p+1}{2q-1}} g_{k,p+1,q-1} - p\sqrt{\frac{2q+1}{2p-1}} g_{k,p-1,q+1} \right) \tag{2.19}$$

The analogous results for the $i = 2$ and $i = 3$ axes follows similarly.

The matrix element now reduces to two terms, each proportional to an overlap of two Gauss-type orbitals:

$$\begin{aligned} \langle g_{k',p',q'} | L_1 | g_{k,p,q} \rangle = -i\hbar \Bigg(& q\sqrt{\frac{2p+1}{2q-1}} \langle g_{k',p',q'} | g_{k,p+1,q-1} \rangle \\ & - p\sqrt{\frac{2q+1}{2p-1}} \langle g_{k',p',q'} | g_{k,p-1,q+1} \rangle \Bigg) \end{aligned} \tag{2.20}$$

Since the overlaps of Gaussian orbitals are known, this can be solved precisely and concludes my analysis of angular momenta of these orbitals.

2.5 The $^9Be^+$-qubit

One possible combination of states to use as the $|1\rangle$ and $|2\rangle$ states for quantum computations using the Beryllium qbit are the $S_{1/2}$, $|F = 2, m = 0\rangle$ and the $S_{1/2}$ $|F = 1, m = 1\rangle$ states, where F is the sum of the nuclear and total electron angular momentum, and m is its projection onto the z-axis, because, at a magnetic field of $119.4428G$, the transition between those two states is independent of the magnetic field to first order, with a transition frequency of $1.2074958GHz$. Another, albeit field-dependent qubit is the one consisting of the $S_{1/2}$ $|F = 1, m = 1\rangle$ and the $P_{1/2}|F = 2, m = 2\rangle$ states. Additionally, the $|3\rangle$-state to be used for the stimulated Raman Transitions introduced below is the $P_{3/2}$, $|F = 3, m = 3\rangle$-state.

2.6 Stimulated Raman Transitions

Stimulated Raman transitions are a mechanism to populate one hyperfine ground state, $|2\rangle$ from another, $|1\rangle$ by coupling them to an exited state $|3\rangle$. This is achieved by exposing the ion to two copropagating laser beams, $\vec{E}_i = \hat{\varepsilon}_i E_i cos(\vec{k}_i \cdot \vec{r} - \omega_i t + \varphi_i)$, for $i \in \{\text{blue, red}\}$. Here, $\hat{\varepsilon}_i$ is the polarisation and E_i the amplitude of the field, and $\vec{k}_i$ the wave vector, ω_i the frequency and φ_i the phase shift. The frequencies of the blue and red fields, ω_b and ω_r are the transition frequencies from the $|1\rangle$ and $|2\rangle$ to the $|3\rangle$ state, ω_{13} and ω_{23} respectively, up to some detuning to account for the doppler shift of the laser frequency in the rest frame of the ion given by $\Delta_b = \omega_b - \omega_{31}$ and $\Delta_r = \omega_r - \omega_{32}$. In this setup, and under certain conditions we can avoid significantly populating the $|3\rangle$-state, effectively achieving a two-level system. Indeed, if we approximate the ion as a dipole, with an atomic dipole moment given by $\vec{d} = e\vec{r}$, make the rotating-wave approximation by neglecting terms with high frequency oscillation such as $e^{\pm i2\omega_i t}$, $i \in \{\text{blue, red}\}$, and approximate to lowest order in the inverse of the detuning frequency of the blue beam, Δ_b, we can achieve a constant amplitude of the excited state of

$$c_3 = \frac{1}{2\Delta_b}(g_b c_1 e^{i(\vec{k}_b \cdot \vec{r} + \varphi_b)} + g_r c_2 e^{i(\vec{k}_r \cdot \vec{r} + \varphi_r)}) \tag{2.21}$$

where $g_b := \frac{1}{\hbar} E_b \langle 3|\vec{d} \cdot \hat{\varepsilon}_b|1\rangle$ and $g_r := \frac{1}{\hbar} E_r \langle 3|\vec{d} \cdot \hat{\varepsilon}_r|2\rangle$ are the single-photon Rabi frequencies, and c_1 and c_2 the amplitudes of the $|1\rangle$ and $|2\rangle$ states. With these approximations, the Hamiltonian of the system is:

$$H_I = \frac{\hbar}{2}\Omega \left(|2\rangle\langle 1| e^{i(\Delta\vec{k} \cdot \vec{r} + \Delta\varphi - \delta t)} + |1\rangle\langle 2| e^{-i(\Delta\vec{k} \cdot \vec{r} + \Delta\varphi - \delta t)} \right) \tag{2.22}$$

where $\Omega = \frac{g_b g_r^*}{2\Delta_b}$, $\Delta\vec{k} = \vec{k}_b - \vec{k}_r$, $\Delta\varphi = \varphi_b - \varphi_r$, and $\delta = \Delta_b - \Delta_r$.

The dynamics of this system are the Rabi oscillations: sinusoidal oscillations of the population of $|1\rangle$ and $|2\rangle$ states with frequency Ω.

2.7 Spin coherence

To measure Spin coherence, the ion is prepared in the $|1\rangle$-state. Then, a $\pi/2$-pulse is applied, which creates a superposition of the two qubit states $|1\rangle$ and $|2\rangle$, evolving in time as: $|\psi\rangle(\varphi) = \frac{1}{\sqrt{2}}(e^{i\varphi}|1\rangle - i|2\rangle)$, where $\varphi = \int_0^T (\omega_R(t) - \omega_0(t))dt$, where ω_R and ω_0 are the frequency difference between the Raman beams, and ω_0 the qubit transition frequency, respectively.

The $\pi/2$-pulse is a single qubit gate operation, which is implemented using the Raman beams introduced in section 2.6 and performs rotations of the state around the Bloch sphere. Mathematically, these rotations are given by

$$\begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} = \hat{R}(\theta, \varphi) \begin{bmatrix} c_1(0) \\ c_2(0) \end{bmatrix}, \text{ with} \tag{2.23a}$$

$$\hat{R}(\theta, \varphi) := \begin{bmatrix} cos\left(\frac{\theta}{2}\right) & -ie^{i\varphi} sin\left(\frac{\theta}{2}\right) \\ -ie^{-i\varphi} sin\left(\frac{\theta}{2}\right) & cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{2.23b}$$

With $\theta = \Omega t$, and $\varphi = \varphi_b - \varphi_r$ a phase. Here, the $\pi/2$-pulse is equivalent to applying $\hat{R}(\pi/2, 0)$ to the qbit.

After letting the system evolve for some time T, we can apply another $\pi/2$-pulse, this time with varying phase, and perform a projective measurement on the resulting state for the probability of being in the $|1\rangle$-state. This, theoretically should behave as $P(|1\rangle) = \frac{1}{2}(1 - cos(\varphi + \varphi_0))$, but since fluctuations in the magnetic field might cause a different behaviour, this measurement can be used as a metric of how well tuned the setup is.

Chapter 3

Experiment

3.1 Proposed Experiments

Once the simulation is complete, we need a way to verify its results. This could be done by comparing to results from experiments. In their phd thesis [12], Lo discuss some theoretical properties of the 9 Be+ qubit as well as some experiments with that same qubit. Because they use a similar system to ours, we could try to replicate these results in our simulation.

Using our implementation of time-varying electromagnetic fields, we could expose the Beryllium qubit to the copropagating raman beams introduced in section 2.6. If the simulation works correctly, we should then observe Rabi oscillations in the populations of the $|1\rangle$ and $|2\rangle$ states.

Additionally, we could try to replicate the experimental result achieved by Lo [12] when they measured the spin coherence of the Beryllium ion and observed the oscillations described in section 2.7 in a similar fashion.

3.2 Angular momenta of Gaussian orbitals

I implemented the function outlined in section 2.4 by getting the relevant overlap matrix elements from pySCF and constructing the L_i-matrices from those. For a 3-Vector B and some pySCF-molecule object mol, I could then construct the angular momentum matrix by calling

```
S = BL(mol, B)
```

However, it was subsequently discovered that pySCF already has a function implemented that calculates this matrix, and so an equivalent result could be achieved by calling

```
Svec = mol.intor('int1e_cg_irxp')
```

This returns an array of angular momenta matrices along the x, y and z axes which can be used to construct the same matrix as above, by summing over the spatial components:

```
S = B[0]*Svec[0,:] + B[1]*Svec[1,:] + B[2]*Svec[2,:]
```

3.3 Energy drift

In the original implementation there was a drift in energy of the ion when subjected to a constant magnetic field that increased with decreasing time step. This was a problem despite the error being rather small, as one would expect it to decrease as we decrease the time step. For this reason, I set out to find the cause of the drift. Some of the things I tried are outlined below.

3.3.1 Elimination of some base transformations

To reach the orthogonality of the basis needed to perform the time integration, a Löwdin orthogonalisation is performed as explained in section 2.2. Originally, this was done by calculating the orthogonalising base change matrices before the time loop, then in each time step, transforming the "self.density_matrix", performing the integration and then, after transforming back, saving the result to "self.density_matrix":

```
...
for i in range(num_time_steps):
            ...
            # transform to orthogonal basis
            density_matrix =
            np.linalg.multi_dot([V, self.density_matrix, V.T])
            # time propagation
            density_matrix =
            np.linalg.multi_dot([U, density_matrix, np.conj(U).T])
            # transform to AO basis
            self.density_matrix =
            np.linalg.multi_dot([X, density_matrix, X.T])
            ...
```

I improved this, by transforming the density matrix saved in the class only once into orthogonality, locally saving the result, and only transforming back in every time step to write to the "self.density_matrix" variable:

```
...
# transform to orthogonal basis
            density_matrix =
            np.linalg.multi_dot([V, self.density_matrix, V.T])
for i in range(num_time_steps):
```

```
...
# time propagation
density_matrix =
np.linalg.multi_dot([U, density_matrix, np.conj(U).T])
# transform to AO basis
self.density_matrix =
np.linalg.multi_dot([X, density_matrix, X.T])
...
```

3.3.2 Second vs. first order time integration scheme

I replaced the explicit second order Magnus propagator used thus far by the modified midpoint unitary transformation, both introduced in section 2.2 and compared the results.

3.3.3 mpmath

mpmath is a Python library that provides a few often-used mathematical functions similar to scipy, but that can compute these functions to arbitrary precision. In two places I replaced a scipy function with the corresponding function provided by mpmath and investigated the effect. Namely, this was in the calculation of the basis transformation matrix, $\mathbf{V}$ and in the calculation of the matrix exponential for the time propagation. Some care needed to be taken to convert the matrices to the correct data types for the respective libraries, but this was done, for example in the case of the matrix exponential, by replacing

```
U = sp.linalg.expm(-1j * dt * F)
```

with the corresponding mpmath function

```
F_tilde = mpmath.expm(mpmath.matrix(-1j * dt * F)
U = np.array(F_tilde, method='taylor'))
U.reshape(F.shape).astype(np.complex128)
```

3.3.4 Changing the magnetic field

So far, all tests have been carried out with the magnetic field being parallel to the z-axis and nonzero. To investigate whether polarisation or along different axes or any presence of the magnetic field whatsoever had any effect on the energy drift, I initialized the simulation with magnetic fields along the x and y axes, or with it turned off completely.

3.3.5 More precise density matrix

Observations the drift of time steps of length $0.05 a.u.$ and higher suggested that truncation errors due to numbers too close to machine precision might

be responsible for the drift. Therefore I can try to improve the code by saving the density matrix with higher precision.

3.3.6 Elimination of base changes

Inspired by the first changes that improved the situation, outlined in section 4.3.1, I looked for a way to eliminate the need for a change of basis altogether. This would be the case if the overlap matrix was the identity to begin with. So I changed the basis from the start, i.e. directly manipulated the NWChem string that is fed to pySCF such that the density matrix was diagonal, up to some entries that were off by an order of 10^{-17}.

Chapter 4

Results

4.1 Choice of basis

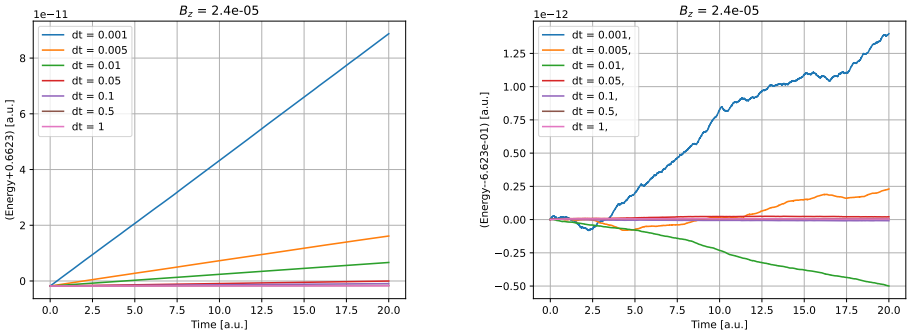
Neither Jensen nor Karlsruhe Basis sets seem suited to our case since both these families of sets were optimized for a different setting to ours. STO-NG and Pople basis sets are computationally cheap, but less precise than the correlation consistent sets. These offer relatively high accuracy at a low computational cost. Additionally they are valence only, which is what we need since the core electrons contribute very little to the physics considered in this project. Thus, of the sets I considered these are the ones most suited for our calculations. The accuracy can be tuned by choosing the number of zetas used, but for this thesis, in almost all the cases and unless noted otherwise, double zeta was used.

4.2 Angular momenta of Gaussian orbitals

Though the discovery of the angular momentum function in pySCF allowed me to verify my calculations and discover that they were correct to within machine precision, it made them and any further analysis of the implementation redundant.

4.3 Energy drift

With experiments outlined above I managed to reduce the energy drift, but I failed to completely eliminate it. A detailed discussion of the results is given below.



(a) Status quo

(b) First improvements

Figure 4.1

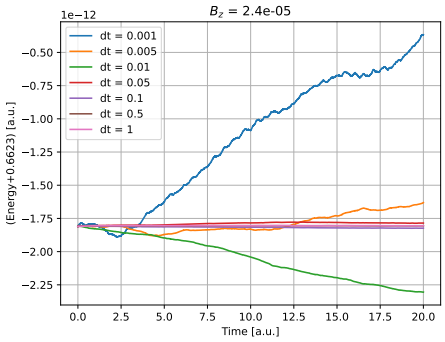


Figure 4.2: Same as Figure 4.1b, but using the first order scheme.

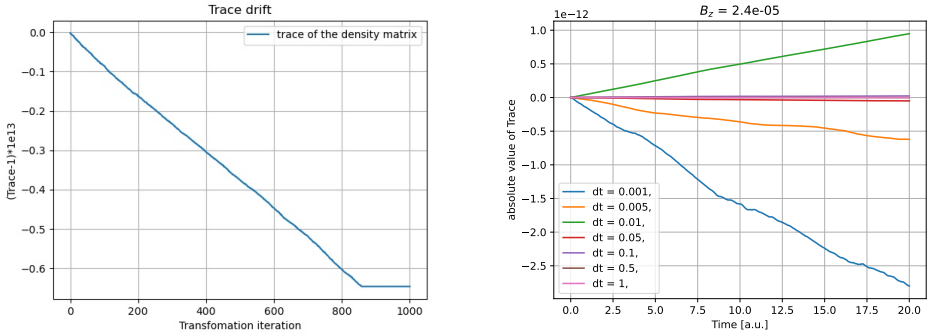
4.3.1 Elimination of some base transformations

Only transforming into orthogonality once instead of in every time step brought the energy drift down from roughly $9 \cdot 10^{-11} a.u./20 a.u.$ to roughly $1.25 \cdot 10^{-11} a.u./20 a.u.$ for a step length of $0.001 a.u.$, shown in Figure 4.1b.

4.3.2 Second vs. first order time integration scheme

As one can tell from comparting Figure 4.2 to subsection 4.3.1, the behaviour of the two integration schemes with respect to the energy drift is identical. Thus, to speed up further experiments, the first order scheme was used.

At this point, I probed the drift a bit more comprehensively than just considering the energy drift, namely by observing the trace of the density matrix during the simulation and the convergence of the different parts of the Fock operator.



(a) Drift of the trace for multiple transformation iterations

(b) Drift of the trace of the density matrix during time integration.

Figure 4.3

4.3.3 Trace drift

From back-and-forth transforming

When the transformation into the orthogonal basis is applied back-and-forth several times consecutively, we can observe a drift in the trace of the density matrix in Figure 4.3a. This confirms that the changes made in subsection 4.3.1 were indeed justified.

During the simulation

Even with one of the transformations done in each timestep eliminated, however, we can still observe a drift in the trace of the density operator during the time integration in Figure 4.3b, meaning some of the operations done to it are not trace preserving. This alone, however cannot be the reason for the drift in energy, as normalizing the trace to unity in each timestep does not fix that issue.

4.3.4 Energy drift of the bigger time steps

Curiously, the energy drift of the bigger time steps (above $0.05 a.u.$) does not seem to be as pronounced as that of the smaller ones (below $0.05 a.u.$). In fact, testing only the bigger ones for a proportionally longer time reveals a drift roughly one order of magnitude lower than expected, see Figure 4.4. This suggests that we have a truncation error somewhere in the calculation due to being too close to machine precision.

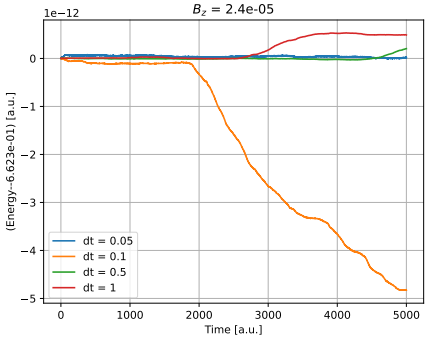


Figure 4.4: Only time steps greater than $0.05 a.u.$.

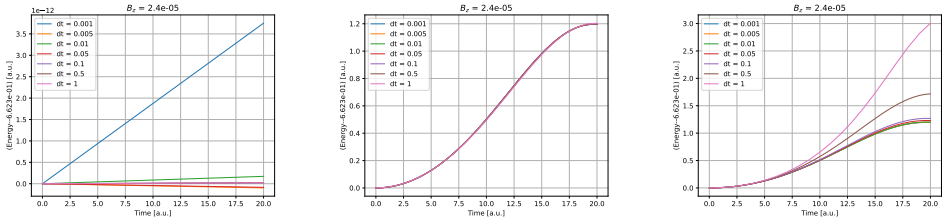


Figure 4.5: Convergence of the core Hamiltonian, Coulomb and exchange matrices, from left to right

4.3.5 Convergence of the Fock operator

Since the Fock operator is made up of three parts, the core Hamiltonian, the exchange integral and the Coulomb integral I can execute the simulation with only one of these parts and observe how they converge in Figure 4.5. The Coulomb and exchange integral converge as expected, but the core Hamiltonian does not converge for diminishing timesteps.

4.3.6 mpmath

Neither of the modifications made to the code by inserting mpmath function where previously the calculations had been done using scipy had any effect on the result. The only thing that did change was the runtime, with the time it took to run a simulation now being orders of magnitude greater than before.

4.3.7 Changing the magnetic field

One would expect a similar behaviour along the x and y axes as along the z axis. However, the behaviour changes radically. At first glance at Figure 4.7, the situation is much worse than along the z-axis, but closer inspection of

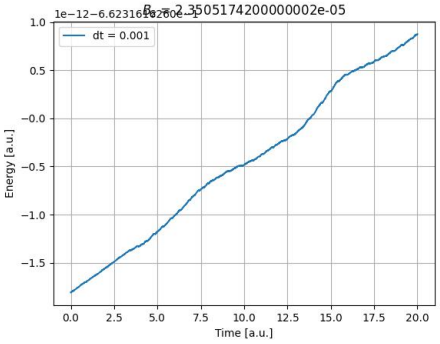


Figure 4.6: Energy drift, as computed with mpmath expm. Only one step length was computed since computing one time step took more than one second.

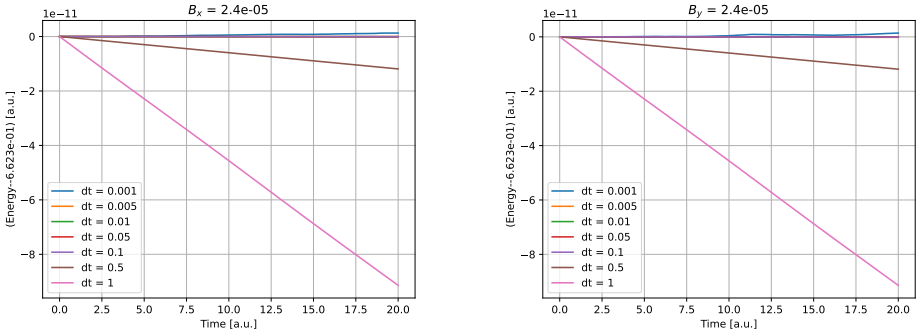


Figure 4.7: Behaviour along x (left) and y (right) axes, for all step lengths.

Figure 4.8 reveals that in this case the drift actually reduces with reducing timestep. At the smaller timesteps the behaviour then looks similar to the z-axis behaviour.

With no magnetic field we get a similar behaviour, where the time integration does not seem to converge, however here the energy drift is one order of magnitude lower, see Figure 4.9.

4.3.8 More precise density matrix

My tests using more precise density matrices showed that the drift all but vanishes even on larger timescales, as shown in figure 4.10a. There the standard deviation of the energy at different time steps is at machine precision, being of order 10^{-16} for time step-length $0.05a.u.$ and $0.1a.u.$, and 10^{-15} for time steps $0.5a.u.$ and $1a.u.$.

Running the same test for all previously probed time steps, however, reveals

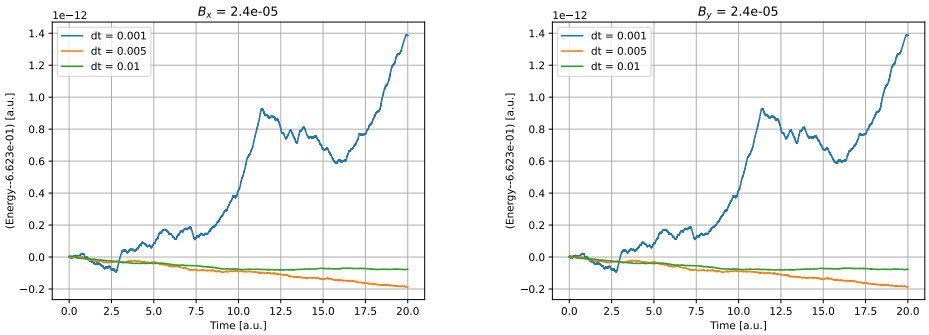


Figure 4.8: Behaviour along x (left) and y (right) axes, for only the short timesteps.

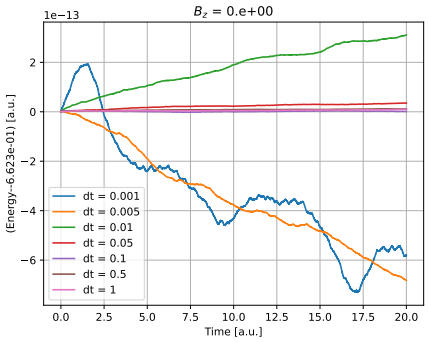
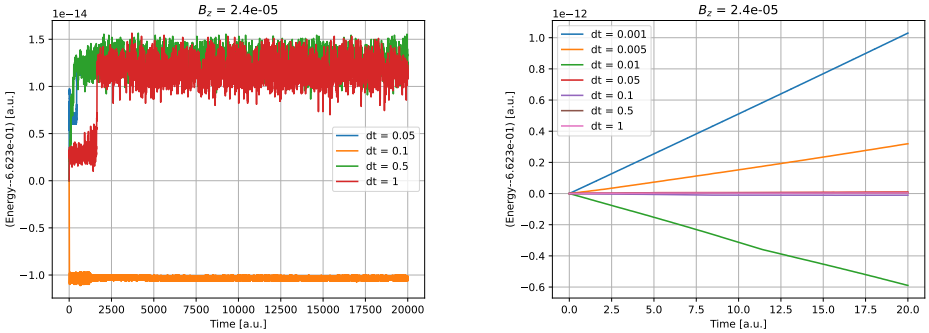


Figure 4.9: No magnetic field



(a) Drift with a density matrix of higher precision

(b) Same as Figure 4.10a, but with additional, smaller time steps.

Figure 4.10

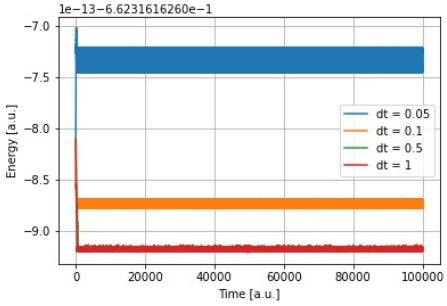
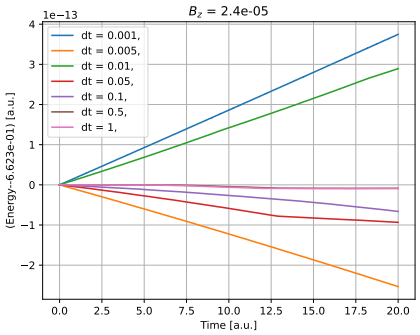
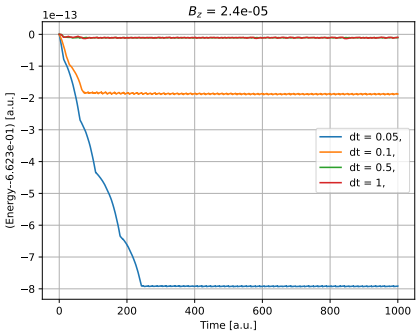


Figure 4.11: Same as Figure 4.10a, but on a longer timescale.



(a) Drift using a basis modified to be orthogonal.



(b) Same as Figure 4.12a, but only larger time steps and over a longer time.

Figure 4.12

in Figure 4.10b that for the smaller ones, this does not solve the issue and the drift still looks rather similar to before the density matrix was saved with additional precision. Additionally we now have a problem when using a magnetic field along an axis other than the z axis, since, as outlined in section 3.3.4, for the simulation to converge there we need smaller time steps. This cannot be reconciled with the need for rather large time steps in order to avoid having the issue with machine precision.

4.3.9 Elimination of base changes

This change further decreased the drift, with the smallest time step now drifting by approximately $3.5 \cdot 10^{-13} a.u./20 a.u.$, see Figure 4.12a. The qualitative behaviour, however, remained unchanged.

This improvement needs to be taken with a grain of salt. As discussed in section 2.3, the coefficients of the basis are optimized for certain tasks. These optimisations do not include the requirement that the basis be orthogonal. To

achieve an orthogonal basis here, I modified the coefficients and perturbed them until the overlap matrix of the basis was the identity. This, however changed the coefficients of the basis, which in turn might have changed the properties of the basis in some unforeseen way. The fact that the ground state energy is the same for both the orthogonal and the unchanged basis does not point towards this being an issue, however there might be some implications onto other parts of the calculation, and simply yielding the same ground state energies is too crude a metric to eliminate doubts into the trustworthiness of this procedure, so further investigation is necessary until this change can be accepted.

Chapter 5

Conclusion

In conclusion, while the project of completely simulating a quantum crystalline beam is not quite complete yet, the tasks carried out for this thesis have brought us closer to that goal.

I have discussed possible choices of a basis for the simulation of the internal degrees of freedom, and chosen one basis that I found to suit our needs most accurately. Although my calculations of the angular momenta of Gaussian orbitals were ultimately of no use to the project, they were correct and so, if nothing else, they were a good exercise.

In the discussion of the stimulated Raman transitions and the spin coherence I have proposed two ways in which a completed simulation could be tested.

More immediately needed for the progress of the project, however, is the continuation of my final task, the energy drift, which I failed to completely resolve. I see two issues that need to be investigated further in order to stop the energy drift once and for all: First of all, I don't understand how changing the direction of the magnetic field changes the convergence of the integration so radically. If one were to find the reason there and achieve similar convergence of both the x and y axes as of the z axis, one could then integrate with time steps above $0.05 a.u.$ and avoid the machine precision errors. Secondly and, in my opinion more urgently, one needs to find the concrete place in the calculation where machine precision errors occur and change the calculations such that they occur no longer. If one could find a transformation that would transform the numbers to be handled by the code into some less extreme orders of magnitude, or extend precision further in some crucial places such that the drift stopped up to some lower time steps than $0.05 a.u.$, then one would also avoid the problems with the other axes.

On the other hand, one could argue that having the error isn't too dramatic. Since all of our errors are on really small scales, and stop growing after a number of time steps, one could just ignore the energy drift and move on.

The total energy drift for all of my tests was close to machine precision, making the drift per time step on the order of machine precision. In the longer simulations the error stopped growing after some time, and stayed constant. However this happened at different times for different time steps, and limited by runtime of the tests I only tested the bigger time steps on these total time scales. Therefore, one thing worthy of trying in the future is testing whether the energy drift stops growing even for the smaller time steps. If that would be the case, then one could truly make the argument that, since the error stays bounded at a very small scale for all step lengths and over a long integration time, it could be ignored.

Appendix A

Source Code

Source Code for this project is available on GitLab: https://gitlab.ethz.ch/heinekas/quantumbeam/-/tree/wepfer_f_bsc_thesis. More specifically, to reproduce the results of chapter 2.4 consider the file *wepfer_f_semester_project/MagneticInteraction.ipynb*, all experiments and plots of chapter 4.3 can be reproduced from the file *wepfer_f_semester_project /EnergyDrift/RunTests.py* by following the instructions in the code.

Bibliography

[1] K. A. Brown and T. Roser. "Towards storage rings as quantum computers". In: *Phys. Rev. Accel. Beams* 23 (5 2020), p. 054701. DOI: 10.1103/PhysRevAccelBeams.23.054701. URL: https://link.aps.org/doi/10.1103/PhysRevAccelBeams.23.054701.

[2] R. T. Husistein. "Simulation of Phonon States in Accelerators". In: (2023). URL: https://gitlab.psi.ch/AMAS-students/Husistein_R/-/blob/737802573465f41cca1aa8b7f5707193266bea2e/master_project/report/thesis.pdf.

[3] M. Solinas. "Extending RT-TDHF to Include Magnetic Field Interactions". In: (2024).

[4] Qiming Sun et al. "PySCF: the Python-based simulations of chemistry framework". In: *WIREs Computational Molecular Science* 8.1 (2018), e1340. DOI: https://doi.org/10.1002/wcms.1340. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1340. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1340.

[5] J. Thijssen. *Computational Physics. 2nd ed.* Cambridge University Press, 2007. DOI: 10.1017/CBO9781139171397.

[6] Xiaosong Li et al. "A time-dependent Hartree-Fock approach for studying the electronic optical response of molecules in intense fields". In: (). DOI: 10.1039/b415849k. URL: www.rsc.org/pccp.

[7] Benjamin P. Pritchard et al. "New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community". In: *Journal of Chemical Information and Modeling* 59.11 (2019). PMID: 31600445, pp. 4814–4820. DOI: 10.1021/acs.jcim.9b00725. eprint: https://doi.org/10.1021/acs.jcim.9b00725. URL: https://doi.org/10.1021/acs.jcim.9b00725.

[8] W. J. Hehre et al. “Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals”. In: *JChPh* 51.6 (1969), pp. 2657–2664. ISSN: 0021-9606. DOI: 10.1063/1.1672392. URL: https://ui.adsabs.harvard.edu/abs/1969JChPh..51.2657H/abstract.

[9] F. Jensen. “Polarization consistent basis sets: Principles”. In: *Journal of Chemical Physics* 115.20 (Nov. 2001), pp. 9113–9125. ISSN: 00219606. DOI: 10.1063/1.1413524. URL: https://www.researchgate.net/publication/234969560_Polarization_consistent_basis_sets_Principles.

[10] Arnim Hellweg and Dmitrij Rappoport. “Development of new auxiliary basis functions of the Karlsruhe segmented contracted basis sets including diffuse basis functions (def2-SVPD, def2-TZVPPD, and def2-QVPPD) for RI-MP2 and RI-CC calculations”. In: *Physical Chemistry Chemical Physics* 17.2 (Dec. 2014), pp. 1010–1017. ISSN: 1463-9084. DOI: 10.1039/C4CP04286G. URL: https://pubs.rsc.org/en/content/articlehtml/2015/cp/c4cp04286g%20https://pubs.rsc.org/en/content/articlelanding/2015/cp/c4cp04286g.

[11] Brian P. Prascher et al. “Gaussian basis sets for use in correlated molecular calculations. VII. Valence, core-valence, and scalar relativistic basis sets for Li, Be, Na, and Mg”. In: *Theoretical Chemistry Accounts* 128.1 (Jan. 2011), pp. 69–82. ISSN: 1432-881X. DOI: 10.1007/S00214-010-0764-0. URL: https://experts.illinois.edu/en/publications/gaussian-basis-sets-for-use-in-correlated-molecular-calculations--2.

[12] Hsiang-Yu Lo. “Creation of Squeezed Schrödinger’s Cat States in a Mixed-Species Ion Trap”. In: (2015). URL: https://api.semanticscholar.org/CorpusID:123980649.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. In consultation with the supervisor, one of the following three options must be selected:

- (●) I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].
- ( ) I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies[2].
- ( ) I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies[3]. In consultation with the supervisor, I did not cite them.

Title of paper or thesis:

Towards a simulation of phonon states in quantum crystalline beams

Authored by:
If the work was compiled in a group, the names of all authors are required.

Last name(s):	**First name(s):**
Wepfer	Florian David

With my signature I confirm the following:
- I have adhered to the rules set out in the Citation Guide.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date	**Signature(s)**
Zürich, 10.6.2024	F. Wepfer

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

[1] E.g. ChatGPT, DALL E 2, Google Bard
[2] E.g. ChatGPT, DALL E 2, Google Bard
[3] E.g. ChatGPT, DALL E 2, Google Bard