University of Cambridge
Department of Physics

Natural Sciences Tripos
Experimental and Theoretical Physics
Part III Project

Quantum Field Theory Simulations

Author:
Carl Andreas Lindstrøm

Supervisor:
Dr Cristopher G Lester

May 2013

Abstract

Quantum Field Theory is a triumph of theoretical physics, but is notoriously hard to visualise. This project developed theoretically and implemented computationally a simplified alternative to the conventional approach by Feynman et al. This allowed realtime simulation of a quantum state in an interactive graphical user interface. To ensure high performance, a number of approximations were applied under the assumption that it did not significantly alter the physics. These included 1-dimensionality, space discretisation and periodicity, and limiting the number of particles considered. Several optimisations were found and implemented, greatly decreasing the number of calculations necessary. Different numerical Schrödinger Equation integrators were investigated and found to be appropriate for different end goals, e.g. high accuracy or long-term energy conservation. The implementation was quantitatively verified to tend to conventional QFT via a comparison with an analytic solution of interacting ϕ^2-theory.

Acknowledgements

Firstly, I would like to thank my project supervisor, Dr Christopher G Lester, for giving me the opportunity to work with him and his group. I greatly appreciated his continuing support and unfailing enthusiasm. It has inspired me to pursue a career in science. Secondly, I wish to thank Charlie Bridge, who worked on a similar Part III project, for his support and for our great discussions. This project was a pleasure to work on from beginning to end.

Contents

Symbols

x_n, p_k	lattice position and momentum	GeV^{-1}, GeV
N	number of lattice points	
L	system length/circumference	GeV^{-1}
$\Delta x, \Delta p$	position and momentum interval	GeV^{-1}, GeV
m	particle mass	GeV
ω_k	energy of momentum mode p_k	GeV
E_i	energy of Fock state indexed i	GeV
$\hat{\phi}_n$	scalar field operator	
$\hat{\pi}_n$	scalar conjugate momentum operator	GeV
$\hat{\psi}_n$	fermion field spinor operator	$\text{GeV}^{\frac{1}{2}}$
$\hat{L}$	Lagrangian operator	GeV
$\hat{\mathcal{L}}$	Lagrangian density operator	GeV^2
$\hat{H}$	Hamiltonian operator	GeV
$\hat{\mathcal{H}}$	Hamiltonian density operator	GeV^2
$\hat{a}_k$, $\hat{a}_k^\dagger$	scalar ladder operators	
$\hat{b}_k$, $\hat{b}_k^\dagger$, $\hat{c}_k$, $\hat{c}_k^\dagger$	fermion and anti-fermion ladder operators	
$\lambda^{(n)}$	interaction couplings	
ξ^μ	1D Dirac γ^μ-matrix equivalent ($\mu = 0, 1$)	
$u(p_k), v(p_k)$	1D Dirac plane wave eigenspinors	$\text{GeV}^{\frac{1}{2}}$
F_k	Sandwich factor for momentum number k	
i	Fock state index	
$c_i(t)$	coefficient of Fock state i	
P	particle number	
$\sigma(N, P)$	number of states with P particles	
$S(N, P)$	number of states with $\leq$ P particles	
	i, σ, S subscripted $_S$ / $_F$ for scalars / fermions.	

Chapter 1

Introduction

1.1 Motivation

In our quest to understand the Universe, we have found a magnificent mathematical frame work we call Quantum Field Theory. It successfully explains most experiments ever conducted, but is notoriously hard to work with. This is mainly caused by our inability to solve it analytically in the presence of interactions. This problem has been solved perturbatively, most famously by Feynman et al. via intuitive diagrams and rules.

Feynman's method does, however, have its shortcomings. It is mainly a tool for calculating cross sections and decay rates, which is key to experimental verification. It presents a picture of interactions via mysterious virtual particles, masking over the complicated nature of the quantum fields.

1.2 Computer Simulation vs. Feynman Diagrams

The most straightforward alternative to Feynman rules and -diagrams is a numerical computer simulation. This allows us to see the time evolution of the field directly, providing greater intuition for the actual mechanisms of QFT. We must, however, note that calculating all these normally hidden variables is a spectacularly inefficient way of calculating cross sections and decay rates. Computational QFT is therefore most appropriate as a visualisation tool for doing virtual experiments.

Simulating quantum fields in this way is effectively doing Nature's job in a computer. Finite computing power severely limits the simulation, and in order to simulate the system in real time we need to apply a number of approximations as well as clever optimisations.

1.2.1 Approximations

The following are the main approximations underlying the method used, assumed not to significantly alter the physics.

- **Discretisation** – QFT works in continuous spacetime, but computers require discreteness. Increasing the number of lattice points severely limits performance, so this number is often small (less than 100 points).
- **1 spatial dimension** – QFT is (3+1)-dimensional, but most interesting phenomena require only (1+1) dimensions. This makes calculation vastly more efficient, although it has interesting consequences e.g. for the treatment of fermions, which are manifestly (3+1)-dimensional.
- **Finite periodic universe** – Although QFT operates in an infinite spacetime, we can approximately emulate this as a finite system with periodic boundary conditions.

We have effectively reduced the Universe to a 1-dimensional ring of points.

1.3 Goal

The goal is to develop a consistent quantum field theory satisfying the above points and implement this computationally. The implementation is aimed for an interactive graphical user interface, as an intuitive visualisation of Quantum Field Theory.

Chapter 2

Theory

We define our system mathematically as satisfying the following:

- 1 spatial dimension, x_n, and 1 time dimension, t.
- Discretised spatial dimension with N lattice points: $x_0, x_1, \ldots, x_{N-1}$ (zero-indexed).
- Periodic spatial boundary conditions: $x_n = x_{n+N}$ and $\phi_n = \phi_{n+N}$.

This can be visualised as a quantum ring shown in Figure 2.1.

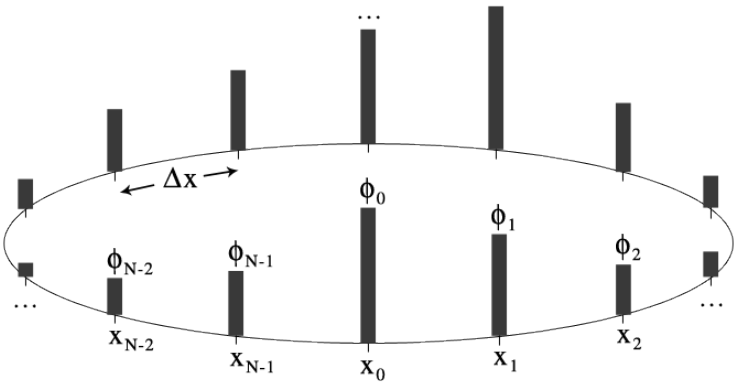

FIGURE 2.1: **Quantum ring**: a 1D discretised quantum system with periodic boundary conditions (x_n denotes position, ϕ_n denotes field value, Δx denotes lattice spacing). *Source: author.*

Sections 2.1-2.4 contain all the necessary theory, assuming a scalar field. In Section 2.5 all relevant formulae are then rederived for fermions. Constructing a consistent 1D discretised quantum field theory constituted a significant portion of the research.

We start with free non-interacting theory, and then introduce interactions[1] [2].

[1] The theory section was inspired by work by project supervisor Dr Lester. Sections where independent research was conducted will be clearly indicated.

[2] Natural units are assumed. For symbols clarification, please refer to the Symbols section.

2.1 Discretisation in 1 Spatial Dimension

We aim to discretise all the relevant formulae necessary for implementation. This is a reasonably straightforward task involving swapping integrals with sums and Dirac-deltas with Kronecker-deltas. Note that the Dirac-delta is not directly equivalent to the Kronecker-delta:

$$\delta^{(1)}(p-q) \quad \rightarrow \quad \frac{\delta_{pq}}{\Delta p} \tag{2.1}$$

from the definition of the Dirac-delta function:

$$\int_{-\infty}^{\infty} \delta^{(1)}(p-q)dp = 1 \quad \rightarrow \quad \sum_{n=0}^{N-1} \frac{\delta_{pq}}{\Delta p}\Delta p = 1 \tag{2.2}$$

2.1.1 Momentum Discretisation

Discretising position space also discretises momentum space. This is because a particle is not allowed to end up anywhere along the ring, only certain points, implying that the particle can only move with certain momenta. It also suggests that if two momenta (e.g. one low and one very high) both take the particle around to the same lattice point, they are equivalent:

$$p_k \equiv p_{k+N} \tag{2.3}$$

From this we can find an expression for the momentum interval:

$$\begin{aligned} e^{ip_k\Delta x} &= e^{i(p_k+N\Delta p)\Delta x} \\ N\Delta p\Delta x &= 2\pi \\ \Delta p = \frac{2\pi}{N\Delta x} &= \frac{2\pi}{L} \end{aligned} \tag{2.4}$$

We write the discritised positions and momenta as:

$$x_n = n\Delta x \tag{2.5}$$

$$p_k = k\Delta p \tag{2.6}$$

2.1.2 Lagrangian and Hamiltonian Densities

The Lagrangian density involves a kinetic term with both time and space derivatives. Time is not discretised, but space is and we must replace the derivative by a difference:

$$\nabla\phi \quad \rightarrow \quad \frac{\phi_{n+\frac{1}{2}} - \phi_{n-\frac{1}{2}}}{\Delta x} \equiv \Delta_x\phi_n \tag{2.7}$$

where Δ_x is the discretised derivative operator. Although non-existent, the mid-point field values $\phi_{n+\frac{1}{2}}$ and $\phi_{n-\frac{1}{2}}$ are chosen to ensure symmetry and ease of calculation. The Lagrangian density[3] discretises to:

$$\mathcal{L} = \frac{1}{2}\pi^2 - \frac{1}{2}(\nabla\phi)^2 - \frac{1}{2}m^2\phi^2 \quad \rightarrow \quad \mathcal{L}_n = \frac{1}{2}\pi_n^2 - \frac{1}{2}\left(\Delta_x\phi_n\right)^2 - \frac{1}{2}m^2\phi_n^2 \tag{2.8}$$

where $\pi = \dot{\phi}$ is the conjugate momentum. Similarly the Hamiltonian discretises to:

$$\mathcal{H} = \frac{1}{2}\pi^2 + \frac{1}{2}(\nabla\phi)^2 + \frac{1}{2}m^2\phi^2 \quad \rightarrow \quad \mathcal{H}_n = \frac{1}{2}\pi_n^2 + \frac{1}{2}\left(\Delta_x\phi_n\right)^2 + \frac{1}{2}m^2\phi_n^2 \tag{2.9}$$

The Lagrangian and Hamiltonian become:

$$L = \int \mathcal{L}\, dx \quad \rightarrow \quad L = \Delta x \sum_{n=0}^{N-1} \mathcal{L}_n \tag{2.10}$$

$$H = \int \mathcal{H}\, dx \quad \rightarrow \quad H = \Delta x \sum_{n=0}^{N-1} \mathcal{H}_n \tag{2.11}$$

2.1.3 Energy

Minimising the action gives us the Euler-Lagrange equations, which when applied to the Lagrangian density become the Klein-Gordon equation[4]. This discretises to:

$$(\partial_\mu\partial^\mu + m^2)\phi = 0 \quad \rightarrow \quad \ddot{\phi}_n + m^2\phi_n - \Delta_x^2\phi_n = 0 \tag{2.12}$$

where

$$\Delta_x^2\phi_n = \frac{\phi_{n+1} - 2\phi_n + \phi_{n-1}}{\Delta x^2} \tag{2.13}$$

Using a trial plane wave solution $\phi_n = \hat{A}e^{i(p_m x_n - E_{pm}t)}$ we can make progress towards finding the energy:

$$\begin{aligned} \ddot{\phi}_n &= -\omega_k^2\phi_n \\ \phi_{n+1} &= e^{ip_k\Delta x}\phi_n \\ \phi_{n-1} &= e^{-ip_k\Delta x}\phi_n \end{aligned}$$

[3] p. 16 in Peskin & Schroeder [1]
[4] p. 17 in Peskin & Schroeder [1]

Substituting this into Eqn (2.12) gives us the discretised energy relation:

$$\begin{aligned} -\omega_k^2\phi_n + m^2\phi_n - \frac{e^{ip_k\Delta x} + e^{-ip_k\Delta x} - 2}{\Delta x^2}\phi_n &= 0 \\ \phi_n \neq 0 \quad \Rightarrow \quad -\omega_k^2 + m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k\Delta x}{2}\right) &= 0 \end{aligned}$$

$$\omega_k = \sqrt{m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k\Delta x}{2}\right)} \tag{2.14}$$

This is invariant under a coordinate transformation $p_k \to p_{k+N}$ as required by periodicity[5] and tends to the familiar continuous limit[6] as $\Delta x \to 0$. As we will see later, Eqn (2.14) also holds for fermions, despite swapping the Klein-Gordon with the Dirac equation.

2.1.4 Quantisation and Commutation Relations

Our treatment so far has been entirely classical. To quantise, we start by discretising the field operator commutation relation[7]:

$$\left[\hat{\phi}(x), \hat{\pi}(y)\right] = i\delta^{(1)}(x-y) \quad \to \quad \left[\hat{\phi}_n, \hat{\pi}_m\right] = i\frac{\delta_{nm}}{\Delta x} \tag{2.15}$$

To make progress it is helpful to Fourier expand the field operator and its conjugate[8]:

$$\hat{\phi}_n = \sum_{k=0}^{N-1} \frac{\Delta p}{2\pi 2\omega_k}\left(\hat{a}_k e^{ip_k x_n} + \hat{a}_k^\dagger e^{-ip_k x_n}\right) \tag{2.16}$$

$$\hat{\pi}_n = -i\sum_{k=0}^{N-1} \frac{\Delta p}{2\pi 2\omega_k}\omega_k\left(\hat{a}_k e^{ip_k x_n} - \hat{a}_k^\dagger e^{-ip_k x_n}\right) \tag{2.17}$$

Note the choice of relativistic normalisation, whereby the sum gets a divisor of $2\omega_k 2\pi$. Any normalisation could have been chosen if consistently applied. Our choice is motivated by Lorentz invariance as well as it producing meaningful dimensionalities[9].

We have now introduced ladder operators, representing creation and annihilation of particles. From Eqns (2.15), (2.16) and (2.17) we obtain the very useful ladder operator

[5] See Appendix A.1 for verification of periodicity.
[6] See Appendix A.2 for verification of the continuous limit energy.
[7] Operators indicated with hats.
[8] p. 20 in Peskin & Schroeder [1]
[9] Discussed in Appendix A.11.

commutation relations[10]:

$$\left[\hat{a}_k, \hat{a}_l^\dagger\right] = 2\omega_k 2\pi \frac{\delta_{kl}}{\Delta p} = 2\omega_k L \delta_{kl} \tag{2.18}$$

and all other commutators are zero.

2.1.5 Fock States and Corresponding Energies

In order to find the eigenstates of the non-interacting Hamiltonian (Fock states) we substitute the Fourier expansions, Eqns (2.16) and (2.17), into the Hamiltonian, Eqn (2.11)[11]:

$$\hat{H}_0 = \sum_{k=0}^{N-1} \frac{\Delta p}{2\pi 2\omega_k} \omega_k \hat{a}_k^\dagger \hat{a}_k + E_{vac} \tag{2.19}$$

$$E_{vac} = \sum_{k=0}^{N-1} \frac{\omega_k}{2} \tag{2.20}$$

The Hamiltonian can safely be shifted to remove the vacuum energy E_{vac}, but it then measures the energy difference to the free vacuum (no particles) and not the theory's intrinsic zero. Interactions lead to eigenstates with less energy than the free vacuum, giving negative energy differences, which is not a problem when keeping in mind the shift.

From Eqn (2.19) we can write down the Fock states:

$$|i\rangle = \prod_{k=0}^{N-1} \frac{\left(\hat{a}_k^\dagger\right)^{l_k}}{\sqrt{(2\omega_k L)^{l_k} l_k!}} |0\rangle \tag{2.21}$$

$$= \frac{\left(\hat{a}_0^\dagger\right)^{l_0} \left(\hat{a}_1^\dagger\right)^{l_1} \dots \left(\hat{a}_{N-1}^\dagger\right)^{l_{N-1}}}{\sqrt{(2\omega_0 L)^{l_0} (2\omega_1 L)^{l_1} \dots (2\omega_{N-1} L)^{l_{N-1}}} \sqrt{l_0! l_1! \dots l_{N-1}!}} |0\rangle \tag{2.22}$$

where the normalisation is chosen to satisfy $\langle m|n\rangle = \delta_{mn}$.

The Fock state energy is a sum of all the particle energies and the vacuum[12]:

$$E_i \equiv \langle i|\hat{H}_0|i\rangle = \sum_{k=0}^{N-1} l_k \omega_k + E_{vac} \tag{2.23}$$

[10] Verified in Appendix A.3
[11] Full derivation in Appendix A.4
[12] Full derivation in Appendix A.5

2.2 Schrödinger Equation Integrators

A general time dependent state $|\chi(t)\rangle$ is a weighted sum of all free eigenstates $|n\rangle$:

$$|\chi(t)\rangle = \sum_{n=0}^{S-1} c_n(t)|n\rangle \tag{2.24}$$

where S Fock states are considered. In the Schrödinger picture, the time evolution is given by the Schrödinger equation:

$$\begin{aligned} i\frac{d}{dt}|\chi(t)\rangle &= \hat{H}|\chi(t)\rangle & (2.25)\\ i\langle n|\sum_{m=0}^{S}\dot{c}_m(t)|m\rangle &= \langle n|\hat{H}\sum_{m=0}^{S-1}c_m(t)|m\rangle & (2.26)\\ i\dot{c}_n(t) &= \sum_{m=0}^{S-1}c_m(t)\langle n|\hat{H}|m\rangle & (2.27)\\ i\dot{c}_n(t) &= E_n c_n(t) + \sum_{m=0}^{S-1}c_m(t)\langle n|\hat{H}_{int}|m\rangle & (2.28) \end{aligned}$$

where we have split up the Hamiltonian into a free and an interacting part: $\hat{H} = \hat{H}_0 + \hat{H}_{int}$. We can now define the interaction Hamiltonian matrix:

$$H_{nm} = \langle n|\hat{H}_{int}|m\rangle \tag{2.29}$$

of $S \times S$ elements, which will be discussed in great detail in the Section 2.3. The equation which needs to be solved numerically is then:

$$\dot{c}_n(t) = -iE_n c_n(t) - i\sum_{m=0}^{S-1} H_{nm}c_m(t) \tag{2.30}$$

This first order equation can be integrated over time given initial coefficients $c_n(0)$.

2.2.1 Second Order Symplectic Integrator

We can now discretise time in order to numerically integrate Eqn (2.30). We start by Taylor expanding $c_n(t+\Delta t)$ and $c_n(t-\Delta t)$ about t:

$$\begin{aligned} c_n(t+\Delta t) &= c_n(t) + \Delta t\dot{c}_n(t) + \frac{\Delta t^2}{2}\ddot{c}_n(t) + O(\Delta t^3) & (2.31)\\ c_n(t-\Delta t) &= c_n(t) - \Delta t\dot{c}_n(t) + \frac{\Delta t^2}{2}\ddot{c}_n(t) + O(\Delta t^3) & (2.32) \end{aligned}$$

Subtracting these and substituting in Eqn (2.30), we obtain a second order integrator:

$$c_n(t+\Delta t) = c_n(t-\Delta t) + 2\Delta t \dot{c}_n(t) + O(\Delta t^3) \tag{2.33}$$

$$c_n(t+\Delta t) = c_n(t-\Delta t) - 2i\Delta t \left(E_n c_n(t) + \sum_{m=0}^{S-1} H_{nm} c_m(t) \right) + O(\Delta t^3) \tag{2.34}$$

Note that this is invariant under the transformation $\Delta t \to -\Delta t$, and therefore has no preferred time direction. Stepping forwards and then backwards in this case loses no information[13] implying that the integrator is symplectic.

2.2.2 Arbitrary Even-Order Integrator

We can generalise this approach to find an integrator of arbitrarily high order[14]. To do this we observe that since we are working in the Schrödinger picture (H_{nm} is time independent), we can differentiate Eqn (2.30) k times to get:

$$\frac{d^{k+1}}{dt^{k+1}} c_n(t) = -iE_n \frac{d^k}{dt^k} c_n(t) - i \sum_{m=0}^{S-1} H_{nm} \frac{d^k}{dt^k} c_m(t) \tag{2.35}$$

This implies we can find any time derivative, provided we do so iteratively: $c_n(t) \to \dot{c}_n(t) \to \ddot{c}_n(t) \to \dddot{c}_n(t) \to \dots$. Taylor expanding $c_n(t+\Delta t)$ and $c_n(t-\Delta t)$ about t to order K:

$$c_n(t+\Delta t) = \sum_{k=0}^{K} \frac{\Delta t^k}{k!} \frac{d^k}{dt^k} c_n(t) + O(\Delta t^{K+1}) \tag{2.36}$$

$$c_n(t-\Delta t) = \sum_{k=0}^{K} \frac{(-\Delta t)^k}{k!} \frac{d^k}{dt^k} c_n(t) + O(\Delta t^{K+1}) \tag{2.37}$$

then subtracting Eqns (2.37) and (2.37) gives us an arbitrary even order integrator:

$$c_n\left(t+\Delta t\right) = c_n\left(t-\Delta t\right) + 2 \sum_{odd\ k=1}^{K-1} \frac{\Delta t^k}{k!} \frac{d^k}{dt^k} c_n(t) + O\left(\Delta t^{K+1}\right) \tag{2.38}$$

where coefficient derivatives are found iteratively by Eqn (2.35). This provides numerical integration of order K at a performance cost of only $\mathrm{O}(K)$.

Note that the first step backwards to $c_n(-\Delta t)$ must be calculated by another integrator, e.g. Eqn (2.37).

[13] Although rounding errors will be introduced.

[14] Arbitrary-order integrators was an area of independent research.

2.3 Interaction Hamiltonians

For any reasonable system (values of N and P_{max})[15] a naïve interaction Hamiltonian computation will be extremely slow. Consider for instance a 3-vertex interaction ϕ_n^3: each Fourier expansion of ϕ_n, as well summing over lattice points, contribute a sum up to N calculated for all $S \times S$ combinations of two Fock states: in total $N^4 S^2$ calculations. Using e.g. $P_{max} = 4, N = 100$ gives of order 10^{21} calculations. This would take current non-supercomputers millions of years, hence the need for heavy optimisation.

Many optimisations are possible, and we can save several orders of N and S. The challenge is to recast our equations into mathematically equivalent, but computationally friendlier forms. Developing these optimisations have been a major focus of this research.

2.3.1 Applying Momentum Conservation

Momentum conservation in QFT shows up as δ-functions, removing one degree of momentum freedom. This saves us two momentum/space sums $\propto O(N^2)$. Consider first the simplest interaction, an additional mass term:

$$\hat{H}_{int}^{(2)} = \Delta x \sum_{n=0}^{N-1} \lambda^{(2)} \hat{\phi}_n^2 \tag{2.39}$$

Substituting in Fourier expansions and rearranging, we get:

$$\begin{aligned}
\hat{H}_{int}^{(2)} &= \lambda^{(2)} \Delta x \sum_{n,k,l=0}^{N-1} \frac{\hat{a}_k e^{ip_k x_n} + \hat{a}_k^\dagger e^{-ip_k x_n}}{2\omega_k L} \frac{\hat{a}_l e^{ip_l x_n} + \hat{a}_l^\dagger e^{-ip_l x_n}}{2\omega_l L} \\
&= \frac{\lambda^{(2)}}{4LN} \sum_{n,k,l=0}^{N-1} \frac{1}{\omega_k \omega_l} (\hat{a}_k \hat{a}_l e^{i(p_k+p_l)x_n} + \hat{a}_k^\dagger \hat{a}_l e^{-i(p_k-p_l)x_n} \\
&\qquad + \hat{a}_k \hat{a}_l^\dagger e^{i(p_k-p_l)x_n} + \hat{a}_k^\dagger \hat{a}_l^\dagger e^{-i(p_k+p_l)x_n})
\end{aligned}$$

where we have used $\Delta p / 2\pi = 1/L$ and $\Delta x / L = 1/N$. Using the identity:

$$\sum_{n=0}^{N-1} e^{i(p_k-p_l)x_n} = N\delta_{kl} \tag{2.40}$$

[15] P_{max} is the maximum number of particles considered. This is further discussed in Section 3.1.

we can now significantly simplify the expression:

$$\hat{H}_{int}^{(2)} = \frac{\lambda^{(2)}}{4LN}\sum_{k,l=0}^{N-1}\frac{N}{\omega_k\omega_l}\left(\hat{a}_k\hat{a}_l\delta_{-k,l} + \hat{a}_k^\dagger\hat{a}_l\delta_{kl} + \hat{a}_k\hat{a}_l^\dagger\delta_{kl} + \hat{a}_k^\dagger\hat{a}_l^\dagger\delta_{-k,l}\right) \quad (2.41)$$

$$= \frac{\lambda^{(2)}}{4}\sum_{k=0}^{N-1}\frac{1}{\omega_k^2}\left(\hat{a}_k\hat{a}_{-k} + \hat{a}_k^\dagger\hat{a}_k + \hat{a}_k\hat{a}_k^\dagger + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right) \quad (2.42)$$

$$= \lambda^{(2)}\sum_{k=0}^{N-1}\left(\frac{1}{2\omega_k} + \frac{\hat{a}_k\hat{a}_{-k} + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger + 2\hat{a}_k^\dagger\hat{a}_k}{L(2\omega_k)^2}\right) \quad (2.43)$$

where the last line is normal ordered (important for another optimisation, Eqn (2.45)). Note that the interaction coupling $\lambda^{(2)}$ is strictly a factor and can be ignored in calculations. This allows realtime weighting of different pre-calculated interaction Hamiltonians.

This method generalises straightforwardly to higher ϕ-order interactions, removing one position and one momentum sum. A minor optimisation is also possible for two or more momentum sums, exploiting symmetry of momentum number interchange. The ϕ^3-term in symmetrised form roughly halves the number of calculations:

$$\frac{\hat{H}_{int}^{(3)}}{\lambda^{(3)}} = 2\sum_{k=2}^{N-1}\sum_{l=1}^{k-1}\hat{h}_{kl} + \sum_{k=1}^{N-1}(2\hat{h}_{k0} + \hat{h}_{kk}) + \hat{h}_{00} \quad (2.44)$$

where the $\hat{h}_{kl}$ is a symmetric normal ordered operator matrix[16].

2.3.2 Avoiding Calculation of Zeros: The Sandwich Factor, F_k

Optimisations so far have been limited to the Hamiltonian itself, by decreasing the number of ladder operators in it[17]. However, we still need to "sandwich" it between $S \times S$ combinations of two Fock states. Most of these combinations will not contribute, reflecting that no amplitude is transferred between e.g. Fock states of different momentum number (momentum conservation) or between odd/even particle number states in ϕ^2-interactions. To be maximally efficient, we cannot test if each combination contributes, but skip directly to calculation of nontrivial elements.

By considering the interaction Hamiltonian $\hat{H}_{int}$ to be a weighted sum of normal ordered ladder operator groups (e.g. $\hat{a}_k^\dagger\hat{a}_l\hat{a}_0$) we can divide and conquer the problem one group at a time. We observe that for a given ket $|m\rangle$ and a ladder operator group, at most one bra $\langle n|$ will give a nonzero contribution. In addition we exploit the ladder operator

[16] Definition of $\hat{h}_{kl}$ in Appendix A.6

[17] This optimisation was found through independent research.

commutator to decouple the momenta into factors. We write the factor F_k from each momentum number k as:

$$F_k(l_k, n_k, m_k) \equiv \left(\langle 0| \frac{(\hat{a}_k)^{\rho_k}}{(2\omega_k L)^{\rho_k}\sqrt{\rho_k!}} \right) \left(\hat{a}_k^\dagger\right)^{n_k} (\hat{a}_k)^{m_k} \left(\frac{\left(\hat{a}_k^\dagger\right)^{l_k}}{(2\omega_k L)^{l_k}\sqrt{l_k!}} |0\rangle \right) \tag{2.45}$$

where non-triviality requires the ladder operator exponents to satisfy:

$$\rho_k = l_k + n_k - m_k \tag{2.46}$$

remembering periodicity: $k \equiv k \pmod{N}$.

The total product for all the momentum factors is then:

$$F_{total}(\{l_k\}, \{n_k\}, \{m_k\}) = \prod_{k=0}^{N-1} F_k(l_k, n_k, m_k) \tag{2.47}$$

Eqn (2.45) can be calculated by programmatically applying ladder operator commutators, but it is far cheaper to reduce it to a closed form mathematical expression. Using the commutator as well as the identity[18]:

$$\langle 0|(\hat{a}_k)^n(\hat{a}_k^\dagger)^n|0\rangle = n!(2\omega_k L)^n \tag{2.48}$$

the following expression was found by inspection:

$$F_k(l_k, n_k, m_k) = \sqrt{\left(\prod_{a=1}^{n_k} (2\omega_k L)(l_k - m_k + a) \right) \left(\prod_{b=1}^{m_k} (2\omega_k L)(l_k + 1 - b) \right)} \tag{2.49}$$

This was a key research finding as it significantly decreases the number of calculations. It has been verified for all $l_k \in \mathbb{N}$ when:

$$\begin{gathered} n_k = 0 \quad , \quad m_k \in \mathbb{N} \\ m_k = 0 \quad , \quad n_k \in \mathbb{N} \\ m_k + n_k \leq 4 \end{gathered}$$

[18] Full derivation in Appendix A.7

2.3.3 Corollary: Calculating Field and Conjugate Momenta Values

The average field value and its conjugate momentum are of a similar forms to interaction Hamiltonians[19]:

$$\phi_n(t) = \langle\psi(t)|\hat{\phi}_n|\psi(t)\rangle = \langle\psi(t)|\sum_{k=0}^{N-1}\frac{\Delta p}{2\pi 2\omega_k}\left(\hat{a}_k e^{ip_k x_n} + \hat{a}_k^\dagger e^{-ip_k x_n}\right)|\psi(t)\rangle \quad (2.50)$$

$$\pi_n(t) = \langle\psi(t)|\hat{\pi}_n|\psi(t)\rangle = \langle\psi(t)|\sum_{k=0}^{N-1}\frac{-i\Delta p}{4\pi}\left(\hat{a}_k e^{ip_k x_n} - \hat{a}_k^\dagger e^{-ip_k x_n}\right)|\psi(t)\rangle \quad (2.51)$$

Expanding the state $|\psi(t)\rangle$ in terms of Fock states:

$$\phi_n(t) = \sum_{i,j=0}^{S-1}\sum_{k=0}^{N-1}\frac{1}{2\omega_k L}c_i^*(t)c_j(t)\langle i|\left(\hat{a}_k e^{ip_k x_n} + \hat{a}_k^\dagger e^{-ip_k x_n}\right)|j\rangle \quad (2.52)$$

$$= \sum_{j=0}^{S-1}\sum_{k=0}^{N-1}\frac{c_j(t)}{2\omega_k L}\left(c_{i_-}^*(t)\langle i_-|\hat{a}_k|j\rangle e^{ip_k x_n} + c_{i_+}^*(t)\langle i_+|\hat{a}_k^\dagger|j\rangle e^{-ip_k x_n}\right) \quad (2.53)$$

where i_- and i_+ are determined by j and k (term dropped if non-existant). Using Eqn (2.45) and noting that $F_k(l_k, 0, 1) = \sqrt{(2\omega_k L)l_k}$ and $F_k(l_k, 1, 0) = \sqrt{(2\omega_k L)(l_k+1)}$ we get:

$$\phi_n(t) = \sum_{j=0}^{S-1}\sum_{k=0}^{N-1}\frac{c_j(t)}{2\omega_k L}\left(c_{i_-}^*(t)F_k\left(l_k(j), 0, 1\right)e^{ip_k x_n} + c_{i_+}^*(t)F_k\left(l_k(j), 1, 0\right)e^{-ip_k x_n}\right)$$

$$= \sum_{j=0}^{S-1}\sum_{k=0}^{N-1}\frac{c_j(t)}{\sqrt{2\omega_k L}}\left(c_{i_-}^*(t)e^{ip_k x_n}\sqrt{l_k(j)} + c_{i_+}^*(t)e^{-ip_k x_n}\sqrt{l_k(j)+1}\right) \quad (2.54)$$

where $l_k(j)$ is the exponent of $\hat{a}_k^\dagger$ in $|j\rangle$. Similarly:

$$\pi_n(t) = \sum_{j=0}^{S-1}\sum_{k=0}^{N-1} i\sqrt{\frac{\omega_k}{2L}}c_j(t)\left(c_{i_+}^*(t)e^{-ip_k x_n}\sqrt{l_k(j)+1} - c_{i_-}^*(t)e^{ip_k x_n}\sqrt{l_k(j)}\right) \quad (2.55)$$

2.4 Finding Interacting Theory Eigenstates

Finding eigenstates of the interacting theory is an essential tool to measuring the correctness of the implementation. Eigenenergies can be measured and compared in ϕ^2-theory, which has an analytical solution.

[19] Calculating field and conjugate momentum values was part of the independent research.

2.4.1 The Shifted Power Iteration Method

We can find the ground state through the shifted power iteration method[20]. This means applying the operator $(1-\hat{H}/E_{max})$ to the state[21], then renormalising, iteratively. This suppresses higher energy eigenstates, hence converging to the ground state. It requires that the initial state has a nonzero ground state component.

$$|\psi\rangle_{i+1} = \frac{(1-\hat{H}/E_{max})|\psi\rangle_i}{||(1-\hat{H}/E_{max})|\psi\rangle_i||} \tag{2.56}$$

$$\lim_{i\to\infty}|\psi\rangle_i = |\tilde{0}\rangle \tag{2.57}$$

where $\tilde{}$ denotes eigenstates. The other eigenstates are found by orthogonality, by subtracting the overlap with lower energy states at each iteration:

$$|\chi_1\rangle = (1-\hat{H}/E_{max})|\psi\rangle_i \tag{2.58}$$

$$|\chi_2\rangle = |\chi_1\rangle - \sum_{e=0}^{n-1}\langle\tilde{e}|\chi_1\rangle|\tilde{e}\rangle \tag{2.59}$$

$$|\psi\rangle_{i+1} = \frac{|\chi_2\rangle}{||\chi_2||} \tag{2.60}$$

$$\lim_{i\to\infty}|\psi\rangle_i = |\tilde{n}\rangle \tag{2.61}$$

where eigenstates are found in order of increasing energy.

The shifted power iteration method converges linearly[22] and is easily implemented[23].

2.4.2 Analytical Solution in ϕ^2-Theory

Using both a mass m and a ϕ^2-interaction is equivalent to an effective mass shift[24]:

$$\mathcal{H}_n \supset \frac{1}{2}m^2\hat{\phi}_n^2 + \lambda^{(2)}\hat{\phi}_n^2 = \frac{1}{2}\mu^2\hat{\phi}_n^2 \tag{2.62}$$

$$\mu = \sqrt{m^2+2\lambda^{(2)}} \tag{2.63}$$

This shift implies expressing eigenstates of mass μ in terms of Fock states of mass m. The ground state of each theory corresponds to no particles of mass μ and the next

[20] p. 204 in Numerical Linear Algebra [2]

[21] E_{max} is the largest eigenvalue of $\hat{H}$.

[22] p. 206 in Numerical Linear Algebra [2]

[23] Alternatives include Rayleigh-quotient iteration, which converges cubically, but requires calculation of the inverse of an $S\times S$ matrix. This is very expensive: $O(S^3)$ per step, and would only outperform power iteration after a large number of steps.

[24] The analytical comparison was developed in collaboration with Charlie Bridge.

eigenstate corresponds to one stationary particle of mass μ. Comparing the vacuum energy-invariant difference between the ground state and the next eigenstate in the two cases allows us to verify the consistency of the implementation:

$$\tilde{E}_1^{\mu} - \tilde{E}_0^{\mu} = \mu = \tilde{E}_1^{m+\phi^2} - \tilde{E}_0^{m+\phi^2} \tag{2.64}$$

where $\tilde{E}_n^{theory}$ denotes the nth eigenenergy on the theory.

Note that this works only for scalars and require consideration of infinitely many particles to be accurate.

2.5 Fermions

In 3D the 4-spinor represents particles and antiparticles of two separate spins[25]. However, in 1D we get a 2-spinor simply representing particles and antiparticles where the spin degree of freedom is lost. This reduces the four γ^{μ}-matrices to two 2×2 matrices we denote ξ^{μ} for clarity, where the spacetime index $\mu = 0, 1$ is consistent with (1+1)-dimensionality.

2.5.1 The Dirac Equation and Dirac γ^{μ}-Matrices

The Dirac equation[26] discretises to:

$$(i\partial_{\mu}\gamma^{\mu} - m)\psi = 0 \quad \rightarrow \quad i\xi^0\dot{\psi}_n - i\xi^1\Delta_x\psi_n - m\psi_n = 0 \tag{2.65}$$

where ξ^{μ} are 2×2-matrices and ψ_n are 2-spinors. One valid choice of ξ^{μ} is picking out the t- and x-directions of γ^{μ}. This decouples the first and fourth spinor index fields from the second and third spinor index fields: $(\psi_1, \psi_2, \psi_3, \psi_4) \rightarrow (\psi_1, \psi_4), (\psi_2, \psi_3)$. This has the advantage of containing no complex numbers, we therefore expect a purely real theory (in this representation):

$$\xi^0 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_3 \qquad \xi^1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \sigma_1\sigma_3 \tag{2.66}$$

Interestingly, choosing the t- and z-directions produce the same results. However, choosing t- and y-components decouples the first and third spinor index fields: $(\psi_1, \psi_2, \psi_3, \psi_4) \rightarrow$

[25] The section on fermions was a result of independent research.
[26] p. 42 in Peskin & Schroeder [1]

$(\psi_1, \psi_3), (\psi_2, \psi_4)$ and gives a different representation where $\xi^1 = \sigma_2\sigma_3$. Using the Pauli matrix anticommutator $\{\sigma_i, \sigma_j\} = 2I\delta_{ij}$ we can show that[27]:

$$\{\xi^\mu, \xi^\nu\} = 2Ig^{\mu\nu} \tag{2.67}$$

verifying that this is indeed a fermionic theory.

Applying the same method to the Dirac equation as to the Klein-Gordon equation in Section 2.1.3 we obtain the same momentum mode energy[28]:

$$\omega_k = \sqrt{m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k \Delta x}{2}\right)} \tag{2.14}$$

2.5.2 The $u(p_k)$ and $v(p_k)$ Spinors

Whereas the scalar eigenstates are single plane waves $e^{ip_k x_n - i\omega_k t}$, 1D fermions have two non-orthogonal plane wave spinors. To determine these, we first write the fermion Hamiltonian[29]:

$$\begin{aligned}
\mathcal{L}_n &= \psi_n^\dagger \xi^0 (i\xi^0 \dot{\psi}_n + i\xi^1 \Delta_x \psi_n - m\psi_n) && (2.68)\\
\pi_n &= \frac{\partial \mathcal{L}_n}{\partial \dot{\psi}_n} = i\psi_n^\dagger && (2.69)\\
\mathcal{H}_n &= \pi_n \psi_n - \mathcal{L}_n = -i\psi_n^\dagger \xi^0 \xi^1 \Delta_x \psi_n + m\psi_n^\dagger \xi^0 \psi_n && (2.70)\\
&= (\psi_{1,n}^*, \psi_{2,n}^*) \begin{pmatrix} m & -i\Delta_x \\ -i\Delta_x & -m \end{pmatrix} \begin{pmatrix} \psi_{1,n} \\ \psi_{2,n} \end{pmatrix} && (2.71)
\end{aligned}$$

This Hamiltonian can be diagonalised, giving us eigenspinors $u(p_k)e^{-ip_k x_n + i\omega_k t}$ and $v(p_k)e^{ip_k x_n - i\omega_k t}$ with eigenenergies ω_k and $-\omega_k$ respectively[30], where:

$$\begin{aligned}
u(p_k) &= \sqrt{\omega_k + m} \begin{pmatrix} 1 \\ -\frac{2\sin\left(\frac{p_k \Delta x}{2}\right)}{\Delta x(\omega_k + m)} \end{pmatrix} && (2.72)\\
v(p_k) &= \sqrt{\omega_k + m} \begin{pmatrix} \frac{2\sin\left(\frac{p_k \Delta x}{2}\right)}{\Delta x(\omega_k + m)} \\ -1 \end{pmatrix} && (2.73)
\end{aligned}$$

[27] Verified in Appendix A.8
[28] Full derivation in Appendix A.9
[29] p. 52 in Peskin & Schroeder [1]
[30] Verified in Appendix A.10

These are fermion- and antifermion plane waves, respectively, and are relativistically normalised such that:

$$u(p_k)^\dagger u(p_k) = v(p_k)^\dagger v(p_k) = 2\omega_k \tag{2.74}$$

$$u(p_k)^\dagger v(-p_k) = v(p_k)^\dagger u(-p_k) = 0 \tag{2.75}$$

2.5.3 Quantising the Theory

To quantise, we invoke the field anticommutator[31]:

$$\left\{\hat{\psi}_{\alpha,n}, \hat{\psi}^\dagger_{\beta,m}\right\} = \delta_{\alpha\beta}\frac{\delta_{nm}}{\Delta x} \tag{2.76}$$

where α, β are spinor indices. With eigenspinors $u(p_k)$ and $v(p_k)$, we can quantise and Fourier expand ψ_n:

$$\hat{\psi}_n = \sum_{k=0}^{N-1} \frac{\Delta p}{2\pi 2\omega_k}\left(u(p_k)\hat{b}_k e^{ip_k x_n} + v(p_k)\hat{c}^\dagger_k e^{-ip_k x_n}\right) \tag{2.77}$$

$$\hat{\psi}^\dagger_n = \sum_{k=0}^{N-1} \frac{\Delta p}{2\pi 2\omega_k}\left(u(p_k)^\dagger\hat{b}^\dagger_k e^{-ip_k x_n} + v(p_k)^\dagger\hat{c}_k e^{ip_k x_n}\right) \tag{2.78}$$

where $\hat{b}_k$ and $\hat{c}_k$ are fermion- and antifermion ladder operators, respectively. Combining Eqns (2.76), (2.78) and (2.78) gives:

$$\left\{\hat{b}_k, \hat{b}^\dagger_l\right\} = \left\{\hat{c}_k, \hat{c}^\dagger_l\right\} = 2\omega_k 2\pi\frac{\delta_{kl}}{\Delta p} = 2\omega_k L\delta_{kl} \tag{2.79}$$

and all other anticommutators are zero. Note that the two types commute normally with each other:

$$\left[\hat{b}_k, \hat{c}^\dagger_l\right] = \left[\hat{b}^\dagger_k, \hat{c}_l\right] = \left[\hat{b}_k, \hat{c}_l\right] = \left[\hat{b}^\dagger_k, \hat{c}^\dagger_l\right] = 0 \tag{2.80}$$

2.5.4 Fock States

Fermionic Fock states are similar to those for scalars, but include both fermion and antifermion ladder operators:

$$|i\rangle = \prod_{k=0}^{N-1} \frac{\left(\hat{b}^\dagger_k\right)^{g_k}\left(\hat{c}^\dagger_k\right)^{h_k}}{\sqrt{(2\omega_k L)^{(g_k+h_k)}}}|0\rangle \tag{2.81}$$

$$= \frac{\left(\hat{b}^\dagger_0\right)^{g_0}\left(\hat{b}^\dagger_1\right)^{g_1}\ldots\left(\hat{b}^\dagger_{N-1}\right)^{g_{N-1}}\left(\hat{c}^\dagger_0\right)^{h_0}\left(\hat{c}^\dagger_1\right)^{h_1}\ldots\left(\hat{c}^\dagger_{N-1}\right)^{h_{N-1}}}{\sqrt{(2\omega_0 L)^{(g_0+h_0)}(2\omega_1 L)^{(g_1+h_1)}\ldots(2\omega_{N-1}L)^{(g_{N-1}+h_{N-1})}}}|0\rangle \tag{2.82}$$

[31] p. 56 in Peskin & Schroeder [1]

where the exponents $g_k, h_k = 0, 1$, reflecting the Pauli exclusion principle. This is due a minus sign from the anticommutator under interchange of two equal ladder operators, requiring zero contribution.

2.5.5 Fermions and the Sandwich Factor, F_k

The optimisation described in Section 2.3.2 also applies to fermions. We require normal ordering of the form:

$$(\hat{b}_0^\dagger \hat{b}_1^\dagger \dots \hat{b}_{N-1}^\dagger \hat{b}_{N-1} \dots \hat{b}_1 \hat{b}_0)(\hat{c}_0^\dagger \hat{c}_1^\dagger \dots \hat{c}_{N-1}^\dagger \hat{c}_{N-1} \dots \hat{c}_1 \hat{c}_0) \tag{2.83}$$

where $\hat{b}_k$'s and $\hat{c}_k$'s are allowed to intermix due to their zero commutator. Sandwiching this between two Fock states allows us to reuse the factor F_k since an even swaps assure no minus signs are introduced when splitting the ladder operators into equal-momentum number factors. It is important to normal order $\hat{b}_k/\hat{c}_k$ based on $k \pmod N$. Additionally, the Pauli exclusion principle dictates $l_k, n_k, m_k = 0, 1$ giving $F_k = 0$ otherwise.

This factor is then used in the same way as for scalars to find interaction Hamiltonians as well as average field and conjugate momentum values.

2.6 Dimensional Analysis

In deriving a 1D quantum field theory, the dimensions of some quantities have changed. Note that discretisation is irrelevant. Dimension changes are summarised[32] in Table 2.1:

Quantity	Dimension in 1D	Dimension in 3D
$\hat{\mathcal{L}}_n$	2	4
$\hat{\mathcal{H}}_n$	2	4
$\hat{\phi}_n$	0	1
$\hat{\pi}_n$	1	2
$\hat{\psi}_n, \hat{\psi}_n^\dagger$	$\frac{1}{2}$	$\frac{3}{2}$
$\hat{a}_k, \hat{a}_k^\dagger$	0	-1
$\hat{b}_k, \hat{b}_k^\dagger, \hat{c}_k, \hat{c}_k^\dagger$	0	-1
$u(p_k), v(p_k)$	$\frac{1}{2}$	$\frac{1}{2}$

TABLE 2.1: **Summary of quantities with changed dimensions:** Mass dimensions of quantities compared in 1D and 3D. All other relevant quantities are dimensionally unchanged. Note that relativistic normalisation is assumed in both 1D and 3D.

[32] Detailed derivation of dimensions in Appendix A.11

It is also worth noting that in simulating the theory, we work in arbitrary mass units. If we choose energy-momentum units GeV, spacetime gets unit GeV^{-1}, reflecting the Heisenberg uncertainty principle. This allows us to "zoom in" to the scale where interactions occurs by setting the dimensionful interaction parameters close to unity and simply redefining the units.

Chapter 3

Implementation

This chapter describes the implementation of the theory in Chapter 2. This includes mainly a treatment of Fock states, since their ordering has not been sufficiently defined, as well as what integrators will be used. Also included is a brief discussion of the choice of programming language and the code base developed.

3.1 State Number Functions and Cut-offs

Even a relatively modest system will have an excessively large number of Fock states; infinitely many in the case of scalars. To be calculable we must apply an appropriate cut-off. There are many ways to do this, but a natural cut-off is by particle number, i.e. considering all Fock States up to and including, say, P_{max} particles. This approximation relies on the assumption that probabilities decrease with increasing particle numbers, i.e. that the energy cost of creating a particle is comparatively high.

For scalars, applying indistinguishability of identical particles, the number of Fock states S_S is given by:

$$S_S(N, P_{max}) \equiv \sum_{P=0}^{P_{max}} \sigma_S(N, P) \tag{3.1}$$

$$\sigma_S(N, P) \equiv \binom{N + P - 1}{P} \tag{3.2}$$

where σ_S is the number of states with a given particle number P. Notice that S_S is unbounded from above with increasing P_{max} for a given N, reflecting that the commutation relation allows infinitely many particles in a finite space.

Due to the Pauli exclusion principle, the number of fermion Fock states S_F is:

$$S_F(N, P_{max}) \equiv \sum_{P=0}^{P_{max}} \sigma_F(N, P) \tag{3.3}$$

$$\sigma_F(N, P) \equiv \binom{2N}{P} \tag{3.4}$$

where the highest possible particle number is $2N$. This occurs when both fermions and antifermions occupy every lattice point.

3.2 Scalar Fock State Labelling

Increasing the particle number significantly limits performance, so we consider mainly Fock states with few particles. It is therefore beneficial to adopt a shorthand notation:

$$|i\rangle_S = |e_0, e_1, \ldots, e_{P-1}\rangle_N \propto \hat{a}^\dagger_{e_0} \hat{a}^\dagger_{e_1} \ldots \hat{a}^\dagger_{e_{P-1}} |0\rangle \tag{3.5}$$

$$e_0 \leq e_1 \leq \ldots \leq e_{P-1}$$

where the $\{e_n\}$ are the momentum numbers of the P particles in state i_S.

3.2.1 Stepping Algorithm

In order to calculate interaction-Hamiltonians, it is key to step efficiently between the Fock states. The algorithm is required to:

- return each state with $P \leq P_{max}$ particles.
- return each state only once.
- terminate after exhausting all viable states.

Although any Fock state ordering would work, it is sensible to step states by increasing particle numbers, starting with no particles. Listing 3.1 shows an algorithm meeting all these requirements (continued overleaf):

```
1 define index, i = 0
2 define empty ladder operator list, e = []
3 WHILE ( i < number of states )
4       PRINT list e
5       define element index, n = last index of e
6       WHILE ( index valid, n >= 0 )
```

```
7           increment element n, e[n]++
8           IF ( element reached max, e[n] == N )
9                 reset element n, e[n] = 0
10                decrement index, n−−
11          ELSE
12                BREAK out of innermost while loop
13    IF( exhausted all elements, n < 0 )
14          reset all elements, e = {0}
15          append another 0 to list, e[last+1] = 0
16    increment index, i++
```

LISTING 3.1: Pseudocode for the scalar Fock state stepping algorithm. This prints out all Fock states in the shorthand notation. See table below for an illustrative example.

This is much easier to understand by example, as presented in Table 3.1:

Scalar index	**Shorthand notation**	**Ladder operators**	**Particle number**	**Momentum number**
0	$\|\rangle_3$	$\|0\rangle$	0	0
1	$\|0\rangle_3$	$\hat{a}_0^\dagger\|0\rangle$	1	0
2	$\|1\rangle_3$	$\hat{a}_1^\dagger\|0\rangle$	1	1
3	$\|2\rangle_3$	$\hat{a}_2^\dagger\|0\rangle$	1	2
4	$\|0,0\rangle_3$	$(\hat{a}_0^\dagger)^2\|0\rangle$	2	0
5	$\|0,1\rangle_3$	$\hat{a}_0^\dagger\hat{a}_1^\dagger\|0\rangle$	2	1
6	$\|0,2\rangle_3$	$\hat{a}_0^\dagger\hat{a}_2^\dagger\|0\rangle$	2	2
7	$\|1,1\rangle_3$	$(\hat{a}_1^\dagger)^2\|0\rangle$	2	2
8	$\|1,2\rangle_3$	$\hat{a}_1^\dagger\hat{a}_2^\dagger\|0\rangle$	2	0
9	$\|2,2\rangle_3$	$(\hat{a}_2^\dagger)^2\|0\rangle$	2	1
10	$\|0,0,0\rangle_3$	$(\hat{a}_0^\dagger)^3\|0\rangle$	3	0
11	$\|0,0,1\rangle_3$	$(\hat{a}_0^\dagger)^2\hat{a}_1^\dagger\|0\rangle$	3	1
12	$\|0,0,2\rangle_3$	$(\hat{a}_0^\dagger)^2\hat{a}_2^\dagger\|0\rangle$	3	2
13	$\|0,1,1\rangle_3$	$\hat{a}_0^\dagger(\hat{a}_1^\dagger)^2\|0\rangle$	3	2
14	$\|0,1,2\rangle_3$	$\hat{a}_0^\dagger\hat{a}_1^\dagger\hat{a}_2^\dagger\|0\rangle$	3	0
15	$\|0,2,2\rangle_3$	$\hat{a}_0^\dagger(\hat{a}_2^\dagger)^2\|0\rangle$	3	1
16	$\|1,1,1\rangle_3$	$(\hat{a}_1^\dagger)^3\|0\rangle$	3	0
17	$\|1,1,2\rangle_3$	$(\hat{a}_1^\dagger)^2\hat{a}_2^\dagger\|0\rangle$	3	1
18	$\|1,2,2\rangle_3$	$\hat{a}_1^\dagger(\hat{a}_2^\dagger)^2\|0\rangle$	3	2
19	$\|2,2,2\rangle_3$	$(\hat{a}_2^\dagger)^3\|0\rangle$	3	0

TABLE 3.1: **Scalar Fock state stepping**. States returned for $N = 3$ and $P_{max} = 3$.

3.2.2 Going Back Again: State Index from Ladder Operators

In efficiently calculating interaction Hamiltonians, it will be necessary to find the state index purely from knowing its ladder operators (the e_k's). By inspection, the following formula was found:

$$i_S\left(\{e_k\}\right) = S_S(N, P-1) + \sum_{k=0}^{P-1} \sum_{m=0}^{(e_p - e_{(k-1)} - 1)} \sigma_S(N - m - e_{(k-1)}, P - 1 - k) \qquad (3.6)$$

where $e_{-1} \equiv 0$. This has been verified for all i_S using (P_{max}, N)-combinations:

P_{max}	N
0	$0 \rightarrow 10^9$
1	$0 \rightarrow 10000$
2	$0 \rightarrow 1000$
3	$0 \rightarrow 200$
4	$0 \rightarrow 100$
5	$0 \rightarrow 60$
6	$0 \rightarrow 40$
$7 \rightarrow 9$	$0 \rightarrow 20$

TABLE 3.2: **Scalar Index Verification**. The scalar index i_S has been verified for all the listed combinations of P_{max} and N. The expression is untested outside these bounds due to calculation time, but is likely to be valid. Our investigations will not exceed these limits.

3.3 Fermion Fock State Labelling

Stepping through fermion Fock states with increasing total particle number is similar to, but more complicated than, scalars. This is due to the mixing of ladder operators $\hat{b}_k^\dagger$ and $\hat{c}_l^\dagger$. Fortunately, the Pauli exclusion principle simplifies the problem by allowing at most one of each ladder operator. We adopt the shorthand notation:

$$|i\rangle_F = |f_0, f_1, \ldots, f_{P_b-1}\,; g_0, g_1, \ldots, g_{P_c-1}\rangle_N \quad \propto \quad \hat{b}_{f_0}^\dagger \hat{b}_{f_1}^\dagger \ldots \hat{b}_{f_{P_b-1}}^\dagger \cdot \hat{c}_{g_0}^\dagger \hat{c}_{g_1}^\dagger \ldots \hat{c}_{g_{P_c-1}}^\dagger |0\rangle \qquad (3.7)$$

$$f_0 < f_1 < \ldots < f_{P_b-1} \quad , \quad g_0 < g_1 < \ldots < g_{P_c-1}$$

where P_b and P_c are number of fermion and antifermions, respectively: $P = P_b + P_c$.

3.3.1 Stepping Algorithm

A stepping algorithm must meet the same requirements as for scalars (see Section 3.2.1). We can also choose to require that for a given total particle number P, we order states by increasing antifermion number P_c, i.e. starting with only fermions, then swapping each fermion with an antifermion. Listing 3.2 shows an algorithm meeting all the above:

```
1  define index, i = 0
2  define empty fermion and antifermion ladder operator lists, f = [] and g = []
3  define total particle number, P = 0 and fermion particle number, P_b = 0
4  WHILE ( i < number of states )
5          PRINT state (f ; g)
6          define fermion element index, n = last index of f
7          WHILE ( index valid, n >= 0 )
8                  increment element n, f[n]++
9                  IF ( n == last index element reached max, f[n] == f[n+1] where f[P_b] = N )
10                         reset element n, f[n] = f[n-1] + 1
11                         decrement index, n--
12                 ELSE
13                         BREAK out of innermost while loop
14         IF( exhausted all fermion elements, n < 0 )
15                 define antifermion element index, m = last index of g
16                 WHILE ( index valid, m >= 0 )
17                         increment element m, g[m]++
18                         IF ( m == last index element reached max, g[m] == g[m+1] where g[P-P_b] = N )
19                                 reset element m, g[m] = g[m-1]+1
20                                 decrement index, m--
21                         ELSE
22                                 BREAK out of innermost while loop
23                 IF( exhausted all antifermion elements, m < 0 )
24                         decrement P_b--
25                         IF ( P_b invalid , P_b < 0 )
26                                 increment P++
27                                 IF ( lattice spaces available, P <= 2N)
28                                         set fermion number P_b = minimum of P and N
29                                 ELSE
30                                         terminate as all states are exhausted
31                                 reset f = [0,1,2..., P_b-1] and g = [0,1,...,P-P_b-1]
32         increment index, i++
```

LISTING 3.2: Pseudocode for the fermion Fock state stepping algorithm. This prints out all Fock states in the shorthand notation. See table below for an illustrative example.

Example values are shown in Table 3.3:

Fermion index	**Shorthand notation**	**Ladder ops.**	P	P_b	P_c	**Mom. number**
0	$\|;\rangle_2$	$\|0\rangle$	0	0	0	0
1	$\|0;\rangle_2$	$\hat{b}_0^\dagger\|0\rangle$	1	1	0	0
2	$\|1;\rangle_2$	$\hat{b}_1^\dagger\|0\rangle$	1	1	0	1
3	$\|;0\rangle_2$	$\hat{c}_0^\dagger\|0\rangle$	1	0	1	0
4	$\|;1\rangle_2$	$\hat{c}_1^\dagger\|0\rangle$	1	0	1	1
5	$\|0,1;\rangle_2$	$\hat{b}_0^\dagger\hat{b}_1^\dagger\|0\rangle$	2	2	0	1
6	$\|0;0\rangle_2$	$\hat{b}_0^\dagger\hat{c}_0^\dagger\|0\rangle$	2	1	1	0
7	$\|1;0\rangle_2$	$\hat{b}_1^\dagger\hat{c}_0^\dagger\|0\rangle$	2	1	1	1
8	$\|0;1\rangle_2$	$\hat{b}_0^\dagger\hat{c}_1^\dagger\|0\rangle$	2	1	1	1
9	$\|1;1\rangle_2$	$\hat{b}_1^\dagger\hat{c}_1^\dagger\|0\rangle$	2	1	1	0
10	$\|;0,1\rangle_2$	$\hat{c}_0^\dagger\hat{c}_1^\dagger\|0\rangle$	2	0	2	1
11	$\|0,1;0\rangle_2$	$\hat{b}_0^\dagger\hat{b}_1^\dagger\hat{c}_0^\dagger\|0\rangle$	3	2	1	0
12	$\|0,1;0\rangle_2$	$\hat{b}_0^\dagger\hat{b}_1^\dagger\hat{c}_1^\dagger\|0\rangle$	3	2	1	1
13	$\|0;0,1\rangle_2$	$\hat{b}_0^\dagger\hat{c}_0^\dagger\hat{c}_1^\dagger\|0\rangle$	3	1	2	0
14	$\|1;0,1\rangle_2$	$\hat{b}_1^\dagger\hat{c}_0^\dagger\hat{c}_1^\dagger\|0\rangle$	3	1	2	1
15	$\|0,1;0,1\rangle_2$	$\hat{b}_0^\dagger\hat{b}_1^\dagger\hat{c}_0^\dagger\hat{c}_1^\dagger\|0\rangle$	4	2	2	0

TABLE 3.3: **Fermion Fock state stepping**. Shows states for a system with $N = 2$ and $P_{max} = 4$ (all states). P, P_b and P_c denote total particle number, fermion number and antifermion number, respectively.

3.3.2 Going Back Again: Fermion Edition

As with scalars, it is necessary to find the fermion state index based on its ladder operators. The following expression was found after lengthy iterative inspection:

$$\begin{aligned} i_F\left(\{f_k\},\{g_k\}\right) &= S_F(N,P-1) + \sum_{k=P_b+1}^{P} \sigma_{FC}(N,k)\sigma_{FC}(N,P-k) \\ &\quad + \sigma_{FC}(N,P_c) \sum_{k=0}^{P_b-1} \sum_{m=0}^{(f_k - f_{(k-1)} - 1)} \sigma_{FC}\left(N-m-f_{(k-1)}-2, P_b-1-k\right) \\ &\quad + \sum_{k=0}^{P_c-1} \sum_{m=0}^{(g_k - g_{(k-1)} - 1)} \sigma_{FC}\left(N-m-g_{(k-1)}-2, P_c-1-k\right) \end{aligned} \tag{3.8}$$

where $f_{-1} = g_{-1} \equiv 0$ and σ_{FC} is the number of states with P particles for a single fermion component (either fermions or antifermions):

$$\sigma_{FC}(N, P) = \binom{N}{P} \tag{3.9}$$

as well as P_b, P_c being the numbers of particles and antiparticles ($P = P_b + P_c$). This has been verified for all i_F using (P_{max}, N)-combinations:

P_{max}	N
0	$0 \rightarrow 10^8$
1	$0 \rightarrow 4000$
2	$0 \rightarrow 400$
3	$0 \rightarrow 100$
$4 \rightarrow 5$	$0 \rightarrow 40$
6	$0 \rightarrow 20$
$7 \rightarrow 12$	$0 \rightarrow 10$
$2N$ (all)	$0 \rightarrow 9$

TABLE 3.4: **Fermion Index Verification**. The fermion index i_F has been verified for all the listed combinations of P_{max} and N. Our investigations will not go beyond these limits, but it is seen as likely to hold in general. The implementation, not the mathematical expression, was found to fail at high P_{max} due to the 4 byte integer limit.

3.4 Integrators Investigated

Two integrators were implemented and investigated:

- Second order integrator: Eqn (2.34). First order first step: Eqn (2.37) with $K = 1$.
- Arbitrary even-order K integrator: Eqn (2.38). Kth order first step: Eqn (2.37).

The second order integrator and the $(K = 2)$-integrator differ only by their first step, allowing us to probe its importance.

3.5 Programming Language and Overall Code Structure

3.5.1 Java: An Ideal Choice

Choice of programming language was motivated by:

- **Visualisation** - Access to interactive graphics frameworks
- **Outreach** - Code extendability and distributability
- **Correctness** - Access to unit testing frameworks
- **Performance** - Speed of heavy realtime calculations

Java provides powerful graphics frameworks and standardised unit testing, as well as being distributable by construction. Although slower than C++[1] or Fortran, Java is much faster than most high level languages[2], and since performance is not the highest priority it is an acceptable choice.

The aim of the code base was to provide both a graphical user interface and an easy-to-use Computational QFT-package. This was achieved by applying an object-oriented software design approach, making it easy to abstract away details of QFT and intuitively manipulate objects like quantum states.

Further details of the extensive code structure are deemed irrelevant, but can be found in Appendix B.1. A graphical user interface guide is included in Appendix B.2.

[1] According to Java/C++ comparison by The Computer Language Benchmark Game [3]
[2] According to Java/Python comparison by The Computer Language Benchmark Game [4]

Chapter 4

Results

To test whether the implementation of the developed theory is consistent with conventional QFT, it is necessary to compare them quantitatively. In addition, many interesting qualitative observations can be made using the graphical user interface.

The quantitative comparison has two parts: free- and interacting theory. The free theory has a simple analytical solution, allowing us to measure the accuracy of the integrators. Building on this, we take advantage of an analytical interacting theory solution to test the validity of the Hamiltonian calculations.

4.1 Free Theory: Integrator Error Analysis

Time evolution in free theory is simply a phase rotation of all Fock state coefficients proportional to their energy:

$$c_i(t)|i\rangle = e^{-iE_i t} c_i(0)|i\rangle \tag{4.1}$$

This needs only a single calculation and is approximated to be errorless. Calculating the same coefficients with a numerical integrator using different time intervals Δt allows us to verify the order of the integrator as well as its symplecticity.

We need only consider either scalars or fermions since they share integrator implementation (see Appendix B.1.2). Scalars are chosen for their simplicity and performance[1].

[1] All the below tests were also applied to fermions, with identical results, as a consistency check. They will not be reproduced here.

4.1.1 Second Order Integrator

The second order integrator, Eqn (2.34), was quantitatively compared to the analytical solution. Plots of relative coefficient error against both time interval (Figure 4.1) and total integration time (Figure 4.2) were produced to test the order of the integrator, as well as an energy against time plot (Figure 4.3) to test for symplecticity (energy conservation). All tests used a single particle wave packet for integration, and the relative error was averaged over all coefficients.

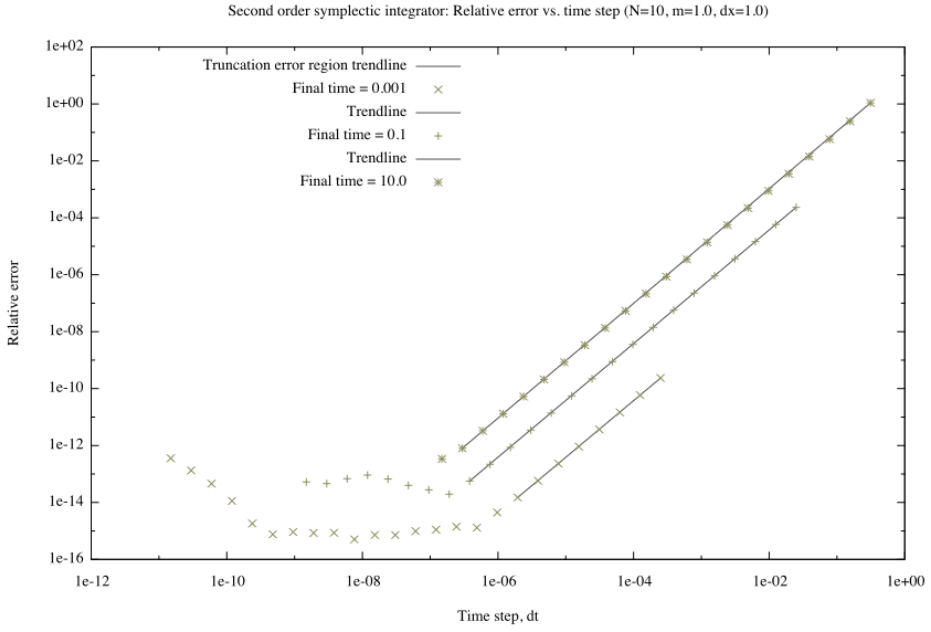

FIGURE 4.1: **Second order integrator, relative error vs. time step (log-log)**: Shown for final (total integration) times of 10, 0.1 and 0.001 (arbitrary inverse mass unit). Trendlines are fitted in the region where truncation errors dominate, with slopes $(2.006 \pm 0.002) \approx 2$ indicating that the integrator is second order.

A linear regression of the trend lines in Figure 4.1 gives a slope and hence an integrator order of (2.0057 ± 0.0022), as expected. The error plateaus and rises again at small time steps due to the relative importance of rounding errors, brought about by the real number binary representation[2].

The trendlines in Figure 4.2 linearly regressed give a slope of 1.0007 ± 0.0009. This is expected for any integration order since we effectively consider the time integral over multiple constant final time calculations $\propto O(t_{final})$.

Figure 4.3 shows oscillation of energy over time. Energy is conserved on average, verifying the second order integrator as symplectic.

[2]Number representation used: Java primitive **double** (4 bytes).

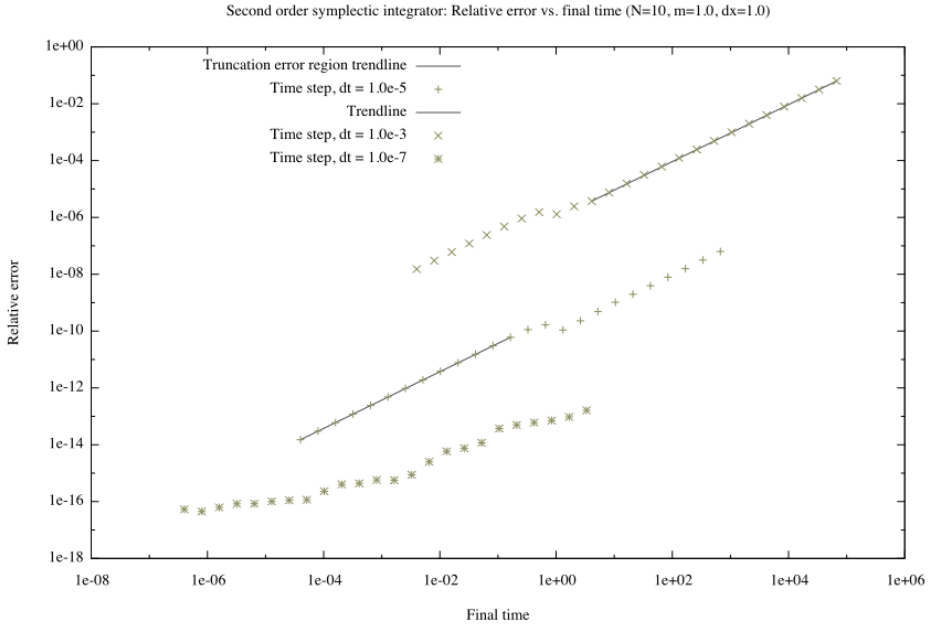

FIGURE 4.2: **Second order integrator, relative error vs. final time (log-log)**: Shown for time steps of 10^{-3}, 10^{-5} and 10^{-7} (arbitrary inverse mass unit). The fitted trendlines have slopes of $(1.0007 \pm 0.0009) \approx 1$. Plateauing is an artefact of rounding errors and binary number representations.

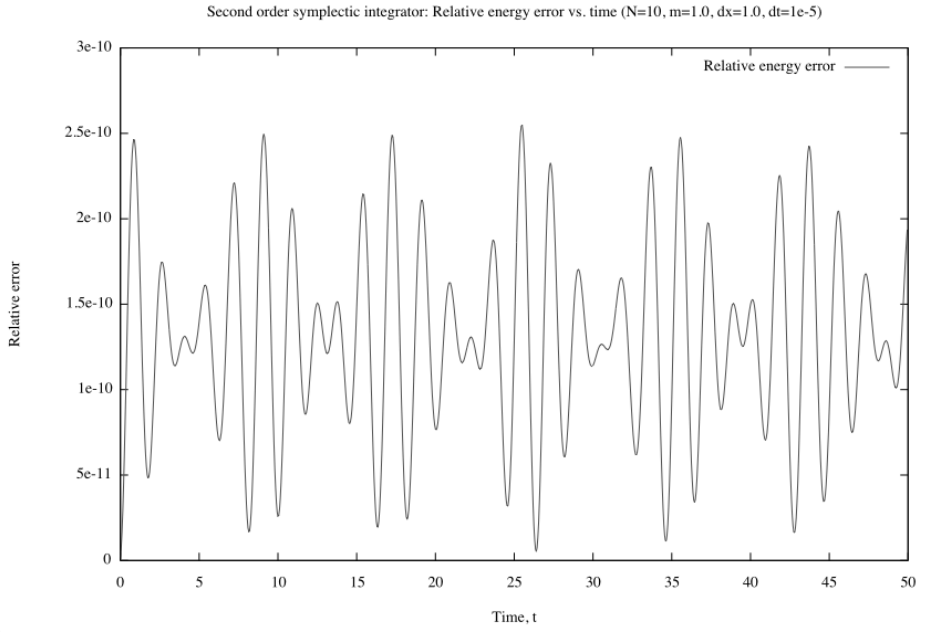

FIGURE 4.3: **Second order integrator, energy vs. time**: This plot shows the relative energy error over time using 1000 data points. It shows oscillation of the energy, but no divergence, clearly indicating conservation of energy (on average) and therefore that the integrator is symplectic. It is interesting to note that the error is always positive.

4.1.2 Arbitrary Even-Order Integrator

The arbitrary even-order integrator, Eqn (2.38), was subjected to the same tests as in Section 4.1.1. For each test, a range of integrator orders K were used. Note that the $(K = 2)$-integrator and the second order integrator are equivalent apart from their first step integration order (first and second order).

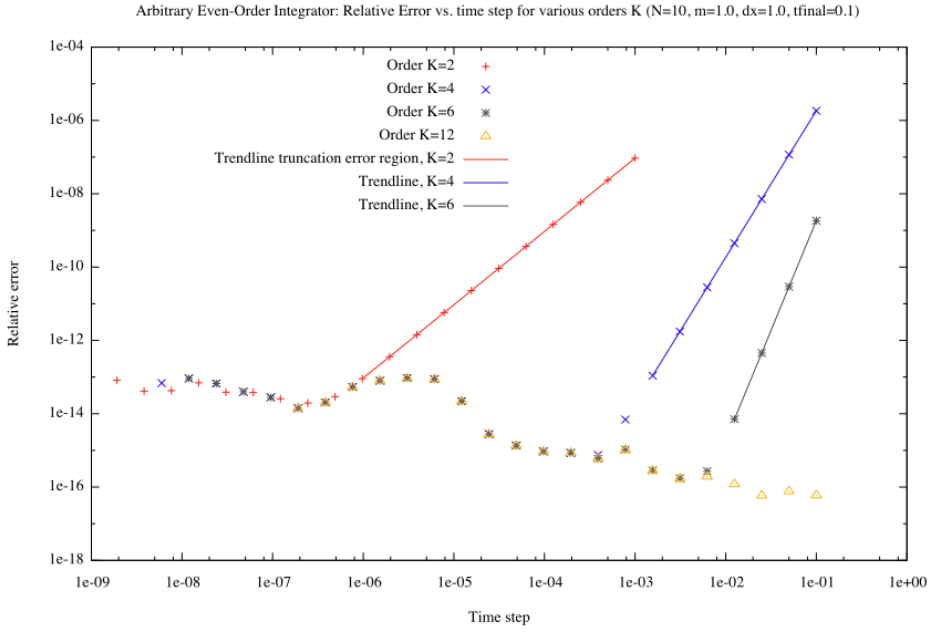

FIGURE 4.4: **Arbitrary even-order integrator, relative error vs. time step (log-log)**: Shown for integration orders $K = 2, 4, 6, 12$. Trendlines are fitted in the region where truncation errors dominate, with slopes $(2.00002 \pm 0.00001) \approx 2$, $(3.999 \pm 0.002) \approx 4$ and $(5.999 \pm 0.005) \approx 6$ for $K = 2, 4, 6$, respectively. Note that for $K = 12$ no truncation error can be seen due to rounding errors. We observe that the highest accuracy calculation was done using a single calculation $(K = 12, \Delta t = 0.1)$.

Figure 4.4 verifies that the integrator is indeed of order K in the truncation error dominated region. Interestingly, we observe that the highest accuracy calculation is also one of the fastest ($K = 12$, one time step). This suggests a new, vastly quicker method of integrating, using a single step while increasing the integration order[3], rather than increasing the number of time steps.

Figure 4.5 shows clearly the integrator error linearly diverges with time, independent of the integrator order K. However, the error constant decreases as $O(const^K)$ with increasing K, whereas the number of calculations increased by $O(K)$, resulting in a net performance gain with higher K.

We see from Figure 4.6 that energy is conserved on average, with an oscillating relative error. Comparing the error amplitude to Figure 4.3 ($\sim 10^{-10}$ and $\sim 10^{-14}$, respectively)

[3]Subject to constraint that all phasors rotate by less than $\sim 2\pi$ in that time step.

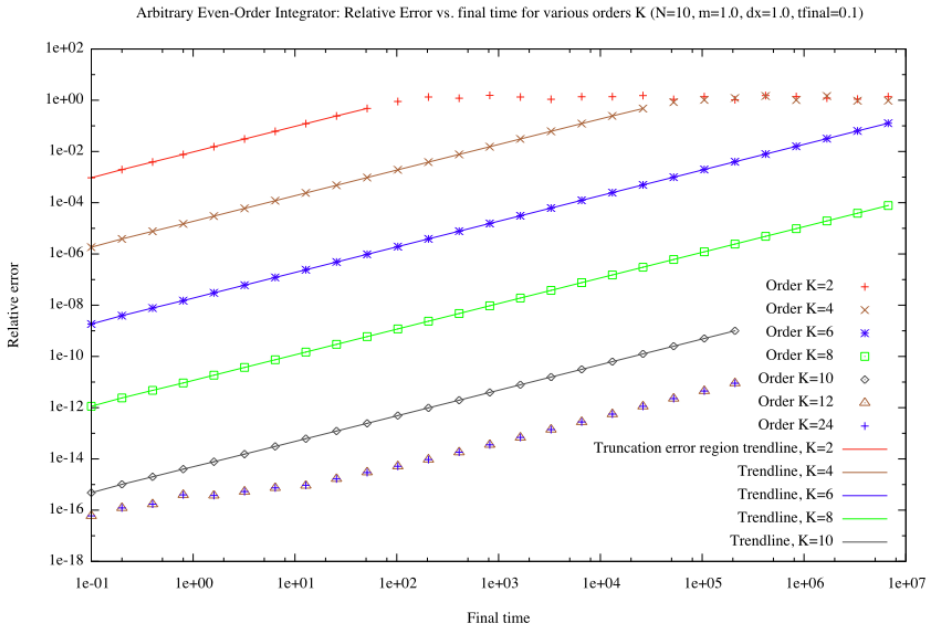

FIGURE 4.5: **Arbitrary even-order integrator, relative error vs. final time (log-log)**: Shown for integration orders $K = 2, 4, 6, 8, 10, 12, 24$. Trendlines are fitted in the region where truncation errors dominate, all with slope $(0.9992 \pm 0.0002) \approx 1$. Higher order integrators are seen to Orders K=2,4 flatten at error ≈ 1 because phasors can only at most point in opposite directions. In this specific case the integration order is irrelevant for $K > 10$ where rounding errors dominate.

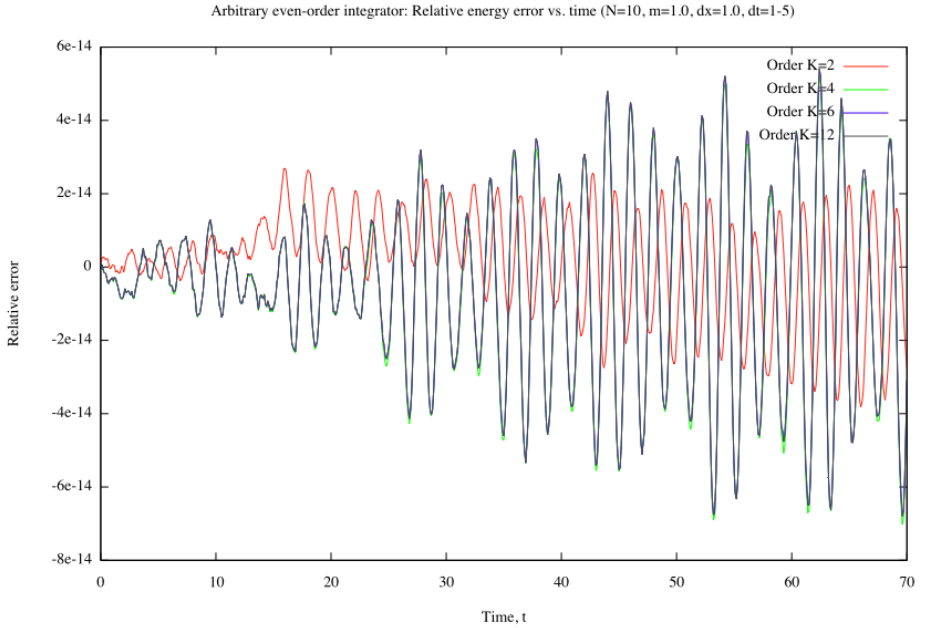

FIGURE 4.6: **Arbitrary even-order order integrator, energy vs. time (free theory)**: Shown for integration orders $K = 2, 4, 6, 12$, each with 700 data points. Integration orders $K = 4, 6, 12$ have all converged onto a single line. All orders have a conserved average, but oscillate with a linearly diverging amplitude.

indicates that using a higher order first step has a significant effect on errors (for $K = 2$). Although second order integrators are symplectic, both $K = 2$ and $K > 2$ shows a linearly diverges oscillation amplitude caused by accumulative rounding errors. Orders $K > 2$ are symplectic in free theory, but not so in interacting theory, as shown below.

4.2 Interacting Theory: Quantitative Comparison to Analytical Solutions

4.2.1 Conservation of Energy and Symplecticity

So far the integrators have only been tested for free theory. In interacting theory, due to the lack of an analytical solution it is not possible to compare Fock state coefficients. However, energy should still be conserved and can be calculated without knowledge of eigenstates.

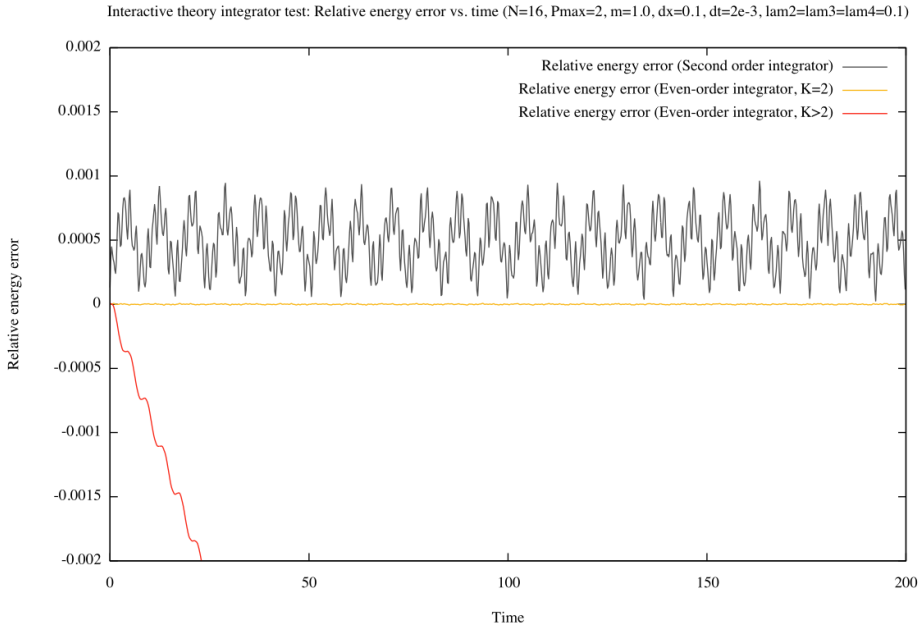

FIGURE 4.7: **Multiple integrators in interacting theory, energy vs. time**: Shown for the second order integrator and even-order integrators $K = 2, > 2$, each with 1000 data points. A highly interacting theory was chosen, with $\lambda^{(2)} = \lambda^{(3)} = \lambda^{(4)} = 0.1$. The linear divergence of $K > 2$ shows clearly that it is not symplectic. $K = 2$ performs better than the second order integrator due to a higher order first step.

Figure 4.7 shows that second order integrators conserve energy and are symplectic in a highly interacting theory. Notice the significance of the first step integration order for the error oscillation amplitude. The divergence of the higher order ($K > 2$)-integrators clearly shows that they are not symplectic.

Based on this comparison, the arbitrary even-order integrator with $(K = 2)$ was chosen for use in the graphical user interface, as it is the most accurate symplectic integrator.

4.2.2 ϕ^2-theory and Effective Mass Comparisons

As discussed in Section 2.4.2, a quantitative test of interacting theory is possible in scalar ϕ^2-theory by finding the two lowest eigenenergies and identifying their difference as an effective mass. An analytical prediction can be made and the error can be measured. The prediction assumes access to all particle numbers. The measured effective mass should therefore tend to the predicted mass with increasing P_{max}.

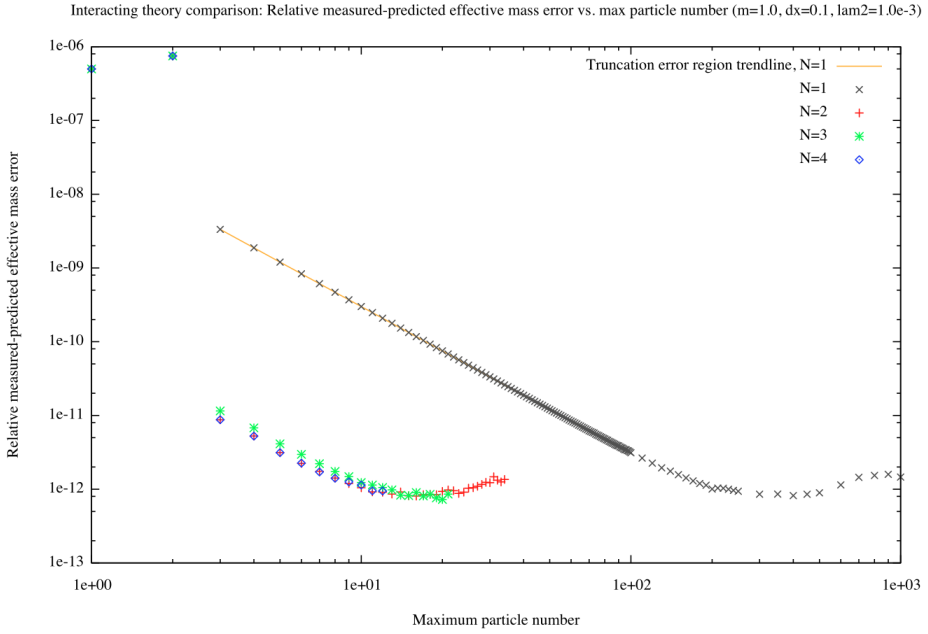

FIGURE 4.8: **Relative measured-to-predicted effective mass error vs. max particle number (log-log)**: This plot shows that the error in the measured effective mass goes down with increasing particle number considered, as expected. The relative error is shown for $1 \leq N \leq 4$, where $N = 1$ is less accurate (by a factor ~ 500) due to the lack of spatial derivatives in the calculation. The error flattens at large particle numbers due to accumulation of rounding errors. The truncation error dominated region has a slope $(-1.991 \pm 0.001) \approx -2$, based on the trend line for $N = 1$.

Using a very small error tolerance level in the Shifted Power Iteration Method to avoid accumulation of rounding errors, a comparison of measured and predicted effective masses were made for increasing P_{max}. This was repeated for $1 \leq N \leq 4$, limited by calculation time, as shown in Figure 4.8. For $N = 1$, the error is significantly higher due to the lack of spatial derivatives in the calculation. However, for all N the error goes as: $error \propto (P_{max})^{-2}$, in the truncation error dominated region. The plot also suggests that at least 3 particles should be considered.

This effectively verifies that the theory developed converges to conventional QFT in the limit $P_{max} \to \infty$.

4.3 Qualitative Observations

Using the graphical user interface, many interesting phenomena were observed. This section includes visual descriptions of some of them (Figures 4.9 - 4.14).

Interesting points noted:

- Throughout all particle propagation and interactions, total momentum was seen to be conserved.
- As wave packets gain momentum, they stop and then start moving backwards (aliasing).
- In fermion interactions ($\bar{\psi}\psi$) the fermion number ($N_{fermions} - N_{antifermions}$) is conserved.
- In even-power interactions, ϕ^{even} or ψ^{even} there is no exchange of probability between odd and even particle numbers.
- The field value[4] vanishes when there are no non-zero coefficients in two consecutive particle numbers.
- When using a negative scalar ϕ^2-term and a positive ϕ^4-term with appropriate couplings, a single particle appears to lose speed (not momentum), suggesting it has acquired additional mass. However, no vacuum expectation value was seen, because of the two previous points.

All the above points, as well as all figures, agree with or are allowed by the theory, but some are not readily derived from the equations, e.g. 2-particle collision outcomes. It is unclear whether the last point has any relation to the Higgs-mechanism.

[4]See Figure B.3 (appendix) for an example of how field values were implemented.

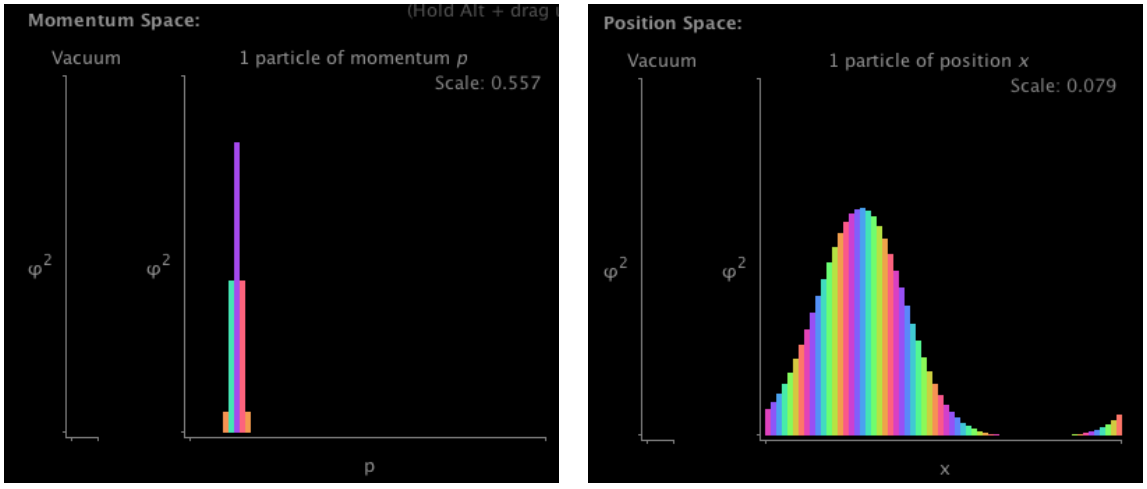

FIGURE 4.9: **Freely propagating single particle**: Height represents probability (Fock state coefficient mod squared) and colour represents complex phase. The gaussian momentum wave packet is stationary, whereas the gaussian position wave packet is moving right at a constant speed (periodic boundary). The vacuum remains empty. When the wave packet momentum is past half max (on right side), the particle moves left; an artefact of sampling in discrete periodic space (aliasing). Fermions behave identically. $[N = 64, P_{max} = 1, \Delta x = 0.1, m = 1]$

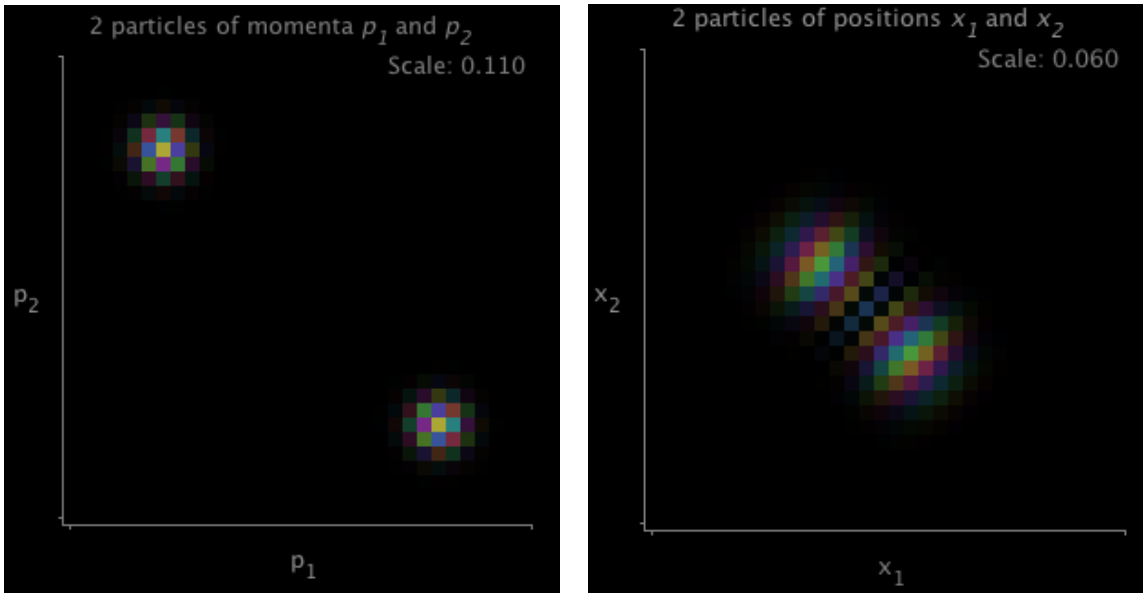

FIGURE 4.10: **Two non-interacting particles passing (left: momentum space, right: position space)**: Colour intensity represents probability. The 2D-gaussian momentum wave packet represents two particles of normally distributed momenta. Notice also the diagonal symmetry, indicating scalar indistinguishability. In position space, the wave packets move diagonally towards each other and pass through, from lack of interactions. $[N = 32, P_{max} = 2, \Delta x = 0.1, m = 1]$

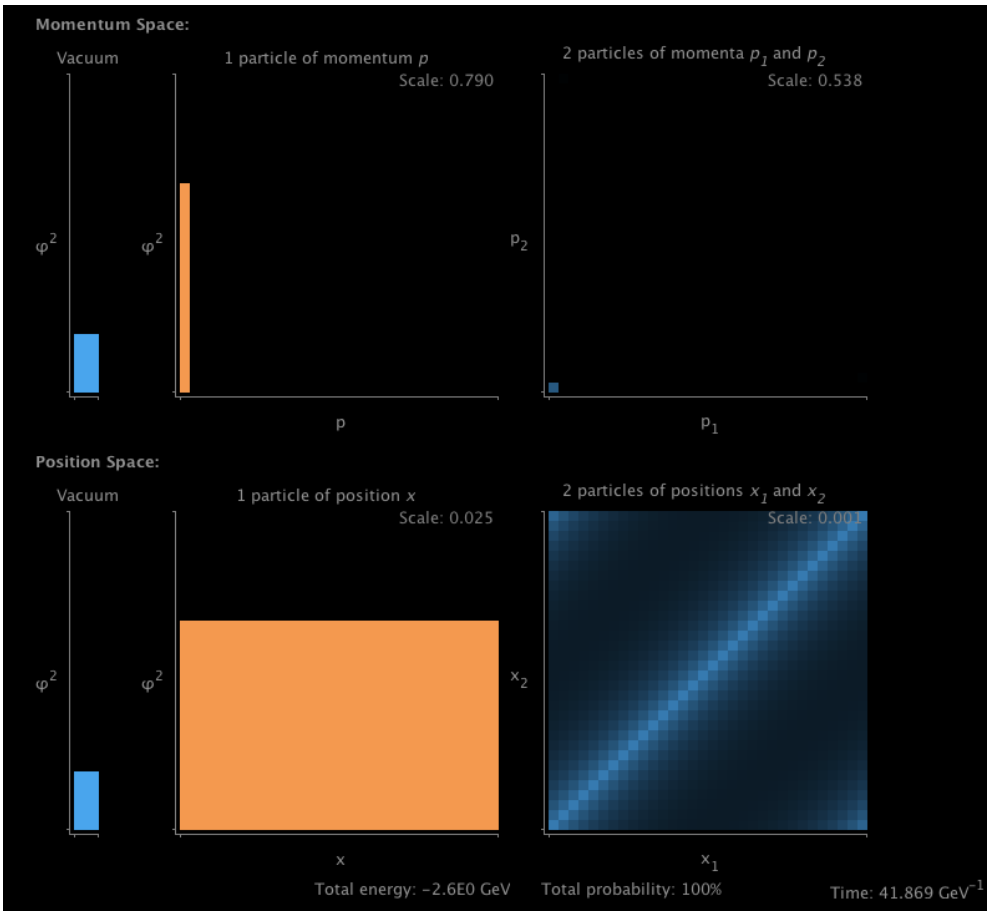

FIGURE 4.11: **Interacting ϕ^3-theory ground state (vacuum)**: Calculating the ground state of the an interacting theory gives contributions to all stationary particles available as well as some minor contributions from two oppositely moving particles (not visible on 2-particle momentum plot), giving a zero net momentum. Observe how definite knowledge of 1-particle momentum ($= 0$) implies no knowledge of its position, by the Heisenberg uncertainty principle. Although particles are interacting, amplitudes are fixed since the system is in an eigenstate. The energy is negative, reflecting the amount of energy it would take to make a no-particle vacuum (2.6 GeV). Please note that the scale is different on each plot, as indicated in their upper right corners. $[N = 32, P_{max} = 2, \Delta x = 0.1, m = 1, \lambda^{(3)} = 1.5]$

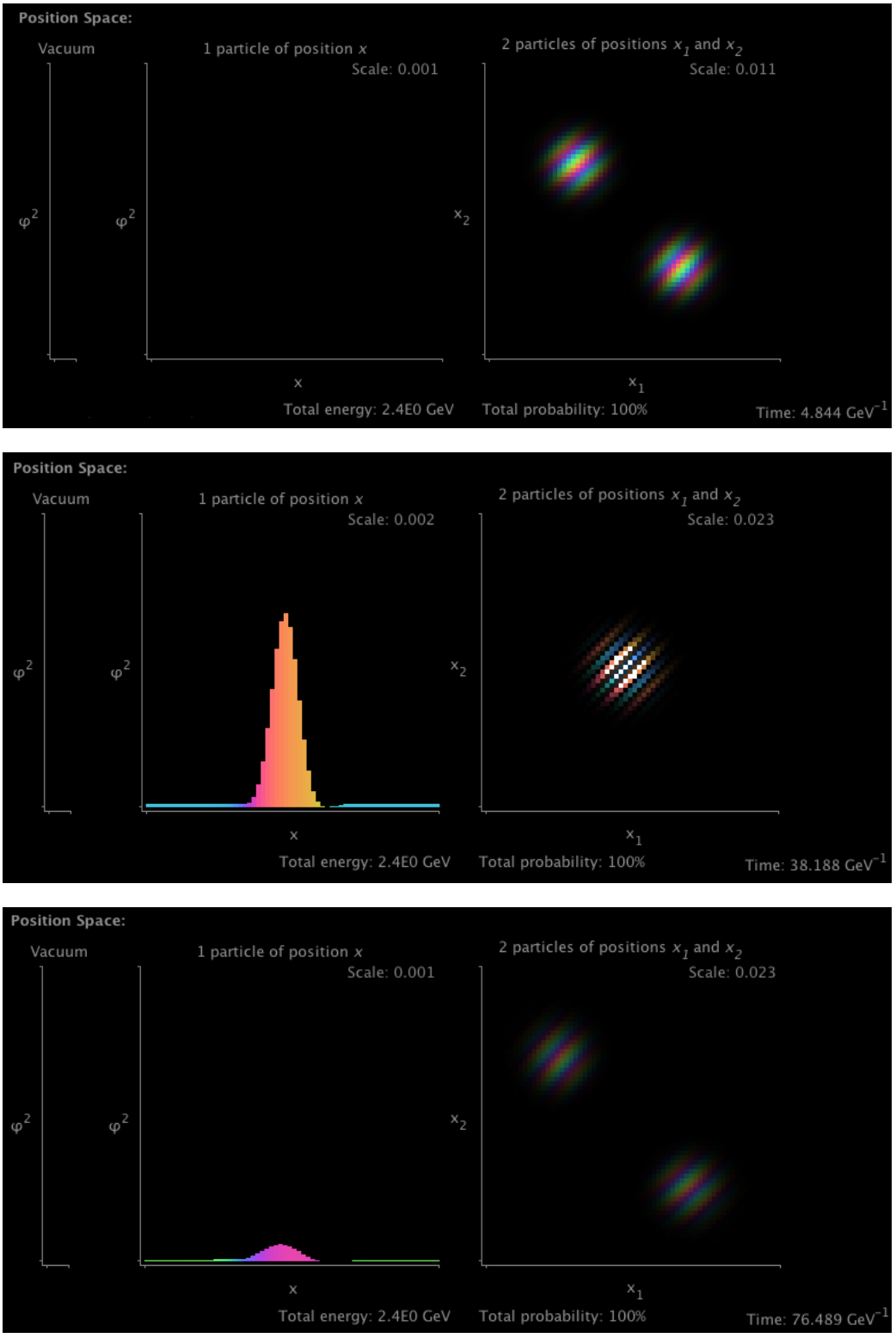

FIGURE 4.12: **Interacting ϕ^3-theory 2-particle collision and "virtual particle" creation**: The upper plot shows two particles at different positions and no single particle. The middle plots shows the two particles interacting and a single "virtual" particle created at the same position. The lower plot shows a small probability left for a single particle having been created. Notice how energy is conserved. Please also note that the scales change, albeit not significantly. $[N = 64, P_{max} = 2, \Delta x = 1.3, m = 1\lambda^{(3)} = 0.23]$

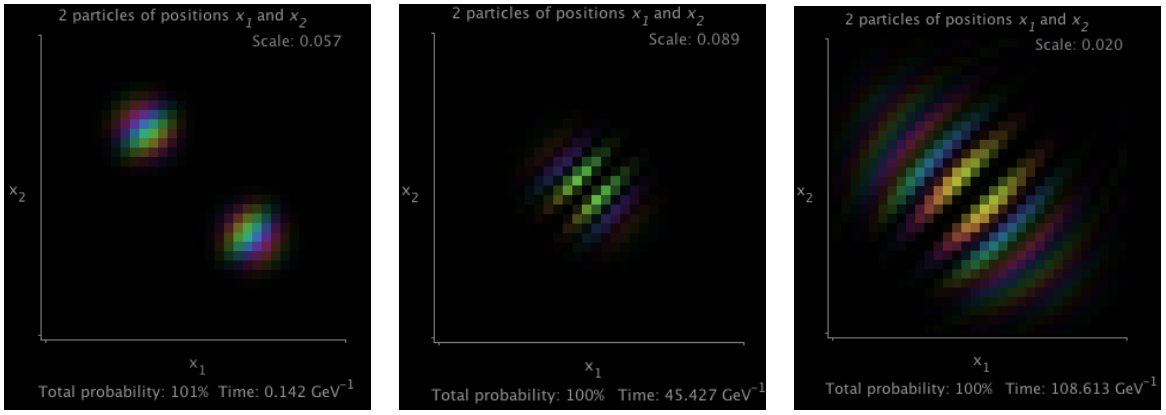

FIGURE 4.13: **Interacting ϕ^4-theory 2-particle collision and momentum redistribution**: The three plots show a 4-vertex interaction (ϕ^4) collision of two particles (time increasing from left to right). The collision leads to a redistribution of particle momenta. No significant probability was observed for other particle numbers. Note that the scale changes. $[N = 32, P_{max} = 2, \Delta x = 1.3, m = 1, \lambda^{(4)} = 1.5]$

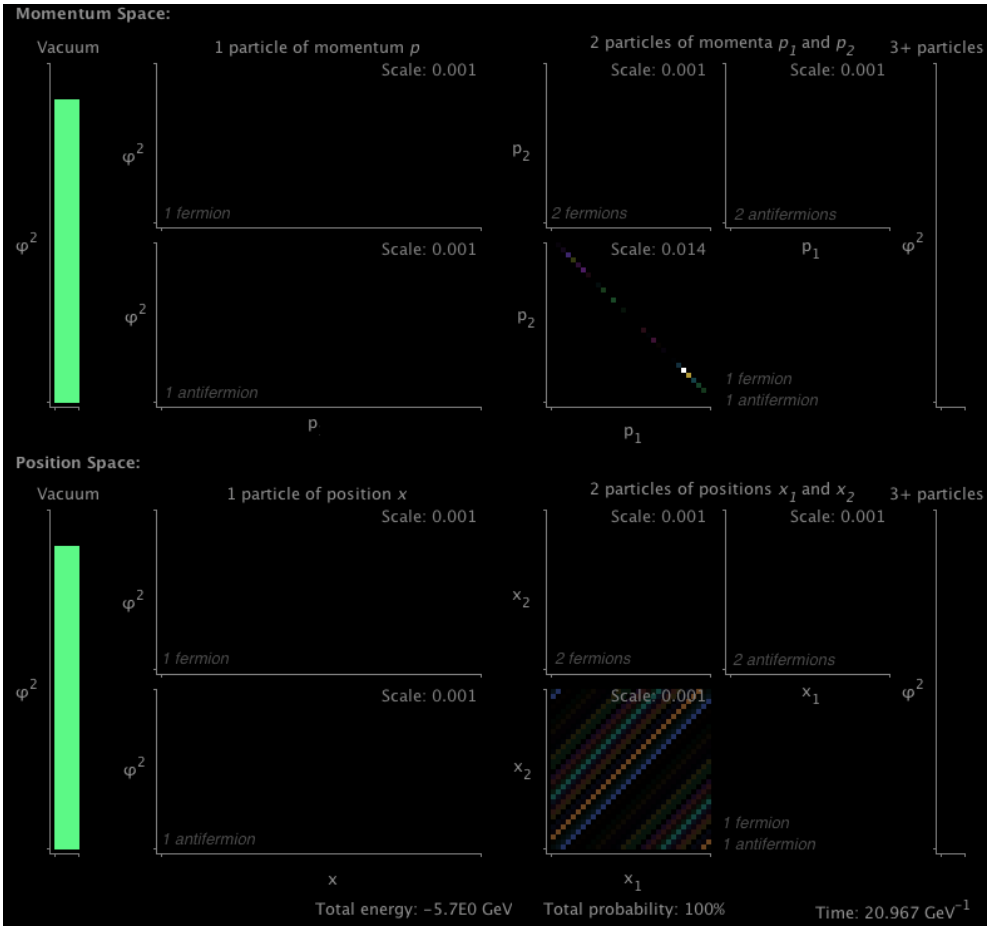

FIGURE 4.14: **Fermion near-vacuum $\bar{\psi}\psi$-interaction**: The above plot shows that starting from a 0-particle vacuum only fermion-antifermion pairs can be made, due to fermion-number conservation. Only states of zero net momentum have non-zero probability. Since the evaluation stops at 3 particles, no multiple fermion-antifermion pairs are considered. $[N = 32, P_{max} = 3, \Delta x = 0.81, m = 1, \lambda^{(2)} = 0.72]$

Chapter 5

Discussion

The simplified version of QFT developed is not, and is not aimed to be, an accurate description of the Universe. However, it was assumed to retain many of the important features of conventional QFT, while also being calculable in realtime by current computers. The following sections discuss the successes and failings of the theory developed with regards to this.

5.1 Theory: Validity of Approximation Assumptions

The quantum field theory developed was found to be mathematically consistent, at least at the level of our investigations. Its relevance to the Universe did, however, rest upon a number of assumptions, which we are now in a position to check the validity of.

- **1-dimensional**: 1D does indeed contain most 3D physics, allowing us to see particle propagation (Figure 4.9) and two-particle collisions in their zero-momentum frame (Figures 4.10 and 4.12). Most equations retain their general form. However, it does not allow particle scattering and it constricts fermions to only particles and antiparticles with no spin-degree of freedom. In addition, some quantities change dimensions, including interaction couplings. This has consequences for the renormalisability of interaction terms, leaving all 1D self-interactions renormalisable as opposed to in 3D. This does not, however, stop us from only considering interactions relevant to 3D.
- **Discreteness and periodicity**: Although some quantities and equations changed, e.g. Eqn (2.14), they all tend to the continuous theory in the limit $N \to \infty$. Discretisation is only apparent at very small N, as seen in Figure 4.8, and is not a

significant source of errors if kept at values $\sim 10 \rightarrow 100$. However, discreteness and periodicity introduces aliasing; a major difference to conventional QFT and only avoided at very low momenta[1]: $p_k << N\Delta p$.

- **Particle number cut-off**: According to Figure 4.8, accuracy improves with number of particles considered, but less so after a certain number ($P_{max} \sim 3$). Some interactions require a minimum of particles considered to contribute, but this is usually low. Larger interaction couplings require higher particle numbers to be considered, implying that for this assumption to hold interactions must be comparatively weak.

The theory developed accomplishes what it was designed for, but only provided it operates within the boundaries specified above.

5.2 Implementation Accuracy and Performance

5.2.1 Integrators

From the comparison of integrators, we find that choice of integrator depends on the goal of the calculation. For high accuracy, few steps with a high order non-symplectic integrator is by far the fastest, as seen in Figure 4.4. This applies to numerical virtual experiments where accuracy is important. Conversely, for long integration times where accuracy is not prioritised, as for the graphical user interface, a second order integrator performs better due to its symplecticity.

5.2.2 Performance

Several crucial optimisations were made, including avoidance of interaction Hamiltonian zero-elements and application of momentum conservation. This reduced the number of calculations sufficiently to allow interesting multi-particle phenomena to be observed, as in Figure 4.12. Unfortunately, the performance decays exponentially with particle number, a severely limiting factor of this simulation method. However, some progress can be made by parallelising the calculation of interaction Hamiltonians and supplying more computing power, as well as pre-rendering simulations rather than displaying them realtime.

[1] Or equivalently at high N.

5.2.3 Qualities as a Visualisation and Teaching Tool

The end goal of the research was to develop a tool for visualising QFT. This was achieved to a satisfactory degree[2] through the successful creation of the interactive graphical user interface. It is the author's sincere hope that the tools developed will be used by many for the purposes of understanding QFT.

5.3 Possible Further Investigations

Interesting extensions building on this research includes:

- **Mixed fields**: Including complex fields (two-component scalar) and scalar-fermion interaction (Higgs/Yukawa-type) in an attempt to visualise the Higgs field and -mechanism. This could follow a generalisation of the approach used to implement two-component fermions.
- **Gauge fields**: By applying a similar approach as to scalars and fermions, gauge fields could be simulated. Optimisations might include an appropriate choice of gauge. If successful, this can be coupled to fermions to demonstrate QED.

[2] Based on feedback from a selection of students.

Chapter 6

Conclusion

In conclusion, a 1D discretised quantum field theory was successfully developed and implemented computationally as an interactive graphical user interface. This was accomplished by rederiving key QFT equations for both scalars and fermions, as well as Schrödinger equation integrators for numerical integration. To ensure high performance, several optimisations were found and implemented, including avoiding calculations of zero-elements in interaction Hamiltonians and applying momentum conservation. This was implemented in Java code and tested extensively both by qualitative means through virtual experiments in the graphical user interface, as well as quantitatively through integrator error analyses and a numerical-analytical comparison in interacting theory. This comparison, of measured and predicted effective masses in interactive ϕ^2-theory using an iterative eigenvalue method, effectively verified the implementation.

Although severely limited by computational power, the graphical user interface provides a simple and intuitive visualisation tool for QFT, presenting an alternative view to the conventional Feynman diagrams. There are several possible extensions to this research, perhaps most intriguingly visualising the Higgs mechanism as it actually appears.

Appendix A

Further Proofs and Derivations

Appendices are considered supplemental to the report and are not offered for examination as part of the Part III project. They are aimed at future students extending this work and other readers interested in a greater level of detail.

This appendix contains proofs and derivations which are not considered crucial, but which have worked through during research and might prove helpful to the reader.

A.1 Discretised Energy Periodicity Invariance

Momentum periodicity implies:

$$p_k \to p_{k+N} = p_k + N\Delta p$$

Subsituting this into the momentum mode energy:

$$\begin{aligned}
\omega_k &\to \sqrt{m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{(p_k + N\Delta p)\Delta x}{2}\right)} \\
&\to \sqrt{m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k\Delta x}{2} + \pi\right)} \\
&\to \sqrt{m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k\Delta x}{2}\right)} = \omega_k
\end{aligned}$$

It is indeed invariant

A.2 Discretised Energy Continuous Limit

In the limit that the lattice spacing tends to zero, the momentum mode energy becomes:

$$\begin{aligned}\lim_{\Delta x\to 0}\omega_k &\to \sqrt{m^2+\frac{4}{\Delta x^2}\left(\frac{p_k\Delta x}{2}-\frac{1}{3!}\left(\frac{p_k\Delta x}{2}\right)^3+...\right)^2}\\ &\to \sqrt{m^2+\frac{4}{\Delta x^2}\left(\frac{p_k\Delta x}{2}\right)^2}\\ &\to \sqrt{m^2+p_k^2}\end{aligned}$$

which is the correct continuous limit.

A.3 Ladder Operator Commutation Relations

Assuming the ladder operator commutation relation, Eqn (2.18):

$$\begin{aligned}[\hat{\phi}_n,\hat{\pi}_m] &= \sum_{k,l=0}\frac{-i\omega_l\Delta p^2}{2\pi 2\omega_k 2\pi 2\omega_l}\left([\hat{a}_k,\hat{a}_l]e^{\cdots}+[\hat{a}_k^\dagger,\hat{a}_l]e^{ip_kx_n-ip_lx_m}-[\hat{a}_k,\hat{a}_l^\dagger]e^{-ip_kx_n+ip_lx_m}-[\hat{a}_k^\dagger,\hat{a}_l^\dagger]e^{\cdots}\right)\\ &= \sum_{k,l=0}\frac{-i\omega_l}{L^2 2\omega_k 2\omega_l}\left(-2L\omega_k\delta_{kl}e^{ip_kx_n-ip_lx_m}-2L\omega_k\delta_{kl}e^{-ip_kx_n+ip_lx_m}\right)\\ &= i\sum_{k,l=0}\frac{\omega_l}{2L}\left(e^{ip_k(x_n-x_m)}+e^{-ip_k(x_n-x_m)}\right)\\ &= i\frac{2N\delta_{nm}}{2L}=i\frac{\delta_{nm}}{\Delta x}\end{aligned}$$

which is consistent.

A.4 Free Hamiltonian in terms of Ladder Operators

Expand Hamiltonian:

$$\begin{aligned}\hat{H}_0 &= \Delta x\sum_{n=0}^{N-1}\hat{\mathcal{H}}_n\\ &= \Delta x\sum_{n=0}^{N-1}\frac{1}{2}\left(\hat{\pi}_n^2+\left(\frac{\hat{\phi}_{n+\frac{1}{2}}-\hat{\phi}_{n-\frac{1}{2}}}{\Delta x}\right)^2+m^2\hat{\phi}_n^2\right)\\ &= \frac{\Delta x}{2}\sum_{n=0}^{N-1}\hat{\pi}_n^2+\frac{1}{2\Delta x}\sum_{n=0}^{N-1}\left(\hat{\phi}_{n+\frac{1}{2}}-\hat{\phi}_{n-\frac{1}{2}}\right)^2+\frac{\Delta x m^2}{2}\sum_{n=0}^{N-1}\hat{\phi}_n^2\end{aligned}$$

Make all terms into a momentum sum:

$$\begin{aligned}
\sum_{n=0}^{N-1}\hat{\pi}_n^2 &= \frac{1}{L^2}\sum_{n,k,l=0}^{N-1}\left(\frac{-i}{2}\right)^2\left(\hat{a}_k e^{ip_k x_n}-\hat{a}_k^\dagger e^{-ip_k x_n}\right)\left(\hat{a}_l e^{ip_l x_n}-\hat{a}_l^\dagger e^{-ip_l x_n}\right)\\
&= -\frac{1}{4L^2}\sum_{k,l=0}^{N-1}\left(\hat{a}_k\hat{a}_l N\delta_{-k,l}-\hat{a}_k^\dagger\hat{a}_l N\delta_{kl}-\hat{a}_k\hat{a}_l^\dagger N\delta_{kl}+\hat{a}_k^\dagger\hat{a}_l^\dagger N\delta_{-k,l}\right)\\
&= -\frac{1}{4L\Delta x}\sum_{k=0}^{N-1}\left(\hat{a}_k\hat{a}_{-k}-\hat{a}_k^\dagger\hat{a}_k-\hat{a}_k\hat{a}_k^\dagger+\hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right)
\end{aligned}$$

$$\begin{aligned}
\sum_{n=0}^{N-1}\hat{\phi}_n^2 &= \frac{1}{L^2}\sum_{n,k,l=0}^{N-1}\frac{1}{4\omega_k\omega_l}\left(\hat{a}_k e^{ip_k x_n}+\hat{a}_k^\dagger e^{-ip_k x_n}\right)\left(\hat{a}_l e^{ip_l x_n}+\hat{a}_l^\dagger e^{-ip_l x_n}\right)\\
&= \frac{1}{L^2}\sum_{k,l=0}^{N-1}\frac{1}{4\omega_k\omega_l}\left(\hat{a}_k\hat{a}_l N\delta_{-k,l}+\hat{a}_k^\dagger\hat{a}_l N\delta_{kl}+\hat{a}_k\hat{a}_l^\dagger N\delta_{kl}+\hat{a}_k^\dagger\hat{a}_l^\dagger N\delta_{-k,l}\right)\\
&= \frac{1}{L\Delta x}\sum_{k=0}^{N-1}\frac{1}{4\omega_k^2}\left(\hat{a}_k\hat{a}_{-k}+\hat{a}_k^\dagger\hat{a}_k+\hat{a}_k\hat{a}_k^\dagger+\hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right)
\end{aligned}$$

$$\begin{aligned}
\sum_{n=0}^{N-1}\left(\hat{\phi}_{n+\frac{1}{2}}-\hat{\phi}_{n-\frac{1}{2}}\right)^2 &= \frac{1}{L^2}\sum_{n,k,l=0}^{N-1}\frac{1}{4\omega_k\omega_l}(\hat{a}_k e^{ip_k x_n}(e^{\frac{ip_k\Delta x}{2}}-e^{\frac{-ip_k\Delta x}{2}})\\
&\quad+\hat{a}_k^\dagger e^{-ip_k x_n}(e^{\frac{-ip_k\Delta x}{2}}-e^{\frac{ip_k\Delta x}{2}}))(\hat{a}_l e^{ip_l x_n}(e^{\frac{ip_l\Delta x}{2}}-e^{\frac{-ip_l\Delta x}{2}})\\
&\quad+\hat{a}_l^\dagger e^{-ip_l x_n}(e^{\frac{-ip_l\Delta x}{2}}-e^{\frac{ip_l\Delta x}{2}}))\\
&= \frac{1}{L^2}\sum_{n,k,l=0}^{N-1}\frac{1}{4\omega_k\omega_l}\left(2i\sin\left(\frac{p_k\Delta x}{2}\right)\hat{a}_k e^{ip_k x_n}-2i\sin\left(\frac{p_k\Delta x}{2}\right)\hat{a}_k^\dagger e^{-ip_k x_n}\right)\\
&\quad\cdot\left(2i\sin\left(\frac{p_l\Delta x}{2}\right)\hat{a}_l e^{ip_l x_n}-2i\sin\left(\frac{p_l\Delta x}{2}\right)\hat{a}_l^\dagger e^{-ip_l x_n}\right)\\
&= \frac{1}{L\Delta x}\sum_{k=0}^{N-1}\frac{\sin^2\left(\frac{p_k\Delta x}{2}\right)}{\omega_k^2}\left(\hat{a}_k\hat{a}_{-k}+\hat{a}_k^\dagger\hat{a}_k+\hat{a}_k\hat{a}_k^\dagger+\hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right)
\end{aligned}$$

Substitute all terms into Hamiltonian again to find it in terms of ladder operators:

$$\begin{aligned}
\hat{H}_0 &= -\frac{\Delta x}{2}\frac{1}{4L\Delta x}\sum_{k=0}^{N-1}\left(\hat{a}_k\hat{a}_{-k} - \hat{a}_k^\dagger\hat{a}_k - \hat{a}_k\hat{a}_k^\dagger + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right) \\
&\quad + \frac{1}{2\Delta x}\frac{1}{L\Delta x}\sum_{k=0}^{N-1}\frac{\sin^2\left(\frac{p_k\Delta x}{2}\right)}{\omega_k\omega_l}\left(\hat{a}_k\hat{a}_{-k} + \hat{a}_k^\dagger\hat{a}_k + \hat{a}_k\hat{a}_k^\dagger + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right) \\
&\quad + \frac{\Delta x m^2}{2}\frac{1}{L\Delta x}\sum_{k=0}^{N-1}\frac{1}{4\omega_k\omega_l}\left(\hat{a}_k\hat{a}_{-k} + \hat{a}_k^\dagger\hat{a}_k + \hat{a}_k\hat{a}_k^\dagger + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right) \\
&= \frac{1}{2L}\sum_{k=0}^{N-1}\frac{1}{4\omega_k^2}(\left(\omega_k^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k\Delta x}{2}\right) + m^2\right)\left(\hat{a}_k^\dagger\hat{a}_k + \hat{a}_k\hat{a}_k^\dagger\right) \\
&\quad + \left(-\omega_k^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{p_k\Delta x}{2}\right) + m^2\right)\left(\hat{a}_k\hat{a}_{-k} + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right)) \\
&= \frac{1}{2L}\sum_{k=0}^{N-1}\frac{1}{4\omega_k^2}2\omega_k^2\left(\hat{a}_k^\dagger\hat{a}_k + \hat{a}_k\hat{a}_k^\dagger\right) = \frac{1}{4L}\sum_{k=0}^{N-1}\left(2\hat{a}_k^\dagger\hat{a}_k + 2\omega_k L\right) \\
&= \sum_{k=0}^{N-1}\frac{\Delta p}{2\pi 2\omega_k}\omega_k\hat{a}_k^\dagger\hat{a}_k + E_{vac} \quad , \quad E_{vac} = \sum_{k=0}^{N-1}\frac{\omega_k}{2}
\end{aligned}$$

A.5 Fock State Energies

The commutator or the ladder operator the Hamiltonian is:

$$[\hat{H}_0, \hat{a}_k^\dagger] = \omega_k\hat{a}_k^\dagger \quad \rightarrow \quad \hat{H}_0\hat{a}_k^\dagger = \hat{a}_k^\dagger(\omega_k + \hat{H}_0)$$

The energy becomes:

$$\begin{aligned}
E_i = \langle i|\hat{H}_0|i\rangle &= \langle i|\hat{H}_0\prod_{k=0}^{N-1}\frac{\left(\hat{a}_k^\dagger\right)^{l_k}}{\sqrt{(2\omega_k L)^{l_k}l_k!}}|0\rangle \\
&= \langle i|\left(\prod_{k=0}^{N-1}\frac{1}{\sqrt{(2\omega_k L)^{l_k}l_k!}}\right)\hat{a}_0^\dagger(\omega_0 + \hat{H}_0)(\hat{a}_0^\dagger)^{l_0-1}(\hat{a}_1^\dagger)^{l_1}\ldots|0\rangle \\
&\quad \ldots[repeat\ for\ all\ \hat{a}_0^\dagger]\ldots \\
&= \langle i|\left(\prod_{k=0}^{N-1}\frac{1}{\sqrt{(2\omega_k L)^{l_k}l_k!}}\right)(\hat{a}_0^\dagger)^{l_0}(l_0\omega_0 + \hat{H}_0)(\hat{a}_1^\dagger)^{l_1}\ldots|0\rangle \\
&\quad \ldots[repeat\ for\ all\ \hat{a}_k^\dagger]\ldots \\
&= \langle i|\left(\prod_{k=0}^{N-1}\frac{1}{\sqrt{(2\omega_k L)^{l_k}l_k!}}\right)(\hat{a}_0^\dagger)^{l_0}\prod_{k=0}^{N-1}(\hat{a}_k^\dagger)^{l_k}(\sum_k^{N-1} l_k\omega_k + E_{vac})|0\rangle \\
&= \langle i|i\rangle(\sum_k^{N-1} l_k\omega_k + E_{vac}) = \sum_k^{N-1} l_k\omega_k + E_{vac}
\end{aligned}$$

A.6 ϕ^3 Interaction Term in Symmetrised Form

The entries in h_{kl}, as seen in Eqn (2.44) are split into 4 parts, defined by $k \neq l \neq 0$ to give:

$$\hat{h}_{00} = \frac{1}{8L^2m^3}\left((\hat{a}_0)^3 + \left(\hat{a}_0^\dagger\right)^3 + 3\hat{a}_0^\dagger(\hat{a}_0)^2 + 3\left(\hat{a}_0^\dagger\right)^2\hat{a}_0\right) + \frac{1}{4Lm^2}\left(3\hat{a}_0 + 3\hat{a}_0^\dagger\right)$$

$$\begin{aligned}\hat{h}_{k0} &= \frac{1}{8L^2m\omega_k^2}\left(\hat{a}_0\hat{a}_k\hat{a}_{-k} + \hat{a}_0^\dagger\hat{a}_k\hat{a}_{-k} + 2\hat{a}_k^\dagger\hat{a}_0\hat{a}_k + 2\hat{a}_k^\dagger\hat{a}_0^\dagger\hat{a}_k + \hat{a}_k^\dagger\hat{a}_{-k}^\dagger\hat{a}_0 + \hat{a}_0^\dagger\hat{a}_k^\dagger\hat{a}_{-k}^\dagger\right)\\ &\quad + \frac{1}{4L\omega_k^2}\left(\hat{a}_0 + \hat{a}_0^\dagger\right)\end{aligned}$$

$$\begin{aligned}\hat{h}_{kk} &= \frac{1}{8L^2\omega_{2k}\omega_k^2}\left((\hat{a}_k)^2\hat{a}_{-2k} + \hat{a}_{2k}^\dagger(\hat{a}_k)^2 + \left(\hat{a}_k^\dagger\right)^2\hat{a}_{2k} + \left(\hat{a}_k^\dagger\right)^2\hat{a}_{-2k}^\dagger\right)\\ &\quad + \frac{1}{8L^2m\omega_k^2}\left(2\hat{a}_k^\dagger\hat{a}_0\hat{a}_k + 2\hat{a}_k^\dagger\hat{a}_0^\dagger\hat{a}_k\right) + \frac{1}{4L\omega_k^2}\left(\hat{a}_0 + \hat{a}_0^\dagger\right)\end{aligned}$$

$$\begin{aligned}\hat{h}_{kl} &= \frac{1}{8L^2\omega_k\omega_l\omega_{(k+l)}}\left(\hat{a}_k\hat{a}_l\hat{a}_{(-k-l)} + \hat{a}_{(k+l)}^\dagger\hat{a}_l\hat{a}_k + \hat{a}_k^\dagger\hat{a}_l^\dagger\hat{a}_{(k+l)} + \hat{a}_k^\dagger\hat{a}_l^\dagger\hat{a}_{(-k-l)}^\dagger\right)\\ &\quad + \frac{1}{8L^2\omega_k\omega_l\omega_{(k-l)}}\left(\hat{a}_k^\dagger\hat{a}_l\hat{a}_{(-k+l)}^\dagger + \hat{a}_k^\dagger\hat{a}_l\hat{a}_{(k-l)} + \hat{a}_l^\dagger\hat{a}_{(k-l)}^\dagger\hat{a}_k + \hat{a}_k^\dagger\hat{a}_{(-k+l)}^\dagger\hat{a}_l\right)\end{aligned}$$

A.7 Ladder Operator Sandwich Identity

Assume:

$$\langle 0|\hat{a}^n(\hat{a}^\dagger)^n|0\rangle = (2EL)^n n!$$

Induction step using the commutator, Eqn (2.18):

$$\begin{aligned}\langle 0|\hat{a}^{n+1}(\hat{a}^\dagger)^{n+1}|0\rangle &= \langle 0|\hat{a}^n(\hat{a}^\dagger\hat{a} + 2EL)(\hat{a}^\dagger)^n|0\rangle\\ &= \langle 0|\hat{a}^n\hat{a}^\dagger\hat{a}(\hat{a}^\dagger)^n|0\rangle + 2EL(2EL)^n n!\\ &= \langle 0|\hat{a}^n\hat{a}^\dagger(\hat{a}^\dagger\hat{a} + 2EL)(\hat{a}^\dagger)^{n-1}|0\rangle + 2EL(2EL)^n n!\\ &= \langle 0|\hat{a}^n(\hat{a}^\dagger)^2\hat{a}(\hat{a}^\dagger)^{n-1}|0\rangle + 2\times(2EL)^{n+1}n!\\ &\quad \ldots[(n+1)\ times]\ldots\\ &= \langle 0|\hat{a}^n(\hat{a}^\dagger)^{n+1}\hat{a}|0\rangle + (n+1)\times(2EL)^{n+1}n!\\ &= (2EL)^{n+1}(n+1)!\end{aligned}$$

Since by definition $\langle 0|0\rangle = 1 = 0!$, this is proved by induction to hold for any $n \geq 0$.

A.8 ξ^μ Anticommutator Relation

The anticommutators of the ξ^μ-matrices are:

$$\begin{aligned}
\{\xi^0,\xi^0\} &= 2\begin{pmatrix}1&0\\0&-1\end{pmatrix}\begin{pmatrix}1&0\\0&-1\end{pmatrix} = 2I = 2g^{00}I \\
\{\xi^0,\xi^1\} &= \begin{pmatrix}1&0\\0&-1\end{pmatrix}\begin{pmatrix}0&1\\-1&0\end{pmatrix}+\begin{pmatrix}0&1\\-1&0\end{pmatrix}\begin{pmatrix}1&0\\0&-1\end{pmatrix} \\
&= \begin{pmatrix}0&1\\1&0\end{pmatrix}+\begin{pmatrix}0&-1\\-1&0\end{pmatrix} = 0 = 2g^{01}I \\
\{\xi^1,\xi^0\} &= \{\xi^0,\xi^1\} = 0 = 2g^{10}I \\
\{\xi^1,\xi^1\} &= 2\begin{pmatrix}0&-1\\1&0\end{pmatrix}\begin{pmatrix}0&-1\\1&0\end{pmatrix} = -2I = 2g^{11}I
\end{aligned}$$

Hence:

$$\{\xi^\mu,\xi^\nu\} = 2g^{\mu\nu}I$$

A.9 Fermion Momentum Mode Energies

Starting with the Dirac equation:

$$\begin{aligned}
i\xi^0\dot{\psi}_n + i\xi^1\Delta_x\psi_n - m\psi_n &= 0 \\
i\xi^0(\pm i\omega_k)\psi_n + i\frac{e^{\frac{ip_k\Delta x}{2}} - e^{-\frac{ip_k\Delta x}{2}}}{\Delta x}\xi^1\psi_n - m\psi_n &= 0 \\
\left(\mp\omega_k\xi^0 - \frac{2}{\Delta x}\sin\left(\frac{ip_k\Delta x}{2}\right)\xi^1 - m\right)\psi_n &= 0 \\
\begin{pmatrix}\mp\omega_k - m & -\frac{2}{\Delta x}\sin\left(\frac{ip_k\Delta x}{2}\right)\\ \frac{2}{\Delta x}\sin\left(\frac{ip_k\Delta x}{2}\right) & \pm\omega_k - m\end{pmatrix}\psi_n &= 0
\end{aligned}$$

but since $\psi_n \neq 0$:

$$\begin{aligned}
\begin{vmatrix}\mp\omega_k - m & -\frac{2}{\Delta x}\sin\left(\frac{ip_k\Delta x}{2}\right)\\ \frac{2}{\Delta x}\sin\left(\frac{ip_k\Delta x}{2}\right) & \pm\omega_k - m\end{vmatrix} &= 0 \\
(\mp\omega_k - m)(\pm\omega_k - m) + \frac{4}{\Delta x^2}\sin^2\left(\frac{ip_k\Delta x}{2}\right) &= 0 \\
\omega_k^2 = m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{ip_k\Delta x}{2}\right) & \\
\omega_k = \sqrt{m^2 + \frac{4}{\Delta x^2}\sin^2\left(\frac{ip_k\Delta x}{2}\right)} &
\end{aligned}$$

A.10 $u(p_k)$ and $v(p_k)$ as Eigenspinors of the Hamiltonian

By writing:

$$\rho_k = \frac{2}{\Delta x}\sin\left(\frac{ip_k\Delta x}{2}\right)$$

we apply the Hamiltonian matrix to $u(p_k)$ and $v(p_k)$ ($N = \sqrt{\omega_k + m}$):

$$\begin{aligned} Hu(p_k)e^{-ip_kx_n} &= \begin{pmatrix} m & -i\Delta_x \\ -i\Delta_x & -m \end{pmatrix} N \begin{pmatrix} 1 \\ \frac{-\rho_k}{\omega_k+m} \end{pmatrix} e^{-ip_kx_n} = \begin{pmatrix} m & -\rho_k \\ -\rho_k & -m \end{pmatrix} N \begin{pmatrix} 1 \\ \frac{-\rho_k}{\omega_k+m} \end{pmatrix} e^{-ip_kx_n} \\ &= N \begin{pmatrix} m - \frac{\rho_k^2}{\omega_k+m} \\ -\rho_k - \frac{m\rho_k}{\omega_k+m} \end{pmatrix} e^{-ip_kx_n} = N\omega_k \begin{pmatrix} 1 \\ \frac{-\rho_k}{\omega_k+m} \end{pmatrix} e^{-ip_kx_n} = \omega_k u(p_k)e^{-ip_kx_n} \end{aligned}$$

$$\begin{aligned} Hv(p_k)e^{ip_kx_n} &= \begin{pmatrix} m & -i\Delta_x \\ -i\Delta_x & -m \end{pmatrix} N \begin{pmatrix} \frac{\rho_k}{\omega_k+m} \\ -1 \end{pmatrix} e^{ip_kx_n} = \begin{pmatrix} m & \rho_k \\ \rho_k & -m \end{pmatrix} N \begin{pmatrix} \frac{\rho_k}{\omega_k+m} \\ -1 \end{pmatrix} e^{ip_kx_n} \\ &= N \begin{pmatrix} \frac{m\rho_k}{\omega_k+m} - \rho_k \\ m + \frac{\rho_k^2}{\omega_k+m} \end{pmatrix} e^{ip_kx_n} = -N\omega_k \begin{pmatrix} \frac{\rho_k}{\omega_k+m} \\ -1 \end{pmatrix} e^{ip_kx_n} = -\omega_k v(p_k)e^{ip_kx_n} \end{aligned}$$

We have shown that the Hamiltonian matrix has to eigenspinors $u(p_k)$ and $v(p_k)$ with eigenvalues ω_k and $-\omega_k$, respectively.

A.11 Dimensional Analysis

Starting with the dimensionless action, $[\hat{S}] = 0$, from its definition[1] we get:

$$[\hat{\mathcal{L}}_n] = [\hat{\mathcal{H}}_n] = 2 \tag{A.1}$$

having used $[\Delta x] = [dt] = -1$.

From the Lagrangian mass term $\propto m^2\hat{\phi}_n^2$ and $[m] = 1$ we obtain:

$$[\hat{\phi}_n] = 0 \tag{A.2}$$

which implies that interaction couplings $\lambda^{(n)}$ satisfy $[\lambda^{(n)}] = 2$ for any n. This has the interesting consequence that any field self-interaction will be renormalisable.

[1] Definition of 1D discretised action: $\hat{S} = \int \sum_n \hat{\mathcal{L}}(x_n, t)dt\Delta x$

Considering the Fourier expansions of $\hat{\phi}_n$ and $\hat{\pi}_n$, Eqns (2.16) and (2.17), and remembering $[\Delta p] = [\omega_k] = 1$, gives us:

$$[\hat{a}_k] = [\hat{a}_k^\dagger] = 0 \tag{A.3}$$

$$[\hat{\pi}_n] = 1 \tag{A.4}$$

where ladder operators are dimensionless due to our choice of relativistic normalisation (would be $\frac{1}{2}$ using non-relativistic normalisation).

The fermionic field also loses one mass dimension. This follows from the Dirac mass term $\propto m\hat{\psi}_n^\dagger \xi^0 \hat{\psi}_n$ in the Lagrangian, since ξ^μ are dimensionless:

$$[\hat{\psi}_n] = \frac{1}{2} \tag{A.5}$$

From the anticommutator, Eqn (2.76), and the Fourier expansion, Eqn (2.76), we obtain:

$$[\hat{c}_k] = [\hat{c}_k^\dagger] = [\hat{b}_k] = [\hat{b}_k^\dagger] = 0 \tag{A.6}$$

$$[u(p_k)] = [v(p_k)] = \frac{1}{2} \tag{A.7}$$

where Eqn (A.7) is consistent with the definition, Eqns (2.72) and (2.73).

Appendix B

Code

Appendices are considered supplemental to the report and are not offered for examination as part of the Part III project. They are aimed at future students extending this work and other readers interested in a greater level of detail.

The code base is too extensive to include in an appendix ($\sim$ 7000 lines of code across $\sim$ 125 classes). All code, including version history, is available online at:

`https://github.com/carlandreaslindstrom/QFT-Simulations/tree/master/QFT%20Simulations/src/uk/ac/cam/cal56`

B.1 Overall Code Structure

Disclaimer: The code base was a joint code project with Charlie Bridge, who was working on a similar Part III project. The graphics and maths packages were shared, whereas the author was mainly responsible for the code development.

Encapsulation, loose coupling and code reusability was important in developing the overall structure. This implies a large package tree with many small loosely coupled classes. The conceptual similarities of the scalar and fermion theories necessitated a heavy reliance on inheritance and interfacing.

The code base[1] was divided into three main packages:

- **Mathematics** - mathematical utilities

[1] All code available online: `https://github.com/carlandreaslindstrom/QFT-Simulations/tree/master/QFT%20Simulations/src/uk/ac/cam/cal56`

- **QFT** - all quantum field theory content
- **Graphics** - graphical user interface

where each package depends on those above it. The overall structure is shown as a class diagram in Figure B.1.

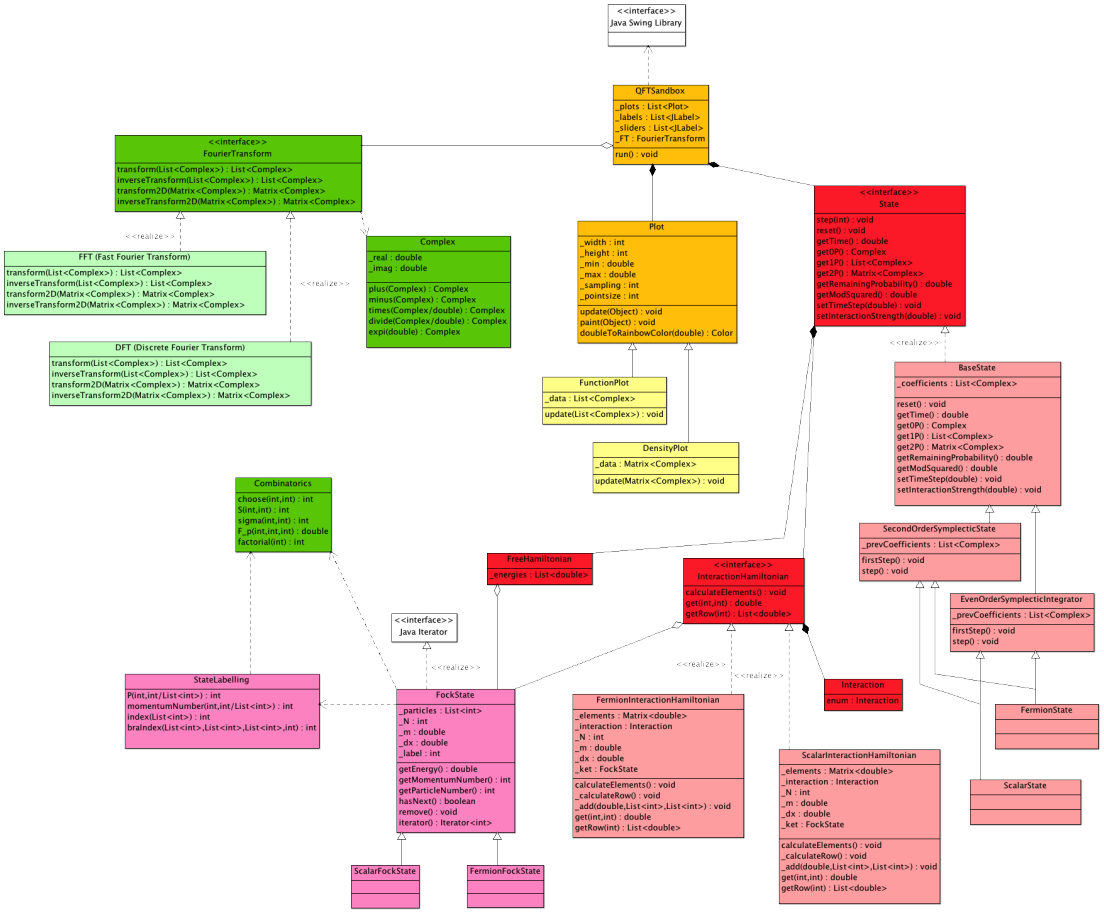

FIGURE B.1: **Class diagram of code base**: The most important classes made are included in this diagram. The three packages are indicated by colours green (mathematics), red (Quantum Field Theory) and yellow (graphics). Every class has one or more unit test classes.

B.1.1 The Mathematics Package

Java has no complex number standard. Although many implementations exist, a complex number class was developed, optimised specifically for QFT. A Fast Fourier Transform (FFT) was implemented using these complex numbers. Additionally, a combinatorics class was implemented, specifically optimised for Eqns (3.1)-(3.4).

B.1.2 The Quantum Field Theory Package

The quantum state is the key class in the QFT-package. All other classes exist to facilitate its time evolution. The state has a big inheritance tree structure, with the common features described in the interface State. The graphics package refers to this interface only, making it possible to use different QFT-packages so long as they all implement this interface. State is then implemented by an abstract BaseState, which has all common functions of scalars and fermions, apart from the integrator functionality. This is encapsulated in the two integrator-states, one integrating by a second order method, the other via the arbitrary even-order method, both extending BaseState. The fermion and scalar states then extend these again. These have all the functions specific to fermions and scalars, and this way we can easy add, say, a gauge field state.

Each State is simulated in a theory. This information is encapsulated in the InteractionHamiltonian class, of which the state has many. These represent interaction terms in the Hamiltonian, without the interaction coupling, which is contained in the State to allow realtime weighting of theories. This InteractionHamiltonians are different for scalars and bosons, but share an interface for easy handling. Each InteractionHamiltonian represents an interaction, and this is represented in a enumeration type Interaction.

The States must start with some initial set of coefficients. This set of coefficients is defined through the WavePacket, which when given information about the shape and phases of a Gaussian wave packet returns a set of coefficients. This is what allows the user to define an initial state from clicking on plots in the graphical user interface.

Calculation of InteractionHamiltonians as well as stepping of the State itself relies on being able to cycle through all the Fock states. This is done via the FockState class, which is an Iterable class (i.e. you can use it in a for-loop: for(index : fockstate) ...;). It implements the stepping algorithms for scalars and fermions in two separate implementations of an interface.

For further details, please refer to the code itself.

B.1.3 The Graphics Package

The graphics package uses the Java Swing framework, including custom-made interactive complex number plots for representing quantum states. Further details are not relevant.

B.2 A Guide to the Graphical User Interface

The graphical user interface[2] presents the user with interactive control over a quantum state, time evolved by QFT. It is split into a display and a control panel, as shown in Figure B.2.

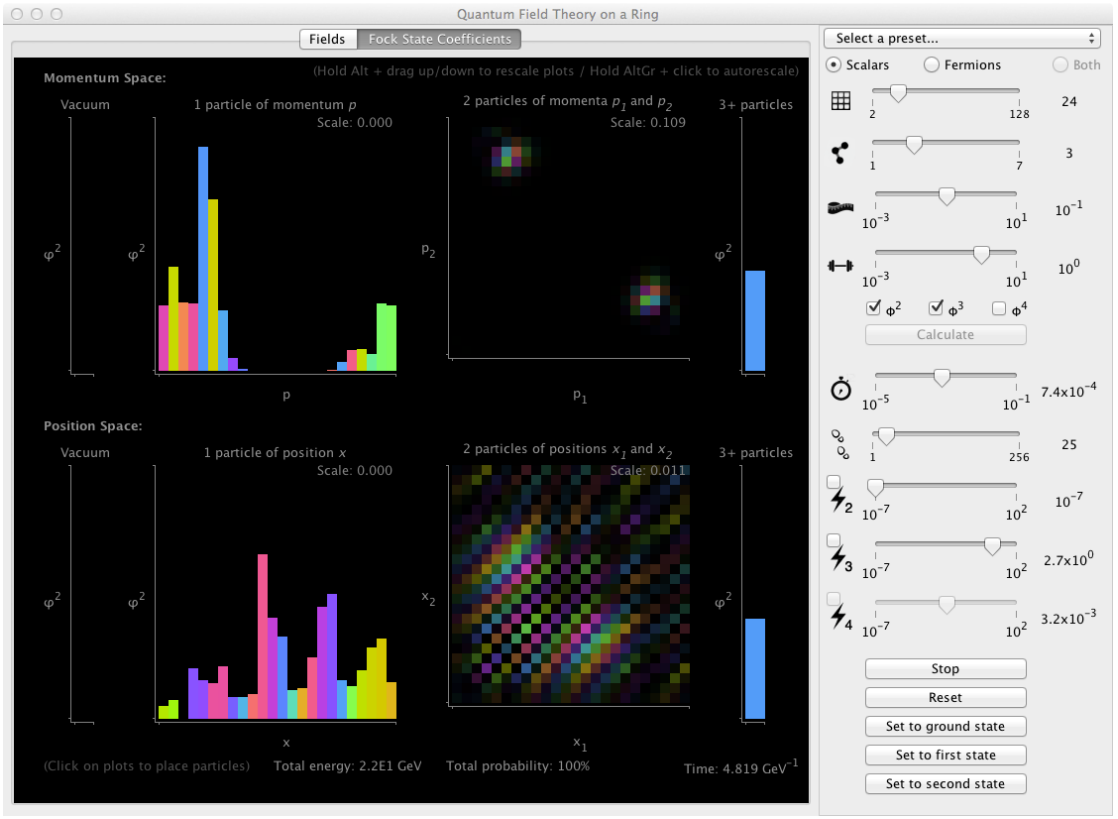

FIGURE B.2: **Graphical user interface window (coefficient view)**: This is the default layout and options, presented to the user at startup. The display (black background) has plots representing the quantum state and allows, in addition to the control panel (grey background), the user to control it.

B.2.1 The Control Panel

The control panel has two basic sections: a compile- and a realtime part. The compile part allows the user to choose scalars/fermions as well as parameters needed in the calculation of the Hamiltonian $(N, P_{max}, \Delta x, m, \lambda^{(2)}, \lambda^{(3)}, ...)$. The "Calculate" button will calculate the Hamiltonian (a one off calculation) and start simulating the state in a default wave packet.

The realtime part allows manipulation without recalculation of the Hamiltonian, and has instant effect on the state as it evolves. This includes the time step Δt used, the

[2]The GUI is (at time of writing) an active code project subject to frequent changes, please expect this guide to be slightly out of date.

number of steps between printouts (effectively speed) and all the interaction couplings $\lambda^{(n)}$. The interaction couplings have an option (checkbox) of being negative.

B.2.2 The Display

The display has two tab views: coefficient view and field view. The most instructive view is the coefficient view, which shows the probabilities of all Fock states (i.e. particles) as well as their complex phase. In function plots, height indicates probability, whereas in the 2D-density plots, probability is indicated by intensity. The upper half shows the momentum coefficients (what is actually calculated) and the lower half shows the position coefficients (found by a Fourier transform). Each space has several plots representing each particle number. The vacuum is represented by a single probability bar plot. The one-particle states of different momenta p are represented by a function plot. Two-particle states are shown in a density plot with axes representing the two momenta p_1 and p_2. Due to the complications of plotting 3D-density plots, all higher particle number states are added into a single probability plot.

Fermions have more plots, representing not only particles, but also antiparticles. It should be clear from context which are which.

The plots are interactive, allowing the user to place particle wave packets by clicking on the plots. Dragging while clicked allows placing the wave packet in the other space as well. This makes it easy to play and create intuition for what happens to "particles" in QFT.

The field view shows the average value of the field and its conjugate momentum, as shown in Figure B.3. These can be real numbers, negative or positive as indicated by the colour of the plot (see legend for sign convention).

Note on plots: Plots are designed to auto-range based on values, but can be overridden manually by Alt-clicking and dragging up/down[3].

All views have additional information about total energy, total probability (useful to tell when integrator fails) and time.

[3]Beware: this might be subject to changes.

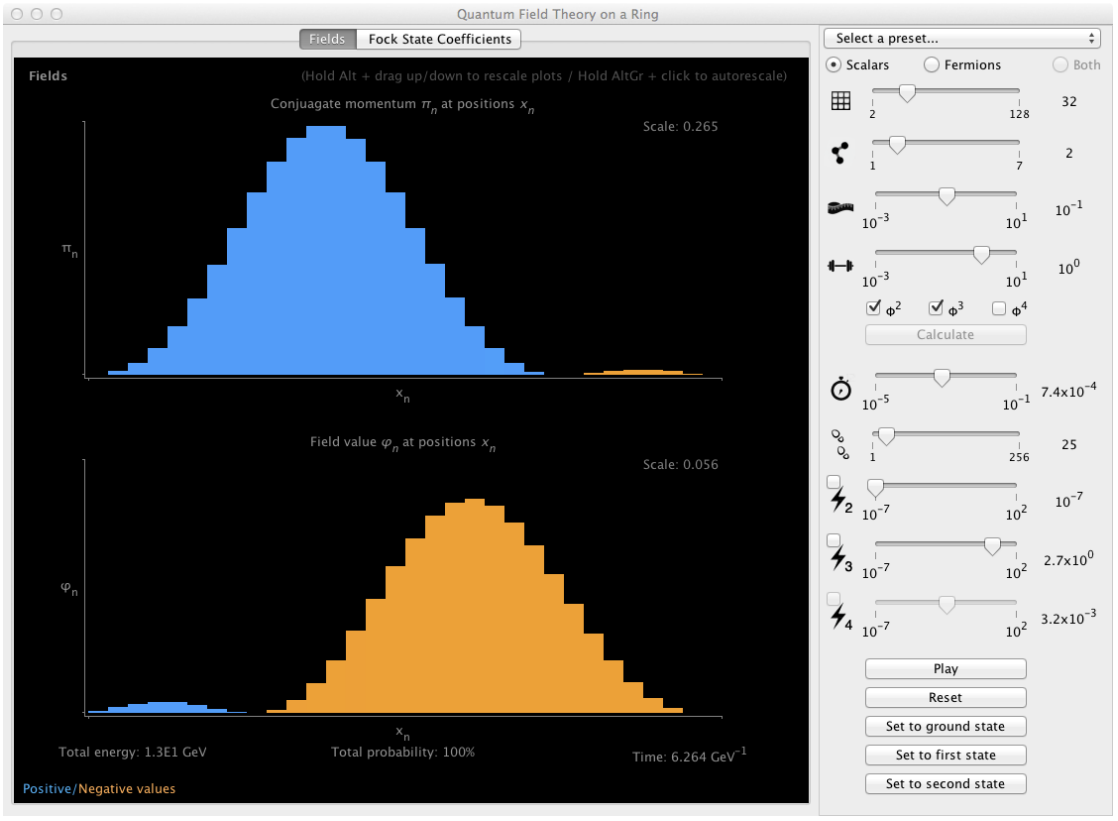

FIGURE B.3: **Graphical user interface window (field view)**: Shows the quantum state in field view, i.e. the value of the field and its conjugate momentum, fluctuating over time.

Bibliography

[1] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to Quantum Field Theory*. Westview Press, Boulder, Canada, 1995.

[2] David Bau and Lloyd N. Trefethen. *Numerical Linear Algebra*. SIAM, Philadelphia, USA, April 1997.

[3] The Computer Language Benchmarks Game. Speed comparison of Java 7 and C++ [accessed May 7, 2013], May 2013. URL `http://benchmarksgame.alioth.debian.org/u64q/java.php`.

[4] The Computer Language Benchmarks Game. Speed comparison of Java 7 and Python [accessed May 7, 2013], May 2013. URL `http://benchmarksgame.alioth.debian.org/u64q/python.php`.