

## QUANTUM LATTICE ALGORITHMS: SIMILARITIES AND CONNECTIONS TO SOME CLASSIC FINITE DIFFERENCE ALGORITHMS

PAUL J. DELLAR<sup>1</sup>

**Abstract.** Quantum lattice algorithms originated with the Feynman checkerboard model for the one-dimensional Dirac equation. They offer discrete models of quantum mechanics in which the complex numbers representing wavefunction values on a discrete spatial lattice evolve through discrete unitary operations. This paper draws together some of the identical, or at least unitarily equivalent, algorithms that have appeared in three largely disconnected strands of research. Treated as conventional numerical algorithms, they are all only first order accurate under refinement of the discrete space/time grid, but may be raised to second order by a unitary change of variables. Much more efficient implementations arise from replacing the evolution through a sequence of unitary intermediate steps with a short path integral formulation that expresses the wavefunction at each spatial point on the most recent time level as a linear combination of values at immediately preceding time levels and neighbouring spatial points. In one dimension, a particularly elegant reformulation replaces two variables at two time levels with a single variable over three time levels. The resulting algorithm is a variational integrator arising from a discrete action principle, and coincides with the Ablowitz–Kruskal–Ladik finite difference scheme for the Klein–Gordon equation.

### 1. INTRODUCTION

Quantum lattice algorithms originated with Feynman’s checkerboard model for the Dirac equation that describes a spin-1/2 particle moving relativistically on a discrete space-time lattice with one spatial dimension [1]. Heisenberg [2] had previously developed a spatially discrete “Gitterwelt” (lattice world) theory. Carazza & Kragh[3] give a fascinating reconstruction of this theory, which was communicated almost entirely in unpublished correspondence.

Subsequent work has followed three parallel lines of development, frequently leading to identical, or at least unitarily equivalent, algorithms under the different names of quantum cellular automata, discrete time quantum walks, and “quantum lattice Boltzmann” algorithms. The first two strands emphasise algorithms for quantum computers, notably  $\mathcal{O}(\sqrt{N})$  search algorithms. The last strand emphasises simulation of quantum phenomena using conventional digital computers by applying techniques used in the derivation of the hydrodynamic algorithms from the Boltzmann equation. They would be better described as “lattice Dirac algorithms” as they have no connection to the quantum Boltzmann equation. All strands offers unitary and readily parallelisable algorithms that are free of the fermion-doubling problem that are commonly found in finite difference or finite element discretisations of quantum mechanical equations. In discrete quantum systems the finite-time evolution operator replaces the Hamiltonian as the principle object of interest. Unitary evolution implies that the

---

<sup>1</sup> OCIAM, Mathematical Institute, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford OX2 6GG, UK

expectation of the evolution operator is itself conserved, offering an exact invariant of the discrete system that is consistent with the energy conservation property of the continuous system implied by Ehrenfest's formula [4].

A fourth line of development treats the Schrödinger equation and its nonlinear extensions as reaction-diffusion equations for complex wavefunctions [5–7], and adapts standard lattice Boltzmann techniques for reaction-diffusion equations [8]. The target partial differential equation then describes slowly varying solutions on some manifold within an enlarged state space for a dissipative system, just as in lattice Boltzmann formulations for hydrodynamics. These formulations are thus neither unitary nor reversible, and will not be considered further.

This paper is based on a tutorial presented at the Discrete Simulation of Fluid Dynamics conference, Yerevan, Armenia, 15–19 July 2013. It covers all three strands above, though devoting most attention to the development of second-order accurate quantum lattice algorithms for simulating the Dirac equation with a potential in multiple space dimensions. Such algorithms are finding new applications beyond quantum electrodynamics and laser-plasma interactions. The Dirac equation also offers a large-scale description of charge carriers within graphene, a two-dimensional hexagonal lattice of carbon atoms, with a vector potential arising from uneven strain within the lattice [9]. Other experimental replications of the Dirac equation use trapped ions in optical lattices. These analogous systems are bringing previously inaccessible phenomena from quantum electrodynamics such as Zitterbewegung and the Klein paradox within experimental and technological reach. A detailed description of a second-order one-dimensional algorithm, and early multi-dimensional algorithms, may be found in the thesis [10].

## 2. THE DIRAC AND SCHRÖDINGER EQUATIONS

The Dirac equation offers a quantum mechanical description of spin-1/2 particles such as electrons that is compatible with special relativity [11–14]. It may be written as the abstract Schrödinger equation

$$i \partial_t \psi = H \psi \quad (1)$$

in convenient “natural” units in which the speed of light  $c = 1$  and the reduced Planck's constant  $\hbar = 1$ . The wavefunction  $\psi = (\psi_1, \psi_2, \psi_3, \psi_4)^T$  has four components. The Hamiltonian is

$$H = (-i \boldsymbol{\alpha} \cdot \nabla + m\beta - g\mathbf{1}), \quad (2)$$

where  $m$  is the particle mass,  $g(\mathbf{x})$  is a time-independent scalar potential,  $\mathbf{1}$  is the  $4 \times 4$  identity matrix,  $\beta$  is a  $4 \times 4$  matrix, and  $\boldsymbol{\alpha}$  is a vector of three  $4 \times 4$  matrices such that  $\boldsymbol{\alpha} \cdot \nabla = \alpha^x \partial_x + \alpha^y \partial_y + \alpha^z \partial_z$ . These matrices may be written in block form as

$$\alpha^i = \begin{pmatrix} 0 & \sigma^i \\ \sigma^i & 0 \end{pmatrix}, \quad \beta = \begin{pmatrix} \mathbf{1} & 0 \\ 0 & -\mathbf{1} \end{pmatrix}, \quad (3)$$

in which  $\mathbf{1}$  is now the  $2 \times 2$  identity matrix. The off-diagonal blocks are the three Pauli spin matrices

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (4)$$

which satisfy the relations [12–15]

$$\sigma^i \sigma^j = \mathbf{1} \delta_{ij} + i \epsilon_{ijk} \sigma^k, \quad (5)$$

where  $\epsilon_{ijk}$  is the alternating Levi-Civita tensor.

This Hamiltonian arose from Dirac's efforts to construct an operator analogue of the relativistic energy-momentum relation  $E^2 = m^2 + |\mathbf{p}|^2$ , in  $c = 1$  units, in which  $\mathbf{p} = -i\nabla$  is the momentum operator. The space-time symmetry expected in special relativity suggests that the Hamiltonian should be a first-order spatial differential operator, to match the single  $\partial_t$  on the left hand side of (1), in contrast to the second-derivative Laplacian that appears in the non-relativistic Schrödinger equation (see below). Naively, one expects the Hamiltonian to be related to  $E = \pm \sqrt{m^2 + |\mathbf{p}|^2}$ . Dirac's introduction of a 4-component wavefunction and a

$4 \times 4$  matrix Hamiltonian let him construct such a square root that was linear in  $\mathbf{p}$ , *i.e.* using only first-order spatial derivatives.

These two constraints, achieving compatibility with  $E^2 = m^2 + |\mathbf{p}|^2$ , and using only first-order spatial derivatives, thus imply not only the existence of quantum mechanical spin, through the appearance of the Pauli spin matrices above, but also the existence of both positive and negative energy states – the positron as well as the electron – arising from the positive and negative  $2 \times 2$  blocks in the  $\beta$  matrix.

Dirac's construction just imposes a set of algebraic relations between the  $\alpha$  and  $\beta$  matrices,

$$\alpha^i \alpha^j + \alpha^j \alpha^i = 2\delta_{ij}, \quad \beta^2 = 1, \quad \alpha^i \beta + \beta \alpha^i = 0, \quad (6)$$

analogous to the relations (5) between the three  $\sigma$  matrices. Any set of matrices satisfying these relations yields a representation of the abstract Dirac algebra. The Dirac equation may thus be written in many apparently different, but ultimately equivalent, forms. The concrete expressions for  $\alpha$  and  $\beta$  given above, and the Hamiltonian (2), give what is commonly called the “standard representation” of the Dirac equation.

### 2.1. From the Dirac to the Schrödinger equation

The most convenient starting point for taking the non-relativistic limit is the Dirac equation for a free particle

$$(\partial_t + \boldsymbol{\alpha} \cdot \nabla) \begin{pmatrix} \psi_+ \\ \psi_- \end{pmatrix} = -im\beta \begin{pmatrix} \psi_+ \\ \psi_- \end{pmatrix} \quad (7)$$

in which  $\psi = (\psi_+, \psi_-)^\top$  has been split into two two-component pairs  $\psi_\pm$ . Equation (7) then separates into

$$\partial_t \psi_+ + \boldsymbol{\sigma} \cdot \nabla \psi_- = -im\psi_+, \quad (8a)$$

$$\partial_t \psi_- + \boldsymbol{\sigma} \cdot \nabla \psi_+ = im\psi_-. \quad (8b)$$

The crossing over of  $\psi_+$  and  $\psi_-$  in the  $\boldsymbol{\sigma} \cdot \nabla$  terms is due to the off-diagonal blocks being non-zero in the  $\alpha$  matrices. In the absence of spatial gradients, we find from (8a,b) that  $\psi_+$  oscillates in proportion to  $e^{-imt}$ , while  $\psi_-$  oscillates in proportion to  $e^{imt}$ .

Following Pauli [11] and Berestetskii *et al.* [13], we break the symmetry between  $\psi_+$  and  $\psi_-$  by seeking solutions whose time-dependence is close to oscillation in proportion to  $e^{-imt}$ . This is equivalent to shifting the energy origin from 0 to  $m$ . Writing

$$\psi_\pm(\mathbf{x}, t) = \phi_\pm(\mathbf{x}, t)e^{-imt} \quad (9)$$

and substituting into (8a,b) gives

$$\partial_t \phi_+ + \boldsymbol{\sigma} \cdot \nabla \phi_- = 0, \quad (10a)$$

$$\partial_t \phi_- + \boldsymbol{\sigma} \cdot \nabla \phi_+ = 2im\phi_-. \quad (10b)$$

The algebraic right hand side now only affects the  $\phi_-$  variable. By analogy with the derivation of hydrodynamics from moments of the Boltzmann equation, we refer to  $\phi_+$  as the slow, or hydrodynamic, variable, and  $\phi_-$  as the fast, or nonhydrodynamic, variable.

The slowly-varying approximation

$$|\partial_t \phi^-| \ll 2m |\phi^-|, \quad (11)$$

holds in the non-relativistic limit, which corresponds to the kinetic energy being much less than the rest energy. Making this approximation in (10b) allows us to solve (10b) for  $\phi_-$  in terms of the gradient of  $\phi_+$ ,

$$\phi^- = -\frac{1}{2m} \mathbf{i}\boldsymbol{\sigma} \cdot \nabla \phi^+. \quad (12)$$

Substituting this relation into the evolution equation (10a) for  $\phi_+$  leads to the one-dimensional Schrödinger equation for a free particle in natural units,

$$i\partial_t\phi^+ = -\frac{1}{2m}\nabla^2\phi^+, \quad (13)$$

since  $(\boldsymbol{\sigma}\cdot\nabla)(\boldsymbol{\sigma}\cdot\nabla) = \nabla^2$  using a property of the  $\boldsymbol{\sigma}$  matrices. As in the derivation of the Navier–Stokes equations from kinetic theory, the derivation of the Schrödinger equation follows from eliminating the fast variable  $\phi^-$  to obtain a closed evolution equation for the slow variable  $\phi^+$  alone [16, 17].

## 2.2. The Foldy–Wouthuysen transformation

Foldy & Wouthuysen developed a more elegant theoretical approach to the non-relativistic limit [12, 18, 19]. They found the exact unitary transformation

$$\psi' = e^{iS}\psi, \quad S = \beta\boldsymbol{\alpha}\cdot\mathbf{p}\theta(|\mathbf{p}|), \quad \tan(2|\mathbf{p}|\theta(|\mathbf{p}|)) = |\mathbf{p}|/m, \quad (14)$$

under which the Dirac Hamiltonian for a free particle becomes

$$H_{\text{NW}} = \beta\sqrt{m^2 + |\mathbf{p}|^2}. \quad (15)$$

This Hamiltonian is the natural operator version of the relation  $E = \sqrt{m^2 + |\mathbf{p}|^2}$  for a relativistic particle, in which  $\mathbf{p}$  is simply the momentum vector, whose components are numbers, rather than the momentum operator. The subscript NW indicates that  $H_{\text{NW}}$  is the Hamiltonian in the Newton–Wigner representation. This representation decouples the four components of the transformed wavefunction into two independent pairs:

$$i\partial_t \begin{pmatrix} \psi_{\text{NW}+} \\ \psi_{\text{NW}-} \end{pmatrix} = H_{\text{NW}} \begin{pmatrix} \psi_{\text{NW}+} \\ \psi_{\text{NW}-} \end{pmatrix} \implies i\partial_t\psi_{\text{NW}\pm} = \pm\sqrt{m^2 + |\mathbf{p}|^2}\psi_{\text{NW}\pm}. \quad (16)$$

For plane wave solutions with momentum  $\mathbf{p}$ , the pairs  $\psi_{\text{NW}\pm}$  thus correspond to energy and momentum eigenstates with energies  $\pm\sqrt{m^2 + |\mathbf{p}|^2}$  respectively. Approximating the Hamiltonian in the Newton–Wigner representation for solutions that vary slowly in space ( $|\mathbf{p}| \ll m$ ) gives

$$H_{\text{NW}} = \beta m \left( 1 + \frac{1}{2} \frac{|\mathbf{p}|^2}{m^2} + \dots \right). \quad (17)$$

Truncating after the  $|\mathbf{p}|^2/m^2$  term gives the usual Schrödinger operator  $|\mathbf{p}|^2/(2m)$  for each of the two blocks, with a superimposed constant potential  $m$  that creates a phase rotation proportional to  $\exp(\pm imt)$  in the wavefunctions  $\psi_{\text{NW}\pm}$ .

Moreover, the position operator  $\mathbf{x}_{\text{NW}}$  in the Newton–Wigner representation satisfies the classical relation for a free particle,

$$\frac{d}{dt}\mathbf{x}_{\text{NW}} = \mathbf{v}_{\text{NW}} \equiv \beta \frac{\mathbf{p}}{\sqrt{m^2 + |\mathbf{p}|^2}}. \quad (18)$$

This is in stark contrast to the velocity operator in the original Dirac–Pauli representation, whose eigenvalues are  $\pm c$  in SI units. This is responsible for the *zitterbewegung* or “trembling motion” of the expectation of the position operator in this representation.

However, the Foldy–Wouthuysen transformation sacrifices the locality of the Dirac Hamiltonian  $\mathbf{H} = \beta m + \boldsymbol{\alpha}\cdot\mathbf{p}$ . It replaces a first order hyperbolic system of partial differential equations with a system involving the spatially non-local operator  $\sqrt{m^2 + |\mathbf{p}|^2}$  defined by its Fourier transform. The Foldy–Wouthuysen transformation is therefore not useful for constructing local discrete (numerical) algorithms. In addition, the Foldy–Wouthuysen transformation for a particle subject to scalar and vector potentials may only be found as an asymptotical

expansion in  $1/c$ . The *minimal coupling* replacement of  $\mathbf{p}$  by  $\mathbf{p} - q\mathbf{A}$  in the presence of a magnetic field holds in the original Dirac–Pauli representation, but *not* in the Newton–Wigner representation. The study of the Foldy–Wouthuysen and related transformations, such as the Douglas–Kroll–Hess expansion, remains active in quantum chemistry as part of many-electron theories for atoms and molecules [20].

### 2.3. The one-dimensional Dirac equation

We consider solutions in which  $\psi$  only depends on  $z$  and  $t$ . Equation (1) then decouples into two separate subsystems,

$$\partial_t \begin{pmatrix} \psi_1 \\ \psi_3 \end{pmatrix} + \partial_z \begin{pmatrix} \psi_3 \\ \psi_1 \end{pmatrix} = -im \begin{pmatrix} \psi_1 \\ -\psi_3 \end{pmatrix}, \quad \partial_t \begin{pmatrix} \psi_2 \\ \psi_4 \end{pmatrix} - \partial_z \begin{pmatrix} \psi_4 \\ \psi_2 \end{pmatrix} = -im \begin{pmatrix} \psi_2 \\ -\psi_4 \end{pmatrix}. \quad (19)$$

These two subsystems describe the evolutions of states with different spins. They decouple due to the absence of spin-changing processes in the one-dimensional Dirac equation for a free particle. Each subsystem may be rewritten in matrix form as

$$\partial_t \begin{pmatrix} \Phi^+ \\ \Phi^- \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \partial_z \begin{pmatrix} \Phi^+ \\ \Phi^- \end{pmatrix} = -im \begin{pmatrix} \Phi^+ \\ -\Phi^- \end{pmatrix}, \quad (20)$$

by writing  $\Phi^+ = \psi_1$  and  $\Phi^- = \psi_3$  for the first subsystem, or  $\Phi^+ = \psi_2$  and  $\Phi^- = -\psi_4$  for the second subsystem.

Following the standard theory of hyperbolic systems, we diagonalise the matrix appearing in (20) as

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \mathbf{U}^\dagger \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \mathbf{U}, \quad (21)$$

using the unitary matrix

$$\mathbf{U} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -i/\sqrt{2} & i/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \mathbf{C}, \quad (22)$$

which is a diagonal multiple of the Hadamard transformation matrix  $\mathbf{C}$  defined in (28) below. Equations (20) may thus be rewritten as

$$\partial_t u + \partial_z u = md, \quad (23a)$$

$$\partial_t d - \partial_z d = -mu. \quad (23b)$$

The extra diagonal matrix with the  $-i$  coefficient in (22) allows this system to be written with real coefficients. The left hand sides now comprise a hyperbolic system in diagonal form. The variables  $u$  and  $d$  are the Riemann invariants [21] of the hyperbolic system defined by the left hand side of (20). They are related to  $\Phi^\pm$  by the unitary transformation

$$\begin{pmatrix} u \\ d \end{pmatrix} = \mathbf{U} \begin{pmatrix} \Phi^+ \\ \Phi^- \end{pmatrix}, \quad (24)$$

and they propagate along the two sets of characteristics  $(z, t) = (z_0 \pm s, t_0 + s)$  parametrised by  $s$ . For example, (23a) becomes  $du/ds = md$  along the  $+$  characteristic, while (23b) becomes  $dd/ds = -mu$  along the  $-$  characteristic,

## 3. QUANTUM ALGORITHMS ON A ONE-DIMENSIONAL LATTICE

### 3.1. Discrete time quantum walks

Classical random walks may be used to construct solutions to Laplace’s equation [22], and Glimm’s [23] random choice method establishes the solution of initial value problems for hyperbolic PDEs. These applications have motivated investigations of algorithmic applications for the quantum random walks introduced independently by Aharonov et al. [24] and Meyer [25]. Due to quantum interference effects, the expected displacement after  $N$  steps of a quantum random walk grows in proportion to  $\sqrt{N}$ , much faster than the growth with  $\sqrt{N}$  for classical random walks. This faster growth makes possible  $\mathcal{O}(\sqrt{N})$  quantum search algorithms. For example,

Grover's [26, 27] quantum search algorithm locates one marked item from among  $N$  items in  $\mathcal{O}(\sqrt{N})$  steps by simulating a discrete time approximation to a Schrödinger equation with an attractive potential localised at the marked item. Childs et al.'s [28] graph traversal algorithm simulates a discrete time approximation to a Schrödinger equation whose Hamiltonian is a scaled adjacency matrix of the graph. The introduction below follows the review of Ambainis [29]. Other reviews include those by Kempe [30] and Kendon [31].

In the classical discrete time random walk on the line, at each time step we move left with probability  $1/2$ , and right with probability  $1/2$ . The natural quantum counterpart considers a basis of states  $|n\rangle$  for  $n \in \mathbb{Z}$  with the transition rule

$$|n\rangle \rightarrow a|n-1\rangle + b|n\rangle + c|n+1\rangle \quad (25)$$

at each step. If one considers an integer lattice with a scalar complex amplitude  $\psi_i$  at each point  $i$ , one may think of  $|n\rangle$  as denoting the state in which  $\psi_n = 1$  and  $\psi_i = 0$  for  $i \neq n$ . The set of  $|n\rangle$  for  $n \in \mathbb{Z}$  thus form a basis for the vector space of complex amplitudes on the lattice. The update rule (25) extends by linearity into an infinite set of linear equations between the coefficients  $\psi_i$  before and after each timestep.

However, Meyer [25] showed that this transition rule is only unitary if one of  $a, b, c$  has unit modulus while the other two vanish, *i.e.* the solutions are  $|a| = 1, b = 0, c = 0$  and permutations. Up to an unimportant global phase, the only unitary operations are deterministic shifts in either direction and the identity operation. This is essentially because the inverse of a general tridiagonal matrix is a full matrix, not another tridiagonal matrix. Meyer [25] identified the very limited subset of tridiagonal matrices that satisfy  $U^{-1} = U^\dagger$ .

This difficulty may be resolved by enlarging the state space to comprise two linearly independent states  $|n, -1\rangle$  and  $|n, 1\rangle$  at each  $n \in \mathbb{Z}$ . Each lattice point  $i$  now holds two complex amplitudes  $\psi_i^+$  and  $\psi_i^-$ . The quantum element then arises through the creation of superpositions.

The one-dimensional discrete time quantum walk algorithm on the line comprises two steps:

- A “coin flip” transformation  $C$ ,

$$C|n, -1\rangle = a|n, -1\rangle + b|n, 1\rangle, \quad C|n, 1\rangle = c|n, -1\rangle + d|n, 1\rangle, \quad (26)$$

- followed by a shift  $S$ ,

$$S|n, -1\rangle = |n-1, -1\rangle, \quad S|n, 1\rangle = |n+1, 1\rangle. \quad (27)$$

Each step of the walk corresponds to applying  $SC$ . The most common choice for the matrix  $C$  is the Hadamard transformation

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}. \quad (28)$$

Applying  $C$  takes  $|n, -1\rangle$  to  $\frac{1}{\sqrt{2}}|n, -1\rangle + \frac{1}{\sqrt{2}}|n, 1\rangle$ , and takes  $|n, 1\rangle$  to  $\frac{1}{\sqrt{2}}|n, -1\rangle - \frac{1}{\sqrt{2}}|n, 1\rangle$ . Applying  $C$  to either basis state thus gives a superposition of the two basis states, each of which will be observed with probability  $1/2$ . This is the sense in which applying  $C$  corresponds to a coin flip.

Shenvi et al. [32] showed that the discrete time quantum walk achieves the same asymptotic performance as Grover's [26, 27] quantum search algorithm. Lovett et al. [33] later showed that the discrete time quantum walk can implement the universal quantum gate set, following Childs' [34] earlier proof of the equivalent result for the continuous time quantum walk. Ryan et al. [35] experimentally demonstrated eight steps of a discrete time quantum walk over three points using an NMR-based quantum computer.

A more general quantum walk uses a biased coin. It replaces the Hadamard matrix (28) with

$$C_\theta = \begin{pmatrix} \cos \theta & -i \sin \theta \\ -i \sin \theta & \cos \theta \end{pmatrix}, \quad (29)$$

for a rotation angle  $\theta \in (0, \pi/2)$ . The composition of **C** followed by **S** defines the unitary mapping

$$\psi^+(n, \tau + 1) = \cos \theta \psi^+(n - 1, \tau) - i \sin \theta \psi^-(n - 1, \tau), \quad (30a)$$

$$\psi^-(n, \tau + 1) = \cos \theta \psi^-(n + 1, \tau) - i \sin \theta \psi^+(n + 1, \tau), \quad (30b)$$

where  $n, t \in \mathbb{Z}$  are discrete space and time coordinates. This algorithm is identical to a quantum lattice gas automaton introduced by Meyer [25, 36].

Strauch [37] reviewed and extended results on the behaviour of this algorithm in the continuum limit. The continuum limit with  $\theta \rightarrow 0$  yields the one-dimensional Dirac equation. This result is originally due to Feynman, and gives his checkerboard path integral formulation for the one-dimensional Dirac equation for a particle with mass controlled by  $\theta$  [1]. The precise limit arises from creating a space-time lattice with  $z = j\Delta t$ ,  $t = n\Delta t$ , putting  $\theta = m\Delta t$  for a particle of mass  $m$ , and taking the limit  $\Delta t \rightarrow 0$  at fixed  $z$  and  $t$ . Conversely, taking  $\theta \rightarrow \pi/2$  in (30a,b) yields the one-dimensional continuous time quantum walk [38]

$$i\partial_t \psi(n, t) = -\gamma \{ \psi(n + 1, t) - 2\psi(n, t) + \psi(n - 1, t) \}. \quad (31)$$

This is recognisable as a spatial finite-difference approximation to the one-dimensional Schrödinger equation for a free particle. The constant  $\gamma$  sets the speed  $\gamma/2$  at which local maxima in  $|\psi|^2$  propagate in the continuum limit.

### 3.2. Quantum cellular automata

The discrete time quantum random walk is just one example of the broader class of quantum cellular automata and quantum lattice gases [25, 36, 39–41]. These systems replace the Boolean variables of classical cellular automata with complex amplitudes, allowing the state of the quantum automaton to be a superposition of the states of the classical automaton. However, only a small subset of update rules produce unitary evolution. For example, the only unitary evolution that may be obtained from the quantum analogs (25) of the common one-dimensional nearest-neighbour cellular automata is a deterministic shift with a constant global phase multiplication (see previous section). An early general formulation for quantum cellular automata by Watrous [42] was later found to allow signal propagation with arbitrarily large velocities, so more restrictive axiomatic formulations with a causality property have been pursued since [43, 44]. Shakeel and Love [45] give a recent review.

Yepez [46] extended Feynman’s one-dimensional path integral formulation to a quantum lattice gas algorithm for the three-dimensional Dirac equation, and Yepez [47] gives a comprehensive exposition of quantum lattice gas algorithms for this and other physical systems. Yepez et al. [48] have performed very large-scale, up to  $5076^3$  points, simulations of the Gross–Pitaevskii (nonlinear Schrödinger) equation using a 2-component quantum lattice gas automata in the diffusive scaling regime in which  $\Delta x \sim \epsilon$  and  $\Delta t \sim \epsilon^2$ . This is the same scaling that leads from the Boltzmann equation directly to the incompressible Navier–Stokes equations in the small Mach number limit at fixed Reynolds number. It was adopted by Inamuro et al. [49] in a lattice Boltzmann context, following earlier work in continuum kinetic theory collected in the book by Sone [50].

### 3.3. The “quantum lattice Boltzmann” algorithm

The hydrodynamic lattice Boltzmann equation may be derived from the discrete Boltzmann PDE through an integration along characteristics [51]. One may apply the same approach to the one-dimensional Dirac equation (23a,b) with a scalar potential,

$$\partial_t u + \partial_z u = md + igu, \quad \partial_t d - \partial_z d = -mu + igd. \quad (32)$$

The potential term  $-g1$  in the original Dirac Hamiltonian  $H$  is a multiple of the identity, so it remains unchanged by the unitary transformations that led from (1) and (2) to (32). Integrating each half along its characteristic

for a timestep  $\Delta t$  gives

$$u(z + \Delta z, t + \Delta t) - u(z, t) = \frac{1}{2}m\Delta t \{d(z + \Delta z, t + \Delta t) + d(z, t)\} \\ + \frac{1}{2}i\Delta t \{g(z + \Delta z, t + \Delta t)u(z + \Delta z, t + \Delta t) + g(z, t)u(z, t)\}, \quad (33a)$$

$$d(z - \Delta z, t + \Delta t) - d(z, t) = -\frac{1}{2}m\Delta t \{u(z - \Delta z, t + \Delta t) + u(z, t)\} \\ + \frac{1}{2}i\Delta t \{g(z - \Delta z, t + \Delta t)d(z - \Delta z, t + \Delta t) + g(z, t)d(z, t)\}. \quad (33b)$$

The left hand sides are exact, while the algebraic terms on the right hand sides have been approximated using the trapezoidal rule. We can now put  $\Delta z = \Delta t$  since the PDE system (32) is written in  $c = 1$  units.

The above pair of algebraic equations is not closed, because  $d(z + \Delta t, t + \Delta t)$  appears on the right hand side of (33a), while  $d(z - \Delta t, t + \Delta t)$  appears on the left hand side of (33b). Similarly,  $u(z - \Delta z, t + \Delta t)$  appears on the right hand side of (33b), while  $u(z + \Delta t, t + \Delta t)$  appears on the left hand side of (33a). The “quantum lattice Boltzmann” (QLB) algorithm [52, 53] alters the locations at which  $u$  and  $d$  are evaluated, and also the location at which the potential  $g$  is evaluated, to obtain the closed pair of algebraic equations

$$u(z + \Delta z, t + \Delta t) - u(z, t) = \frac{1}{2}m\Delta t \{d(z - \Delta z, t + \Delta t) + d(z, t)\} \\ + \frac{1}{2}i\Delta t g(z, t) \{u(z + \Delta z, t + \Delta t) + u(z, t)\}, \quad (34a)$$

$$d(z - \Delta z, t + \Delta t) - d(z, t) = -\frac{1}{2}m\Delta t \{u(z + \Delta z, t + \Delta t) + u(z, t)\} \\ + \frac{1}{2}i\Delta t g(z, t) \{(z - \Delta z, t + \Delta t) + d(z, t)\}. \quad (34b)$$

Solving these two linear equations determines

$$u(z + \Delta z, t + \Delta t) = a(z, t)u(z, t) + b(z, t)d(z, t), \quad (35a)$$

$$d(z - \Delta z, t + \Delta t) = a(z, t)d(z, t) - b(z, t)u(z, t), \quad (35b)$$

with coefficients

$$a(z, t) = \frac{1 - \frac{1}{4}\Delta t^2(m^2 - g(z, t)^2)}{1 - ig(z, t)\Delta t + \frac{1}{4}\Delta t^2(m^2 - g(z, t)^2)}, \quad b(z, t) = \frac{m\Delta t}{1 - ig(z, t)\Delta t + \frac{1}{4}\Delta t^2(m^2 - g(z, t)^2)}. \quad (36)$$

These coefficients satisfy  $|a|^2 + |b|^2 = 1$  and  $ab^* = a^*b$ , so the right hand sides of (35a,b) are equivalent to multiplying the vector  $(u, d)^\top$  by a unitary matrix. Meyer [25] showed that this algorithm for a free particle is unitarily equivalent to the discrete time quantum walk described above.

#### 4. CONVERGENCE PROPERTIES

The above algorithm, in its three different guises, evolves the wavefunction values  $u$  and  $d$  on a discrete space/time lattice through a sequence of unitary operations. However, it provides only a first-order accurate approximation to the one-dimensional Dirac equation. That the accuracy is only first order may most easily be seen from the approximation of  $u(z - \Delta z, t + \Delta t)$  by  $u(z + \Delta z, t + \Delta t)$ , and of  $d(z + \Delta z, t + \Delta t)$  by  $d(z - \Delta z, t + \Delta t)$  in the derivation in Sec. 3.3. Alternatively, the first-order accuracy is due to the approximation of the simultaneous evolution through both spatial derivatives and algebraic terms by the sequential evolution under these terms separately, as expressed by the SC pair in Sec. 3.1.

Figure 1 shows the resulting solution for a particle with mass  $m = 20$  in the periodic potential  $g(z) = -20\sin^2(\pi z/10)$  on the interval  $z \in [-5, 5)$  with periodic boundary conditions. The initial conditions

$$\Phi_0^+ = \frac{1}{(2\pi\Delta_0^2)^{1/4}} \exp \left[ -\frac{(z - z_0)^2}{4\Delta_0^2} + i\kappa(z - z_0) \right] \quad (37)$$

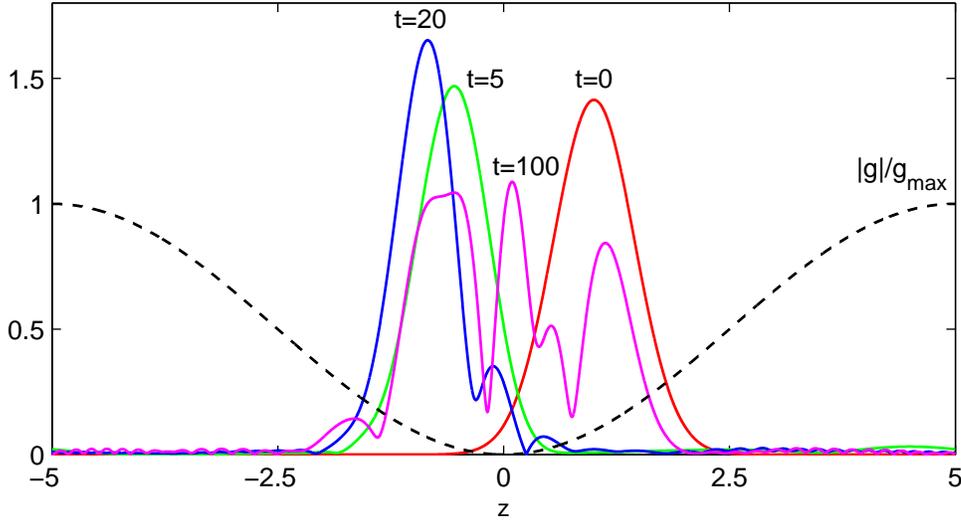


FIGURE 1. Evolution of  $|\Phi^+|$  for an initial Gaussian wavepacket centred at  $z = 1$  in a sinusoidal potential (shown dotted and normalised to a maximum of 1).

and  $\Phi_0^- = 0$  approximate a non-relativistic wavepacket centred around  $z_0 = 1$  with spread  $\Delta_0^2 = 0.1$  and momentum  $\kappa = 0.5$ . The plot also shows the spatial variation of the potential  $g(z)$ , which is an attractive potential for the  $\Phi^+$  component. Figure 2 shows the errors in  $\psi = (u, d)^\top$  computed using the Succi–Benzi QLB scheme, and the two second-order schemes described below. The quantity plotted is the root-mean-square error

$$\|\Delta\psi\|_2 = \left[ \frac{1}{N} \sum_{i=1}^N \left( |u_i - u_i^{\text{ref}}|^2 + |d_i - d_i^{\text{ref}}|^2 \right) \right]^{1/2} \quad (38)$$

relative to an exponentially accurate reference solution  $(u_i^{\text{ref}}, d_i^{\text{ref}})$  on the same spatial lattice. As expected, the standard quantum lattice algorithms, and the unitarily equivalent Succi–Benzi QLB scheme, are only first order accurate, while the two improved schemes described below are second order accurate.

#### 4.1. Computation of reference solutions

If we represent  $u$  and  $d$  by vectors of values at a uniformly spaced set of points  $\mathbf{x} = \{x_1, \dots, x_N\}$ ,  $\mathbf{u} = \{u_1, \dots, u_N\}$ ,  $\mathbf{d} = \{d_1, \dots, d_N\}$ , with periodic boundary conditions, we may write a semi-discrete approximation to the one-dimensional Dirac equation in matrix form as

$$\frac{d}{dt} \begin{pmatrix} \mathbf{u} \\ \mathbf{d} \end{pmatrix} = \mathbf{E} \begin{pmatrix} \mathbf{u} \\ \mathbf{d} \end{pmatrix}, \quad \text{where} \quad \mathbf{E} = \begin{pmatrix} -\mathbf{D} - \mathbf{G} & m \mathbf{1} \\ -m \mathbf{1} & \mathbf{D} - \mathbf{G} \end{pmatrix}. \quad (39)$$

The  $2N \times 2N$  evolution matrix  $\mathbf{E}$  is made up of  $N \times N$  blocks involving the Fourier differentiation matrix  $\mathbf{D}$ , the identity matrix  $\mathbf{1}$ , and the diagonal matrix  $\mathbf{G} = \text{diag}(g(x_1), \dots, g(x_N))$  whose  $i$ th entry is  $g(x_i)$ .

The solution of this matrix ODE system with initial conditions  $\mathbf{u}_0$  and  $\mathbf{d}_0$  is

$$\begin{pmatrix} \mathbf{u}(t) \\ \mathbf{d}(t) \end{pmatrix} = \exp(t\mathbf{E}) \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{d}_0 \end{pmatrix}, \quad (40)$$

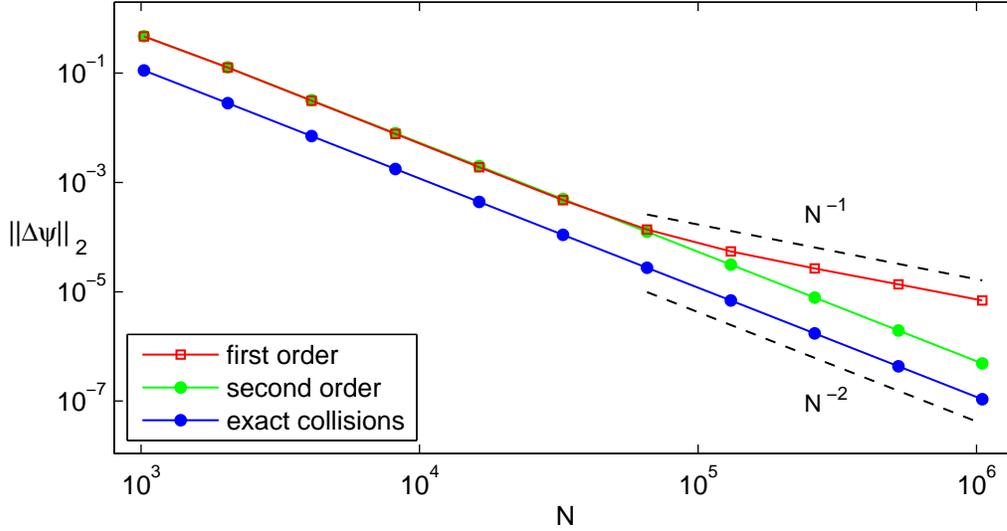


FIGURE 2. Convergence in the discrete  $\ell_2$  norm of  $\psi = (u, d)^\top$  at  $t = 100$  towards an exponentially accurate reference solution with increasing numbers of gridpoints  $N$ .

where the matrix exponential is defined by the convergent series

$$\exp(t\mathbf{E}) = \sum_{n=0}^{\infty} \frac{(t\mathbf{E})^n}{n!}. \quad (41)$$

The efficient and accurate computation of the matrix exponential has been the subject of two noteworthy reviews by Moler & van Loan [54, 55]. The most common approach uses scaling and squaring, or repeated application of the property

$$\exp(t\mathbf{E}) = [\exp(\frac{1}{2}t\mathbf{E})]^2. \quad (42)$$

We determine an integer  $r$  such that  $2^{-r}t\mathbf{E}$  is small enough for a truncation of the series (41) to give sufficient accuracy with a reasonable number of terms, and then square the resulting matrix  $r$  times. Greater accuracy in finite precision arithmetic may be obtained by using a Padé approximation for  $\exp(2^{-r}t\mathbf{E})$  in place of a truncated power series [56, 57].

The reference solutions used in the convergence plot were computed using 512 Fourier collocation points. The matrix  $\mathbf{E}$  is small enough for a direct computation of  $\exp(t\mathbf{E})$  by scaling and squaring using the `expokit` package [56], while the Fourier differentiation matrix provides exponential accuracy. The reference solution values  $u_i^{\text{ref}}$  and  $d_i^{\text{ref}}$  were constructed by zero-extending the Fourier coefficients to form a Fourier series on the fine grid, then computing an inverse FFT on the fine grid.

The Fourier transform of the Gaussian initial conditions (37) decays rapidly like  $\exp(-\Delta_0^2 k^2)$  with increasing  $|k|$ . Figure 3 shows the moduli of the Fourier coefficients  $\hat{u}(k_n, t)$  for  $k_n = 2\pi n/10$  for these initial conditions, and for solution at  $t = 100$ . The broadening of the spectrum, linked to the increasing structure in the solutions shown in figure 1, is sufficiently slow that the solutions for 128 and 512 Fourier collocation points are still close to indistinguishable at  $t = 100$ .

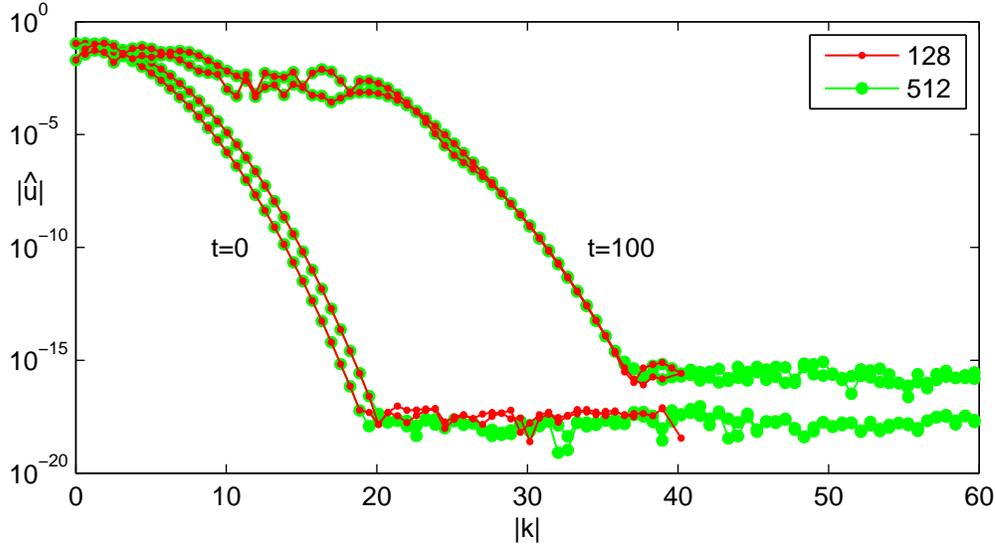


FIGURE 3. The rapid decay of  $|\hat{u}(k, t)|$  with increasing  $|k|$  persists from  $t = 0$  to  $t = 100$ . The spectral solutions computed by matrix exponentiation using 128 and 512 Fourier modes are thus close to indistinguishable. The two different curves for each  $t$  arise from positive and negative  $k$ .

## 5. PLANE WAVE SOLUTIONS

One may also investigate the accuracy of these discrete algorithms from the behaviour of individual Fourier modes, following the approach of Strauch [37] for quantum random walks, or the equivalent von Neumann approach for finite difference schemes [58, 59]. The spatial Fourier transform of the one-dimensional Dirac equation for a free particle may be written as

$$i\partial_t \begin{pmatrix} \hat{u} \\ \hat{d} \end{pmatrix} = \mathbf{H}_k \begin{pmatrix} \hat{u} \\ \hat{d} \end{pmatrix}, \quad (43)$$

where  $\hat{u}$  is the spatial Fourier transform of  $u$ ,

$$\hat{u}(k, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u(z, t) e^{-ikz} dz, \quad (44)$$

and the Hamiltonian is the  $2 \times 2$  matrix

$$\mathbf{H}_k = \begin{pmatrix} k & im \\ -im & -k \end{pmatrix} = k\sigma^z + m\sigma^y. \quad (45)$$

The solution to (43) with initial conditions  $\hat{u}_0$  and  $\hat{d}_0$  may be written as

$$\begin{pmatrix} \hat{u} \\ \hat{d} \end{pmatrix} = \exp(-it\mathbf{H}_k) \begin{pmatrix} \hat{u}_0 \\ \hat{d}_0 \end{pmatrix}. \quad (46)$$

The exponential of the Hamiltonian is

$$\exp(-it\mathbf{H}_k) = \cos(\Omega t) \mathbf{1} - (i/\Omega) \sin(\Omega t) \mathbf{H}_k, \quad (47)$$

with  $\Omega = \sqrt{m^2 + k^2}$ , using properties of the Pauli spin matrices.

### 5.1. Plane wave solutions of the discrete algorithm

In contrast, substituting  $u(z, t) = \hat{u}(t)e^{ikz}$  and  $d(z, t) = \hat{d}(t)e^{ikz}$  into the discrete algorithm (35) yields

$$\begin{pmatrix} \hat{u}(t + \Delta t) \\ \hat{d}(t + \Delta t) \end{pmatrix} = \tilde{\mathbb{E}} \begin{pmatrix} \hat{u}(t) \\ \hat{d}(t) \end{pmatrix}, \quad (48)$$

where the discrete evolution matrix is

$$\tilde{\mathbb{E}} = \frac{1}{1 + \frac{1}{4}m^2\Delta t^2} \begin{pmatrix} e^{-ik\Delta t} \left(1 - \frac{1}{4}m^2\Delta t^2\right) & e^{-ik\Delta t}m\Delta t \\ -e^{ik\Delta t}m\Delta t & e^{ik\Delta t} \left(1 - \frac{1}{4}m^2\Delta t^2\right) \end{pmatrix}. \quad (49)$$

The coefficients of the matrix  $\tilde{\mathbb{E}}$  agree up to and including terms of  $\mathcal{O}(\Delta t)$  with those of the exact evolution matrix  $\mathbb{E} = \exp(-it\mathbf{H}_k)$  defined above. This is consistent with the observed first order global accuracy.

The eigenvalues of  $\tilde{\mathbb{E}}$  may be written as  $\tilde{\lambda}_{\pm} = \exp(\pm i\tilde{\Omega}\Delta t)$  where

$$\tilde{\Omega} = \frac{1}{\Delta t} \cos^{-1} \left\{ \cos(k\Delta t) \left( \frac{1 - \frac{1}{4}m^2\Delta t^2}{1 + \frac{1}{4}m^2\Delta t^2} \right) \right\} = \Omega - \frac{m^2(m^2 + 2k^2)}{12\Omega} \Delta t^2 + \mathcal{O}(\Delta t^4). \quad (50)$$

The discrete eigenvalues thus differ from the eigenvalues of the exact Dirac evolution matrix by an  $\mathcal{O}(\Delta t^2)$  error. Figure 4(a) compares  $\tilde{\Omega}$  and  $\Omega$  for the parameters  $m = 3$  and  $\Delta t = 0.1$ . The product  $m\Delta t$  is relatively large, to make differences between  $\Omega$  and  $\tilde{\Omega}$  easily visible.

The group velocity for waves in the numerical scheme is

$$\frac{\partial \tilde{\Omega}}{\partial k} = \frac{\sin(k\Delta t)}{\sin(\tilde{\Omega}\Delta t)} \left( \frac{1 - \frac{1}{4}m^2\Delta t^2}{1 + \frac{1}{4}m^2\Delta t^2} \right), \quad (51)$$

while the group velocity for waves in the Dirac equation is  $\partial\Omega/\partial k = k/\Omega$ . These two expressions are plotted in Fig. 4(b). The numerical group velocity has the same sign as the true group velocity everywhere except at the points  $k\Delta t = \pm\pi$  where the numerical group velocity vanishes. The discrete frequency  $\tilde{\Omega}$  thus varies monotonically over the intervals  $0 \leq k\Delta t \leq \pi$  and  $-\pi \leq k\Delta t \leq 0$ , so the quantum lattice algorithm is free of the fermion doubling problem that affects many finite difference and finite element schemes for the Dirac equation.

### 5.2. Fermion doubling

The fermion doubling problem may be illustrated with the first order wave equation  $u_t + u_z = 0$ , as obtained in Sec. 2.3 for a massless ( $m = 0$ ) particle. Approximating the spatial derivative  $u_z$  using the natural centred finite difference scheme on a spatial lattice with points  $z_n = n\Delta z$ , while leaving time continuous, gives

$$\frac{d}{dt}u_n + \frac{u_{n+1} - u_{n-1}}{2\Delta z} = 0. \quad (52)$$

This equation has plane wave solutions  $u_n(t) = \exp(i(kn\Delta z - \omega t))$  with  $\omega = \sin(k\Delta z)/\Delta z$  as shown in figure 5(a) [58]. For  $k\Delta z \ll 1$  we recover the dispersion relation  $\omega = k$  for the wave equation. However, the frequency  $\omega$  returns to zero when  $k\Delta z = \pi$ , and there are two wavenumbers  $k$  for each frequency  $\omega \in [0, 1/\Delta z]$ . This is the

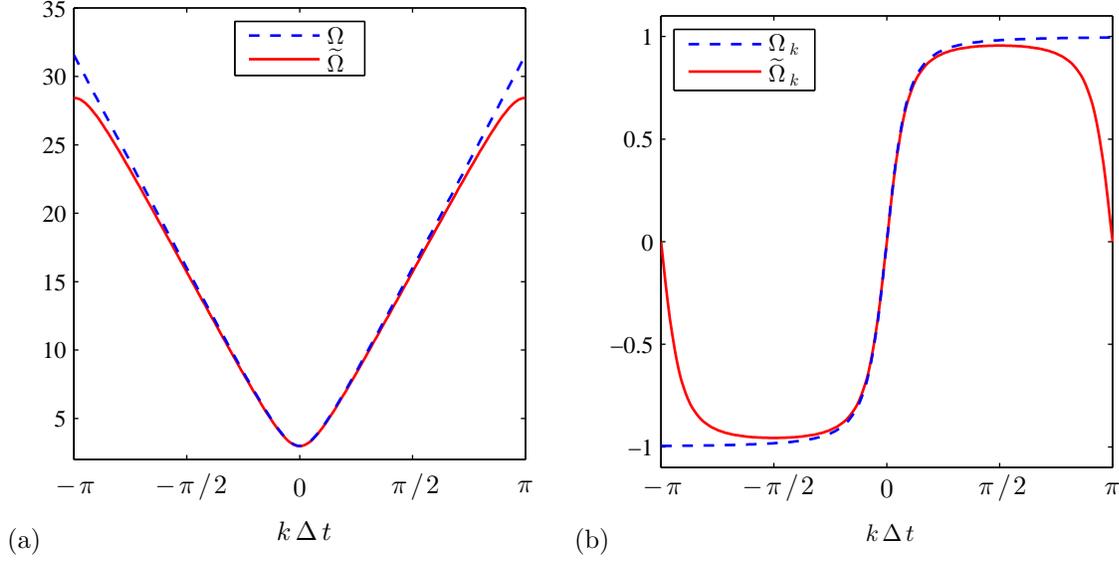


FIGURE 4. (a) Discrete and continuous dispersion relations for the parameters  $m = 3$  and  $\Delta t = 0.1$ . (b) Discrete and continuous group velocities  $\partial\Omega/\partial k$  and  $\partial\tilde{\Omega}/\partial k$  for the same parameters.

origin of the name “fermion doubling”. Moreover, the group velocity  $c_g = d\omega/dk$  is positive for  $k\Delta z \in (0, \pi/2)$  and negative for  $k\Delta z \in (\pi/2, \pi)$ .

The second set of solutions to the dispersion relation appear because the spatial stencil is twice as wide as the spacing  $\Delta z$  between lattice points. The natural solution is to reduce the width of the stencil by using the upwind differencing formula

$$\frac{d}{dt}u_n + \frac{u_n - u_{n-1}}{\Delta z} = 0. \quad (53)$$

This gives the complex dispersion relation

$$\omega = \frac{2}{\Delta z} \sin\left(\frac{k\Delta z}{2}\right) \exp\left(-i \frac{k\Delta z}{2}\right), \quad (54)$$

with expansion

$$\omega = k - \frac{1}{2}i k^2 \Delta z - \frac{1}{6}k^3 \Delta z^2 + \dots \quad (55)$$

The first imaginary term gives a diffusive decay of the solution over time, as a consequence of the upwind differencing. This is incompatible with unitary and time-reversible evolution in a quantum system. The analogous downwind differencing  $(u_{n+1} - u_n)/\Delta z$  would give exponential growth due to an effective negative diffusivity.

Nielsen & Ninomiya [60–62] formalised these concrete examples into a “no-go” theorem for the non-existence of a spatial discretisation that gives unitary evolution (no spurious growth or decay) in continuous time, and a 1-to-1 relation between frequency and wavenumber.

The quantum lattice algorithms avoid these difficulties by discretising both space and time, so that (53) becomes

$$\frac{u_n(t + \Delta t) - u_n(\Delta t)}{\Delta t} + \frac{u_n(t) - u_{n-1}(t)}{\Delta z} = 0, \quad (56)$$

and synchronising the space and time steps ( $\Delta z = \Delta t$ ) so that (56) becomes  $u_n(t + \Delta t) = u_{n-1}(t)$ , which is the exact solution of  $u_t + u_z = 0$ .

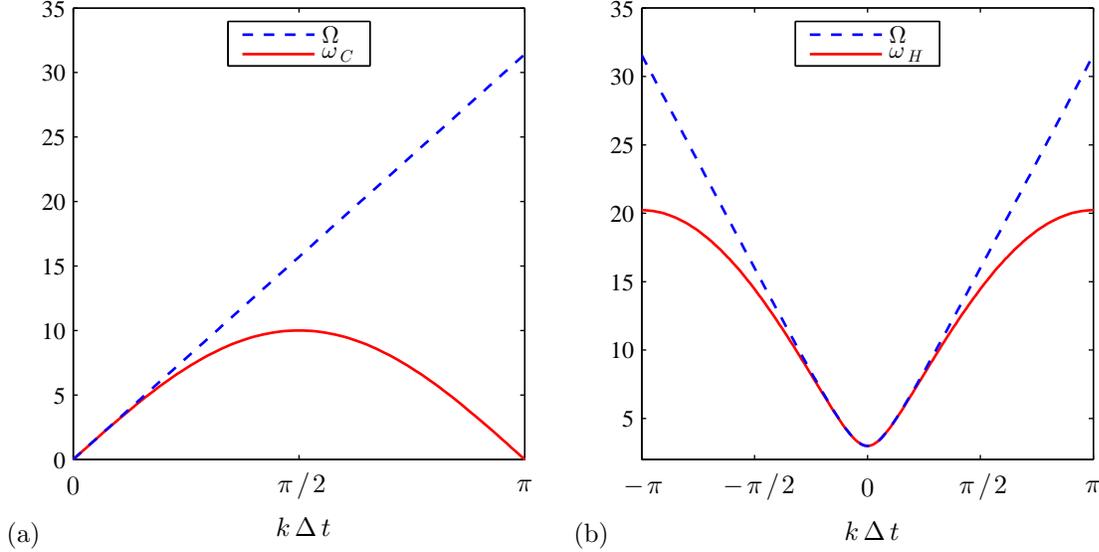


FIGURE 5. (a) Dispersion relation for the centred finite difference approximation to the wave equation showing fermion doubling. (b) Dispersion relation for Heisenberg's Gitterwelt with parameters  $m = 3$  and  $\Delta t = 0.1$ . The numerical dispersion relations are shown as solid curves, and those of the underlying PDEs are shown dotted.

### 5.3. Heisenberg's Gitterwelt

Heisenberg [2] proposed the spatial discretisation

$$\frac{d^2}{dt^2}u_n - \frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta z^2} + m^2u_n = 0, \quad (57)$$

for the Klein–Gordon equation. This semi-discrete equation has dispersion relation

$$\omega^2 = m^2 + (2/\Delta z)^2 \sin^2(k\Delta z/2), \quad (58)$$

which is shown in figure 5(b) for the parameters  $m = 3$  and  $\Delta z = 0.1$ . The effective mass of the particle represented by a solution of this dispersion relation is determined by  $dc_g/dk = d^2\omega/dk^2$ . This gives the rate at which the group velocity  $c_g$  changes with the momentum  $k$ , and hence the reciprocal of the mass. At  $k = 0$  the curvature  $d^2\omega/dk^2 = 1/m$  as expected, while  $d^2\omega/dk^2 \sim -\Delta z/2$  for  $k = \pi/\Delta z$  if  $m\Delta z \ll 1$ . The dispersion relation near  $k = \pi/\Delta z$  then corresponds to that of a particle with the opposite charge and a much larger effective mass [2, 3].

Heisenberg's dispersion relation shows some qualitative similarities with the dispersion relation for the quantum lattice algorithm shown in figure 4, notably that  $\omega$  is a monotonic function of  $k$ . However, the curvatures near  $k = 0$  and  $k = \pi/\Delta z$  are different in magnitude, and time is continuous rather than discrete.

## 6. A SECOND-ORDER ACCURATE ONE-DIMENSIONAL ALGORITHM

The discrete algebraic system (33a,b) may be extended into a second-order accurate algorithm using a change of variables equivalent to the  $f_i$  to  $\tilde{f}_i$  introduced by He et al. [51] in their second-order accurate derivation of hydrodynamic lattice Boltzmann algorithms. However, the alternative derivation of this transformation recently

introduced by Dellar [63] for hydrodynamic lattice Boltzmann algorithms using Strang [64] splitting is simpler and more natural in a quantum context, as it guarantees that the overall algorithm is unitary.

The idea of splitting a unitary evolution into a product of simpler unitary operators is natural in quantum computation [46]. However, the quantum computation literature is primarily concerned with achieving any kind of convergence for algorithms adapted to quantum computers, rather than achieving second-order convergence with algorithms for conventional digital computers.

Separating the one-dimensional Dirac equation with a scalar potential into two operators,

$$\partial_t \begin{pmatrix} u \\ d \end{pmatrix} = \begin{pmatrix} -\partial_z & 0 \\ 0 & \partial_z \end{pmatrix} \begin{pmatrix} u \\ d \end{pmatrix}, \quad \partial_t \begin{pmatrix} u \\ d \end{pmatrix} = \begin{pmatrix} ig & m \\ -m & ig \end{pmatrix} \begin{pmatrix} u \\ d \end{pmatrix}, \quad (59)$$

is equivalent to splitting the Hamiltonian into its spatial derivative and algebraic parts. The solutions to the two parts may be written using the operators

$$S = \begin{pmatrix} \exp(-\Delta t \partial_z) & 0 \\ 0 & \exp(\Delta t \partial_z) \end{pmatrix}, \quad C = \exp \left\{ \Delta t \begin{pmatrix} ig & m \\ -m & ig \end{pmatrix} \right\} = e^{ig\Delta t} \begin{pmatrix} \cos m\Delta t & \sin m\Delta t \\ -\sin m\Delta t & \cos m\Delta t \end{pmatrix}. \quad (60)$$

The  $\exp(\pm\Delta t \partial_z)$  coefficients on the diagonal of  $S$  come from the identity

$$\psi(z + \Delta z) = \sum_{n=0}^{\infty} \frac{\Delta z^n}{n!} (\partial_z^n \psi) \Big|_z = \exp(\Delta z \partial_z) \psi \Big|_z, \quad (61)$$

and  $\Delta z = \pm\Delta t$  in natural units. Combining the  $S$  and  $C$  operators gives the update rule

$$\psi(t + \Delta t) = SC\psi(t), \quad (62)$$

where  $\psi(t) = (u(0, t), \dots, u(n\Delta z, t), d(0, t), \dots, d(n\Delta z, t))^T$  is the vector of  $u$  and  $d$  values at lattice points. This is the form that appears in the discrete time quantum walks of Sec. 3.1.

Instead of using the above exact expression, we may approximate the exponential of the collision (or coin flip) operator using the Crank–Nicolson form

$$\left(1 - \frac{1}{2}i\Delta t H_m\right) \left(1 + \frac{1}{2}i\Delta t H_m\right)^{-1} = \exp(-i\Delta t H_m) + \mathcal{O}(\Delta t^3). \quad (63)$$

The expression on the left hand side is unitary whenever  $H_m$  is Hermitian. It defines

$$\tilde{C} = \frac{1}{1 - i\Delta t g - \frac{1}{4}\Delta t^2(g^2 - m^2)} \begin{pmatrix} 1 + \frac{1}{4}\Delta t^2(g^2 - m^2) & m\Delta t \\ -m\Delta t & 1 + \frac{1}{4}\Delta t^2(g^2 - m^2) \end{pmatrix}, \quad (64)$$

which gives the same unitary and second-order accurate approximation to the algebraic terms that appears in the quantum lattice Boltzmann algorithm [52].

Either algorithm is unitary, but the combinations  $SC$  and  $S\tilde{C}$  are only first order accurate due to the splitting error between the non-commuting operators  $S$  and  $C$  or  $\tilde{C}$ . We may construct a second-order accurate algorithm using Strang's [64] splitting formula

$$\psi(t + \Delta t) = C^{1/2}SC^{1/2}\psi(t) \quad (65)$$

or the equivalent using  $\tilde{C}$  instead of  $C$ . Applying this algorithm for  $n$  timesteps gives

$$\psi(t + n\Delta t) = C^{1/2}SC^{1/2}C^{1/2}SC^{1/2} \dots C^{1/2}SC^{1/2}\psi(t), \quad (66)$$

and we may combine adjacent pairs using  $C^{1/2}C^{1/2} = C$  to obtain

$$\psi(t + n\Delta t) = C^{1/2}SCSC \dots SC^{1/2}\psi(t). \quad (67)$$

The integration along characteristics approach leads naturally to collisions preceding streaming, so we rewrite the above as

$$\psi(t + n\Delta t) = \mathbf{C}^{1/2}(\mathbf{SC})^n \mathbf{C}^{-1/2} \psi(t), \quad (68)$$

using the identity  $\mathbf{C}^{1/2} = \mathbf{C}\mathbf{C}^{-1/2}$ . We thus recover what looks like the previous first-order splitting,

$$\tilde{\psi}(t + n\Delta t) = (\mathbf{SC})^n \tilde{\psi}(t), \quad (69)$$

but applied to the transformed wavefunctions

$$\tilde{\psi} = \mathbf{C}^{-1/2} \psi, \quad \psi = \mathbf{C}^{1/2} \tilde{\psi}. \quad (70)$$

This transformation raises the accuracy from first order to second order in  $\Delta t$ . The operators  $\mathbf{C}^{\pm 1/2}$  are applied only once each per computation, not once per timestep, so we may tolerate  $\mathcal{O}(\Delta t^2)$  errors in  $\mathbf{C}^{\pm 1/2}$  while preserving global second-order accuracy. For example, approximating  $\mathbf{C}^{1/2} \approx \frac{1}{2}(1 + \mathbf{C})$  leads to the same  $f_i \rightarrow \bar{f}_i$  transformation introduced by He et al. [51] for the hydrodynamic lattice Boltzmann algorithm [63].

However, algorithms for quantum mechanics should be unitary, which  $\frac{1}{2}(1 + \mathbf{C})$  is not, so we use either the exact formula

$$\mathbf{C}^{\pm 1/2} = \exp \left\{ \pm \frac{1}{2} \Delta t \begin{pmatrix} ig & m \\ -m & ig \end{pmatrix} \right\} = e^{ig\Delta t/2} \begin{pmatrix} \cos m\Delta t/2 & \pm \sin m\Delta t/2 \\ \mp \sin m\Delta t/2 & \cos m\Delta t/2 \end{pmatrix}. \quad (71)$$

or the Crank–Nicolson approximation

$$\tilde{\mathbf{C}}^{\pm 1/2} = (1 \mp \frac{1}{4} i \Delta t H_m) (1 \pm \frac{1}{4} i \Delta t H_m)^{-1}. \quad (72)$$

Either transformation raises the previous first-order algorithms to second-order accuracy, as shown in figure 2. In terms of our earlier Fourier analysis, we replace  $\tilde{\mathbf{E}}$  with  $\mathbf{C}^{1/2} \tilde{\mathbf{E}} \mathbf{C}^{-1/2}$ . This similarity transformation leaves the eigenvalues, which were already second-order accurate, unchanged, but corrects the  $\mathcal{O}(\Delta t)$  terms in the eigenvectors to agree with the eigenvectors of the exact evolution operator  $\mathbf{E}$ .

## 7. A SECOND-ORDER ACCURATE ALGORITHM WITH EXACT COLLISIONS

While the formulation (3.3) with its  $a$  and  $b$  coefficients arises naturally from integrating the complete one-dimensional Dirac equations along characteristics, the splitting approach in (6) shows that one could equally well employ the exact solution (60) of the decoupled system (59) involving only  $m$  and  $g$  instead of the second-order Crank–Nicolson approximation (64). This gives the alternative algorithm

$$u(z + \Delta z, t + \Delta t) = e^{ig(z)\Delta t} (\cos(m\Delta t)u(z, t) + \sin(m\Delta t)d(z, t)), \quad (73a)$$

$$d(z - \Delta z, t + \Delta t) = e^{ig(z)\Delta t} (\cos(m\Delta t)d(z, t) - \sin(m\Delta t)u(z, t)), \quad (73b)$$

for a time-independent potential. For a free particle, this is equivalent to replacing  $m$  with an effective numerical mass

$$\tilde{m} = \frac{2}{\Delta t} \tan \left( \frac{1}{2} m \Delta t \right) \sim m \left( 1 + \frac{1}{12} m^2 \Delta t^2 + \frac{1}{120} m^4 \Delta t^4 + \dots \right). \quad (74)$$

The tangent half-angle formulae then imply

$$\frac{\tilde{m} \Delta t}{1 + \tilde{m}^2 \Delta t^2 / 4} = \sin(m\Delta t), \quad \text{and} \quad \frac{1 - \tilde{m}^2 \Delta t^2 / 4}{1 + \tilde{m}^2 \Delta t^2 / 4} = \cos(m\Delta t), \quad (75)$$

so the Crank–Nicolson collision matrix  $\tilde{\mathbf{C}}$  evaluated for  $\tilde{m}$  coincides with the exact collision matrix  $\mathbf{C}$  for  $m$ .

In terms of computational cost,  $\cos(m\Delta t)$  and  $\sin(m\Delta t)$  are constants that may be precomputed. This just leaves the evaluation of  $e^{ig(z)\Delta t}$ , while the Crank–Nicolson matrix requires the reciprocal of the complex denominator  $1 - ig(z, t)\Delta t + \frac{1}{4}\Delta t^2(m^2 - g(z, t)^2)$ . On modern hardware these two operations are of comparable cost, and both are typically negligible compared with loading and storing data. The complex exponential is usually best evaluated as  $\cos(g(z)\Delta t) + i\sin(g(z)\Delta t)$ , due to the availability of fast routines to compute simultaneously both sin and cos of a real argument.

In concrete implementations on an Nvidia K20c GPU with error correcting code (ECC) memory disabled, both algorithms offer about 2200 million lattice updates per second using 64 bit floating point arithmetic. This corresponds to a useful memory bandwidth of 136 GBytes per second, based on the complex  $u$  and  $d$  being read and written once per lattice point, and the real  $g$  being read once. The algorithm using exponentiation is about 3% slower, but the resulting errors are about 4 times smaller, as shown by the lower blue line in figure 2. Again, one needs to pre- and post-process using  $C^{\pm 1/2}$  to achieve second-order accuracy.

This is the quantum equivalent of the hydrodynamic lattice Boltzmann algorithm with exact collisions in [63]. However, the standard hydrodynamic lattice Boltzmann algorithm relies upon a cancellation between the splitting error and the error in the Crank–Nicolson approximation of collisions to achieve grid-scale Reynolds numbers above unity. The errors in the hydrodynamic part of the solution are larger if one uses the exact solution of the collision step in place of the Crank–Nicolson approximation. In particular, the Crank–Nicolson approximation yields the correct hydrodynamic viscosity even for  $\Delta t \gg \tau$ , while the algorithm with exact collisions gives an incorrect viscosity unless  $\Delta t < \tau$ .

## 8. MULTIPLE SPACE DIMENSIONS

The quantum lattice Boltzmann approach was inspired by the structural similarities between the Dirac equation in the form

$$(\partial_t + \boldsymbol{\alpha} \cdot \nabla) \psi = i(g1 - m\beta)\psi, \quad (76)$$

and the discrete Boltzmann equation

$$(\partial_t + \boldsymbol{\xi}_i \cdot \nabla) f_i = \sum_{j=0}^n \Omega_{ij} (f_j - f_j^{(0)}), \quad (77)$$

that forms the basis of hydrodynamic lattice Boltzmann algorithms. Both are linear, symmetric, hyperbolic systems with algebraic source terms. However, the discrete Boltzmann equation (77) is highly unusual among multi-dimensional hyperbolic systems in possessing one-dimensional characteristic curves of the form  $(\mathbf{x}, t) = (\mathbf{x}_0 + s\boldsymbol{\xi}_i, t_0 + s)$  with parameter  $s$ . Writing (77) out in full gives

$$\left[ \partial_t + \begin{pmatrix} \xi_{0x} & 0 & \cdots & 0 \\ 0 & \xi_{1x} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_{nx} \end{pmatrix} \partial_x + \begin{pmatrix} \xi_{0y} & 0 & \cdots & 0 \\ 0 & \xi_{1y} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_{ny} \end{pmatrix} \partial_y + \begin{pmatrix} \xi_{0z} & 0 & \cdots & 0 \\ 0 & \xi_{1z} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_{nz} \end{pmatrix} \partial_z \right] \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} = \boldsymbol{\Omega} \begin{pmatrix} f_0 - f_0^{(0)} \\ f_1 - f_1^{(0)} \\ \vdots \\ f_n - f_n^{(0)} \end{pmatrix}. \quad (78)$$

Each component of  $\boldsymbol{\xi}_i \cdot \nabla = \xi_{ix}\partial_x + \xi_{iy}\partial_y + \xi_{iz}\partial_z$  involves a diagonal matrix, which is highly unusual. All one may expect in general is the existence of three-dimensional characteristic surfaces in four-dimensional  $(\mathbf{x}, t)$  space [21].

The three matrices  $\alpha^x$ ,  $\alpha^y$ ,  $\alpha^z$  do not commute, and so are not simultaneously diagonalisable. The characteristics of the Dirac equation are thus three-dimensional surfaces, the light cones of special relativity, and discontinuities in the initial conditions are confined to propagating along these surfaces. There are no one-dimensional characteristic curves that one may integrate along, as in the approach of He et al. [51] that leads from the discrete Boltzmann equation to a lattice Boltzmann equation.

Instead, one may proceed by splitting (76) into three parts, each containing one of the  $\alpha^i$  matrices and  $1/3$  of the algebraic right hand side

$$(\partial_t + \alpha^x \partial_x) \psi = \frac{1}{3}i(g1 - m\beta)\psi, \quad (79a)$$

$$(\partial_t + \alpha^y \partial_y) \psi = \frac{1}{3}i(g1 - m\beta)\psi, \quad (79b)$$

$$(\partial_t + \alpha^z \partial_z) \psi = \frac{1}{3}i(g1 - m\beta)\psi. \quad (79c)$$

This follows the general approach of Succi and Benzi [52] and Palpacelli and Succi [65], but reformulated to use the standard representation of the Dirac equation instead of the Majorana representation. The latter applies a transformation that has the effect of exchanging  $\beta$  and  $\alpha^y$ , so a real matrix appears on the left hand side of the analogue of (79b), but the three right hand sides become non-diagonal.

Each  $\alpha^i$  matrix is Hermitian, and so diagonalisable using one of the three unitary matrices

$$X = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad Y = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & i & 0 & 1 \\ -i & 0 & i & 0 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -i \end{pmatrix}, \quad Z = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}. \quad (80)$$

This enables the left hand side of each of (79a,b,c) to be diagonalised using

$$X^\dagger \alpha^x X = Y^\dagger \alpha^y Y = Z^\dagger \alpha^z Z = \beta. \quad (81)$$

For example, we multiply (79c) on the left by  $Z^\dagger$

$$Z^\dagger (\partial_t + \alpha^z \partial_z) \psi = Z^\dagger \frac{1}{3}i(g1 - m\beta)\psi. \quad (82)$$

and insert a factor of  $ZZ^\dagger = 1$ ,

$$Z^\dagger (\partial_t + \alpha^z \partial_z) ZZ^\dagger \psi = Z^\dagger \frac{1}{3}i(g1 - m\beta)ZZ^\dagger \psi, \quad (83)$$

so that  $\alpha^z$  is diagonalised,

$$(\partial_t + \beta \partial_z) Z^\dagger \psi = Z^\dagger \frac{1}{3}i(g1 - m\beta)ZZ^\dagger \psi. \quad (84)$$

This is now in the form of a one-dimensional Dirac equation, with a diagonal differential operator  $(\partial_t + \beta \partial_z)$ , so it may be integrated along its characteristic curves using the approach of Sec. 3.3. The solution of (79c) over a timestep  $\Delta t$  is thus given by rotating from the standard representation to the diagonal-in-z representation by applying  $Z^\dagger$ , evolving using  $\tilde{E}_z$ , and rotating back to the standard representation by applying  $Z$ , giving  $Z\tilde{E}_z Z^\dagger$  in all.

A discrete approximation to the evolution under the three-dimensional Dirac equation is thus given by

$$\psi^{n+1} = Z\tilde{E}_z Z^\dagger Y\tilde{E}_y Y^\dagger X\tilde{E}_x X^\dagger \psi^n \quad (85)$$

with a splitting error proportional to  $\Delta t^2$ . The overall algorithm is thus only first order accurate. The original formulation [52, 65] applied the  $Z$  and  $Z^\dagger$  and the other matrix pairs in the wrong order, and thus suffered from  $\mathcal{O}(1)$  errors, most notably through a loss of isotropy for spherically symmetric initial conditions [66]. The corrected algorithm is isotropic, and converges with first order accuracy towards independent spectral solutions of the three-dimensional Dirac equation [67].

In fact, Yopez [46] had previously presented a three-dimensional first-order algorithm that is unitarily equivalent to the corrected algorithm (85) above:

$$\psi' = \mathcal{Y}_{\pi/4}^{(2)} S_x \mathcal{Y}_{\pi/4}^{(2)\dagger} \mathcal{X}_{\pi/4}^{(2)\dagger} S_y \mathcal{X}_{\pi/4}^{(2)} S_z \mathcal{X}_{m\Delta t}^{(1)} \psi, \quad (86)$$

where  $S_x, S_y, S_z$  are the classical translation operators along the 3 coordinate axes, and the  $\mathcal{X}$  and  $\mathcal{Y}$  matrices are

$$\begin{aligned}\mathcal{X}_\theta^{(1)} &= e^{i\theta\sigma^x} \otimes \mathbf{1} = \begin{pmatrix} \cos\theta & 0 & i\sin\theta & 0 \\ 0 & \cos\theta & 0 & i\sin\theta \\ i\sin\theta & 0 & \cos\theta & 0 \\ 0 & i\sin\theta & 0 & \cos\theta \end{pmatrix}, \\ \mathcal{X}_\theta^{(2)} &= \mathbf{1} \otimes e^{i\theta\sigma^x} = \begin{pmatrix} \cos\theta & i\sin\theta & 0 & 0 \\ i\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & i\sin\theta \\ 0 & 0 & i\sin\theta & \cos\theta \end{pmatrix}, \\ \mathcal{Y}_\theta^{(2)} &= \mathbf{1} \otimes e^{i\theta\sigma^y} = \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{pmatrix}.\end{aligned}\tag{87}$$

These correspond to the  $Y^\dagger X$  and other matrices in (85). This algorithm achieves some simplification by applying the evolution due to the whole algebraic term first using  $\mathcal{X}_{m\Delta t}^{(1)}$ , then computing the evolution due to the pure streaming operator  $(\partial_t + \boldsymbol{\alpha} \cdot \nabla)\psi = 0$ . Yepez et al. [48] used a related three-dimensional, two-component formulation to simulate the Gross-Pitaevskii (nonlinear Schrödinger) in the diffusive scaling with  $\Delta x \sim \epsilon$  and  $\Delta t \sim \epsilon^2$ .

## 9. A SECOND-ORDER MULTI-DIMENSIONAL FORMULATION USING SHORT PATH INTEGRALS

A more elegant implementation uses the reversability of the individual streaming steps to precompute a multidimensional streaming operator in finite difference form. For simplicity we consider the two-dimensional Dirac equation for a massless particle, also known as the Weyl equation, with a scalar potential,

$$(\partial_t + \boldsymbol{\sigma} \cdot \nabla)\psi = ig\psi.\tag{88}$$

Here  $\psi = (u, d)^\top$  is a two-component wavefunction, and  $\boldsymbol{\sigma} \cdot \nabla = \sigma^x \partial_x + \sigma^y \partial_y$ . Besides offering a simplified example, this equation describes the low energy behaviour of charge carriers in graphene, the two-dimensional form of carbon [9].

In the absence of the potential, the evolution of the wavefunction  $\psi$  at a given lattice point  $i, j$  may be approximated with second-order accuracy by the result of applying  $\mathcal{X}^{1/2} \mathcal{Y} \mathcal{X}^{1/2}$ . We interpret  $\mathcal{X}^{1/2}$  as producing shifts by  $\frac{1}{2}\Delta x$  in the positive and negative directions, *i.e.*  $\mathcal{X}^{1/2} = \mathbf{X} S_x^{1/2} \mathbf{X}^{-1}$  where  $\mathbf{X}$  is a matrix that diagonalises  $\sigma^x$ . The result of applying the  $\mathcal{X}^{1/2} \mathcal{Y} \mathcal{X}^{1/2}$  operator to  $\psi = (u, d)^\top$  is

$$\left(\mathcal{X}^{1/2}\psi\right)_{i,j} = \frac{1}{2} \begin{pmatrix} u_{i-1/2,j} + d_{i-1/2,j} + u_{i+1/2,j} - d_{i+1/2,j} \\ u_{i-1/2,j} + d_{i-1/2,j} - u_{i+1/2,j} + d_{i+1/2,j} \end{pmatrix}.\tag{89}$$

This is a ‘‘pull’’ formulation that calculates  $\mathcal{X}^{1/2}\psi$  at the  $i, j$  lattice point, as opposed to the previous ‘‘push’’ formulation that propagates  $\psi_{i,j}$  to adjacent lattice points through a sequence of unitary operations.

Similarly, the result of applying  $\mathcal{Y} = \mathbf{Y} S_y \mathbf{Y}^{-1}$  is

$$(\mathcal{Y}\psi)_{i,j} = \frac{1}{2} \begin{pmatrix} u_{i,j-1} + u_{i,j+1} - i d_{i,j-1} + i d_{i,j+1} \\ d_{i,j-1} + d_{i,j+1} + i u_{i,j-1} - i u_{i,j+1} \end{pmatrix}.\tag{90}$$

Each application of  $S_x^{1/2}$  involves shifts by 0 or  $\pm 1/2$  points, so the combined application of  $\mathcal{X}^{1/2} \mathcal{Y} \mathcal{X}^{1/2}$  only involves shifts by 0 or  $\pm 1$  points. We can thus express its operation as a finite-difference formula involving  $u$

and  $d$  at the point  $i, j$  and its neighbours:

$$\left(\mathcal{X}^{1/2}\mathcal{Y}\mathcal{X}^{1/2}\psi\right)_{ij} = \begin{pmatrix} u_{i,j}^{n+1} \\ d_{i,j}^{n+1} \end{pmatrix}, \quad (91)$$

where

$$u_{i,j}^{n+1} = \frac{1}{4} \left[ u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} \right] + \frac{1}{4} \left[ d_{i-1,j-1} + d_{i-1,j+1} - d_{i+1,j-1} - d_{i+1,j+1} \right] + \frac{1}{2}i \left[ d_{i,j+1} - d_{i,j-1} \right], \quad (92a)$$

$$d_{i,j}^{n+1} = \frac{1}{4} \left[ d_{i-1,j-1} + d_{i+1,j-1} + d_{i-1,j+1} + d_{i+1,j+1} \right] + \frac{1}{4} \left[ u_{i-1,j-1} + u_{i-1,j+1} - u_{i+1,j-1} - u_{i+1,j+1} \right] - \frac{1}{2}i \left[ u_{i,j+1} - u_{i,j-1} \right]. \quad (92b)$$

The superscript  $n$  is omitted on the right hand sides for clarity. The half points such as  $i + 1/2, j$  only appear in the intermediate stages to construct the formulae above. This construction is equivalent to a short path integral formulation, since we sum contributions from neighbouring points such as  $u_{i\pm 1, j\pm 1}^n$  to find  $u_{i,j}^{n+1}$  and  $u_{i,j}^{n+1}$ . A full path integral formulation would extend the sum back to include only contributions from the initial conditions  $u_{i,j}^0$  and  $d_{i,j}^0$  at  $t = 0$ .

Equations (92a,b) define a second-order accurate unitary streaming operator  $S$  that approximates the exponential  $\exp(-\Delta t \boldsymbol{\sigma} \cdot \nabla)$  of the two-dimensional Weyl operator. The scalar potential may be included by combining  $S$  with a collision operator  $C$  using the above second-order splitting  $C^{1/2}SC^{1/2}$  with adjacent applications of  $C^{1/2}$  combined. This is equivalent to multiplying the right hand sides of (92a,b) by  $\exp(i\Delta t g_{i,j})$  in the timestepping algorithm,

$$u_{i,j}^{n+1} = \exp(i\Delta t g_{i,j}) \left\{ \frac{1}{4} \left[ u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} \right] + \frac{1}{4} \left[ \dots \right] + \frac{1}{2}i \left[ d_{i,j+1} - d_{i,j-1} \right] \right\}, \quad (93a)$$

$$d_{i,j}^{n+1} = \exp(i\Delta t g_{i,j}) \left\{ \frac{1}{4} \left[ d_{i-1,j-1} + d_{i+1,j-1} + d_{i-1,j+1} + d_{i+1,j+1} \right] + \frac{1}{4} \left[ \dots \right] - \frac{1}{2}i \left[ u_{i,j+1} - u_{i,j-1} \right] \right\}, \quad (93b)$$

while applying  $\exp(\pm \frac{1}{2}i\Delta t g_{i,j})$  as pre- and post-processing to maintain second order accuracy. One could include the exponentials in the intermediate steps that define  $S$ , at the complexity and expense of evaluating  $\exp(\frac{1}{2}i\Delta t g_{i,j})$  at many more points, including the fictitious intermediate points  $i \pm 1/2, j$ . For a massive particle one would replace  $g_{ij}$  by  $g_{ij} \pm m$ .

This algorithm is more memory-intensive than those described in Sec. 8. It requires storage for  $\psi_{i,j}^{n+1}$  as well as  $\psi_{i,j}^n$ , like a naive implementation of a hydrodynamic lattice Boltzmann algorithm, while the separate streaming and collision steps in Sec. 8 may be implemented in-place. The fraction of additional memory required may be reduced by decomposing the computational domain into parts that are updated sequentially, so only one sub-domain's worth of additional memory is necessary.

On the other hand, each timestep of this algorithm requires only one pass through memory, and  $\psi_{i,j}^{n+1}$  for each different pair  $i, j$  may be computed independently. The removal of multiple passes through memory, and of the sequential execution order required for in-place streaming, brings substantial performance improvements over the algorithms in Sec. 8, and make it vastly more suitable for implementation on graphical processing units (GPUs). A CUDA Fortran implementation of the algorithm (93a,b) achieves about 1660 million lattice updates per second for a  $4096 \times 4096$  lattice on an Nvidia K20c GPU with error correcting code (ECC) memory

disabled, equivalent to about 124 GBytes per second of useful memory traffic (calculated assuming each  $u$ ,  $d$ , and  $g$  value at time  $n$  is read once, and each  $u$  and  $d$  value at time  $n + 1$  is written once). These figures increase to 2370 million lattice updates per second and 141 GBytes per second for the algorithm in (92a,b) for a free particle that omits the  $\exp(i\Delta t g_{i,j})$  factor that requires an additional read from memory and a transcendental floating point operation.

## 10. THREE TIME LEVEL FORMULATION

A similar reformulation is possible in one spatial dimensional without introducing any additional storage. One may derive the Klein–Gordon equation

$$u_{tt} - u_{zz} = -m^2 u \quad (94)$$

from the one-dimensional Dirac equation for a free particle by applying  $(\partial_t - \partial_z)$  to the first order PDE for  $u$  and eliminating  $d$ . Applying an equivalent approach to the discrete algorithm

$$u_{i+1}^{n+1} = au_i^n + bd_i^n, \quad (95a)$$

$$d_{i-1}^{n+1} = ad_i^n - bu_i^n. \quad (95b)$$

leads to an algorithm that evolves  $u$  alone across three time levels.

We may solve (95a) for

$$d_i^n = (u_{i+1}^{n+1} - au_i^n)/b, \quad (96)$$

substitute into (95b), and solve for

$$d_{i-1}^{n+1} = (a/b)u_{i+1}^{n+1} - (b + a^2/b)u_i^n. \quad (97)$$

Equations (96) and (97) imply two different expressions for  $d_{i-1}^n$ . Imposing consistency between them determines

$$u_i^{n+1} = -(a^2 + b^2)u_i^{n-1} + a(u_{i-1,n} + u_{i+1,n}), \quad (98)$$

or

$$u_i^{n+1} = u_{i+1}^n + u_{i-1}^n - u_i^{n-1} - \frac{1}{2}\hat{m}^2\Delta t^2(u_{i-1,n} + u_{i+1,n}), \quad (99)$$

after defining the adjusted mass  $\hat{m} = m(1 + \Delta t^2 m^2/4)^{-1/2}$ . This is the Ablowitz–Kruskal–Ladik [68] scheme for the Klein–Gordon equation with the adjusted mass  $\hat{m}$ . For spatially uniform solutions, it reduces to the standard Verlet [69] leapfrog scheme

$$u^{n+1} - 2u^n + u^{n-1} = -\Delta t^2 \hat{m}^2 u^n, \quad (100)$$

for a harmonic oscillator  $u_{tt} = -\hat{m}^2 u$  with the adjusted mass  $\hat{m}$ . Alternatively, when  $m = 0$  it reduces to a standard finite difference scheme for the wave equation [70].

The scheme (99) is distinguished by evaluating the algebraic mass term using the average  $\frac{1}{2}(u_{i-1}^n + u_{i+1}^n)$  of the points used to approximate the spatial derivative  $u_{xx}$ , rather than introducing the additional point  $u_i^n$  into the stencil, as shown in figure 6. The scheme (99) thus describes evolution on two interwoven but uncoupled space-time grids connecting the values  $u_i^n$  for which  $i + n$  is either always even or always odd.

The discrete scheme (99) is an exact rewriting of the original quantum lattice algorithm (95a,b) for  $u_i^n$  and  $d_i^n$ . Instead of computing  $u_{i,\dots,N}^{n+1}$  and  $d_{i,\dots,N}^{n+1}$  from  $u_{i,\dots,N}^n$  and  $d_{i,\dots,N}^n$ , one stores two time levels  $u_{i,\dots,N}^n$  and  $u_{i,\dots,N}^{n-1}$  and computes  $u_{i,\dots,N}^{n+1}$ . One may reconstruct  $d_{i,\dots,N}^{n+1}$  from  $u_{i,\dots,N}^{n+1}$  and  $u_{i,\dots,N}^n$  using (97).

Moreover, the values at the  $n - 1$  time level only appear through  $u_i^{n-1}$  in the formula (99) for computing  $u_i^{n+1}$ . Since the two  $i$  values are the same, a concrete implementation may overwrite the value  $u_i^{n-1}$  stored in

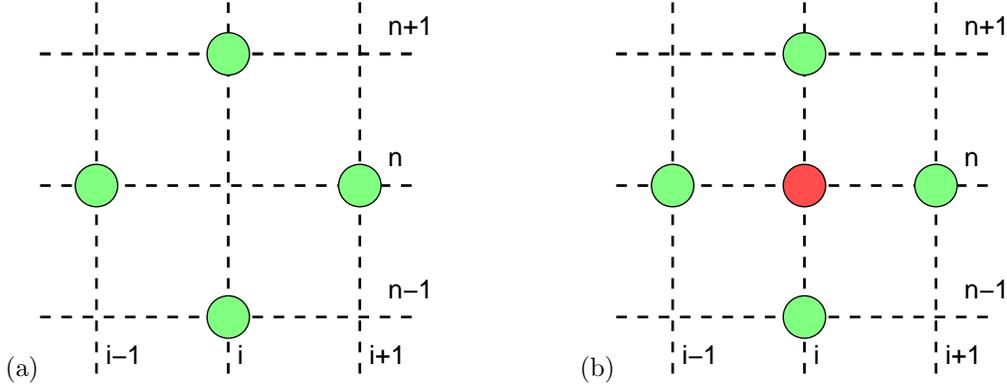


FIGURE 6. (a) Four point stencil for the three time level algorithm that computes  $u_i^{n+1}$  from  $u_{i\pm 1}^n$  and  $u_i^{n-1}$ . The algebraic mass term is computed from the average  $\frac{1}{2}(u_{i-1}^n + u_{i+1}^n)$ , rather than by enlarging the stencil to include the central point  $u_i^n$  shown in (b).

the array holding  $u_{i,\dots,N}^{n-1}$  with the newly computed value for  $u_i^{n+1}$ . This reformulation is therefore again much better suited to a multi-threaded parallel implementations, or implementations on GPUs since the calculations of the  $u_i^{n+1}$  over  $i = 1, \dots, N$  may be performed in any order. This is in distinct contrast to the requirement for either two sets of arrays or a specialised data layout and streaming algorithm (such as the algorithm by Bailey et al. [71]) for the direct implementation of (95a,b) on a GPU.

## 11. DISCRETE ACTION PRINCIPLE

The complex Klein–Gordon equation (94) may be obtained from the action integral

$$S = \int dt \mathcal{L} = \iint dt dz L = \iint dt dz \left( \frac{1}{2}|u_t|^2 - \frac{1}{2}|u_z|^2 - \frac{1}{2}m^2|u|^2 \right) \quad (101)$$

using the complex Hamilton's principle that varies  $u$  and its complex conjugate  $u^*$  independently [12, 14]

$$0 = \frac{\delta \mathcal{L}}{\delta u} = \frac{\partial}{\partial t} \frac{\partial L}{\partial u_t} + \frac{\partial}{\partial z} \frac{\partial L}{\partial u_z} - \frac{\partial L}{\partial u}, \quad 0 = \frac{\delta \mathcal{L}}{\delta u^*} = \frac{\partial}{\partial t} \frac{\partial L}{\partial u_t^*} + \frac{\partial}{\partial z} \frac{\partial L}{\partial u_z^*} - \frac{\partial L}{\partial u^*}. \quad (102)$$

This is equivalent to writing  $u = u_r + iu_i$  and varying the two real fields  $u_r$  and  $u_i$  independently. The action integral is the time integral of a Lagrangian functional  $\mathcal{L}$ , which in turn is the spatial integral of a Lagrangian density  $L$ .

The above three time level scheme (99) is the variational integrator derived from the discrete Hamilton's principle

$$\frac{\partial S}{\partial u_i^n} = 0, \quad \frac{\partial S}{\partial u_i^{n*}} = 0, \quad (103)$$

for the discrete action

$$S = \sum_{i,n} \left( \frac{1}{2} |u_i^{n+1} - u_i^n|^2 - \frac{1}{2} |u_{i+1}^n - u_i^n|^2 - \frac{1}{4} \tilde{m}^2 \Delta t^2 (u_i^n u_{i+1}^{n*} + u_i^{n*} u_{i+1}^n) \right). \quad (104)$$

We may identify each of the three terms as discrete approximations to the three terms representing kinetic, elastic, and potential energy in the continuous action (101) using the three points  $u_i^n$ ,  $u_{i+1}^n$ ,  $u_i^{n+1}$  and their complex conjugates.

The same discrete action may be rewritten more symmetrically as

$$S' = \sum_{i,n} \frac{1}{2} u_i^{n*} \left( u_{i-1}^n + u_{i+1}^n - u_i^{n+1} - u_i^{n-1} - \frac{1}{2} \tilde{m}^2 \Delta t^2 (u_{i-1}^n + u_{i+1}^n) \right) + \text{complex conjugate}. \quad (105)$$

This form emphasises the symmetry between the grid points  $u_{i\pm 1}^n$  and time levels  $u_i^{n\pm 1}$  in (99). Moreover, this expression for the action is written as  $u_i^{n*}$  multiplying the discrete evolution equation for  $u_i^n$ , plus  $u_i^n$  multiplying the discrete evolution equation for  $u_i^{n*}$ , so one may read off the evolution equations directly from the action.

The two expressions  $S$  and  $S'$  for the discrete action differ by the telescoping sum

$$S - S' = \sum_{i,n} \left( |u_i^{n+1}|^2 - |u_{i+1}^n|^2 + \frac{1}{4} \tilde{m}^2 \Delta t^2 (u_i^{n*} u_{i-1}^n - u_{i+1}^{n*} u_i^n + u_i^n u_{i-1}^{n*} - u_{i+1}^n u_i^{n*}) \right. \\ \left. + \frac{1}{2} (u_i^{n*} u_{i+1}^n - u_{i-1}^{n*} u_i^n + u_i^n u_{i+1}^{n*} - u_{i-1}^n u_i^{n*} + u_i^{n*} u_i^{n-1} - u_i^{n+1*} u_i^n + u_i^n u_i^{n-1*} - u_i^{n+1} u_i^{n*}) \right) \quad (106)$$

A telescoping sum is the discrete analogue of a null Lagrangian density  $L'$  for which  $\iint dt dz L' = 0$  in the presence of suitable (*e.g.* periodic) boundary conditions. For example, the complex Klein–Gordon equation may also be obtained from the continuous action

$$S' = -\frac{1}{4} \iint dt dz (u^* (u_{tt} - u_{zz} + m^2 u) + u (u_{tt}^* - u_{zz}^* + m^2 u^*)), \quad (107)$$

which is related to the previous expression  $S$  in (101) by

$$S - S' = \frac{1}{4} \iint dt dz \left( \frac{\partial}{\partial t} (u^* u_t + u u_t^*) - \frac{\partial}{\partial z} (u^* u_z + u u_z^*) \right). \quad (108)$$

## 12. DISCRETE ENERGY CONSERVATION PROPERTY

The quantum lattice algorithms also satisfy a discrete energy conservation property that complements the above discrete action principle [4]. We write the expectation of a general operator  $A$  as  $\langle A \rangle = \langle \psi | A \psi \rangle$  for a normalised state with  $\langle \psi | \psi \rangle = 1$ . This asymmetric notation is required when  $A$  is not Hermitian. The continuous time evolution of an expectation is given by Ehrenfest's formula [15]

$$\frac{d}{dt} \langle A \rangle = i \langle [H, A] \rangle + \langle \partial_t A \rangle, \quad (109)$$

where  $[H, A] = HA - AH$  is the commutator of  $A$  and  $H$ . The  $\langle \partial_t A \rangle$  term captures any explicit time-dependence in  $A$ . The Hamiltonian for a closed quantum system cannot depend explicitly on time, so Ehrenfest's formula establishes that  $\langle H \rangle$  is constant. This expresses the energy conservation property of a closed quantum system evolving in continuous time.

By contrast, the quantum lattice algorithms generate evolution in discrete time steps of length  $\Delta t$ . We may write  $|\psi^{n+1}\rangle = \tilde{E} |\psi^n\rangle$ , where  $|\psi^n\rangle$  and  $|\psi^{n+1}\rangle$  are the abstract states at times  $n$  and  $n+1$ . The discrete evolution operator  $\tilde{E}$  is unitary for a quantum system with no explicit time dependence. Unitary evolution implies conservation of probability,

$$\langle \psi^{n+1} | \psi^{n+1} \rangle = \langle \tilde{E} \psi^n | \tilde{E} \psi^n \rangle = \langle \psi^n | \tilde{E}^\dagger \tilde{E} \psi^n \rangle = \langle \psi^n | \psi^n \rangle, \quad (110)$$

where the third step  $\tilde{E}^\dagger \tilde{E} = 1$  uses the unitary property of  $\tilde{E}$ . Computing the expectation of the evolution operator itself gives

$$\langle \tilde{E} \rangle_{n+1} = \langle \psi^{n+1} | \tilde{E} \psi^{n+1} \rangle = \langle \tilde{E} \psi^n | \tilde{E} \tilde{E} \psi^n \rangle = \langle \psi^n | \tilde{E}^\dagger \tilde{E} \tilde{E} \psi^n \rangle = \langle \psi^n | \tilde{E} \psi^n \rangle = \langle \tilde{E} \rangle_n, \quad (111)$$

so the unitary evolution in discrete time steps generated by  $\tilde{\mathbf{E}}$  conserves the expectation  $\langle \tilde{\mathbf{E}} \rangle = \langle \psi | \tilde{\mathbf{E}} \psi \rangle$ .

The discrete evolution operator  $\tilde{\mathbf{E}}$  approximates the formal solution operator  $\mathbf{E}_{\Delta t} = \exp(-i\Delta t \mathbf{H})$  for the evolution of the continuous-time system  $i\partial_t \psi = \mathbf{H}\psi$  over a timestep  $\Delta t$ . This evolution conserves the expectation  $\langle \mathbf{E}_{\Delta t} \rangle$ , since  $\mathbf{E}_{\Delta t}$  commutes with  $\mathbf{H}$ . Formally expanding  $\langle \mathbf{E}_{\Delta t} \rangle$  for small  $\Delta t$  gives

$$\langle \psi | \exp(-i\Delta t \mathbf{H}) \psi \rangle = \langle \psi | \psi \rangle - i\Delta t \langle \psi | \mathbf{H} \psi \rangle - \Delta t^2 \langle \mathbf{H} \psi | \mathbf{H} \psi \rangle + \mathcal{O}(\Delta t^3). \quad (112)$$

The leading order term  $\langle \psi | \psi \rangle$  is the total probability, while the first correction (and the leading order imaginary part) is proportional to the expectation of the Hamiltonian. Conservation of the expectation of the evolution operator  $\mathbf{E}_{\Delta t}$  thus encapsulates both conservation of total probability (unitary evolution) and conservation of energy.

The discrete evolution operator  $\tilde{\mathbf{E}}$  is a unitary approximation to  $\mathbf{E}_{\Delta t}$ , so the conservation of  $\langle \tilde{\mathbf{E}} \rangle$  under the discrete evolution implies conservation of a consistent discrete approximation to  $\langle \mathbf{H} \rangle$ . If  $\tilde{\mathbf{E}}$  is a first-order in  $\Delta t$  approximation to  $\mathbf{E}_{\Delta t}$ ,

$$i \frac{1}{\Delta t} \langle \tilde{\mathbf{E}} - 1 \rangle = \langle \mathbf{H} \rangle + \mathcal{O}(\Delta t), \quad (113)$$

and if  $\tilde{\mathbf{E}}$  is a second-order approximation,

$$\text{Re} \left\{ i \frac{1}{\Delta t} \langle \tilde{\mathbf{E}} - 1 \rangle \right\} = \langle \mathbf{H} \rangle + \mathcal{O}(\Delta t^2). \quad (114)$$

We now apply these general results to the quantum lattice algorithms for the one-dimensional Dirac equations. Some three-dimensional applications may be found in [4].

### 12.1. One-dimensional Dirac equation

The action integral for the Dirac equation may be written in a form analogous to (107) as [12, 14]

$$\mathcal{S}_{\text{Dirac}} = \iint dt dV \psi^\dagger \left( i \frac{\partial \psi}{\partial t} - \mathbf{H} \psi \right). \quad (115)$$

Treating  $\psi$  and  $\psi^\dagger$  as independent variables and varying  $\psi^\dagger$  gives the Dirac equation in the form  $i\partial_t \psi = \mathbf{H}\psi$ . This action may be written in a more symmetric form between  $\psi$  and  $\psi^\dagger$ , analogous to (101), by adding an exact time derivative. The second term in the Lagrangian density is minus the expectation of the Hamiltonian operator,

$$\langle \mathbf{H} \rangle = \langle \psi | \mathbf{H} \psi \rangle = \int dz \psi^\dagger \mathbf{H} \psi, \quad (116)$$

which is equal to the Hamiltonian functional in classical Lagrangian particle mechanics.

For the 1D Dirac equation written in the form (32) the expectation of the Hamiltonian is

$$\langle \mathbf{H} \rangle = \int dz \left( u^* (-iu_z + imd - gu) + d^* (id_z - imu - gd) \right). \quad (117)$$

This is easier to interpret in the variables  $\Phi^\pm = (u \pm id)/\sqrt{2}$ ,

$$\langle \mathbf{H} \rangle = \int dz \left( m (|\Phi^+|^2 - |\Phi^-|^2) - g (|\Phi^+|^2 + |\Phi^-|^2) - i (\Phi^{+\ast} \partial_z \Phi^- + \Phi^{-\ast} \partial_z \Phi^+) \right), \quad (118)$$

as the mass terms for  $\Phi^\pm$  describe positive and negative energy states. The terms involving spatial derivatives in both expressions may be written symmetrically between the variables  $u$  and  $d$ , or  $\Phi^\pm$ , and their complex conjugates by adding exact derivatives with respect to  $z$ .

These expressions differ from the Hamiltonian functional for the Klein–Gordon equation one would obtain from the action (101). In particular, the Hamiltonian density  $(1/2)(|u_t|^2 + |u_z|^2 + m^2|u|^2)$  for the Klein–Gordon equation is positive definite, while (117) and (118) are not. Each component of the wavefunction in the Dirac equation individually satisfies the Klein–Gordon equation, which just expresses the relativistic relation  $E^2 = p^2 + m^2$  in natural units. The distinguishing property of the Dirac equation as a description for spin-1/2 particles lies in its structure as a first-order system for a multi-component wavefunction [14].

Figure 7 shows the evolution of the expectation of a natural discrete approximation to the Hamiltonian (117),

$$\langle \tilde{H} \rangle = \Delta z \sum_j \left( u_j^* (-i u_{zj} + i m d_j - g_j u_j) + d_j^* (i d_{zj} - i m u_j - g_j d_j) \right), \quad (119)$$

for the initial value problem described in Sec. 4.1 simulated using the second order “exact collisions” formulation of Sec. 7 on three different lattices. The quantities  $u_j$ ,  $d_j$ ,  $g_j$  are the discrete approximations to  $u$ ,  $d$ ,  $g$  at lattice points obtained by applying  $\mathcal{C}^{1/2}$  to the variables in the stream-collide algorithm. The derivatives  $u_{zj}$  and  $d_{zj}$  were obtained using the Fourier differentiation matrix described in Sec. 4.1. The sum over  $j$  corresponds to an approximation of the spatial integral in (117) by the trapezoidal rule. This is exponentially accurate for a periodic domain, being equivalent to expressing the integrand as a Fourier series and integrating each term analytically.

The expectations of  $\tilde{H}$  in figure 7 show oscillations whose amplitudes are proportional to  $\Delta z^2$ . By contrast, the discrete approximations to  $\langle H \rangle$  given by  $-\text{Im} \langle \tilde{E} \rangle_n / \Delta t$  with

$$\langle \tilde{E} \rangle_n = \sum_j u_j^{n*} u_j^{n+1} + d_j^{n*} d_j^{n+1} \quad (120)$$

show no oscillations when plotted on the same axes. The differences between these approximations and the analytical value of  $\langle H \rangle$  for the given initial conditions is also proportional to  $\Delta z^2$ , although the numerical coefficient is substantially larger than for  $\langle \tilde{H} \rangle$ . However,  $\langle \tilde{E} \rangle_n$  is a discrete invariant, while  $\langle \tilde{H} \rangle$  is not.

Conversely, figure 8 shows the evolution of  $\langle \tilde{H} \rangle$  for three different algorithms – first order, second order, and second order with exact collisions – for the initial value problem in Sec. 4.1 on a lattice with 4096 points. Although figure 2 shows that the second order algorithm with Crank–Nicolson collisions gives virtually no improvement in the error in the wavefunction at this resolution, this second order algorithm gives much smaller oscillations in  $\langle \tilde{H} \rangle$ . To the extent that the constancy of  $\langle \tilde{H} \rangle$  is a useful measure of solution quality, this measure is greatly improved by adopting a second order algorithm even for coarse resolutions.

### 13. CONCLUSION

Quantum lattice algorithms have followed three parallel and largely disconnected lines of development under the names discrete time quantum walks, quantum lattice gas automata, and quantum lattice Boltzmann equations. The states in quantum lattice gas automata are already continuous complex variables, so there is no distinction like that between the Boolean variables of lattice gas automata and the continuous real variables in their Boltzmann approximations. However, the first two of the three are currently primarily concerned with “in principle” properties of algorithms suitable for implementation on quantum computers, rather than constrained by finite computational resources on conventional digital computers. In particular, there has been little interest in improving upon the formal first order in time accuracy achieved from a simple juxtaposition SC of operators generating “coin flips” C and classical deterministic shift operations S.

The accuracy in one dimension may be improved to second order by pre- and post-processing the states using the square root of the coin flip transformation. This is very closely related to the  $f_i$  to  $\bar{f}_i$  transformation introduced by [51], which has since been rederived as a second-order splitting between decoupled streaming and collision steps [63]. However, the second-order splitting interpretation leads naturally to a unitary relation between barred and unbarred variables, and motivates the use of the exact solution for the evolution under the

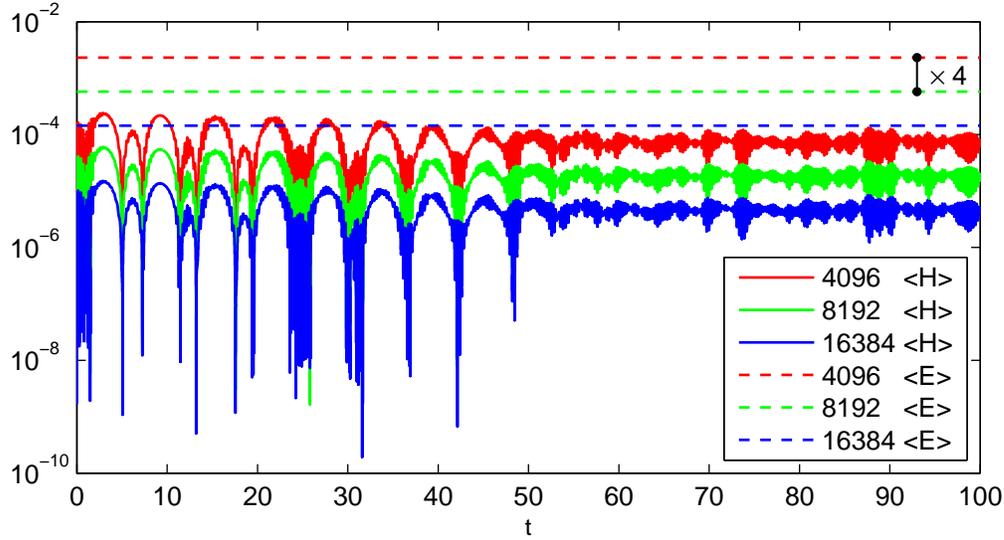


FIGURE 7. Evolution in the differences between  $\langle \tilde{H} \rangle$  and  $-\text{Im} \langle \tilde{E} \rangle / \Delta t$  from the exact  $\langle H \rangle$  of the initial conditions for three different resolutions with 4096, 8192, and 16384 points. The natural discrete approximation  $\langle \tilde{H} \rangle$  oscillates (solid lines), with an amplitude that decreases with increasing resolution, while  $-\text{Im} \langle \tilde{E} \rangle / \Delta t$  remains constant (dotted lines).

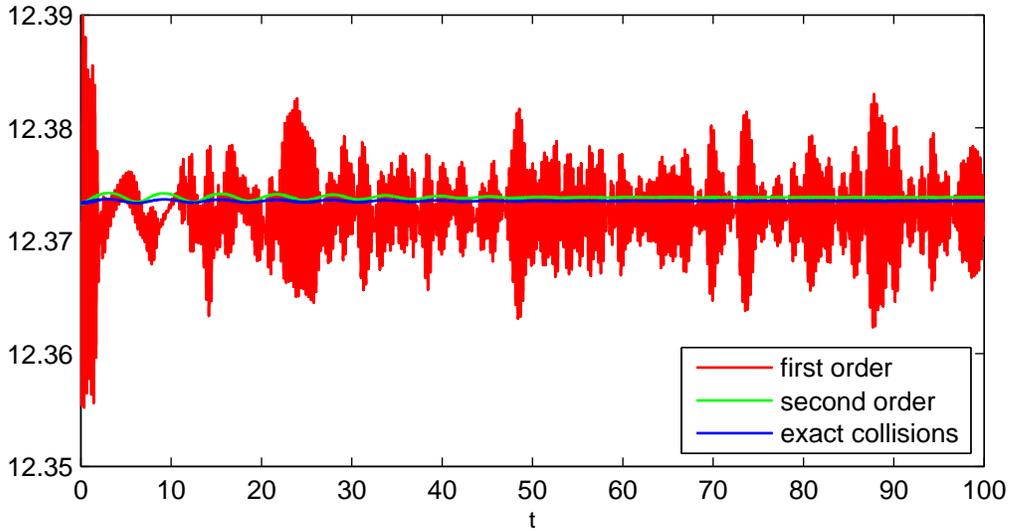


FIGURE 8. Evolution of  $\langle \tilde{H} \rangle$  for the first order, second order, and exact collision algorithms, all with 4096 points.

coin flip transformation in place of the Crank–Nicolson approximation uses in the “quantum lattice Boltzmann” algorithm. This incurs a negligible performance penalty on modern hardware such as recent Nvidia GPUs.

Turning to multiple spatial dimensions, the Dirac equation differs from the discrete Boltzmann equation through having characteristic surfaces, the light cones of special relativity, rather than characteristic curves.

A multi-dimensional implementation must therefore rely upon splitting between dimensions, which is typically only first-order accurate. An approach based on multiple applications of  $(\mathcal{X}^{1/2}\mathcal{Y}\mathcal{X}^{1/2})^n = \mathcal{X}^{1/2}(\mathcal{Y}\mathcal{X})^n\mathcal{X}^{-1/2}$  may be applied in two dimensions, but does not extend into three dimensions.

Instead, a different approach replaces the evolution through a series of unitary intermediate steps with an expression for the wavefunction values at the most recent time level  $n + 1$  in terms of values of wavefunctions at a few preceding time levels,  $n$  or  $n$  and  $n - 1$ , and neighbouring grid points, which is equivalent to a short path integral formulation. These expressions take the form of finite difference formulae that approximate the exponential  $\exp(-\Delta t\sigma \cdot \nabla)$  of the Dirac streaming operator, with a particular structure that ensures overall unitary evolution. Requiring only a single pass through memory per timestep, and allowing asynchronous updates of different grid points without ordering requirements, these algorithms provide excellent performance on GPU hardware.

#### 14. ACKNOWLEDGEMENTS

Some of this research was supported by an Advanced Research Fellowship from the UK Engineering and Physical Sciences Research Council [grant number EP/E054625/1]. The computations employed the Advanced Research Computing facilities at the University of Oxford, and the Emerald GPU cluster at the Science & Engineering South (SES) Centre for Innovation.

#### REFERENCES

- [1] R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals* (McGraw–Hill, New York, 1965).
- [2] W. Heisenberg, *Die Selbstenergie des Elektrons*, Z. Phys. **65**, 4 (1930).
- [3] B. Carazza and H. Kragh, *Heisenberg’s lattice world: The 1930 theory sketch*, Amer. J. Phys. **63**, 595 (1995).
- [4] P. J. Dellar, *An exact energy conservation property of the quantum lattice Boltzmann algorithm*, Phys. Lett. A **376**, 6 (2011).
- [5] H.-L. Lai and C.-F. Ma, *An implicit scheme of lattice Boltzmann method for sine-Gordon equation*, Chinese Phys. Lett. **25**, 2118 (2008).
- [6] B. Shi and Z. Guo, *Lattice Boltzmann model for nonlinear convection-diffusion equations*, Phys. Rev. E **79**, 016701 (2009).
- [7] L. Zhong, S. Feng, P. Dong, and S. Gao, *Lattice Boltzmann schemes for the nonlinear Schrödinger equation*, Phys. Rev. E **74**, 036704 (2006).
- [8] S. Ponce Dawson, S. Chen, and G. D. Doolen, *Lattice Boltzmann computations for reaction-diffusion equations*, J. Chem. Phys. **98**, 1514 (1993).
- [9] A. H. Castro Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov, and A. K. Geim, *The electronic properties of graphene*, Rev. Mod. Phys. **81**, 109 (2009).
- [10] D. Lapitski, *Development of the Quantum Lattice Boltzmann method for simulation of quantum electrodynamics with applications to graphene*, DPhil thesis, University of Oxford (2014).
- [11] W. Pauli, *General Principles of Quantum Mechanics* (Springer, Berlin; Heidelberg; New York, 1980), translated by P. Achuthan & K. Venkatesan from articles in Handbuch der Physik (1933 and 1958).
- [12] J. D. Bjorken and S. D. Drell, *Relativistic Quantum Fields* (McGraw–Hill, New York; London, 1964).
- [13] V. B. Berestetskii, E. M. Lifshitz, and L. P. Pitaevskii, *Quantum Electrodynamics* (Butterworth–Heinemann, 1982), 2nd ed.
- [14] L. H. Ryder, *Quantum Field Theory* (Cambridge University Press, Cambridge, 1996), 2nd ed.
- [15] L. D. Landau and E. Lifshitz, *Quantum Mechanics: Non-Relativistic Theory* (Butterworth–Heinemann, 1977), 3rd ed.
- [16] N. G. van Kampen, *Chapman–Enskog as an application of the method for eliminating fast variables*, J. Statist. Phys. **46**, 709 (1987).

- [17] P. J. Dellar, *Macroscopic descriptions of rarefied gases from the elimination of fast variables*, Phys. Fluids **19**, 107101 (2007).
- [18] L. L. Foldy and S. A. Wouthuysen, *On the Dirac theory of spin 1/2 particles and its non-relativistic limit*, Phys. Rev. **78**, 29 (1950).
- [19] J. P. Costella and B. H. J. McKellar, *The Foldy–Wouthuysen transformation*, Amer. J. Phys. **63**, 1119 (1995).
- [20] M. Reiher, *Relativistic Douglas–Kroll–Hess theory*, WIREs Comput. Mol. Sci. **2**, 139 (2012).
- [21] G. B. Whitham, *Linear and Nonlinear Waves* (Wiley Interscience, New York, 1974).
- [22] J. F. Reynolds, *A proof of the random-walk method for solving Laplace’s equation in 2-D*, Math. Gazette **49**, 416 (1965).
- [23] J. Glimm, *Solutions in the large for nonlinear hyperbolic systems of equations*, Comm. Pure Appl. Math. **18**, 697 (1965).
- [24] Y. Aharonov, L. Davidovich, and N. Zagury, *Quantum random walks*, Phys. Rev. A **48**, 1687 (1993).
- [25] D. A. Meyer, *From quantum cellular automata to quantum lattice gases*, J. Statist. Phys. **85**, 551 (1996).
- [26] L. K. Grover, *A fast quantum mechanical algorithm for database search*, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 1996), STOC ’96, pp. 212–219.
- [27] L. K. Grover, *From Schrödinger’s equation to the quantum search algorithm*, Amer. J. Phys. **69**, 769 (2001).
- [28] M. A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and A. D. Spielman, *Exponential algorithmic speedup by a quantum walk*, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 2003), STOC ’03, pp. 59–68.
- [29] A. Ambainis, *Quantum walks and their algorithmic applications*, Int. J. Quant. Inf. **1**, 507 (2003).
- [30] J. Kempe, *Quantum random walks: An introductory overview*, Contemp. Phys. **44**, 307 (2003).
- [31] V. Kendon, *Decoherence in quantum walks – a review*, Math. Struct. Comput. Sci. **17**, 1169 (2007).
- [32] N. Shenvi, J. Kempe, and K. B. Whaley, *Quantum random-walk search algorithm*, Phys. Rev. A **67**, 052307 (2003).
- [33] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon, *Universal quantum computation using the discrete-time quantum walk*, Phys. Rev. A **81**, 042330 (2010).
- [34] A. M. Childs, *Universal computation by quantum walk*, Phys. Rev. Lett. **102**, 180501 (2009).
- [35] C. A. Ryan, M. Laforest, J. C. Boileau, and R. Laflamme, *Experimental implementation of a discrete-time quantum random walk on an NMR quantum-information processor*, Phys. Rev. A **72**, 062317 (2005).
- [36] D. A. Meyer, *Quantum mechanics of lattice gas automata: One-particle plane waves and potentials*, Phys. Rev. E **55**, 5261 (1997).
- [37] F. W. Strauch, *Relativistic effects and rigorous limits for discrete- and continuous-time quantum walks*, J. Math. Phys. **48**, 082102 (2007).
- [38] E. Farhi and S. Gutmann, *Quantum computation and decision trees*, Phys. Rev. A **58**, 915 (1998).
- [39] I. Bialynicki-Birula, *Weyl, Dirac, and Maxwell equations on a lattice as unitary cellular automata*, Phys. Rev. D **49**, 6920 (1994).
- [40] B. M. Boghosian and W. Taylor, *Quantum lattice-gas model for the many-particle Schrödinger equation in  $D$  dimensions*, Phys. Rev. E **57**, 54 (1998).
- [41] D. A. Meyer, *Quantum computing classical physics*, Phil. Trans. Roy. Soc. A: **360**, 395 (2002).
- [42] J. Watrous, *On one-dimensional quantum cellular automata*, in *36th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, USA, 1995), pp. 528–537.
- [43] B. Schumacher and R. F. Werner, *Reversible quantum cellular automata*, arXiv:quant-ph/0405174 (2004).
- [44] P. Arrighi, V. Nesme, and R. Werner, *One-dimensional quantum cellular automata over finite, unbounded configurations*, Lect. Notes Comput. Sci. **5196**, 64 (2008).
- [45] A. Shakeel and P. J. Love, *When is a quantum cellular automaton (QCA) a quantum lattice gas automaton (QLGA)?*, J. Math. Phys. **54**, 092203 (2013).

- [46] J. Yepez, *Relativistic path integral as a lattice-based quantum algorithm*, Quant. Informat. Process. **4**, 471 (2005).
- [47] J. Yepez, *Quantum Algorithms for Computational Physics: Volume 3 of Lattice Gas Dynamics*, Tech. Rep. AFRL-VS-HA-TR-2006-1143, Air Force Research Laboratory, Hanscom, MA (2007), available from <http://www.dtic.mil/docs/citations/ADA474659>.
- [48] J. Yepez, G. Vahala, L. Vahala, and M. Soe, *Superfluid turbulence from quantum Kelvin wave to classical Kolmogorov cascades*, Phys. Rev. Lett. **103**, 084501 (2009).
- [49] T. Inamuro, M. Yoshino, and F. Ogino, *Accuracy of the lattice Boltzmann method for small Knudsen number with finite Reynolds number*, Phys. Fluids **9**, 3535 (1997).
- [50] Y. Sone, *Kinetic Theory and Fluid Dynamics* (Birkhäuser, Boston, 2002).
- [51] X. He, S. Chen, and G. D. Doolen, *A novel thermal model of the lattice Boltzmann method in incompressible limit*, J. Comput. Phys. **146**, 282 (1998).
- [52] S. Succi and R. Benzi, *Lattice Boltzmann equation for quantum mechanics*, Physica D **69**, 327 (1993).
- [53] S. Succi, *Numerical solution of the Schrödinger equation using discrete kinetic theory*, Phys. Rev. E **53**, 1969 (1996).
- [54] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev. **20**, 801 (1978).
- [55] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev. **45**, 3 (2003).
- [56] B. R. Sidje, *Expokit: a software package for computing matrix exponentials*, ACM Trans. Math. Softw. **24**, 130 (1998).
- [57] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM. J. Matrix Anal. Appl. **26**, 1179 (2005).
- [58] R. Vichnevetsky and J. B. Bowles, *Fourier Analysis of Numerical Approximations of Hyperbolic Equations* (SIAM, Philadelphia, 1982).
- [59] L. Trefethen, *Group velocity in finite difference schemes*, SIAM Rev. **24**, 113 (1982).
- [60] H. B. Nielsen and M. Ninomiya, *Absence of neutrinos on a lattice: (I) Proof by homotopy theory*, Nucl. Phys. B **185**, 20 (1981).
- [61] H. B. Nielsen and M. Ninomiya, *Absence of neutrinos on a lattice: (II) Intuitive topological proof*, Nucl. Phys. B **193**, 173 (1981).
- [62] H. Nielsen and M. Ninomiya, *A no-go theorem for regularizing chiral fermions*, Phys. Lett. B **105**, 219 (1981).
- [63] P. J. Dellar, *An interpretation and derivation of the lattice Boltzmann method using Strang splitting*, Comput. Math. Applic. **65**, 129 (2013).
- [64] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal. **5**, 506 (1968).
- [65] S. Palpacelli and S. Succi, *Numerical validation of the quantum lattice Boltzmann scheme in two and three dimensions*, Phys. Rev. E **75**, 066704 (2007).
- [66] P. J. Dellar, D. Lapitski, S. Palpacelli, and S. Succi, *Isotropy of three-dimensional quantum lattice Boltzmann schemes*, Phys. Rev. E **83**, 046706 (2011).
- [67] D. Lapitski and P. J. Dellar, *Convergence of a three-dimensional quantum lattice Boltzmann scheme towards solutions of the Dirac equation*, Phil. Trans. Roy. Soc. A **369**, 2155 (2011).
- [68] M. Ablowitz, M. Kruskal, and J. Ladik, *Solitary wave collisions*, SIAM J. Appl. Math. **36**, 428 (1979).
- [69] L. Verlet, *Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*, Phys. Rev. **159**, 98 (1967).
- [70] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations* (Cambridge University Press, Cambridge, 1996).
- [71] P. Bailey, J. Myre, S. D. C. Walsh, D. J. Lilja, and M. O. Saar, *Accelerating lattice Boltzmann fluid flow simulations using graphics processors*, in *International Conference on Parallel Processing* (IEEE Computer Society, Washington, DC, USA, 2009), pp. 550–557.