Solving the Nonlinear Vlasov Equation on a Quantum Computer

Tamás Vaszary[1,2,3], Animesh Datta[1], Tom Goffrey[1], and Brian Appelbe[2]

[1]Department of Physics, University of Warwick, Coventry CV4 7AL, United Kingdom

[2]Department of Physics, Imperial College London, London SW7 2AZ, United Kingdom

[3]Mathematical Institute, University of Oxford, Oxford OX2 6GG, United Kingdom

We present a mapping of the nonlinear, electrostatic Vlasov equation with Krook-type collision operators, discretized on a $(1+1)$ dimensional grid, onto a recent Carleman linearization-based quantum algorithm for solving ordinary differential equations (ODEs) with quadratic nonlinearities. We derive upper bounds for the query- and gate complexities of the quantum algorithm in the limit of large grid sizes. We conclude that these are polynomially larger than the time complexity of the corresponding classical algorithms. We find that this is mostly due to the dimension, sparsity and norm of the Carleman linearized evolution matrix. We show that the convergence criteria of the quantum algorithm places severe restrictions on potential applications. This is due to the high level of dissipation required for convergence, that far exceeds the physical dissipation effect provided by the Krook operator for typical plasma physics applications.

1   Introduction

The ability to effectively simulate and predict plasma behaviour is key to our understanding of a rich variety of physical phenomena, such as high-energy astrophysics, as well as magnetic- and inertial confinement fusion. However, the capacities of current classical supercomputers often prove to be insufficient for large-scale plasma simulations because of the inherently multi-scale nature of the problem, where fluid and kinetic scales are coupled across spatial and temporal domains. Alternatives to classical computation are thus desirable. One of the candidates is quantum computation, which offers more efficient algorithms for certain problems in terms of computational complexity [24, 45]. Consequently, quantum computation is beginning to attract attention in the plasma physics community. Recent reviews [13, 29, 56] identify some of the challenges of mapping plasma problems onto the architecture of quantum computers. The most notable difficulty that arises in the construction of such mappings is the fact that while plasma models are generally nonlinear, quantum computers are naturally suited to linear problems only. From a mathematical viewpoint this marks an enormous difference: something that is explicitly written vs. something that has no closed-from solution. So far this difficulty was mitigated by only concentrating on plasma problems in the linearized regime [2, 10, 14, 37–39]. However, the practicality of this approach is limited as most plasma phenomena of scientific relevance, such as turbulence, shock waves and wave-particle interactions are strongly nonlinear [9].

Tamás Vaszary: Tamas.Vaszary@maths.ox.ac.uk

Animesh Datta: animesh.datta@warwick.ac.uk

Quantum approaches to nonlinear dynamics have gained more attention in the field of computational fluid mechanics. For the Navier-Stokes equation, a specific method was introduced in Refs. [19, 20], with a broader discussion provided in Ref. [4]. The convergence of the Carleman linearization for the Navier-Stokes equations was examined in Ref. [23], following a methodology similar to the one we describe later. The quantum solution of simplified fluid models, such as the lattice-Boltzmann formulation and Burger's equation, have also been successfully achieved using Carleman linearization, a technique we revisit in this work [12, 27, 28, 33, 41, 44].

The Vlasov equation, which we focus on in this paper, has been particularly important for the simulation of nonlinear plasma phenomena. Unlike macroscopic fluid-based descriptions, the Vlasov equation emerges from kinetic theory and provides a significantly more detailed way of capturing plasma dynamics. Classical algorithms that solve the Vlasov equation (coupled to Maxwell's equations) include grid-based and operator-splitting methods, as well as particle-in-cell simulations [5, 8, 11, 46]. The Vlasov-Maxwell system is a family of coupled nonlinear partial differential equations (PDEs).

The first quantum algorithm to solve nonlinear differential equations had a runtime scaling exponentially with simulation time [32]. This was later improved to quadratic [36] but without proving the correctness of their approach. More recently, the mapping of nonlinear classical dynamics to linear quantum dynamics has been investigated [15, 21, 30, 54, 55], however these typically do not provide generally applicable methods to perform the mappings of arbitrary (polynomially) nonlinear dynamics.

Another direction with the aim of procedural implementation lies in embedding methods such as, homotopy perturbation [53] or Carleman linearization [6, 22, 31, 34, 35, 48]. These capitalize on a fairly thorough understanding of quantum algorithms for solving linear differential equations [3]. They lead to quantum algorithms offering favourable query- and gate complexity scaling with respect to simulation time, the number of dynamical variables and the allowed error on the solution. This scaling and the procedural applicability comes at the cost of two limitations. Firstly, the linear part of the evolution is required to be dissipative, meaning that the norm of the solution vector decreases in time[1]. Secondly, the ratio of the combined strength of nonlinear and inhomogeneous effects to that of linear dissipation is required to be small. This is typically formulated as $R < 1$ for some so-called convergence parameter R quantifying the mentioned ratio.

In this paper, we provide a quantum algorithm for solving the nonlinear Vlasov equation in the non-relativistic, electrostatic limit (Eq. (1)). It is based on a quantum algorithm due to Krovi [31]. For simplicity, we focus on one position and one velocity dimension. The two-dimensional continuous phase space is discretized and the Vlasov equation is mapped to a finite difference equation, compatible with the mathematical framework of Ref. [31]. The electrostatic interaction is captured by Gauss's law[2]. This results in a family of ordinary differential equations (ODEs) with a quadratic nonlinearity. We embed these into an infinitely large family of linear ODEs through Carleman linearization [7]. After truncation to a finite family, it is evolved by a Taylor series-based high-order time integrator. Due to

---

[1] While this limitation is captured differently across the literature in mathematical terms, qualitatively they capture the same principle. This also holds for the next limitation. For an error analysis without the dissipativity condition, see Ref. [52].

[2] We show that when the Vlasov equation is coupled to Gauss's law, the constraints mentioned in the previous paragraph are satisfied for certain regimes of plasma parameters as long as collisions make the system dissipative enough. On the other hand, when coupling it to the otherwise physically equivalent Ampere's law, we find these convergence criteria to be unsatisfiable regardless of the values of the plasma parameters. See Section 8.

the time discretization, the dynamics is represented as a matrix inversion problem, which we solved with a Quantum Linear Solver Algorithm (QLSA).

Our work reaches the following conclusions. First, we find that the dissipation mechanism of the plasma (originating from collisions and modeled by a Krook-type collision operator) directly translates into the dissipation of the nonlinear ODE, which is required for the convergence of Carleman linearization. We also find that physically it is far too weak to balance out the R parameter for realistic grid-sizes. This is because its numerator contains $\|F^{(2)}\|$, the norm of the matrix encoding the nonlinearity. It contains a double integral in our case that computes the electric field from Gauss's law, and causes $\|F^{(2)}\|$ to grow with grid-size. Subsection 5.3 gives an explicit upper bound on the number of velocity grid points in terms of plasma parameters, such as temperature and the size of the phase space simulation box. We explicitly use typical numerical parameters from inertial confinement fusion and interstellar plasma to illustrate the severity of this restriction.

Second, we find that the query- and gate complexities are polynomially larger than the time complexity of the simple classical implementation of the same finite difference scheme, with respect to error and simulation time, in the asymptotic limit of large grid sizes. The main contributors to these complexities are the number of required Carleman linearization steps enlarging the dimension of the linearized system, the norm of the matrix encoding that system, and finally the sparsity. The latter scales linearly with system size due to Gauss's law containing a double integral over phase space. The upper bounds for the final complexities are presented in Eqs. (71-72).

Feed into quantum algorithm

1. Choose plasma model:
$$\frac{\partial}{\partial t} f(x, v, t) = H(f(x', v', t), x, v)$$

2. Discretize on a grid: $f(x_i, v_j, t) \to f_{ij}(t)$ st.:
$$\frac{d}{dt} f_{ij} = h(f_{i'j'}, i, j)$$

3. Derive state vector u and matrices $F^{(k)}$ st.:
$$\frac{du}{dt} = F^{(2)} u^{\otimes 2} + F^{(1)} u + F^{(0)}$$

4. Check for convergence of quantum algorithm:
$$\mu(F^{(1)}) < 0, \quad R < 1$$

5. Carleman linearize ODE system:
$$\frac{dx}{dt} = Ax + b$$

6. Discretize time evolution to obtain:
$$L|y\rangle = |\psi_{\text{in}}\rangle$$

7. Apply quantum linear solver:
$$|y\rangle = L^{-1}|\psi_{\text{in}}\rangle$$
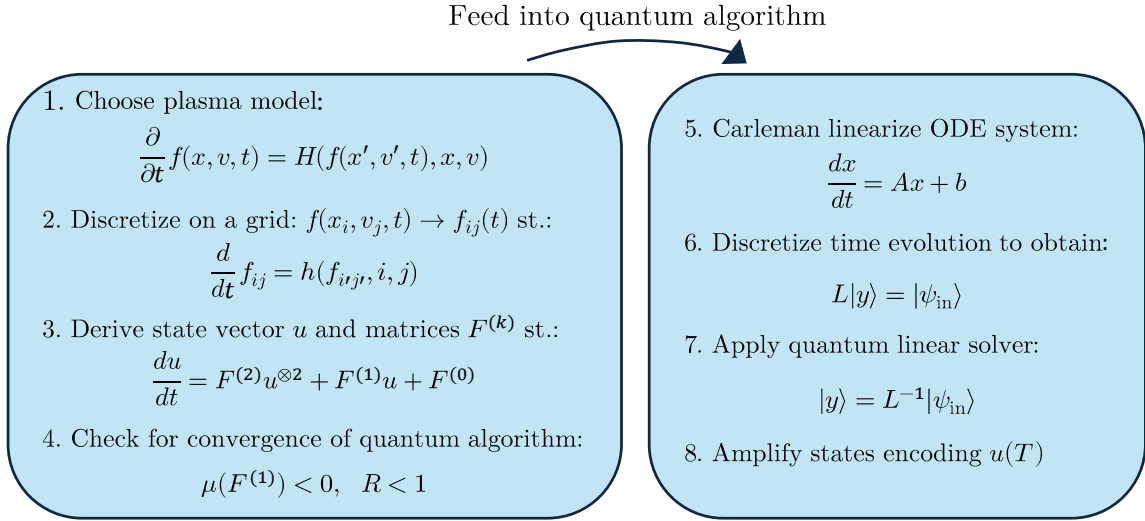
8. Amplify states encoding $u(T)$

Figure 1: A schema of the procedure used in this paper. Steps 7 and 8 are the uniquely quantum ones.

Our procedure is summarized in Fig. 1. It presents the first step on the path to solving general plasma physics problems on quantum computers. The approach taken builds on the preliminary results developed in previous work [50].

This paper is structured as follows. For clarity, we first establish the notation used in the paper in Table 1. In Section 2, we present the finite difference equation to be solved. In Section 3, we summarize the relevant parts of the quantum algorithm from Ref. [31] for completeness. In Section 4, we construct the structure of the mapping between the Vlasov equation coupled to Gauss's law and the input of Ref. [31]. In Section 5, we investigate the behaviour of the convergence criteria specifically for the finite difference equations in Section 2.3. In Section 6, we calculate the propagation of the phase space discretization

error through the algorithm. In Section 7, we present the upper bounds of the complexities of our quantum algorithm. In Section 8 we analyse how coupling to Ampere's law instead of Gauss's law affects the convergence of Carleman linearization. We finish by discussing our results in Section 9 and concluding in Section 10.

Symbol	Meaning	Symbol	Meaning
v_n	n-th entry of vector v	$\kappa(A)$	condition number of matrix A
$\|v\|$	l_2 norm of vector v	$\text{vec}(A)$	vectorization of matrix A
A_{ij}	ij-th entry of matrix A	$\mathbb{I}$	identity matrix
$\|A\|_k$	induced l_k norm of matrix A	$\lceil x \rceil$	ceiling of number x
$\|A\| = \|A\|_2$	spectral norm of matrix A	$x /\!/ y$	$x - y\left(\lceil x/y \rceil - 1\right)$
$\|A\|_F$	Frobenius norm of matrix A	$\text{erf}(x)$	$(2/\sqrt{\pi})\int_0^x \exp(-t^2)dt$
$\mu(A)$	log-norm of matrix A	δ_{ij}	Kronecker delta
$\alpha(A)$	spectral abscissa of matrix A	$\oplus$	direct sum

Table 1: Summary of the nontrivial notation. More details are in Appendix A

In Appendix A we define the nontrivial mathematical expressions. In Appendix B we state the entries of the matrices the mapping resulted in. In Appendices C and D we derive quantities from these needed for the complexity. Finally, in Appendix E we carry out the error and complexity analysis of the classical solution of the problem.

2 The Vlasov equation

2.1 The model

The Vlasov equation describes the time evolution of the phase space distribution function $f(x, v, t) \in \mathbb{R}_{\geq 0}$, where $x \in \mathbb{R}$ is position and $v \in \mathbb{R}$ is velocity. We restrict ourselves to one space and one velocity dimension and apply it to a plasma with one dynamical species, electrons, with charge $-q < 0$ and mass m_e. An additional fixed, neutralising and stationary ion background species is denoted by $f^{\text{bg}}(x, v) = f^{\text{bg}}(v)$, which is taken to be uniform in space. This choice does not affect the convergence of our quantum algorithm. Also note that we use SI units throughout this paper. Then the Vlasov equation for $f(x, v, t)$ in the electrostatic limit reads

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} - \frac{qE}{m_e}\frac{\partial f}{\partial v} = C[f], \tag{1}$$

where $C[f]$ is a collision operator that relaxes f into a target distribution, conventionally chosen to be a Maxwellian. $E(x, t)$ is the electric field generated by the total (electron and ion) charge distribution. Combining with Gauss's law the full system reads

$$\frac{\partial f}{\partial t} = -v\frac{\partial f}{\partial x} + \frac{q^2}{m_e\varepsilon_0}\frac{\partial f}{\partial v}\int^x dx' \int dv' \left(f^{\text{bg}} - f\right) + C[f]. \tag{2}$$

However, we can choose to treat the field as a dynamical variable and evolve it using Ampere's law. Then the resulting system is

$$\frac{\partial f}{\partial t} = -v\frac{\partial f}{\partial x} + \frac{q}{m_e}E\frac{\partial f}{\partial v} + C[f],$$
$$\frac{\partial E}{\partial t} = \frac{q}{\varepsilon_0}\int v\left(f - f^{\mathrm{bg}}\right)dv. \tag{3}$$

We take periodic boundary conditions (BCs) in position with periodicity $x_{\max}$, as well as fixed ones in velocity such that $f(x, v, t)$ goes to zero outside $|v| = v_{\max}$, for some $v_{\max}$:[3]

$$f(0, v, t) = f(x_{\max}, v, t), \qquad\qquad \forall\, v, t, \tag{4a}$$
$$f(x, v, t) = 0, \qquad\qquad \forall\, |v| > v_{\max} \text{ and } \forall\, x, t, \tag{4b}$$
$$E(0, t) = E(x_{\max}, t) = 0, \qquad\qquad \forall\quad t. \tag{4c}$$

Finally, we use a generalized version of the Krook collision operator given by

$$C\left[f\right] \equiv \nu(v)\left(f^M - f\right), \tag{5}$$

where $f^M = f^M(v) \sim \exp(-bv^2)$ is a Maxwellian distribution, which is a Gaussian in one dimension, and $\nu(v)$ is a velocity dependent collision frequency. The Krook collision operator pushes f towards f^M, the target Maxwellian, over timescale $1/\nu(v)$. Physically, the parameter b can be expressed as $b = m_e/(2k_B\mathcal{T})$, where k_B is the Boltzmann constant and $\mathcal{T}$ is the temperature. We take $\nu(v)$ to be in the general form

$$\nu(v) \equiv \nu_0 + h(v), \tag{6}$$

where ν_0 is a base-line collision frequency and $h(v) = h(-v) > 0$ is a small (positive) variation on top of it. This form is chosen such that $\min_v \nu(v) = \nu_0 > 0$ and

$$\max_{|v|\leq v_{\max}} h(v) \ll \nu_0, \tag{7}$$

which will be used in proving the convergence of the quantum algorithm.

The velocity dependence of $\nu(v)$ in the Krook model was proposed in Ref. [47] and further investigated in Refs. [1, 25]. Their conclusion is that by enforcing certain conditions on the function $\nu(v)$ (which we do not explicitly do here), the Krook operator can mimic the behaviour of the physically more meaningful Fokker–Planck operator [43] despite being more simple. Furthermore, this can be done while maintaining some desired properties of the system such as the conservation of particle number, total momentum and energy.

2.2   Discretization of phase space

We discretize the distribution function $f(x, v, t)$ on an $N = N_x \times N_v$ grid, leading to

$$f \equiv \begin{pmatrix} f_{1,1} & \cdots & f_{1,N_v} \\ \vdots & \ddots & \vdots \\ f_{N_x,1} & \cdots & f_{N_x,N_v} \end{pmatrix}, \tag{8}$$

---

[3]These are chosen for mathematical simplicity and do not affect the conclusion of this work.

where $f_{i,j} \equiv f(x_i, v_j, t)$ and the grid is spanned by N_x independent spatial coordinates indexed by i and N_v velocity coordinates indexed by j. We define

$$x_i \equiv (i-1)\Delta x, \quad i = 1, \ldots, N_x,$$
$$\text{such that} \quad x_1 = 0, \, x_{N_x} = x_{\max} - \Delta x \qquad \text{and} \qquad x_{\max} \equiv N_x \Delta x, \tag{9}$$

and

$$v_j \equiv -v_{\max} + (j-1)\Delta v, \quad j = 1, \ldots, N_v,$$
$$\text{such that} \quad v_1 = -v_{\max}, \, v_{N_v} = v_{\max} \qquad \text{and} \qquad v_{\max} \equiv \frac{(N_v - 1)\Delta v}{2}. \tag{10}$$

Furthermore, we choose to approximate the differential operators $\partial/\partial x$, $\partial/\partial v$ with second order central derivatives and the integrals with the trapezoidal rule, as given in Appendix A.5. These introduce an error of $\mathcal{O}(\Delta x^2 + \Delta v^2)$ on df_{ij}/dt.

2.3 Finite difference equations

This paper, except for Section 8, uses Gauss's law as a coupling between the Vlasov equation and Maxwells equations, i.e. aims to solve Eq. (2). Applying the above discretization to that system results in the finite difference equations

$$\frac{d}{dt}f_{ij} = -\frac{q^2}{m_e \varepsilon_0} \left.\frac{\partial}{\partial v}\right|_{ij} f \cdot \iint^{x_i} f_{IJ}\, dx_I dv_J \qquad\qquad = \mathcal{O}\left(f^2\right) \tag{11a}$$

$$- v_j \left.\frac{\partial}{\partial x}\right|_{ij} f + \frac{q^2}{m_e \varepsilon_0} \left.\frac{\partial}{\partial v}\right|_{ij} f \cdot \iint^{x_i} f_{IJ}^{\text{bg}}\, dx_I dv_J \qquad\qquad = \mathcal{O}\left(f^1\right) \tag{11b}$$

$$- \nu(v_j) f_{ij} \qquad\qquad = \mathcal{O}\left(f^1\right) \tag{11c}$$

$$+ \nu(v_j) f_j^M \qquad\qquad = \mathcal{O}\left(f^0\right), \tag{11d}$$

where the discretized target Maxwellian is

$$f_j^M = \frac{\mathcal{N}}{2x_{\max}\Delta v} \left[\sum_{J=1}^{N_v} \exp\left(-bv_J^2\right) \right]^{-1} \exp\left(-bv_j^2\right). \tag{12}$$

The prefactor in the above equation ensures that $\iint f dx dv = \mathcal{N}$ under the trapezoidal rule for the integration on the whole grid ($0 \le x \le x_{\max} = x_{N_x+1}$), where $\mathcal{N}$ is the number of electrons per unit area of the other 2 position dimensions. Said otherwise, $\mathcal{N} = \int n(x)dx$ where $n(x)$ is the 3 dimensional spatial density, taken to be a function of x. By assuming that the static background is uniform in space, the integral in Eq. (11b) becomes

$$\iint^{x_i} f_{IJ}^{\text{bg}}\, dx_I dv_J = \frac{i-1}{N_x}\mathcal{N}, \tag{13}$$

where we take the same $\mathcal{N}$ as for electrons to ensure net charge neutrality.

3 The quantum algorithm

The quantum algorithm onto which we map our discretized Vlasov equation was developed recently by Krovi [31]. In [31, Theorem 7], a linear ODE solver is presented, while in [31, Theorem 8], it is applied to ODE systems with quadratic nonlinearities using Carleman linearization. We summarize the relevant parts of the quantum algorithm below.

3.1 Problem statement

The quantum algorithm in question solves an ODE system of the form

$$\frac{du}{dt} = F^{(2)}u^{\otimes 2} + F^{(1)}u + F^{(0)}, \qquad u(0) = u_{\text{in}}, \tag{14}$$

where the state vector $u \equiv [u_1, \ldots, u_d]^\mathsf{T} \in \mathbb{R}^d$ is a d dimensional column vector and $u^{\otimes 2} \equiv u \otimes u = [u_1^2, u_1 u_2, \ldots, u_1 u_d, u_2 u_1, \ldots, u_d u_{d-1}, u_d^2]^\mathsf{T} \in \mathbb{R}^{d^2}$ is a d^2 dimensional column vector containing all the quadratic nonlinearities. T stands for transpose. Each $u_k = u_k(t)$ is a function of time t on the interval $t \in [0, T]$. The matrices $F^{(2)} \in \mathbb{R}^{d \times d^2}$ and $F^{(1)} \in \mathbb{R}^{d \times d}$, and the vector $F^{(0)} \in \mathbb{R}^d$ are all time independent and have sizes such that the matrix multiplications in Eq. (14) are well defined.

Other quantum algorithms based on procedural linear embedding apply to slightly different systems. For instance, Refs. [53] and [35] require $F^{(0)} = 0$ and Ref. [34] requires $F^{(0)} = F^{(0)}(t)$, i.e., an explicit time dependence in the inhomogeneous term. This would correspond to an explicit time dependence in the target Maxwellian in the collision operator, i.e., $f^{M,(s)}(v) \to f^{M,(s)}(v, t)$. Finally, Ref. [48] is focused on higher order nonlinearities.

3.2 Convergence criteria

The quantum algorithm reviewed in Section 3.3 applies to the ODE system in Eq. (14) under two conditions. Firstly, $F^{(1)}$ must have a negative log-norm:

$$\mu\left(F^{(1)}\right) < 0, \tag{15}$$

which corresponds to the system being dissipative. Secondly, the convergence parameter R, given by

$$R \equiv \frac{1}{|\mu(F^{(1)})|} \left(\left\|F^{(2)}\right\| \|u_{\text{in}}\| + \frac{\left\|F^{(0)}\right\|}{\|u_{\text{in}}\|} \right), \tag{16}$$

must satisfy $R < 1$. Mathematically, these are both required to bound the error due to Carleman linearization [18]. The norms are defined in Appendix A.1.

3.3 The algorithm

3.3.1 Input information

The input consists of maps for the rescaled elements of each expression in Eq. (14). Let us define the rescaling through the positive number γ as

$$u \to \bar{u} \equiv \frac{u}{\gamma}. \tag{17}$$

This rescaling changes the $F^{(2)}$, $F^{(0)}$ matrices as

$$F^{(2)} \to \bar{F}^{(2)} \equiv \gamma F^{(2)}, \qquad F^{(0)} \to \bar{F}^{(0)} \equiv \frac{F^{(0)}}{\gamma}. \tag{18}$$

Note that the rescaling does *not* affect $F^{(1)}$ and R, and hence they will not carry the bar. The factor γ must be chosen such that $\bar{u}$ has no physical dimension and satisfies $\|\bar{u}_{\text{in}}\| < 1$, as well as that

$$\left|\mu\left(F^{(1)}\right)\right| > \left\|\bar{F}^{(2)}\right\| + \left\|\bar{F}^{(0)}\right\|. \tag{19}$$

These are required when bounding the error due to the Carleman linearization. Both these conditions can be met by [34]

$$\gamma \equiv \sqrt{\|u_{\text{in}}\|r_+}, \quad r_+ \equiv \frac{-\mu\left(F^{(1)}\right) + \sqrt{\mu\left(F^{(1)}\right)^2 - 4\|F^{(2)}\|\|F^{(0)}\|}}{2\|F^{(2)}\|}. \tag{20}$$

The rescaled system is used henceforth and accessed through the oracles $O_{F^{(2)}}, O_{F^{(1)}}, O_{F^{(0)}}$ and O_x, with the last one preparing the Carleman linearized state vector x defined below.

3.3.2 Conceptual steps

1. The Carleman linearized system has a state vector z, given by

$$z \equiv \begin{bmatrix} \bar{u} \\ \bar{u}^{\otimes 2} \\ \vdots \\ \bar{u}^{\otimes N_C} \end{bmatrix} = \left[\bar{u}^{\mathsf{T}}, \, \bar{u}^{\otimes 2\,\mathsf{T}}, \dots, \bar{u}^{\otimes N_C\,\mathsf{T}}\right]^{\mathsf{T}}, \tag{21}$$

where N_C is the number of Carleman linearization steps. This vector is evolved by the linear equation

$$\frac{dz}{dt} = Az + b, \qquad z(0) = \left[\bar{u}_{\text{in}}^{\mathsf{T}}, \, \bar{u}_{\text{in}}^{\otimes 2\,\mathsf{T}}, \dots, \bar{u}_{\text{in}}^{\otimes N_C\,\mathsf{T}}\right]^{\mathsf{T}}. \tag{22}$$

For $N_C \to \infty$, Eq. (22) encodes Eq. (14) exactly, meanwhile for finite N_C it only approximates it. A and b can be constructed from $\bar{F}^{(2)}$, $F^{(1)}$ and $\bar{F}^{(0)}$ as

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{N_C-1} \\ z_{N_C} \end{bmatrix} = \begin{pmatrix} A_1^1 & A_2^1 & & & & \\ A_1^2 & A_2^2 & A_3^2 & & & \\ & A_2^3 & A_3^3 & A_4^3 & & \\ & \ddots & \ddots & \ddots & & \\ & & & A_{N_C-2}^{N_C-1} & A_{N_C-1}^{N_C-1} & A_{N_C}^{N_C-1} \\ & & & & A_{N_C-1}^{N_C} & A_{N_C}^{N_C} \end{pmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{N_C-1} \\ z_{N_C} \end{bmatrix} + \begin{bmatrix} \bar{F}^{(0)} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \tag{23}$$

with $z_k = \bar{u}^{\otimes k}$ as above. A is a $d_A \equiv \left(d^{N_C+1} - d\right)/(d-1)$ dimensional, square and block-tridiagonal matrix. The matrices inside are

$$\begin{aligned} A_{j-1}^j &\equiv \bar{F}^{(0)} \otimes \mathbb{I}^{\otimes j-1} + \mathbb{I} \otimes \bar{F}^{(0)} \otimes \mathbb{I}^{\otimes j-2} + \dots + \mathbb{I}^{\otimes j-1} \otimes \bar{F}^{(0)}, \\ A_j^j &\equiv F^{(1)} \otimes \mathbb{I}^{\otimes j-1} + \mathbb{I} \otimes F^{(1)} \otimes \mathbb{I}^{\otimes j-2} + \dots + \mathbb{I}^{\otimes j-1} \otimes F^{(1)}, \\ A_{j+1}^j &\equiv \bar{F}^{(2)} \otimes \mathbb{I}^{\otimes j-1} + \mathbb{I} \otimes \bar{F}^{(2)} \otimes \mathbb{I}^{\otimes j-2} + \dots + \mathbb{I}^{\otimes j-1} \otimes \bar{F}^{(2)}. \end{aligned} \tag{24}$$

2. The analytical solution of Eq. (22) is

$$z(T) = \exp(AT)z(0) + \left(\int_0^T \exp(As)ds\right)b = \exp(AT)z(0) + T\sum_{j=0}^{\infty} \frac{(AT)^j}{(j+1)!}b. \tag{25}$$

Time is discretized with increments of h such that $t_l = lh$ with $l = 0, \dots, m = T/h$, where for simplicity we assumed that h and T are set so that the number of timesteps m is an integer. The algorithm approximates Eq. (25) at every timestep as

$$y_{l+1} = T_k(Ah)y_l + S_k(Ah)hb, \tag{26}$$

where y_l approximates $z(t_l)$. The matrices in the above equation are Taylor series truncated at level k as

$$T_k(w) \equiv \sum_{j=0}^{k} \frac{w^j}{j!} \quad , \quad S_k(w) \equiv \sum_{j=1}^{k} \frac{w^{j-1}}{j!}. \tag{27}$$

Then Eq. (26) is implemented as a linear system of equations

$$L|y\rangle = |\psi_{\text{in}}\rangle, \tag{28}$$

with a matrix L formulated as

$$
\begin{aligned}
L &= \mathbb{I} - N \\
N &= \sum_{i=0}^{m} |i+1\rangle\langle i| \otimes M_2(\mathbb{I} - M_1)^{-1} + \sum_{i=m+1}^{m+p-1} |i+1\rangle\langle i| \otimes \mathbb{I} \\
M_1 &= \sum_{j=0}^{k-1} |j+1\rangle\langle j| \otimes \frac{Ah}{j+1} \\
M_2 &= \sum_{j=0}^{k} |0\rangle\langle j| \otimes \mathbb{I}.
\end{aligned}
\tag{29}
$$

The RHS of Eq. (28) is

$$|\psi_{\text{in}}\rangle = \frac{1}{\sqrt{\|z_0\|^2 + mh^2\|b\|^2}} \left(|0, 0, z_0\rangle + h \sum_{i=0}^{m-1} |i, 1, b\rangle \right). \tag{30}$$

The first register, indexed by i, contains the time step. The second one, indexed by j, is the Taylor sub-time-step of the time integrator and finally the third contains the states on which A acts. In the above state $|z_0\rangle$ encodes the initial condition of the linearized system [34]

$$|z_0\rangle = \frac{1}{\sqrt{V}} \sum_{j=1}^{N_C} \|\bar{u}_{\text{in}}\|^j |j\rangle \otimes |\bar{u}_{\text{in}}\rangle^{\otimes j} \otimes |0\rangle^{\otimes N_C - j}, \quad \text{where} \quad V = \sum_{j=1}^{N_C} \|\bar{u}_{\text{in}}\|^{2j}. \tag{31}$$

The $|0\rangle$ state in the third term of the tensor product above is d dimensional. This ensures that the dimension of $|\bar{u}_{\text{in}}\rangle^{\otimes j} \otimes |0\rangle^{\otimes N_C - j}$ is fixed to be $d^{N_C} \; \forall j$ for mathematical consistency. The state preparation of $|\psi_{\text{in}}\rangle$ is described in Ref. [34].

3. The last step is inverting Eq. (28) to obtain the state

$$
\begin{aligned}
|y\rangle &= L^{-1}|\psi_{\text{in}}\rangle \\
&= \sum_{i=0}^{m} |i, 0, y_i\rangle + \sum_{i=m+1}^{m+p-1} |i, 0, y_m\rangle.
\end{aligned}
\tag{32}
$$

Here the point of having the parameter p becomes apparent: it increases the amplitude of the $|y_m\rangle$ states we are looking for. In fact, the first d components of $|y_m\rangle$, which we call $|y_{1,m}\rangle$, is level 1 in the Carleman embedding, hence it encodes $\bar{u}(T)$, the rescaled solution of (14). To ensure that the success probability for measuring $|y_{1,m}\rangle$ is constant, further amplitude amplification is performed on Eq. (32).

3.4 Errors

There are three sources of errors in the quantum algorithm in Section 3.3. Firstly, the truncation of the Carleman linearization at level N_C in Eqs. (21-22) introduces an error δ relative to the exact solution, defined as

$$\|z_1(T) - \bar{u}(T)\| \leq \delta \|\bar{u}(T)\|, \tag{33}$$

provided that N_C is chosen according to [31, Lemma 17] as

$$N_C \geq \left\lceil \frac{2\log\left(T\left\|\bar{F}^{(2)}\right\|/\delta\|\bar{u}(T)\|\right)}{\log(1/\|\bar{u}_{\text{in}}\|)} \right\rceil, \tag{34}$$

for a desired δ. Secondly, the truncation of the temporal Taylor series in Eq. (26) at level k introduces a relative error δ' on the Carleman linearized state vector as

$$\|y_m - z(T)\| \leq \delta' \|z(T)\|, \tag{35}$$

where $z(T)$ is the exact solution of the linear system from Eq. (25) and y_m approximates it. The above inequality is conditioned on choosing k in terms of δ' according to [31, Theorem 3] as

$$(k+1)! \geq \frac{me^3}{\delta'}\left(1 + \frac{Te^2\|b\|}{\|z(T)\|}\right), \tag{36}$$

and on having $\|Ah\| \leq 1$ (which is satisfied due to the choice of h later). Note that Eq. (35) is also a bound on the temporal error of the first block components of y_m, namely $y_{1,m}$. Hence, by bounding $\|z(T)\|$ on the RHS of Eq. (35) as $\|z(T)\| \leq (1+\delta)\sqrt{N_C}\|\bar{u}(T)\|$ via [31, Theorem 8], we may rewrite it as

$$\|y_{1,m} - z_1(T)\| \leq \delta'(1+\delta)\sqrt{N_C}\|\bar{u}(T)\|. \tag{37}$$

Combining Eq. (33) with Eq. (37) we obtain the bound

$$\begin{aligned} \|y_{1,m} - \bar{u}(T)\| &= \|y_{1,m} - z_1(T) + z_1(T) - \bar{u}(T)\| \\ &\leq \|y_{1,m} - z_1(T)\| + \|z_1(T) - \bar{u}(T)\| \\ &\leq \left(\delta'(1+\delta)\sqrt{N_C} + \delta\right)\|\bar{u}(T)\|. \end{aligned} \tag{38}$$

Finally, the QLSA introduces a purely quantum error denoted by ε_q. We therefore choose δ, δ' such that $\delta'(1+\delta)\sqrt{N_C} + \delta \leq \varepsilon_q/2$, since then according to [31, Theorem 8], the full error on the normalized state also satisfies

$$\left\|\frac{y_{1,m}}{\|y_{1,m}\|} - \frac{\bar{u}(T)}{\|\bar{u}(T)\|}\right\| \leq \varepsilon_q. \tag{39}$$

3.5 Complexity of the QLSA

The Carleman linearized system is put into the form of a system of linear equations after the time discretization, as given in Eq. (28). The QLSA applied to that linear system produces the state $L^{-1}|\psi_{\text{in}}\rangle$ with an error of ε_q and has a query complexity of

$$\mathcal{O}\left(s_A k\kappa(L)\text{polylog}\left[k, m, d_A, \kappa(L), \frac{1}{\varepsilon_q}\right]\right), \tag{40}$$

and a gate complexity that is larger by a factor of

$$\mathcal{O}\left(\text{polylog}\left[k, m, \frac{1}{\varepsilon_q}\right]\right), \tag{41}$$

as it was shown in [31, Theorem 6]. In the above, s_A is the sparsity of A, d_A is its dimension, $\kappa(L)$ is the condition number of the linear system, k is the truncation level of the temporal Taylor series and m is the number of time-steps. We now apply the QLSA to the Carleman linearized system.

3.6 Complexity of the nonlinear solver

The quantum algorithm first prepares $|\psi_{\text{in}}\rangle$ from Eq. (30), then applies the QLSA from above to produce $L^{-1}|\psi_{\text{in}}\rangle$ and finally amplifies the amplitude of the $|y_{1,m}\rangle$ states. If $\mu(F^{(1)}) < 0$ and $R < 1$, the quantum algorithm produces a quantum state satisfying Eq. (39) with a query complexity [31, Theorem 8]

$$\mathcal{O}\left(\sqrt{N_C(\delta)}g_u s N_C(\delta)T\|A\|\frac{\log\Omega(\delta',\delta)}{\log\log\Omega(\delta',\delta)}\text{polylog}\left[\log\Omega(\delta',\delta), T\|A\|, d^{N_C(\delta)}, \frac{1}{\varepsilon_q}\right]\right), \tag{42}$$

and a gate complexity that is larger by a factor of

$$\mathcal{O}\left(\text{polylog}\left[\log\Omega(\delta',\delta), T\|A\|, \frac{1}{\varepsilon_q}\right]\right). \tag{43}$$

1. The above complexities assume that the following internal parameters are picked:

$$h = \frac{T}{\lceil T\|A\|\rceil}, \quad m = p = \frac{T}{h} = \lceil T\|A\|\rceil, \quad \delta + (1+\delta)\delta'\sqrt{N_C(\delta)} \leq \varepsilon_q/2,$$

$$k(\delta',\delta) = \left\lceil\frac{2\log\Omega(\delta',\delta)}{\log\log\Omega(\delta',\delta)}\right\rceil, \quad \Omega(\delta',\delta) = \frac{e^3 T\|A\|}{\delta'}\left(1 + \frac{Te^2\|\bar{F}^{(0)}\|}{\|\bar{u}(T)\|}\right), \tag{44}$$

$$N_C(\delta) = \left\lceil\frac{2\log\left(T\|\bar{F}^{(2)}\|/\delta\|\bar{u}(T)\|\right)}{\log(1/\|\bar{u}_{\text{in}}\|)}\right\rceil.$$

2. $\Omega(\delta',\delta)$ above is defined such that $(k(\delta',\delta)+1)! > \Omega(\delta',\delta)$ and therefore Eq. (36) is satisfied. We included the explicit δ' and δ dependence for clarity, noting that the δ dependence originates from the dependence of the norms in $\Omega(\delta',\delta)$ on $N_C(\delta)$.

3. Similarly, setting $N_C(\delta)$ according to Eq. (44) ensures that Eq. (33) is satisfied.

4. The choice for δ and δ' in Eq. (44) makes the l_2 error of the solution state ε_q as shown in Eq. (39).

5. The denominator in $k(\delta',\delta)$, namely $\log\log\Omega(\delta',\delta)$, vanishes asymptotically when $k(\delta',\delta)$ is in logarithms in Eqs. (40-41). Hence the polylog($\cdot$)'s contain no $\log\log\Omega(\delta',\delta)$ terms.

6. The factor $sN_C(\delta)$ in Eq. (42) originates from the sparsity of A being $s_A = 3sN_C(\delta)$, where s denotes the sparsity of $\bar{F}^{(2)}, F^{(1)}, \bar{F}^{(0)}$.

7. $\|A\|$ can be bounded as

$$\|A\| \le N_C(\delta)\left(\left\|\bar{F}^{(0)}\right\| + \left\|F^{(1)}\right\| + \left\|\bar{F}^{(2)}\right\|\right). \tag{45}$$

8. The condition number of L is asymptotically bounded as $\kappa(L) = \mathcal{O}(m+p) = \mathcal{O}(T\|A\|)$ using [31, Theorem 4].

9. The dimension of A is $d_A = \mathcal{O}(d^{N_C(\delta)})$. The contribution of this term to the query complexity is hence polynomial in $N_C(\delta)$.

10. The factor $\sqrt{N_C(\delta)}g_u$ in Eq. (42) is the result of applying amplitude amplification on the $|y_{1,m}\rangle$ states, as the probability to measure the $y_{1,m}$ states can be lower bounded as

$$P[\text{measure } y_{1,m}] \ge \mathcal{O}\left(\frac{1}{N_C g_u^2}\right), \qquad g_u \equiv \frac{\|u_{\text{in}}\|}{\|u(T)\|}. \tag{46}$$

11. The the polylog$(\cdot)$ expressions are polynomials of the logarithms of their inputs, and this polynomial can only be determined entirely during the actual implementation of the quantum algorithm.

12. Note that s appears in Ref. [31] within the polynomial term, but it is actually linear only. That is because the underlying QLSA used is linear in s, as indicated in Eq. (40).

4 Mapping onto the quantum algorithm

We now introduce the mapping between our discretized Vlasov equation, as formulated in Eqs. (11a-11d), onto the ODE system in Eq. (14). The contents of the $\mathcal{O}(f^2)$ line encode the nonlinearity, hence they will determine $F^{(2)}$. Similarly, the $\mathcal{O}(f^1)$ lines determine $F^{(1)}$ and the $\mathcal{O}(f^0)$ one determines $F^{(0)}$.

The key idea to carry out this mapping is to place the dynamical variables in a vector and treat that as the state vector u. In the Gauss's law case it is the vectorization of the distribution matrix f from Eq. (8), while in the Ampere's law case the E_i field variables would also be part of u. The latter is discussed in Section 8. This sort of 'packaging' can be done in many different ways, but here we employ row-major vectorization, as explained in Appendix A.2. We show in Section 9 that the choice of the vectorization method does not affect the complexity of the problem. After the construction of u, the entries of the matrices $F^{(2)}$, $F^{(1)}$, $F^{(0)}$ can be obtained procedurally from the finite difference equations.

4.1 State vector

When coupling to Gauss's law, the row-major vectorization of the distribution matrix f from Eq. (8) gives us the state vector u as

$$u = \text{vec}(f) = [f_{1,1}, \cdots, f_{1,N_v}, f_{2,1}, \cdots, f_{2,N_v}, \ldots, f_{N_x,N_v}]^\mathsf{T}, \tag{47}$$

with the length $N = N_x N_v$ being the total number of grid-points. Elements of u can be expressed from those of f and vice versa as

$$u_n = f_{\lceil n/N_v \rceil,\, n /\!/ N_v}, \qquad \text{where} \qquad \underbrace{\lceil n/N_v \rceil}_{\text{computes row index } i}, \qquad \underbrace{n /\!/ N_v}_{\text{computes column index } j}, \tag{48a}$$

$$f_{ij} = u_{(i-1)N_v + j}, \qquad \text{where} \qquad \underbrace{(i-1)N_v}_{\text{goes over } i-1 \text{ full rows}}, \qquad \underbrace{j}_{\text{gives } j\text{-th element of row } i}, \tag{48b}$$

where the $/\!/$ notation was defined in Table 1. It computes the second index of f, from n. Note that the ranges of the indices are $i = 1, \ldots, N_x$; $j = 1, \ldots, N_v$; $n = 1, \ldots, N$. The above equations align with the fact that the index n entirely identifies the tuple (i, j) and vice versa. Then, $u^{\otimes 2}$ becomes

$$u^{\otimes 2} = \left[f_{1,1} u^{\mathsf{T}}, \, \cdots, \, f_{1,N_v} u^{\mathsf{T}}, \, f_{2,1} u^{\mathsf{T}}, \, \cdots, \, f_{2,N_v} u^{\mathsf{T}}, \, \ldots, \, f_{N_x,N_v} u^{\mathsf{T}} \right]^{\mathsf{T}}, \tag{49}$$

where each 'element' above is an N dimensional vector. For example, the first one reads $f_{11} u = [f_{11} u_1, f_{11} u_2, \ldots, f_{11} u_N]^{\mathsf{T}}$. Hence, $u^{\otimes 2}$ is N^2 dimensional. Its elements can be associated to those of u and f, and vice versa, via

$$u_n^{\otimes 2} = u_{\lceil n/N \rceil} \cdot u_{n /\!/ N}, \qquad \text{where} \qquad \underbrace{\lceil n/N \rceil}_{\text{gives first term of product}}, \qquad \underbrace{n /\!/ N}_{\text{gives second term of product}}, \tag{50a}$$

$$u_a u_b = u_{N(a-1)+b}^{\otimes 2}, \qquad \text{where} \qquad \underbrace{N(a-1)}_{\text{goes over terms } u_1 u, \ldots u_{a-1} u}, \qquad \underbrace{b}_{\text{gives } b\text{-th element of } u_a u}, \tag{50b}$$

$$f_{ab} f_{cd} = u_{N[(a-1)N_v + b - 1] + (c-1)N_v + d}^{\otimes 2}, \quad \text{where} \quad \underbrace{N[(a-1)N_v + b - 1]}_{\text{goes over terms } f_{1,1} u, \ldots f_{a,b-1} u}, \quad \underbrace{(c-1)N_v + d}_{\text{gives element within } f_{a,b} u}. \tag{50c}$$

Note that Eqs. (50b-50c) are not unique since the Kronecker product places each $f_{ab} \cdot f_{cd}$ term twice into $u^{\otimes 2}$, except the square terms. The above formulas can be realized from the structure of the Kronecker product and u.

4.2 Method to find the matrices

Upon substituting the conversion formulas (48b,50c) into the finite difference equations (11a-11d) and simplifying the expressions, one obtains an evolution equation of the form

$$\frac{du_n}{dt} = \sum_{m \in S^{(n)(2)}} C_m^{(n)(2)} u_m^{\otimes 2} + \sum_{m \in S^{(n)(1)}} C_m^{(n)(1)} u_m + \sum_{m \in S^{(n)(0)}} C_m^{(n)(0)}, \tag{51}$$

with coefficients $C_m^{(n)(2)}$, $C_m^{(n)(1)}$, $C_m^{(n)(0)}$. The first two encode a *weighting* for the elements of $u^{\otimes 2}$ and u, respectively, while $C_m^{(n)(0)}$ encodes the inhomogeneity. The sets $S^{(n)(2)}$ and $S^{(n)(1)}$ describe *which elements* of the mentioned vectors take part in evolving u_n. The superscript (n) denotes which element of u is being evolved.

The key point is that the above sums are *linear combinations* of the elements of $u^{\otimes 2}$ and u. These linear combinations can be identified as matrix multiplications with suitable matrices $F^{(2)}$ and $F^{(1)}$. By also denoting the inhomogeneous terms with a matrix (a vector to be exact), called $F^{(0)}$, one obtains an ODE system of the form of Eq. (14) where the matrices have entries

$$\left[F^{(2)} \right]_{n,k} = \sum_{m \in S^{(n)(2)}} C_m^{(n)(2)} \delta_{km}, \tag{52a}$$

$$\left[F^{(1)} \right]_{n,k} = \sum_{m \in S^{(n)(1)}} C_m^{(n)(1)} \delta_{km}, \tag{52b}$$

and the vector

$$\left[F^{(0)} \right]_n = \sum_{m \in S^{(n)(0)}} C_m^{(n)(0)}. \tag{53}$$

The takeaway is that the above analytical maps are directly deducible from the finite difference equations in a procedural way, once the conversion formulas are known.

4.3 The matrices

The above procedure results in maps for the entries of the $F^{(k)}$ matrices. In Appendix B all three are written out explicitly with the Gauss's law coupling, and $F^{(1)}$ with the Ampere's law coupling.

The linear contribution is partitioned into two parts as $F^{(1)} \equiv F^{(1a)} + F^{(1b)}$. Let $F^{(1a)}$ encode the collisional part of the linear evolution, Eq. (11c), as well as $F^{(1b)}$ encode the rest of the linear contribution, Eq. (11b). An identical mapping for Ampere's law is discussed in Section 8. The full $F^{(1)}$ matrices are visually represented on Fig. 2 for a (4,4) grid. Both couplings result in a diagonal $F^{(1a)}$ and off-diagonal $F^{(1b)}$, however the inner structures of the matrices are different. In Fig. 2 (a), which is the Gauss's law version, $F^{(1b)}$ is antisymmetric, otherwise known as skew-symmetric. This can be analytically confirmed from the maps given in Eqs. (97-98). In Fig. 2 (b), which is the Ampere's law version, the $F^{(1b)}$ contribution has no symmetry structure in it. This version has empty columns, which carry special importance form the point of view of the Carleman linearization as it is investigated later in Section 8.

For $F^{(2)}$ with Gauss's law, Fig. 3 provides an illustration for a (3,4) grid. Its inner structure is worth discussing as it reveals how the ODE system encodes the nonlinearity and the numerical integration in particular. Its building unit is the trapezoidal rule row vector, denoted by $\mathbb{T}$, that contains all the information about the numerical integration scheme. Just as the integration scheme in Eq. (94), $\mathbb{T}^{[i]}$ carries a free index $i = 1, \ldots, N_x$, which is the last position index still being integrated over. Mathematically, it reads

$$\mathbb{T}^{[i]} \equiv \begin{cases} \begin{cases} \mathsf{T}\,\mathrm{vec}(0_{N_x,N_v}) = [\underbrace{0, 0 \ldots, 0}_{N_x N_v \text{ times}}], & \text{for } i = 1, \\ \\ \mathsf{T}\,\mathrm{vec}\left[\left.\begin{bmatrix} 2 & 2 & \cdots & 2 & 2 \\ 4 & 4 & \cdots & 4 & 4 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 4 & 4 & \cdots & 4 & 4 \\ 2 & 2 & \cdots & 2 & 2 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}\right\} \begin{matrix} i \text{ rows} \\ \\ \\ N_x - i \text{ rows} \end{matrix}\right] & \text{for } i > 1. \\ \\ = [\underbrace{2, \ldots, 2}_{N_v \text{ times}}, \ \underbrace{4, \ldots, 4}_{N_v(i-2) \text{ times}}, \underbrace{2, \ldots, 2}_{N_v \text{ times}}, \ \underbrace{0, \ldots, 0}_{(N_x-i)N_v \text{ times}}], \end{cases} \end{cases} \tag{54}$$

Here the transpose T was put in front of the vectorization for readability. For $i = 1$, all slots are 0, hence $\mathbb{T}^{[1]}$ is just a vector of 0's. For all i, $\mathbb{T}^{[i]}$ is row vector of length N. The point of this object is that the electric field generated by the electrons at position x_i can be computed by matrix multiplying $\mathbb{T}^{[i]}$ by u as

$$E(x_i)^{(\mathrm{e})} \propto \iint^{x_i} f_{IJ}\,dx_I dv_J = \frac{\Delta x \Delta v}{4}\,\mathbb{T}^{[i]}u. \tag{55}$$
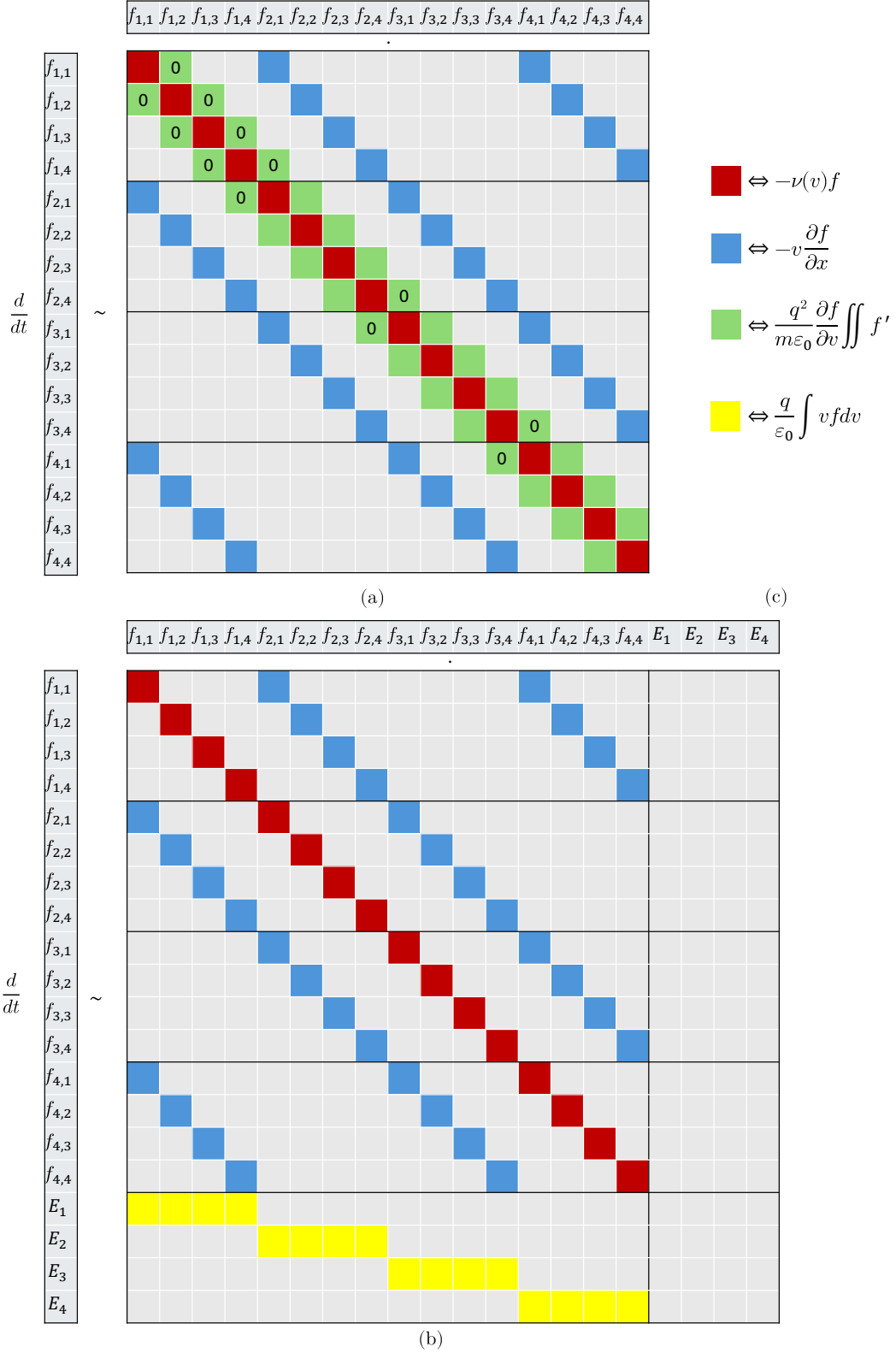
Figure 2: Illustration of the $F^{(1)}$ matrix for a $(4, 4)$ grid. (a): with Gauss's law; (b): with Ampere's law; (c): color codes for elements. On the left of (a) and (b), the evolved vector u is written out explicitly, meanwhile on top, u represents how the matrix multiplication mixes its elements. All gray elements are zero. Cells with the same colors have generally different numerical values.

The above equation essentially says, that one can force element-wise array multiplication of two matrices (the trapezoidal rule and f in this case) to be a matrix multiplication by vectorizing both (into $\mathbb{T}$ and u in this case).

Figure 3: Illustration of the Gauss's law version of $F^{(2)}$. The middle matrix is $F^{(2)}$ up to a constant factor of $-q^2\Delta x/(8m_e\varepsilon_0)$, for a $(3,4)$ grid. On the left, the evolved vector u is written out explicitly The small cells are row vectors with length $N = 12$. On top $u^{\otimes 2}$ represents how the matrix multiplication mixes its elements. Gray elements are 0.

Using this one may express $F^{(2)}$ as a block-diagonal sequence as

$$F^{(2)} = -\frac{q^2\Delta x}{8m_e\varepsilon_0}\cdot\bigoplus_{i=1}^{N_x}B^{[i]} = -\frac{q^2\Delta x}{8m_e\varepsilon_0}\cdot B^{[1]}\oplus B^{[2]}\oplus\cdots\oplus B^{[N_x]}, \tag{56}$$

where the direct sum is defined in Appendix A.3.

It may be seen on Fig. 3, our example, that it consists of $N_x = 3$ blocks, $B^{[1]}$, $B^{[2]}$, $B^{[3]}$, which only differ in the superscript i on $\mathbb{T}^{[i]}$. Generally, block $B^{[i]}$ has size $(N_v, N_xN_v^2)$ and is given by

$$B^{[i]} \equiv \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix} \otimes \mathbb{T}^{[i]}, \tag{57}$$

where the left term in the tensor product has size (N_v, N_v). The upper and lower diagonals encode the two 'legs' of the finite difference derivative stencil of $\partial f/\partial v$ from Eq. (93). Together with $\mathbb{T}^{[i]}$ encoding the double integral of f, $B^{[i]}$ compactly expresses the nonlinearity in Eq. (11a). We are now ready to analyze the convergence criteria of the quantum algorithm given these matrices.

5 Convergence

5.1 Strategy

The convergence of the quantum algorithm from Section 3 is investigated in the limit of $N_x, N_v \gg 1$. For the sake of the convergence analysis, since ν_0 (from Eq. (6)) controls the strength of the dissipation in the system, it is beneficial to treat it similarly to N_x, N_v, which are *control parameters*. This means that the scaling of each mathematical expression is realized with respect to these three parameters. On the other hand variables such as $x_{\max}$, $v_{\max}$, $\mathcal{N}$ and b are treated as fixed. Moreover, the function $h(v)$ (from Eq. (6)) is also treated as fixed and independent of the control parameters.

Finally, N_v is assumed to be even from now on. These assumptions simplify the computation of certain terms to some extent as it is shown later. Remember that Eq. (9) fixes Δx from N_x and $x_{\max}$ as well as that Eq. (10) fixes Δv from N_v and $v_{\max}$.

We shall show later that the value of ν_0 puts a restriction on the maximum value N_v may take. Hence, within the calculation below we also treat ν_0 as a potential large valued.

5.2 Ingredients for R

In Appendix C.1 we compute the exact $\|F^{(2)}\|$ and in Appendix C.2 we compute the asymptotic scaling of $\|F^{(0)}\|$.

As for $\mu(F^{(1)})$, recall from Subsection 4.3 the decomposition $F^{(1)} = F^{(1a)} + F^{(1b)}$, where $F^{(1a)}$ is diagonal and $F^{(1b)}$ is antisymmetric. Then we have

$$
\begin{aligned}
\mu\left(F^{(1)}\right) &= \lambda_{\max} \left\{ \frac{F^{(1a)} + F^{(1b)} + \left(F^{(1a)} + F^{(1b)}\right)^T}{2} \right\} \\
&= \lambda_{\max} \left\{ F^{(1a)} \right\} \\
&= \max_{j=1,\ldots,N_v} \left\{ -\nu(v_j) \right\} \\
&\leq -\nu_0.
\end{aligned}
\tag{58}
$$

Therefore $\mu(F^{(1)}) \leq -\nu_0 < 0$, as required. We see that without collisions, this condition would not be satisfied in our model. The simplicity of the above calculation is the advantage behind using the Krook collision operator.

The choice of possible initial conditions is restricted by two factors. Firstly, $\|u_{\text{in}}\|$ needs to be exactly computable as it appears multiple times, in both numerators and denominators of expressions. Secondly, u_{in} must be more *concentrated*[4] in phase space than the Maxwellian (Gaussian) thermal state $f^M(v)$ encoded in $F^{(0)}$. This is fundamentally because the ratio $\|F^{(0)}\|/\|u_{\text{in}}\|$ must decrease as the system size (N_x, N_v) grows. Recall from Eq. (11d) and correspondingly Eq. (96) that the Gaussian in $F^{(0)}$ is weighted by the values of $\nu(v)$. This weighting can be thought of as being 'cancelled' by the ν_0 in $\mu(F^{(1)})$ in the denominator of R as shown in the above paragraph. With this cancellation the ratio $\|F^{(0)}\|/\|u_{\text{in}}\|$ becomes the norm of a Gaussian over the norm of u_{in}. The more evenly distributed the elements of a vector are, the smaller its l_2 norm becomes. Consequently, u_{in} must be more concentrated in phase space than $f^M(v)$.

The above two conditions are satisfied for a number of initial conditions. We choose a scenario in which two spatially uniform beams with opposite, definite velocities collide

---

[4]Concentration in phase space refers to $f(x, v)$ being large in the vicinity of certain x and v values, and close to 0 elsewhere.

head-on. This occurs in nature in the form of astrophysical jet collisions [16, 40]. Then we have $f_{ij}(t = 0) \propto \delta_{j,J} + \delta_{j,N_v - J + 1}$, which encodes the two beams with velocities v_J and $-v_J$ for some J. The lack of i dependence ensures uniformity in x. This may be illustrated as

$$
u_{\text{in}} \propto \text{vec} \begin{pmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}, \tag{59}
$$

where the 1's are in the J'th and $(N_v - J + 1)$'th columns. The proportionality constant can be fixed from requiring that $\iint f_{ij}(t = 0) = \mathcal{N}$ on the full grid, using the trapezoidal rule for integration. This results in the map given in Appendix C.3.

Upon substituting $\mu(F^{(1)})$ from Eq. (58) and $\|F^{(2)}\|$, $\|F^{(0)}\|$, $\|u_{\text{in}}\|$ from Appendices C.1-C.3 into the R parameter given in Eq. (16), in the asymptotic limit one arrives at

$$
\lim_{N_x, N_v \gg 1} R = \frac{q^2 \mathcal{N}}{2\sqrt{2} m_e \varepsilon_0 v_{\text{max}}} \frac{N_v^{3/2}}{\nu_0} = \mathcal{O}\left(\frac{N_v^{3/2}}{\nu_0}\right). \tag{60}
$$

As a result of our choice for u_{in}, the factor $\|F^{(0)}\|/[\|u_{\text{in}}\|\mu(F^{(1)})]$ entirely vanishes in the asymptotic limit. If that was not the case, then regardless of the value of ν_0, the R parameter would have either a constant $\mathcal{O}(1)$ contribution from the mentioned factor, or would explicitly increase with system size (N_x, N_v).

5.3 Connection to plasma physics

To gain insight into the level of restriction on plasma parameters Eq. (60) requires, we will employ a simple model for the collision frequency that is commonly used in plasma physics [9]. It reads

$$
\nu_0 \approx \frac{q^4 \bar{n} \log(\Lambda)}{(4\pi\varepsilon_0)^2 m_e^{1/2} (\mathcal{T} k_B)^{3/2}}, \tag{61}
$$

where $\log(\Lambda)$ is the Coulomb logarithm, whose value is around $\log(\Lambda) \approx 10$. We can combine this with Eq. (60), set $R < 1$ and $\mathcal{N} = \bar{n} x_{\text{max}}$. Note that we choose $v_{\text{max}} = 10 \cdot v_p$, with $v_p = 1/\sqrt{b}$ being the most probable velocity and $b = m_e/(2k_B \mathcal{T})$ being the decay factor of the Maxwellian. Then this gives us the bound

$$
N_v \lesssim \left(\frac{25}{\pi^2} \frac{q^2}{\varepsilon_0 k_B} \frac{1}{x_{\text{max}} \mathcal{T}}\right)^{2/3}. \tag{62}
$$

We will first use typical physical values of astrophysical plasma within interstellar medium. For this setup our BCs and the $(1 + 1)$ dimensional approximation is relatively accurate when considering the flow of ionized matter far from its sources, e.g. between distant stars. The concrete example we shall use is warm ionized medium (WIM) made up of electrons and ionized hydrogen, which takes up 20-50 % of the interstellar matter of the Milky Way. Its typical temperature is $\mathcal{T} = 8000$ K [17]. Additionally, we choose $x_{\text{max}} = 1000$ km, which, using Eq. (62), gives

$$
N_v \lesssim 1.6 \cdot 10^{-9}, \tag{63}
$$

which is many orders of magnitude below grid sizes of interest.

Another setup would be inertial confinement fusion with physical parameters in the order of $\mathcal{T} = 50$ million K and $x_{\max} = 10^{-4}$ m [9], giving us

$$N_v \lesssim 2.24 \cdot 10^{-5}, \tag{64}$$

which is again unrealistic.

We may also try substituting a reasonable velocity grid size of $N_v \geq 100$ into Eq. (62). Using the numerical values of q, ε_0 and k_B, then gives the bound

$$x_{\max}\mathcal{T} \lesssim 5.31 \cdot 10^{-7} \text{ mK}. \tag{65}$$

The typical value of the product of $x_{\max}$ and $\mathcal{T}$ in real-world setups are orders of magnitude above what this inequality would require.

Consequently, we conclude that the region of convergence of the quantum algorithm does not include scenarios of physical interest. In order to solve the system with reasonable grid sizes, collisions have to be unphysically strong in our model.

6 Errors

In this section we analyze the errors that arise when solving the finite difference equations, Eqs. (11a-11d), with the quantum algorithm discussed. We assume the plasma parameters are set such that the convergence criteria investigated in the above section are satisfied. The errors that arise will be compared to those of the classical solution derived in Appendix E. A total of four different errors are contained within the output state of the quantum algorithm, as shown in Table 2.

Source of error	Size	
Phase space discretization	$\varepsilon_c \sim T/(N_x^2 + N_v^2)$	
Truncating Carleman linearization	δ	$\Rightarrow \delta + (1+\delta)\delta'\sqrt{N_C(\delta)}$
Truncating temporal Taylor series	δ'	
QLSA	ε_q	

Table 2: Summary of the errors that enter the final quantum state produced by the quantum algorithm applied to the discretized Vlasov equation.

The finite difference derivatives and integral carry an (absolute) error of $\mathcal{O}(\Delta x^2 + \Delta v^2)$, which enters the $F^{(j)}$ matrices on the RHS of the quadratic ODE in Eq. (14). It is furthermore directly pushed into the Carleman linearized evolution matrix A in Eq. (23). Then the system of linear equations described by the matrix L encodes the $T_k(Ah)$ and $S_k(Ah)$ matrices, as indicated in Eqs. (27-29). As the matrix A enters these two matrix functions, the error is multiplied by h and accompanied by another error originating from truncating the matrix functions at level k. The latter is already contained in δ', hence we do not include it in the below equation. The nonzero entries of L therefore carry a grid discretization error of

$$\mathcal{O}\left(h[\Delta x^2 + \Delta v^2]\right). \tag{66}$$

Note that this is analogous to the same error on the classical solution at time $t = h$ given in Eq. (134). This is then amplified by the condition number of L, which is bounded by

[31, Theorem 4]

$$\kappa(L) \leq (m + p)C(A)(1 + \delta)e(1 + e), \tag{67}$$

where

$$C(A) = \sup_{t \in [0,T]} \|\exp(At)\| \tag{68}$$

is a measure of the growth in the state vector z induced by the matrix A. According to [31, Lemma 16], we have $C(A) \leq 1$ after the rescaling described in Subsection 3.3.1. Additionally, we have $m = p = T/h$ according to Eq. (44), hence $\kappa(L) = \mathcal{O}(T/h)$. Note how multiplying Eq. (66) by $\kappa(L)$ is entirely analogous to the temporal accumulation of the error in the fully classical case (when going from Eq. (134) to Eq. (135)). This all means that the error originating from phase space discretization within the quantum algorithm scales as

$$\varepsilon_c = \mathcal{O}\left(\frac{T}{N_x^2 + N_v^2}\right). \tag{69}$$

The other classical errors, δ and δ' are set according to the size of ε_q, as discussed in Subsection 3.4. This way we may write the total error on the entries of the state $|y_{1,m}\rangle$ as $\mathcal{O}(\varepsilon_q + \varepsilon_c)$.

7 Complexity

We now present the query- and gate complexities for our problem specifically for the case of a grid with a fixed N_x and N_v ratio, i.e with $N_x^2 = \mathcal{O}(N) = N_v^2$. The complexities are obtained from Eqs. (42) and (43) by inserting the quantities derived in Appendix D, as shown in Appendix D.7. Now we choose

$$\delta = \frac{\varepsilon_q}{4} = \mathcal{O}(\varepsilon_q),$$

$$\delta' = \frac{\varepsilon_q}{4 + \varepsilon_q} \frac{1}{\sqrt{N_C(\delta = \varepsilon_q/4)}} = \mathcal{O}\left(\varepsilon_q \left[\frac{\log(1/\|\bar{u}_{\text{in}}\|)}{\log\left(T\|\bar{F}^{(2)}\|/\varepsilon_q\|\bar{u}(T)\|\right)}\right]^{1/2}\right). \tag{70}$$

This ensures asymptotically that $\delta + (1 + \delta)\delta'\sqrt{N_C(\delta)} = \varepsilon_q/2$. As for the phase space discretization error, we set $N = \mathcal{O}(T/\varepsilon_c)$ according to Eq. (69). Additionally, whenever the parameter ν_0 appears, we substitute $\nu_0 \geq \mathcal{O}(N_v^{3/2}) = \mathcal{O}(N^{3/4})$ from Eq. (60) to stay within the boundary of convergence, i.e. to keep $R \lesssim 1$. Defining the parameter $\eta \equiv T/(\varepsilon_q\varepsilon_c)$, the above procedure leads to a query complexity of [5]

$$\mathcal{O}\left(T^{9/2}\varepsilon_c^{-7/2} \frac{\log^{\frac{5}{2}}(\eta)}{\log[\log(\eta)]} \text{poly}\left[\frac{T^{1/2}}{\varepsilon_c^{1/2}}\log(\eta)\right] \text{polylog}\left[\log(\eta), \frac{T}{\varepsilon_c}\log(\eta), \frac{T}{\varepsilon_c}, \frac{1}{\varepsilon_q}\right]\right), \tag{71}$$

and a gate complexity which is larger than the above by the factor

$$\mathcal{O}\left(\text{polylog}\left[\log(\eta), \frac{T}{\varepsilon_c}\log(\eta), \frac{1}{\varepsilon_q}\right]\right). \tag{72}$$

---

[5] We use the identity $\log(X^a Y^b) = \mathcal{O}(\log[XY])$ with some fixed $a, b > 0$ for $X, Y \to \infty$, as well as $\log[X\log(X)] = \mathcal{O}(\log[X])$ for $X \to \infty$ to simplify our expressions.

The leading order query- and gate complexities without logarithmic terms read

$$\tilde{\mathcal{O}}\left(T^{9/2}\varepsilon_c^{-7/2}\mathrm{poly}\left[T^{1/2}\varepsilon_c^{-1/2}\right]\right). \tag{73}$$

This may now be compared to the time complexity of the classical solution of the same finite difference equation. In Appendix E we showed that to be

$$\mathcal{O}\left(T^2\varepsilon_c^{-1}\right), \tag{74}$$

where the error ε_c is completely analogous to the classical error in the quantum complexities that is denoted the same. From comparing Eq. (73) and Eq. (74), we conclude that regardless of the exact form the unknown polynomial takes, the quantum solution of the problem is polynomially less efficient than the classical one.

8 Coupling to Ampere's law

In this section we explore why coupling the Vlasov equation to Ampere's law would lead to $\mu(F^{(1)}) \geq 0$, which would cause Carleman linearization to not converge.

8.1 Finite difference equations

Discretizing Eq. (3) using the phase space grid described in Section 2.2 gives the finite difference equations

$$\frac{d}{dt}f_{ij} = +\frac{q}{m_e}E_i\left.\frac{\partial}{\partial v}\right|_{ij}f \qquad\qquad = \mathcal{O}\left(E^1f^1\right) \tag{75a}$$

$$-v_j\left.\frac{\partial}{\partial x}\right|_{ij}f \qquad\qquad = \mathcal{O}\left(f^1\right) \tag{75b}$$

$$-\nu(v_j)f_{ij} \qquad\qquad = \mathcal{O}\left(f^1\right) \tag{75c}$$

$$+\nu(v_j)f_j^M \qquad\qquad = \mathcal{O}\left(f^0\right), \tag{75d}$$

$$\frac{d}{dt}E_i = +\frac{q}{\varepsilon_0}\int v_J f_{iJ}\,dv_J \qquad\qquad = \mathcal{O}\left(f^1\right) \tag{75e}$$

$$-\frac{q}{\varepsilon_0}\int v_J f_J^{\mathrm{bg}}\,dv_J \qquad\qquad = \mathcal{O}\left(f^0\right), \tag{75f}$$

where $E_i(t) = E(x_i, t)$ is understood. The contents of the $\mathcal{O}(E^1f^1)$ line encode the nonlinearity, hence they determine $F^{(2)}$. Similarly, the $\mathcal{O}(f^1)$ lines determine $F^{(1)}$ and the $\mathcal{O}(f^0)$ ones determine $F^{(0)}$.

8.2 State vector in the Ampere's law case

When coupling to Ampere's law, the electric field is treated as a dynamical object similarly to the distribution matrix. The state vector is chosen to be formulated by appending the E_i's at the end of $\mathrm{vec}(f)$ as

$$u = [f_{1,1},\ \cdots,\ f_{1,N_v},\ f_{2,1},\ \cdots,\ f_{2,N_v},\ \ldots,\ f_{N_x,N_v},\ E_1,\ldots,\ E_{N_x}]^{\mathsf{T}}, \tag{76}$$

which has length $N_x N_v + N_x = N_x(N_v + 1)$. We have invertible maps similarly to the Gauss's law case, in the form

$$u_n = \begin{cases} f_{\lceil n/N_v \rceil, \, n /\!/ N_v}, & \text{for } 1 \le n \le N_x N_v, \\ E_{n - N_x N_v}, & \text{else,} \end{cases} \tag{77a}$$

$$f_{ij} = u_{(i-1)N_v + j}, \tag{77b}$$

$$E_i = u_{N_x N_v + i}. \tag{77c}$$

The ranges of the indices are $i = 1, \ldots, N_x$; $j = 1, \ldots, N_v$; $n = 1, \ldots, N_x(N_v + 1)$. Other maps, including those concerning $u^{\otimes 2}$ are not necessary to reach our conclusion on why coupling to Ampere's law would not work. To find the matrices, we must again follow the procedure described in Section 4.2 and insert the conversion formulas from above into Eqs. (75a-75f). We explicitly write the map for the entries of $F^{(1)}$ that arises from this in Appendix B.4.

8.3 Convergence

The fundamental problem with coupling to Ampere's law is that $\mu(F^{(1)}) < 0$ is not satisfied. To see this, let us write the *linear* part of the time evolution from Eq. (3) in matrix notation as

$$\frac{\partial}{\partial t} \begin{bmatrix} f \\ E \end{bmatrix} = \begin{pmatrix} -v\frac{\partial}{\partial x} - \nu(v) & 0 \\ \frac{q}{\varepsilon_0} \int v \, dv & 0 \end{pmatrix} \begin{bmatrix} f \\ E \end{bmatrix} + \ldots \tag{78}$$

Note how the electric field E does not take part in the linear evolution at all. The resulting $F^{(1)}$ matrix is going to have empty columns, regardless of our choice of coordinate discretization and packaging the discrete variables into a state vector u. We chose a specific discretization scheme, schematically written in Eqs. (75a-75f) and visualized on Fig. 2 (b). We can see that the number of empty columns is N_x, since $E(x)$ is represented by N_x discrete variables. This is consistent with what Eq. (78) implies. As a result, there are at least N_x eigenvalues of $F^{(1)}$ that are 0, hence $\alpha(F^{(1)}) \ge 0$, where $\alpha(\cdot)$ is defined in Appendix A.1. Now Eq. (87a) tells us that

$$0 \le \alpha\left(F^{(1)}\right) \le \mu\left(F^{(1)}\right), \tag{79}$$

and consequently our quantum algorithm would not converge, regardless of the values of the parameters in Eqs. (75a-75f).

9 Discussion

In our procedure, the value of the R parameter only depends on N_v and not N_x in the asymptotic limit. This is a result of the difference between the scalings of the spectral- and Frobenius norms of $F^{(2)}$, as stated in Appendix C.1. R depending on N_v only means that if our physical system has some base-line collision frequency ν_0, then it puts a restriction on the maximum value N_v may take but puts no restriction on N_x. When using a physical model for ν_0, we may formulate that as an inequality between N_v, temperature $\mathcal{T}$ and position periodicity $x_{\max}$, as shown in Eq. (62). This is qualitatively a similar restriction as the well-known CFL condition in numerical analysis that connects computational and physical parameters [42]. We also would like to note that our formula for R, as given in

Eq. (60), strongly resembles the same formula derived for the Carleman linearization of the Navier-Stokes equations in Ref. [23]. In that system, linear dissipativity is provided by viscosity, that is qualitatively analogous to the collisions in plasma.

Eq. (44) suggests that the number of required Carleman steps N_C is logarithmic. However, $1/\|\bar{u}_{\text{in}}\|$ asymptotically approaches 1 in the denominator and hence blows up the expression. This is due to the fact that γ approaches $\|u_{\text{in}}\|$ asymptotically. To resolve the issue, in Appendix D.3 the second order contributions were collected and the logarithm was Taylor expanded with respect to them. Therefore, N_C is actually polynomial instead of logarithmic with respect to system size. Consequently, d, the dimension of the Carleman linearized system from Eq. (23), is exponential in N_x and N_v. As a result of the query- and gate complexities of the underlying QLSA being polylogarithmic in d, the final complexities of our algorithm gain a polynomial factor in N_x and N_v, which is then converted into a polynomial factor in T and ε_c. This could be hypothetically resolved by using a QLSA whose complexities scale as $\log \log d$. This is highly unlikely, as the Hilbert space can only store an exponential amount of information with respect qubit number, instead of super-exponential.

Another large addition to the complexity of the problem is the sparsity of A that inherits the maximum of the sparsities of the $F^{(k)}$ matrices. Since $F^{(0)}$ is entirely filled due to the Maxwellian in the Krook collision operator and since $F^{(2)}$ encodes two double integrals in each row, they have $s = N, 2N$, respectively. On the other hand, $F^{(1)}$ encodes finite difference derivatives and its sparsity is fixed to five. As a result, the sparsity of the linearized system grows linearly with system size, making it non-sparse. Note that the sparsity of $F^{(0)}$ is the consequence of our choice for the collision operator but the sparsity of $F^{(2)}$ is inherently part of the system, due to the inversion of Gauss's law. The factor the sparsity adds to the complexity could be potentially reduced by using a QLSA that is optimized with respect to sparsity [51].

Furthermore, as it was mentioned in Subsection 5.2, the choice for the initial condition u_{in} of the system is heavily restricted. This is due to the requirement that if one wishes to control the value of R in the algorithm with the collision frequency ν_0, the terms independent of ν_0 must vanish in the asymptotic limit. This is a consequence of the mathematical form of the collision operator. While the Krook one contains an inhomogeneous term, others such as the Fokker-Planck do not. In those cases the restriction on u_{in} is expected to be significantly looser, since then $F^{(0)} = 0$. On the contrary, the Fokker-Planck operator would make $F^{(2)}$ significantly more complicated if multiple species are treated as dynamic.

The final output state of the quantum algorithm (after discarding the first and second registers) is $|y_{1,m}\rangle$, which encodes the vector $u(T)$ and hence the distribution matrix $f(T)$. It can be used for the computation of plasma observables of interest, in particular scalars that are linear or quadratic in the distribution function, via evaluating linear or quadratic forms of the state $|y_{1,m}\rangle$, respectively.

It is important to discuss whether the choice of vectorization affects the convergence of the quantum algorithm. In Section 4 we stated that our choice is row-major vectorization, but many other choices exist that are equally valid from a mathematical point of view, e.g. column-major vectorization. The output of different vectorizations of the same matrix (or multidimensional array in general) are permutations of each other. This means that if $u = \text{vec}_1(f)$ and $\tilde{u} = \text{vec}_2(f)$ are different vectorizations of f, then there exists a permutation matrix (also known as commutation matrix) P such that $u = P\tilde{u}$. The matrix P has exactly one 1 in each row and column, with 0's everywhere else. Furthermore, it is orthogonal in the sense that $P^\mathsf{T} P = P P^\mathsf{T} = \mathbb{I}$. Substituting $u = P\tilde{u}$ into Eq. (14) and

multiplying by P^T, we obtain

$$\frac{d\tilde{u}}{dt} = \tilde{F}^{(2)}\left(\tilde{u} \otimes \tilde{u}\right) + \tilde{F}^{(1)}\tilde{u} + \tilde{F}^{(0)}, \tag{80}$$

where the permuted matrices are

$$\tilde{F}^{(2)} = P^\mathsf{T} F^{(2)}\left(P \otimes P\right), \qquad \tilde{F}^{(1)} = P^\mathsf{T} F^{(1)} P, \qquad \tilde{F}^{(0)} = P^\mathsf{T} F^{(0)}. \tag{81}$$

The computational complexity of the problem is formulated in terms of the norms of the above matrices, meaning that as long as unitarily/orthogonally invariant norms are used, the complexity is invariant under the choice of vectorization and is an intrinsic property of the finite difference scheme. The spectral norm used in this paper, as well as other norms characterized by singular values only[6], satisfy this property [26]. For these norms, $\|UAV\| = \|A\|$ holds for all unitary/orthogonal U, V, which means that $\left\|\tilde{F}^{(k)}\right\| = \left\|F^{(k)}\right\|$ holds in Eq. (81) for $k = 0, 1, 2$, leaving the complexity invariant. Upon substituting $F^{(1)} = P\tilde{F}^{(1)}P^\mathsf{T}$ into the eigenvalue equation satisfied by it, we also see that the permutation leaves the eigenvalues invariant. This is also true for the eigenvalues of the symmetric part of $F^{(1)}$, i.e. the log norm $\mu(F^{(1)})$ is invariant as well.

The above argument holds regardless of the dimension of the original array to vectorized, i.e. when vectorizing the six dimensional array containing the values of $f_{i_1,i_2,i_3,j_1,j_2,j_3}$ that arises when discretizing the distribution function in 3+3 dimensional phase space, any output vector is the permutation of any other possible output.

Last but not least, it is worth noting that the $R < 1$ requirement is not strict in practice, as explored in Ref. [34]. They performed numerical simulations concerning the convergence of Carleman linearization for a toy system[7] with $R \approx 44$ and found that the error still decreases exponentially as N_C is increased. We anticipate that this conclusion would also hold for the Vlasov equation, meaning that the restrictions on the plasma parameters discussed in Subsection 5.3 could be relaxed.

10   Conclusion and prospects

We showed for the first time how the nonlinear Vlasov equation can be solved on a future quantum computer, using the Carleman linearization based quantum algorithm in Ref. [31]. In Section 2 we stated the finite difference equations to be solved and in Section 4 we outlined how they were mapped onto the input of the quantum algorithm, Eq. (14). Then, in Section 5 we found that there exist plasma parameters for which the quantum algorithm converges when the Vlasov equation is coupled to Gauss's law. However, convergence requires the collision frequency to be unphysically large to allow realistic grid sizes. On the other hand, when coupling to Ampere's law the convergence criteria were found to be mathematically unsatisfiable within our framework. This is a result of the fact that the electric field does not participate in inducing the linear part of the evolution, as it only acts nonlinearly through the Lorentz force. Hence the linear part of the evolution is not dissipative, leading to unbounded error during the Carleman linearization.

The query- and gate complexities of the problem were analyzed with the Gauss's law coupling and with the assumption that the convergence criteria are met. It was found

---

[6]Examples besides the spectral norm are the Frobenius norm and the trace/nuclear norm. Counterexamples are the 1- and ∞ norms.

[7]The system is the one dimensional Burger's equation. They used the Euler scheme for the temporal update, which corresponds to $k = 1$ in our framework.

that their upper bounds of both are polynomially larger than the time complexity of the most straightforward classical algorithm that solves the same finite difference equation. The largest contributors to the quantum complexities are $\|A\|$, the norm of the Carleman linearized evolution matrix; the sparsity of A that is linear in the size of the phase-space grid; and finally N_C, the number of Carleman linearization steps which grows polynomially with system size as well.

The development of quantum algorithms with the same input formalism as Eq. (14) is an active area of research, partially because higher dimensional polynomial nonlinearities can be transformed into quadratic ones [18]. Hence it is likely that quantum algorithms with less strict constraints and lower complexities will be put forward, and then our mapping will be applicable to those without major modifications required. An example would be applying QLSA's that are optimized with respect to system dimension, or sparsity.

Future work could be in multiple directions. As it may be seen from the complexities, the dependence on the error originating from phase space discretization is polynomial, meanwhile others are only logarithmic. This might be mitigated by solving the PDE directly on a quantum device rather than discretizing the PDE into an ODE system and solving the latter. Performing this would require an embedding method for nonlinear PDE systems, as a generalization of those for ODE ones (i.e a generalization of Carleman linearization or homotopy perturbation). A prototype for such a mathematical framework was recently proposed by us in Ref. [49]. However, it is yet unclear whether such a high-dimensional family of linear PDEs can be efficiently and accurately solved on analog quantum simulators operating with continuous variables. If that can be achieved, the error dependence of the complexity of our algorithm would be relaxed from polynomial to logarithmic.

Another direction could be to generalize our procedure to the full (3+3) dimensional Vlasov-Maxwell system. However, since that is formulated in terms of integrals of the phase space distribution function within the past lightcones (in terms of retarded time), the system is actually made of *delay* differential equations. It can be put into normal (integro-) differential equation form by working in the non-relativistic limit. By carrying this out, one might find that the linear part of the evolution not being dissipative when coupling to Ampere's/Faraday's law is just a consequence of working in one dimension and having no magnetic field. It would also be insightful to know how the complexity of the problem varies with dimension.

Furthermore, implementing more complex and practical collision operators is also to be done. That might change the structure of the $F^{(k)}$ matrices in a way that the regime of convergence with $R < 1$ is actually closer to physically meaningful scenarios.

11   Acknowledgements

References

[1] A. Alekseenko and Craig Euler. A Bhatnagar–Gross–Krook kinetic model with velocity-dependent collision frequency and corrected relaxation of moments. *Continuum Mechanics and Thermodynamics*, 01 2013. DOI: 10.1007/s00161-014-0407-0.

[2] Abtin Ameri, Erika Ye, Paola Cappellaro, Hari Krovi, and Nuno F. Loureiro. Quantum algorithm for the linear Vlasov equation with collisions. *Phys. Rev. A*, 107:062412, Jun 2023. DOI: 10.1103/PhysRevA.107.062412.

[3] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao. A theory of quantum differential equation solvers: limitations and fast-forwarding. 2023. DOI: 10.48550/arXiv.2211.05246.

[4] *Investigation of quantum algorithms for direct numerical simulation of the Navier-Stokes equations*, 12 2019. Annual Research Briefs 2019. DOI: 10.13140/RG.2.2.22657.81762.

[5] T.D. Arber and R.G.L. Vann. A critical comparison of eulerian-grid-based vlasov solvers. *Journal of Computational Physics*, 180(1):339–357, 2002. ISSN 0021-9991. DOI: 10.1006/jcph.2002.7098.

[6] Noah Brustle and Nathan Wiebe. Quantum and classical algorithms for nonlinear unitary dynamics. *Quantum*, 9:1741, May 2025. ISSN 2521-327X. DOI: 10.22331/q-2025-05-13-1741.

[7] Torsten Carleman. Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Mathematica*, 59(none):63 – 87, 1932. DOI: 10.1007/BF02546499.

[8] A.V. Chankin, D.P. Coster, and G. Meisl. Development and Benchmarking of a New Kinetic Code for Plasma Periphery (KIPP). *Contributions to Plasma Physics*, 52(5-6): 500–504, 2012. DOI: 10.1002/ctpp.201210039.

[9] Francis F Chen. *Introduction to plasma physics and controlled fusion*, volume 1. Springer, 2016. DOI: 10.1007/978-3-319-22309-4. Third Edition.

[10] Pedro C. S. Costa, Stephen Jordan, and Aaron Ostrander. Quantum algorithm for simulating the wave equation. *Phys. Rev. A*, 99:012323, Jan 2019. DOI: 10.1103/PhysRevA.99.012323.

[11] John M. Dawson. Particle simulation of plasmas. *Rev. Mod. Phys.*, 55:403–447, Apr 1983. DOI: 10.1103/RevModPhys.55.403.

[12] Reuben Demirdjian, Thomas Hogancamp, and Daniel Gunlycke. An Efficient Decomposition of the Carleman Linearized Burgers' Equation. 2025. DOI: 10.48550/arXiv.2505.00285.

[13] I. Y. Dodin and E. A. Startsev. On applications of quantum computing to plasma simulations. *Physics of Plasmas*, 28(9):092101, 09 2021. ISSN 1070-664X. DOI: 10.1063/5.0056974.

[14] Alexander Engel, Graeme Smith, and Scott E. Parker. Quantum algorithm for the Vlasov equation. *Phys. Rev. A*, 100:062315, Dec 2019. DOI: 10.1103/PhysRevA.100.062315.

[15] Alexander Engel, Graeme Smith, and Scott E. Parker. Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms. *Physics of Plasmas*, 28(6):062305, 06 2021. ISSN 1070-664X. DOI: 10.1063/5.0040313.

[16] Attilio Ferrari. Modeling extragalactic jets. *Annual Review of Astronomy and Astrophysics*, 36(Volume 36, 1998):539–598, 1998. ISSN 1545-4282. DOI: 10.1146/annurev.astro.36.1.539.

[17] Katia M. Ferrière. The interstellar environment of our galaxy. *Rev. Mod. Phys.*, 73: 1031–1066, Dec 2001. DOI: 10.1103/RevModPhys.73.1031.

[18] Marcelo Forets and Amaury Pouly. Explicit Error Bounds for Carleman Linearization. 2017. DOI: 10.48550/arXiv.1711.02552.

[19] Frank Gaitan. Finding flows of a Navier–Stokes fluid through quantum computing. *npj Quantum Information*, 6(1):61, 2020. DOI: 10.1038/s41534-020-00291-0.

[20] Frank Gaitan. Finding Solutions of the Navier-Stokes Equations through Quantum Computing—Recent Progress, a Generalization, and Next Steps Forward. *Advanced Quantum Technologies*, 4(10):2100055, 2021. DOI: 10.1002/qute.202100055.

[21] Dimitrios Giannakis, Abbas Ourmazd, Philipp Pfeffer, Jörg Schumacher, and Joanna Slawinska. Embedding classical dynamics in a quantum computer. *Phys. Rev. A*, 105: 052404, May 2022. DOI: 10.1103/PhysRevA.105.052404.

[22] Abeynaya Gnanasekaran, Amit Surana, and Hongyu Zhu. Variational quantum framework for nonlinear pde constrained optimization using carleman linearization. 2024. DOI: 10.48550/arXiv.2410.13688.

[23] Javier Gonzalez-Conde, Dylan Lewis, Sachin S. Bharadwaj, and Mikel Sanz. Quantum Carleman linearisation efficiency in nonlinear fluid dynamics. 2024. DOI: 10.48550/arXiv.2410.23057.

[24] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917855. DOI: 10.1145/237814.237866.

[25] Jeffrey Haack, C. Hauck, Christian Klingenberg, Marlies Pirner, and Sandra Warnecke. A consistent BGK model with velocity-dependent collision frequency for gas mixtures. *Journal of Statistical Physics*, 184, 09 2021. DOI: 10.1007/s10955-021-02821-2.

[26] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985. DOI: 10.1017/CBO9780511810817.

[27] Wael Itani and Sauro Succi. Analysis of Carleman Linearization of Lattice Boltzmann. *Fluids*, 7(1), 2022. ISSN 2311-5521. DOI: 10.3390/fluids7010024.

[28] Wael Itani, Katepalli R. Sreenivasan, and Sauro Succi. Quantum Algorithm for Lattice Boltzmann (QALB) Simulation of Incompressible Fluids with a Nonlinear Collision Term. 2023. DOI: 10.48550/arXiv.2304.05915.

[29] I. Joseph, Y. Shi, M. D. Porter, A. R. Castelli, V. I. Geyko, F. R. Graziani, S. B. Libby, and J. L. DuBois. Quantum computing for fusion energy science applications. *Physics of Plasmas*, 30(1):010501, 01 2023. ISSN 1070-664X. DOI: 10.1063/5.0123765.

[30] Ilon Joseph. Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics. *Phys. Rev. Res.*, 2:043102, Oct 2020. DOI: 10.1103/PhysRevResearch.2.043102.

[31] Hari Krovi. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum*, 7:913, 2023. DOI: 10.22331/q-2023-02-02-913.

[32] Sarah K Leyton and Tobias J Osborne. A quantum algorithm to solve nonlinear differential equations. 2008. DOI: 10.48550/arXiv.0812.4423.

[33] Xiangyu Li, Xiaolong Yin, Nathan Wiebe, Jaehun Chun, Gregory K. Schenter, Margaret S. Cheung, and Johannes Mülmenstädt. Potential quantum advantage for simulation of fluid dynamics. 2024. DOI: 10.48550/arXiv.2303.16550.

[34] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, 118(35): e2026805118, 2021. DOI: 10.1073/pnas.2026805118.

[35] Jin-Peng Liu, Dong An, Di Fang, Jiasu Wang, Guang Hao Low, and Stephen Jordan. Efficient quantum algorithm for nonlinear reaction–diffusion equations and energy estimation. *Quantum*, 5:502, July 2023. ISSN 2521-327X. DOI: 10.1007/s00220-023-04857-9.

[36] Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. Quantum algorithm for nonlinear differential equations. 2020. DOI: 10.48550/arXiv.2011.06571.

[37] Koichi Miyamoto, Soichiro Yamazaki, Fumio Uchida, Kotaro Fujisawa, and Naoki Yoshida. Quantum algorithm for the Vlasov simulation of the large-scale structure formation with massive neutrinos. *Phys. Rev. Res.*, 6:013200, Feb 2024. DOI: 10.1103/PhysRevResearch.6.013200.

[38] I. Novikau, E. A. Startsev, and I. Y. Dodin. Quantum signal processing for simulating cold plasma waves. *Phys. Rev. A*, 105:062444, Jun 2022. DOI: 10.1103/PhysRevA.105.062444.

[39] Ivan Novikau, Ilya Y. Dodin, and Edward A. Startsev. Encoding of linear kinetic plasma problems in quantum circuits via data compression. 2024. DOI: 10.48550/arXiv.2403.11989.

[40] George K Parks. *Physics Of Space Plasmas: An Introduction*. CRC Press, 1 edition, 1995. DOI: 10.1201/9780429301674.

[41] John Penuel, Amara Katabarwa, Peter D. Johnson, Collin Farquhar, Yudong Cao, and Michael C. Garrett. Feasibility of accelerating incompressible computational fluid dynamics simulations with fault-tolerant quantum computers. 2024. DOI: 10.48550/arXiv.2406.06323.

[42] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, USA, 3 edition, 2007. ISBN 0521880688. URL https://dl.acm.org/doi/10.5555/1403886.

[43] Marshall N. Rosenbluth, William M. MacDonald, and David L. Judd. Fokker-Planck equation for an inverse-square force. *Phys. Rev.*, 107:1–6, Jul 1957. DOI: 10.1103/PhysRev.107.1.

[44] Claudio Sanavio and Sauro Succi. Lattice boltzmann–carleman quantum algorithm and circuit for fluid flows at moderate reynolds number. *AVS Quantum Science*, 6(2):023802, 04 2024. ISSN 2639-0213. DOI: 10.1116/5.0195549.

[45] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. DOI: 10.1137/S0097539795293172.

[46] N.J. Sircombe and T.D. Arber. VALIS: A split-conservative scheme for the relativistic 2D Vlasov–Maxwell system. *Journal of Computational Physics*, 228(13):4773–4788, 2009. ISSN 0021-9991. DOI: 10.1016/j.jcp.2009.03.029.

[47] Henning Struchtrup. The BGK-model with velocity-dependent collision frequency. *Continuum Mechanics and Thermodynamics*, 9:23–31, 02 1997. DOI: 10.1007/s001610050053.

[48] Amit Surana, Abeynaya Gnanasekaran, and Tuhin Sahai. An efficient quantum algorithm for simulating polynomial differential equations. 2023. DOI: 10.48550/arXiv.2212.10775.

[49] Tamás Vaszary. Carleman Linearization of Partial Differential Equations. 2024. DOI: 10.48550/arXiv.2412.00014.

[50] Tamás Vaszary. Solving the Nonlinear Vlasov Equation on a Quantum Computer (Dissertation), May 2024. URL https://doi.org/10.5281/zenodo.11200239.

[51] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. Quantum linear system algorithm for dense matrices. *Phys. Rev. Lett.*, 120:050502, Jan 2018. DOI: 10.1103/PhysRevLett.120.050502.

[52] Hsuan-Cheng Wu, Jingyao Wang, and Xiantao Li. Quantum Algorithms for Nonlinear Dynamics: Revisiting Carleman Linearization with No Dissipative Conditions. 2024. DOI: 10.48550/arXiv.2405.12714.

[53] Cheng Xue, Yu-Chun Wu, and Guo-Ping Guo. Quantum homotopy perturbation method for nonlinear dissipative ordinary differential equations. *New Journal of Physics*, 23(12):123035, dec 2021. DOI: 10.1088/1367-2630/ac3eff.

[54] Julien Zylberman, Giuseppe Di Molfetta, Marc Brachet, Nuno F. Loureiro, and Fabrice Debbasch. Quantum simulations of hydrodynamics via the Madelung transformation. *Phys. Rev. A*, 106:032408, Sep 2022. DOI: 10.1103/PhysRevA.106.032408.

[55] Julien Zylberman, Giuseppe Di Molfetta, Marc Brachet, Nuno F. Loureiro, and Fabrice Debbasch. Hybrid quantum-classical algorithm for hydrodynamics. 2022. DOI: 10.48550/arXiv.2202.00918.

[56] Óscar Amaro and Diogo Cruz. A living review of quantum computing for plasma physics. 2023. DOI: 10.48550/arXiv.2302.00001.

A Mathematical definitions

A.1 Norm definitions

The l_k, and in particular the l_2 norm of vector a v with length N is

$$\|v\|_k \equiv \left(\sum_{n=1}^{N} |v_n|^k \right)^{1/k}, \qquad \|v\| \equiv \|v\|_2. \tag{82}$$

Similarly for matrices, the l_k norm, and in particular the l_2 spectral norm is defined as

$$\|A\|_k \equiv \max_{\|x\|_k \neq 0} \frac{\|Ax\|_k}{\|x\|_k}, \qquad \|A\| \equiv \|A\|_2. \tag{83}$$

The l_1, l_2 and l_∞ norms simplify to

$$\|A\|_1 \equiv \max_j \sum_i |A_{ij}|, \tag{84a}$$

$$\|A\| \equiv \sqrt{\lambda_{\max} (A^\dagger A)}, \tag{84b}$$

$$\|A\|_\infty \equiv \max_i \sum_j |A_{ij}|, \tag{84c}$$

where $\lambda_{\max}(\cdot)$ returns the largest eigenvalue of its input and $\dagger$ denotes the Hermitian conjugate. An alternative measure is the Frobenius norm:

$$\|A\|_F \equiv \sqrt{\sum_{n=1}^{N} \sum_{m=1}^{M} |A_{nm}|^2}, \tag{85}$$

for a matrix of size (N, M). These matrix norms exist for all matrices.

The (l_2) log-norm of a square matrix A is given by

$$\mu(A) \equiv \lim_{h \to 0^+} \frac{\|\mathbb{I} + hA\| - 1}{h} = \lambda_{\max} \left\{ \frac{A + A^\mathsf{T}}{2} \right\}. \tag{86}$$

Finally, $\alpha(A) \equiv \max\{\mathrm{Re}(\lambda)\}$ is the spectral abscissa, the largest one of the real parts of the eigenvalues λ of A. Unlike $\|A\|_k$ and $\|A\|_F$, $\mu(A)$ and $\alpha(A)$ can take negative values and are only applicable to square matrices.

Key relations we use are

$$\alpha(A) \leq \mu(A) \leq \|A\|, \tag{87a}$$

$$\|A\| \leq \|A\|_F, \tag{87b}$$

$$\|A\| \leq \sqrt{\|A\|_1 \|A\|_\infty}. \tag{87c}$$

A.2 Vectorization

The vectorization operator takes a matrix and outputs a column vector which contains each matrix element once. We choose to work with *row-major* vectorization, that places the elements of the input matrix into the vector row by row going from top to bottom. For an arbitrary matrix A with size (N, M) it acts as

$$\mathrm{vec}(A) = \mathrm{vec} \begin{pmatrix} A_{1,1} & \cdots & A_{1,M} \\ \vdots & \ddots & \vdots \\ A_{N,1} & \cdots & A_{N,M} \end{pmatrix} \equiv [A_{1,1}, \ldots, A_{1,M}, A_{2,1}, \ldots, A_{2,M}, \ldots, A_{N,M}]^\mathsf{T}. \tag{88}$$

The resulting $\text{vec}(A)$ has length NM. Note that vectorization is entirely invertible, as long as the size of the target matrix is specified.

A.3   Direct sum

The direct sum takes matrices and places them in a higher dimensional matrix in the form of a block-diagonal chain. For multiple matrices $\left\{A^{(i)}\right\}$ with $i = 1, \ldots, I$, the direct sum is

$$
\bigoplus_{i=1}^{I} A^{(i)} \equiv A^{(1)} \oplus A^{(2)} \oplus \cdots \oplus A^{(I)} \equiv \begin{pmatrix} A^{(1)} & 0 & \cdots & 0 \\ 0 & A^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A^{(I)} \end{pmatrix}. \tag{89}
$$

The sizes of the zeros can be identified uniquely based on the sizes of the matrices $\left\{A^{(i)}\right\}$.

A.4   Identities used

1. Transpose of a Kronecker product is the Kronecker product of the transposes: $(A \otimes B)^{\mathsf{T}} = A^{\mathsf{T}} \otimes B^{\mathsf{T}}$.

2. The eigenvalues of the tensor product $A \otimes B$ are products of eigenvalues $\lambda_i \omega_j$ if λ_i is an eigenvalue of A and ω_j is that of B.

3. The only non-zero eigenvalue of the outer product vv^{T} of the column vector v with itself is the inner product $\lambda = v^{\mathsf{T}} v = \|v\|^2$.

4. The eigenvalues of $F(A)$ are $F(\lambda_i)$ if $F(x)$ has a Taylor expansion near $x = 0$.

5. The eigenvalues of the (n, n) sized tridiagonal Toeplitz matrix

$$
\begin{pmatrix} \alpha & \beta & 0 & \cdots & 0 \\ \gamma & \alpha & \beta & \cdots & 0 \\ 0 & \gamma & \alpha & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \beta \\ 0 & 0 & \cdots & \gamma & \alpha \end{pmatrix} \tag{90}
$$

are

$$
\lambda_k = \alpha + 2\beta \sqrt{\frac{\gamma}{\beta}} \cos\left(\frac{k\pi}{n+1}\right), \qquad k = 1, \ldots, n. \tag{91}
$$

A.5   Discrete calculus

The finite difference schemes operate with central difference derivatives given by

$$
\left.\frac{\partial}{\partial x}\right|_{ij} f \equiv -\frac{1}{2\Delta x} \begin{cases} f_{i+1,j} - f_{i+N_x-1,j}, & \text{for } i = 1, \\ f_{i-N_x+1,j} - f_{i-1,j}, & \text{for } i = N_x, \\ f_{i+1,j} - f_{i-1,j}, & \text{else}, \end{cases} \tag{92}
$$

$$
\left.\frac{\partial}{\partial v}\right|_{ij} f \equiv -\frac{1}{2\Delta v} \begin{cases} f_{i,j+1}, & \text{for } j = 1, \\ -f_{i,j-1}, & \text{for } j = N_v, \\ f_{i,j+1} - f_{i,j-1}, & \text{else}. \end{cases} \tag{93}
$$

where the different cases arise due to the specific BCs. The double integral in Eq. (11a), that is cumulative in x and goes over the whole range of v, is approximated with the 2 dimensional trapezoidal rule [42], given by

$$\iint^{x_i} f_{IJ}\, dx_I dv_J \equiv \frac{\Delta x \Delta v}{2} \cdot \begin{cases} \begin{cases} 0, & \text{for } i = 1, \\ \sum_{J=1}^{N_v} (f_{1,J} + f_{i,J}) + 2\sum_{J=1}^{N_v}\sum_{I=2}^{i-1} f_{I,J}, & \text{for } 2 \leq i \leq N_x. \end{cases} \end{cases} \tag{94}$$

Similarly, the single integral in Eq. (75e) is approximated as

$$\int v_J f_{iJ}\, dv_J \equiv \Delta v \sum_{j=1}^{N_v} v_J f_{iJ}. \tag{95}$$

Note that despite using the trapezoidal rule, the $j = 1$ and $j = N_v$ gridpoints are not treated differently from the others, unlike the $i = 1$ and $i = N_x$ ones. This is due to the v integral going over the range $-\infty < v < \infty$, so in theory the f_{ij} values at $|v| > v_{\max}$ velocities (with $f(x, |v| > v_{\max}) = 0$) also enter the integrals but have vanishing contributions. Hence all of the above approximations carry an error of $\mathcal{O}(\Delta x^2 + \Delta v^2)$.

B Entries of $F^{(2)}$, $F^{(1)}$, $F^{(0)}$

B.1 Gauss's law coupling $F^{(0)}$

From Eq. (11d), we realize the elements of $F^{(0)}$ as

$$\left[F^{(0)}\right]_n = \frac{\mathcal{N}}{2x_{max}\Delta v}\left[\sum_{J=1}^{N_v}\exp\left(-bv_J^2\right)\right]^{-1}\cdot \nu(v_{n/\!\!/N_v})\exp\left(-bv_{n/\!\!/N_v}^2\right), \tag{96}$$

with $n = 1, \ldots, N$. Note that the n dependence on the RHS only appears in the form of $n /\!\!/ N_v = j$ as expected from a collision operator without x dependence.

B.2 Gauss's law coupling $F^{(1)}$

As said earlier, linear contribution is partitioned into two parts as $F^{(1)} = F^{(1a)} + F^{(1b)}$, where $F^{(1a)}$ encodes the collisional part, Eq. (11c), and $F^{(1b)}$ encodes the rest of the linear contributions, Eq. (11b). Therefore,

$$\left[F^{(1a)}\right]_{n,k} = -\nu(v_{n/\!\!/N_v})\delta_{k,n}, \tag{97}$$

where the $\delta_{k,n}$ encodes the proportionality with f_{ij}. Hence $F^{(1a)}$ is diagonal. Note that $n, k = 1, \ldots, N$. For $F^{(1b)}$, the elements read

$$\left[F^{(1b)}\right]_{n,k} = -\frac{v_{n/\!\!/N_v}}{2\Delta x}\begin{cases} \delta_{k,n+N_v} - \delta_{k,n+N_v(N_x-1)}, & \text{for } n \leq N_v, \\ \delta_{k,n+N_v} - \delta_{k,n-N_v}, & \text{for } N_v < n \leq N_v(N_x-1), \\ \delta_{k,n-N_v(N_x-1)} - \delta_{k,n-N_v}, & \text{else}, \end{cases}$$
$$+\frac{q^2\mathcal{N}}{2m_e\varepsilon_0\Delta v}\frac{\lceil n/N_v\rceil - 1}{N_x}\begin{cases} \delta_{k,n+1}, & \text{for } n /\!\!/ N_v = 1, \\ -\delta_{k,n-1}, & \text{for } n /\!\!/ N_v = N_v, \\ \delta_{k,n+1} - \delta_{k,n-1}, & \text{else}. \end{cases} \tag{98}$$

The first term encodes $-v_j\,\partial/\partial x|_{ij}f$ and has 3 cases, which are equivalent to the 3 cases of Eq. (92), the discretized derivative operator. The second term encodes $(q^2/m_e\varepsilon_0)\partial/\partial v|_{ij}f\cdot\iint^{x_i} f^{\mathrm{bg}}(x_I,v_J)dx_I dv_J$, in which the value of the integral was substituted from Eq. (13). Note how $\lceil n/N_v\rceil$ computes i. The cases correspond to those of Eq. (93).

B.3 Gauss's law coupling $F^{(2)}$

The map below captures Eq. (11a). The cases are explained as follows. Recall that the state vector index n and the grid indices (i,j) are related by $\lceil n/N_v\rceil = i$, $n \,/\!/\, N_v = j$ and $n = (i-1)N_v + j$. The row and column indices are in the ranges $n = 1,\dots,N$ and $k = 1,\dots,N^2$, respectively.

1. Case $\{n < N_v + 1\}$ evolves f_{ij} with $i = 1$, when the integral of the ion background is 0 by definition.

2. Case $\{\lceil n/N_v\rceil \geq 2$ and $n \,/\!/\, N_v = 1\}$ evolves f_{ij} with $i \geq 2$ and $j = 1$, where the left 'leg' of $\partial/\partial v|_{ij}f$ vanishes due to BCs.

3. Case $\{\lceil n/N_v\rceil \geq 2$ and $n \,/\!/\, N_v = N_v\}$ evolves f_{ij} with $i \geq 2$ and $j = N_v$, where the right 'leg' of $\partial/\partial v|_{ij}f$ vanishes due to BCs.

4. The last case is the general one, in which both 'legs' exist.

$$
\left[F^{(2)}\right]_{n,k} = -\frac{q^2\Delta x}{4m_e\varepsilon_0}\cdot
\begin{cases}
0, & \text{for } n < N_v + 1, \\[2ex]
\begin{cases}
\displaystyle\sum_{J=1}^{N_v}\left(\delta_{k,Nn+J} + \delta_{k,Nn+(\lceil n/N_v\rceil-1)N_v+J}\right) \\
\displaystyle +2\sum_{I=2}^{\lceil n/N_v\rceil-1}\sum_{J=1}^{N_v}\delta_{k,Nn+(I-1)N_v+J},
\end{cases}
& \begin{array}{l}\text{for } \left\lceil\dfrac{n}{N_v}\right\rceil \geq 2 \\[1ex] \text{and } n \,/\!/\, N_v = 1,\end{array} \\[5ex]
\begin{cases}
\displaystyle\sum_{J=1}^{N_v}\left(\delta_{k,N(n-2)+(\lceil n/N_v\rceil-1)N_v+J} + \delta_{k,N(n-2)+J}\right) \\
\displaystyle +2\sum_{I=2}^{\lceil n/N_v\rceil-1}\sum_{J=1}^{N_v}\delta_{k,N(n-2)+(I-1)N_v+J},
\end{cases}
& \begin{array}{l}\text{for } \left\lceil\dfrac{n}{N_v}\right\rceil \geq 2 \\[1ex] \text{and } n \,/\!/\, N_v = N_v,\end{array} \\[5ex]
\begin{cases}
\left[\displaystyle\sum_{J=1}^{N_v}\left(\delta_{k,Nn+J} + \delta_{k,Nn+(\lceil n/N_v\rceil-1)N_v+J}\right) \right. \\
\left. \displaystyle +2\sum_{I=2}^{\lceil n/N_v\rceil-1}\sum_{J=1}^{N_v}\delta_{k,Nn+(I-1)N_v+J}\right] - \\
\left[\displaystyle\sum_{J=1}^{N_v}\left(\delta_{k,N(n-2)+(\lceil n/N_v\rceil-1)N_v+J} + \delta_{k,N(n-2)+J}\right)\right. \\
\left.\displaystyle +2\sum_{I=2}^{\lceil n/N_v\rceil-1}\sum_{J=1}^{N_v}\delta_{k,N(n-2)+(I-1)N_v+J}\right],
\end{cases}
& \text{else,}
\end{cases}
\tag{99}
$$

B.4 Ampere's law coupling $F^{(1)}$

Again, the collision part, Eq. (75c), is packaged into $F^{(1a)}$ and the rest of the linear evolution, Eqs. (75b,75e), is packaged into $F^{(1b)}$. Hence $F^{(1)} = F^{(1a)} + F^{(1b)}$ as in the Gauss's law case and we have

$$\left[F^{(1a)}\right]_{n,k} = \begin{cases} -\nu(v_{n/\!/N_v})\delta_{k,n}, & \text{for } n \leq N_v N_x, \\ 0, & \text{else}, \end{cases} \tag{100}$$

and

$$\left[F^{(1b)}\right]_{n,k} = \begin{cases} -\dfrac{v_{n/\!/N_v}}{2\Delta x} \cdot \begin{cases} \delta_{k,n+N_v} - \delta_{k,n+N_v(N_x-1)}, & \text{for } n \leq N_v, \\ \delta_{k,n+N_v} - \delta_{k,n-N_v}, & \text{for } N_v < n \leq N_v\,(N_x-1), \\ \delta_{k,n-N_v(N_x-1)} - \delta_{k,n-N_v}, & \text{for } N_v\,(N_x-1) < n \leq N_v N_x, \end{cases} \\ \dfrac{\Delta v q}{\varepsilon_0} \displaystyle\sum_{J=1}^{N_v} \delta_{k,(n-N_x N_v-1)N_v+J}\, v_J, & \text{else}, \end{cases} \tag{101}$$

with $n, k = 1, \ldots N_x(N_v + 1)$. The first 3 cases in Eq. (101) correspond to those of Eq. (92), meanwhile the last line integrates $\int v_J f_{iJ} dv_J$.

C Derivation of norms from Gauss's law coupling

C.1 $\|F^{(2)}\|$

We first write the spectral norm of Eq. (56) according to its definition:

$$\begin{aligned} \left\|F^{(2)}\right\|^2 &= \left(\frac{q^2\Delta x}{8m_e\varepsilon_0}\right)^2 \lambda_{\max}\left\{\bigoplus_{i=1}^{N_x} B^{[i]\mathsf{T}} B^{[i]}\right\} \\ &= \left(\frac{q^2\Delta x}{8m_e\varepsilon_0}\right)^2 \max_{i=1,\ldots,N_x} \left\|B^{[i]}\right\|^2, \end{aligned} \tag{102}$$

and then

$$\begin{aligned} \left\|B^{[i]}\right\|^2 &= \lambda_{\max}\left\{B^{[i]\mathsf{T}} B^{[i]}\right\} \\ &= -\lambda_{\min}\left\{\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix}^2\right\} \cdot \left\|\mathbb{T}^{[i]}\right\|^2, \end{aligned} \tag{103}$$

where we used Identities 1, 2 and 3 in this order. Now from Eq. (54) we can obtain

$$\left\|\mathbb{T}^{[i]}\right\|^2 = \begin{cases} 0, & \text{for } i = 1, \\ 2N_v \cdot 2^2 + N_v(i-2) \cdot 4^2, & \text{else}. \end{cases} \tag{104}$$

Using Identities 4 and 5, we also get

$$\lambda_{\min}\left\{\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix}^2\right\} = \min_{k=1,\dots,N_v}\left\{\left(0 + 2\sqrt{-1}\cos\left(\frac{k\pi}{1+N_v}\right)\right)^2\right\} \tag{105}$$

$$= -4\cos^2\left(\frac{\pi}{1+N_v}\right).$$

Therefore substituting Eq. (104) and Eq. (105) into Eq. (103) results in

$$\left\|B^{[i]}\right\|^2 = \begin{cases} 0, & \text{for } i = 1, \\ 32\cos^2\left(\frac{\pi}{1+N_v}\right)N_v(2i-3), & \text{else.} \end{cases} \tag{106}$$

Putting this back into Eq. (102) and taking the square root we arrive at

$$\left\|F^{(2)}\right\| = \frac{q^2 x_{\max}}{\sqrt{2}m_e\varepsilon_0}\cos\left(\frac{\pi}{1+N_v}\right)\frac{\sqrt{N_v(2N_x-3)}}{N_x}, \tag{107}$$

which asymptotically approaches

$$\lim_{N_x,\,N_v\gg1}\left\|F^{(2)}\right\| = \frac{q^2 x_{\max}}{m_e\varepsilon_0}\sqrt{\frac{N_v}{N_x}} = \mathcal{O}\left(\sqrt{\frac{N_v}{N_x}}\right). \tag{108}$$

For reference, the Frobenius norm, that upper bounds the spectral norm (Eq. (87b)) and significantly more straightforward to compute, scales asymptotically as

$$\lim_{N_x,\,N_v\gg1}\left\|F^{(2)}\right\|_F = \mathcal{O}(N_v), \tag{109}$$

which is polynomially larger than the spectral norm and would put even more severe restrictions on the convergence of the quantum algorithm than the one found in Subsection 5.3.

C.2 $\|F^{(0)}\|$

This is a vector so it's fast to numerically compute its exact norm. Hence the task is really just to get the norm's asymptotic scaling. Making use of the fact that N_v is even and writing the l_2 norm of Eq. (96) in terms of j only, we get

$$\left\|F^{(0)}\right\| = \frac{\mathcal{N}}{2x_{\max}\Delta v}\left[2\sum_{J=N_v/2+1}^{N_v}\exp\left(-bv_J^2\right)\right]^{-1}\sqrt{2N_x\sum_{j=N_v/2+1}^{N_v}\nu^2(v_j)\exp\left(-2bv_j^2\right)}. \tag{110}$$

We bound this part by part: first the J sum from below, or equivalently its reciprocal from above as

$$\sum_{J=N_v/2+1}^{N_v}\exp\left(-bv_J^2\right) \geq \int_{N_v/2+1}^{N_v+1}\exp\left(-bv_J^2\right)dJ \tag{111}$$

$$= \frac{\sqrt{\pi}}{2\Delta v\sqrt{b}}\left[\operatorname{erf}\left(\Delta v\sqrt{b}\frac{N_v+1}{2}\right) - \operatorname{erf}\left(\Delta v\sqrt{b}\frac{1}{2}\right)\right].$$

The sum over j must be upper bounded. Recall that $\mathcal{O}(\nu[v_{\max}]) = \mathcal{O}(\nu_0)$ due to Eq. (7), so that we can pull $\nu(v_j)$ out of the sum to get the scaling:

$$\sum_{j=N_v/2+1}^{N_v} \exp\left(-2bv_j^2\right) \leq \int_{N_v/2}^{N_v} \exp\left(-2bv_j^2\right) dj$$

$$= \frac{1}{2\sqrt{b}\Delta v} \frac{\sqrt{2\pi}}{2} \left[\operatorname{erf}\left(\Delta v \sqrt{b/2}(N_v-1)\right) + \operatorname{erf}\left(\Delta v \sqrt{b/2}\right) \right]. \tag{112}$$

Putting Eq. (111) and Eq. (112) together, we get

$$\lim_{N_x,\,N_v\gg1} \left\|F^{(0)}\right\| = \frac{\mathcal{N} N_v \nu_0}{4x_{\max}v_{\max}} \left[2\frac{N_v\sqrt{\pi}}{4v_{\max}\sqrt{b}} \operatorname{erf}\left(v_{\max}\sqrt{b}\right) \right]^{-1} \sqrt{2N_x \frac{\sqrt{2\pi}N_v}{8\sqrt{b}v_{\max}} \operatorname{erf}\left(v_{\max}\sqrt{2b}\right)}$$

$$= \frac{\mathcal{N}}{x_{\max}\sqrt{v_{\max}}} \left(\frac{b}{2^7\pi}\right)^{1/4} \frac{\sqrt{\operatorname{erf}\left(v_{\max}\sqrt{2b}\right)}}{\operatorname{erf}\left(v_{\max}\sqrt{b}\right)} \nu_0 \sqrt{N_x N_v}$$

$$= \mathcal{O}\left(\nu_0 \sqrt{N_x N_v}\right). \tag{113}$$

C.3 $\|u_{\text{in}}\|$

The map corresponding to Eq. (59) is

$$\left[u_{\text{in}}\right]_n = \frac{\mathcal{N}}{2x_{\max}\Delta v} \left(\delta_{n/\!/N_v,J} + \delta_{n/\!/N_v,N_v-J+1}\right), \tag{114}$$

with an associated norm

$$\|u_{\text{in}}\| = \frac{\mathcal{N}\sqrt{N_x}}{\sqrt{2}x_{\max}\Delta v}. \tag{115}$$

Asymptotically, it scales as

$$\lim_{N_x,\,N_v\gg1} \|u_{\text{in}}\| = \frac{\mathcal{N}\sqrt{N_x}N_v}{2\sqrt{2}v_{\max}x_{\max}} = \mathcal{O}\left(\sqrt{N_x}N_v\right). \tag{116}$$

C.4 $\|F^{(1)}\|$

We use the l_1 and l_∞ norms from Eqs. (84a-84c) and the fact that the offdiagonal part of $F^{(1)}$ is antisymmetric, so $F_{ij}^{(1)} = (-1)^{\delta_{ij}+1}F_{ji}^{(1)}$. Then we apply this to the identity Eq. (87c) and get

$$\left\|F^{(1)}\right\| \leq \sqrt{\|F^{(1)}\|_1 \|F^{(1)}\|_\infty}$$

$$= \left\|F^{(1)}\right\|_1$$

$$\leq \nu(v_{\max}) + \frac{q^2\mathcal{N}}{2m_e\varepsilon_0\Delta v} \frac{N_x-1}{N_x} \cdot 2 + \frac{v_{\max}}{2\Delta x} \cdot 2 \tag{117}$$

$$\leq \mathcal{O}(N_v^{3/2} + N_x),$$

where $\nu_0 \geq \mathcal{O}(N_v^{3/2})$ was substituted from Eq. (60) to keep $R \lesssim 1$.

D   Derivation of quantities for complexity

D.1   Scaling with γ

We are interested in the large grid size limit. Hence substituting the results from the above subsections into Eq. (20) we get

$$\lim_{N_x,\,N_v \gg 1} \gamma^2 = \frac{\mathcal{N} m_e \varepsilon_0 N_x \sqrt{N_v}}{4\sqrt{2} v_{\max} x_{\max}^2 q^2} \left(\nu_0 + \sqrt{\nu_0^2 - \left(\frac{b}{2^7 \pi}\right)^{1/4} \frac{\sqrt{\mathrm{erf}\left(v_{\max}\sqrt{2b}\right)}}{\mathrm{erf}\left(v_{\max}\sqrt{b}\right)} \frac{4 q^2 \mathcal{N}}{m_e \varepsilon_0 \sqrt{v_{\max}}} \nu_0 N_v} \right).$$

(118)

If we again substitute $\nu_0 \geq \mathcal{O}(N_v^{3/2})$, the expression simplifies heavily and the second term in the square root vanishes, giving

$$\lim_{N_x,\,N_v \gg 1} \gamma = \frac{\mathcal{N}}{2\sqrt{2} v_{\max} x_{\max}} N_x^{1/2} N_v = \mathcal{O}(N_x^{1/2} N_v).$$

(119)

We see that γ goes to $\|u_{\mathrm{in}}\|$ asymptotically.

D.2   Solution decay

The fraction $\|u_{\mathrm{in}}\|/\|u(T)\|$, denoted as g_u, measures the decay of the norm of the solution and is unchanged by the rescaling so we ignore that here. To estimate this, we take $u(T)$ to be a Maxwellian, which is a very good approximation for $T \gtrsim 1/\nu_0$. Remember that $F^{(0)}$ in Eq. (96) encodes a Maxwellian that is weighted by values of $\nu(v)$. Therefore, if we set $\nu_0 = 1$ in $\|F^{(0)}\|$, the norm, we end up with $\|u(T)\| = \mathcal{O}(N_v^{1/2} N_x^{1/2})$. Then, substituting $\|u_{\mathrm{in}}\|$ from Eq. (116) results in

$$g_u = \frac{\|u_{\mathrm{in}}\|}{\|u(T)\|} \approx \frac{\mathcal{O}\left(N_v N_x^{1/2}\right)}{\mathcal{O}\left(N_v^{1/2} N_x^{1/2}\right)} = \mathcal{O}\left(N_v^{1/2}\right).$$

(120)

D.3   Carleman steps

The number of Carleman linearization steps $N_C(\delta)$ is defined in Eq. (44). The leading order contribution gives $\log(1)$ in the denominator To resolve this issue, we collect the second order contributions to the quantities that affect the denominator:

$$\nu_0 \geq \mathcal{O}\left(N_v^{3/2} + N_v\right),$$
$$\mu\left(F^{(1)}\right) = -\nu_0 - \mathcal{O}(1) = -\mathcal{O}\left(N_v^{3/2} + N_v\right),$$
$$\|u_{\mathrm{in}}\| = \mathcal{O}\left(N_x^{1/2} N_v - N_x^{1/2}\right),$$
$$\left(\frac{1}{\|\bar{u}_{\mathrm{in}}\|}\right)^2 = \frac{-\mu\left(F^{(1)}\right) + \sqrt{\mu\left(F^{(1)}\right)^2 - 4\|F^{(2)}\|\|F^{(0)}\|}}{2\|u_{\mathrm{in}}\|\|F^{(2)}\|}$$
$$= \mathcal{O}\left(1 + N_x^{-1} + N_v^{-1/3}\right),$$
$$\log(1/\|\bar{u}_{\mathrm{in}}\|) = \mathcal{O}\left(N_x^{-1} + N_v^{-1/3}\right).$$

(121)

The second order scalings of $\|F^{(2)}\|$ and $\|F^{(0)}\|$ are not needed because they fall out, their leading order size appears as the second order contribution to the full expression

above. Note that the second order contributions originate from terms such as $1/\Delta v = 2v_{\max}/(N_v - 1) = \mathcal{O}(N_v^{-1} + N_v^{-2})$. Substituting the above scalings into Eq. (44) yields

$$N_C(\delta) = \mathcal{O}\left(\left(N_x + N_v^{1/3}\right)\log\left(\frac{TN_v^2}{\delta}\right)\right). \tag{122}$$

D.4 $\|A\|$

Recall that Eq. (45) bounds $\|A\|$. To compute that, we first perform the rescaling discussed in Subsection 3.3.1:

$$\left\|\bar{F}^{(2)}\right\| = \mathcal{O}\left(N_v^{1/2}N_x^{-1/2}N_x^{1/2}N_v\right) = \mathcal{O}\left(N_v^{3/2}\right),$$
$$\left\|\bar{F}^{(0)}\right\| = \mathcal{O}\left(\frac{\nu_0 N_x^{1/2}N_v^{1/2}}{N_x^{1/2}N_v}\right) = \mathcal{O}\left(N_v\right), \tag{123}$$

where we used $\nu_0 \geq \mathcal{O}(N_v^{3/2})$. Then we obtain

$$\|A\| \leq N_c(\delta)\left(\underbrace{\mathcal{O}\left(N_v\right)}_{\|\bar{F}^{(0)}\|} + \underbrace{\mathcal{O}\left(N_v^{3/2} + N_x\right)}_{\|F^{(1)}\|} + \underbrace{\mathcal{O}\left(N_v^{3/2}\right)}_{\|\bar{F}^{(2)}\|}\right)$$
$$= \mathcal{O}\left(\left(N_x^2 + N_v^{11/6} + N_x N_v^{3/2}\right)\log\left(\frac{TN_v^2}{\delta}\right)\right), \tag{124}$$

where $\|F^{(1)}\|$ is unaffected by the rescaling and ends up being the largest, and with $N_C(\delta)$ substituted from Eq. (122).

D.5 Sparsity

The sparsity of $\bar{F}^{(0)}$ is N since it is fully filled, that of $F^{(1)}$ is 5 and that of $\bar{F}^{(2)}$ is $2N$ due to the 2 double integrals encoded per row. Hence,

$$s = \max\left\{N, 5, 2N\right\} = 2N = \mathcal{O}(N). \tag{125}$$

D.6 $\Omega(\delta', \delta)$

The expression $\Omega(\delta', \delta)$ is defined in Eq. (44). In the asymptotic limit it takes the value

$$\Omega(\delta', \delta) = \mathcal{O}\left(\frac{T^2 N_v^{3/2}}{\delta'}\left(N_x^2 + N_v^{11/6} + N_x N_v^{3/2}\right)\log\left(\frac{TN_v^2}{\delta}\right)\right). \tag{126}$$

D.7 Complexity in terms of grid size

Substituting the above results into Eq. (42) and Eq. (43) we obtain the complexities. We use the identity $\log\left(X^a Y^b\right) = \mathcal{O}(\log(XY))$ as $X, Y \to \infty$ with fixed powers $a, b > 0$ to simplify our expressions. This gives us the query complexity

$$\mathcal{O}\Bigg(TN_x N_v^{3/2}\left(N_x^{7/2} + N_x^{3/2}N_v^{11/6} + N_x^{5/2}N_v^{3/2} + N_x^2 N_v^2 + N_x N_v^{7/2} + N_v^{23/6}\right) \times$$
$$\times \log^{\frac{3}{2}}\left(\frac{TN_v}{\delta}\right)\frac{\log\Omega(\delta', \delta)}{\log\log\Omega(\delta', \delta)}\mathrm{poly}\left[\left(N_x + N_v^{1/3}\right)\log\left(\frac{TN_v}{\delta}\right)\right] \times \tag{127}$$
$$\times \mathrm{polylog}\left[\log\Omega(\delta', \delta), T\left(N_x + N_v\right)\log\left(\frac{TN_v}{\delta}\right), N_x N_v, \frac{1}{\varepsilon_q}\right]\Bigg),$$

and the gate complexity that is larger by a factor of

$$\mathcal{O}\left(\text{polylog}\left[\log\Omega(\delta',\delta), T\left(N_x + N_v\right)\log\left(\frac{TN_v}{\delta}\right), \frac{1}{\varepsilon_q}\right]\right), \tag{128}$$

where asymptotically we have

$$\log\Omega(\delta',\delta) = \mathcal{O}\left(\log\left[\frac{TN_v}{\delta'}(N_x + N_v)\log\left(\frac{TN_v}{\delta}\right)\right]\right). \tag{129}$$

Now, for a grid with a fixed N_x and N_v ratio, i.e substituting $N_x^2 = \mathcal{O}(N) = N_v^2$, the query complexity simplifies to

$$\mathcal{O}\left(TN^{7/2}\log^{\frac{3}{2}}\left(\frac{TN}{\delta}\right)\frac{\log\Omega(\delta',\delta)}{\log\log\Omega(\delta',\delta)}\times\right.$$
$$\left.\times\,\text{poly}\left[N^{1/2}\log\left(\frac{TN}{\delta}\right)\right]\text{polylog}\left[\log\Omega(\delta',\delta), TN\log\left(\frac{TN}{\delta}\right), N, \frac{1}{\varepsilon_q}\right]\right), \tag{130}$$

and the gate complexity to the above times the factor

$$\mathcal{O}\left(\text{polylog}\left[\log\Omega(\delta',\delta), TN\log\left(\frac{TN}{\delta}\right), \frac{1}{\varepsilon_q}\right]\right), \tag{131}$$

with the simplified $\log\Omega(\delta',\delta)$ being

$$\log\Omega(\delta',\delta) = \mathcal{O}\left(\log\left[\frac{TN}{\delta'}\log\left(\frac{TN}{\delta}\right)\right]\right). \tag{132}$$

E   Classical complexity and error analysis

To solve the finite difference scheme given in Eqs. (11a-11d) on a classical computer for the distribution matrix at time $T = mh$, i.e. for $f(T)$, one needs to perform

$$\mathcal{O}\left(kmN\right) \tag{133}$$

elementary operations, where k is the order of the time integrator, m is the number of time-steps and $N = N_x N_v$ is the total number of grid-points. Note that the double integral is cumulative in x, hence it does not introduce any additional N_x or N_v factors. The (classical) error at each time-step of the integrator is the same as order as the one introduced at the very first time-step, which is

$$\varepsilon_c(h) = \mathcal{O}\left(h^{k+1} + h\left(\Delta x^2 + \Delta v^2\right)\right), \tag{134}$$

since the phase-space discretization is second order. After m steps it accumulates to

$$\varepsilon_c(T) = \mathcal{O}\left(m\left[h^{k+1} + h\left(\Delta x^2 + \Delta v^2\right)\right]\right). \tag{135}$$

Now, typically stability conditions require $h \leq \mathcal{O}(\Delta x^q + \Delta v^q)$, where $q \geq 1$ is some low integer. k is usually fixed in the range $[2, 5]$. In these practical cases then the first contribution in Eq. (135) is either asymptotically smaller than the second one or scales

the same way. These two cases result in the same scaling in big $\mathcal{O}$ notation. Therefore, for a grid with a fixed N_x and N_v ratio, i.e with $N_x^2 = \mathcal{O}(N) = N_v^2$, we have

$$\varepsilon_c(T) \approx \mathcal{O}\left(\frac{T}{N}\right), \tag{136}$$

because $\Delta x = \mathcal{O}\left(1/\sqrt{N}\right) = \Delta v$ and $m = T/h$. So the classical time complexity is approximately

$$\approx \mathcal{O}\left(m \cdot \frac{T}{\varepsilon_c}\right) = \mathcal{O}\left(T^2 \varepsilon_c^{-1}\right), \tag{137}$$

where $\varepsilon_c = \varepsilon_c(T)$ is understood. This is now a function of ε_c and T only, hence mathematically comparable to the quantum gate complexity. k can be ignored as it is fixed.