

Quantum Linear PDE Solution Methods

René Steijl

James Watt School of Engineering
University of Glasgow
James Watt South Building
G12 8QQ Glasgow
United Kingdom
rene.steijl@glasgow.ac.uk

ABSTRACT

Finite-Volume discretization methods for linear PDEs that lead to circulant matrices are discussed in this set of lecture notes. The quantum-circuit implementation of these sparse matrices is detailed. The resulting quantum-circuits represent the time-update by a single time-step that can be used in the time-integration of linear Partial Differential Equations (PDEs). A key feature of the sparse-matrix resulting from the finite-volume discretization is that in general it is a non-unitary matrix. It is shown how such a non-unitary matrix can be embedded in a matrix of double the original size. For the linear advection equation with periodic boundary conditions, the discretization as well as the quantum-circuit construction are detailed. Finally, time-integration methods for systems of linear PDEs are discussed, in particular a method based on the 'reservoir' technique. This latter part forms a first step toward the quantum-circuit implementation of the Discrete-Velocity Method (DVM) for the kinetic Boltzmann equation, as discussed in a following set of lectures notes in the current Lecture Series.

Contents

1.0 Quantum algorithms for linear PDEs using a circulant matrix	2
1.1 Linear advection equation	2
1.2 Review of key properties of unitary matrices	2
1.3 Toeplitz and circulant matrices	2
1.4 Quantum circuit implementation based on unitary dilation	4
2.0 Solution methods for systems of linear PDEs	8
2.1 Reservoir technique as basis for quantum algorithms for linear PDEs	8
2.2 Finite-Volume method for one-dimensional collisionless Boltzmann equation	9
2.3 Reservoirs, CFL counters and variable time-step	9
3.0 Summary and further developments	11

1.0 QUANTUM ALGORITHMS FOR LINEAR PDES USING A CIRCULANT MATRIX

1.1 Linear advection equation

Spatial discretization of the linear advection equation using the Finite-Volume Method (FVM) on a uniform one-dimensional mesh with N cells (spacing Δx) leads to the following set of linear differential equations,

$$\begin{aligned} \frac{\partial f}{\partial t} + c \frac{\partial f}{\partial x} &= 0 \\ \Rightarrow \frac{\partial \vec{f}}{\partial t} &= \mathcal{A} \vec{f} ; \quad \vec{f} = (f_0, f_1, \dots, f_{N-1})^T \end{aligned} \quad (1)$$

where f_i represent the cell-centre value of f in cell i and \mathcal{A} represents a sparse $N \times N$ matrix resulting from the discretization. As is well-described in textbooks on the topic, discretization of time using a first-order accurate forward (explicit) time-stepping approach to Equation (1) leads to the following result,

$$\frac{\vec{f}^{n+1} - \vec{f}^n}{\Delta t} = \mathcal{A} \vec{f}^n \Rightarrow \vec{f}^{n+1} = \vec{f}^n + \Delta t \mathcal{A} \vec{f}^n = (\mathcal{I} + \Delta t \mathcal{A}) \vec{f}^n \quad (2)$$

where n represents the (current) time level and Δt the time step used. \mathcal{I} is the $N \times N$ identity matrix. **For a quantum computer implementation of the operation performed on the solution vector a key challenge is the fact that in general $\mathcal{I} + \Delta t \mathcal{A}$ will not be a unitary matrix.**

1.2 Review of key properties of unitary matrices

The conjugate transpose (or Hermitian adjoint matrix) of a complex matrix is the result of transposing the matrix and replacing its elements by their conjugates. A Hermitian matrix is a square matrix with complex entries that is equal to its own conjugate transpose. A real matrix is Hermitian if it is symmetric. A unitary matrix is a matrix whose inverse equals its conjugate transpose.

1.3 Toeplitz and circulant matrices

An $N \times N$ Toeplitz matrix is a matrix of the form,

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \dots & t_{-(N-3)} & t_{-(N-2)} & t_{-(N-1)} \\ t_1 & t_0 & t_{-1} & \dots & t_{-(N-4)} & t_{-(N-3)} & t_{-(N-2)} \\ t_2 & t_1 & t_0 & \dots & t_{-(N-5)} & t_{-(N-4)} & t_{-(N-3)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ t_{N-3} & t_{N-4} & t_{N-5} & \dots & t_0 & t_{-1} & t_{-2} \\ t_{N-2} & t_{N-3} & t_{N-4} & \dots & t_1 & t_0 & t_{-1} \\ t_{N-1} & t_{N-2} & t_{N-3} & \dots & t_2 & t_1 & t_0 \end{pmatrix} \quad (3)$$

so that the $N \times N$ Toeplitz matrix is fully described by the $2N-1$ entries of its first row and column. Mahasinghe and Wang[1] associate the following array to the Toeplitz matrix (using an extra zero element):

$$\psi_T = (t_0, t_{-1}, t_{-2}, \dots, t_{-(N-2)}, t_{-(N-1)}, 0, t_{N-1}, t_{N-2}, \dots, t_3, t_2, t_1)$$

A special kind of Toeplitz matrices involve matrices where for each row the elements have been moved right by one position relative to the previous row. This characteristic defines a *circulant matrix*.

A key property of circulant matrices is their diagonalization using the discrete Fourier transform matrix F_N . For an $N \times N$ circulant matrix C , diagonalization gives,

$$\begin{aligned} C &= F_N^* \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_{N-1}) F_N \\ \lambda_j &= c_0 + c_1 \omega^j + c_2 \omega^{2j} + \dots + c_{N-1} \omega^{j(N-1)} \end{aligned} \quad (4)$$

where λ_j ($j \in [0, N - 1]$) is the j th eigenvalue of C . Here, c_0, \dots, c_{N-1} are the N elements on the first row of matrix C and $\omega = \exp(2\pi i/N)$. In Equation (4), F_N and F_N^* represent the Discrete Fourier transform and the matrix for its reverse, respectively.

Circulant matrices are important in the discretization of the one-dimensional linear advection on a uniformly-spaced mesh.

Applying a standard finite-volume 1st-order accurate upwind-based discretization to the model problem considered here leads to an operator of the form,

$$\mathcal{I} + \Delta t \mathcal{A}_{upw} = \begin{pmatrix} 1 - \frac{c\Delta t}{\Delta x} & 0 & 0 & \dots & 0 & 0 & +\frac{c\Delta t}{\Delta x} \\ +\frac{c\Delta t}{\Delta x} & 1 - \frac{c\Delta t}{\Delta x} & 0 & \dots & 0 & 0 & 0 \\ 0 & +\frac{c\Delta t}{\Delta x} & 1 - \frac{c\Delta t}{\Delta x} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 - \frac{c\Delta t}{\Delta x} & 0 & 0 \\ 0 & 0 & 0 & \dots & +\frac{c\Delta t}{\Delta x} & 1 - \frac{c\Delta t}{\Delta x} & 0 \\ 0 & 0 & 0 & \dots & 0 & +\frac{c\Delta t}{\Delta x} & 1 - \frac{c\Delta t}{\Delta x} \end{pmatrix} \quad (5)$$

where periodic boundary conditions and $c > 0$ have been assumed. This discretization matrix is also a *circulant matrix*. For the matrix shown in Equation (5), the eigenvalues are:

$$\lambda_j = 1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x} \omega^{j(N-1)}; \quad j \in [0, N - 1] \quad (6)$$

Although a Toeplitz matrix is not circulant in general, any Toeplitz matrix T can be embedded in a circulant matrix defined by,

$$C_T = \begin{pmatrix} T & B_T \\ B_T & T \end{pmatrix}; \quad C_T \begin{pmatrix} \psi \\ 0 \end{pmatrix} = \begin{pmatrix} T & B_T \\ B_T & T \end{pmatrix} \begin{pmatrix} \psi \\ 0 \end{pmatrix} = \begin{pmatrix} T\psi \\ B_T\psi \end{pmatrix} \quad (7)$$

where,

$$B_T = \begin{pmatrix} 0 & t_{N-1} & t_{N-2} & \dots & t_3 & t_2 & t_1 \\ t_{-(N-1)} & 0 & t_{N-1} & \dots & t_4 & t_3 & t_2 \\ t_{-(N-2)} & t_{-(N-1)} & 0 & \dots & t_5 & t_4 & t_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{-3} & t_{-4} & t_{-5} & \dots & t_0 & t_{N-1} & t_{N-2} \\ t_{-2} & t_{-3} & t_{-4} & \dots & t_{-(N-1)} & t_0 & t_{N-1} \\ t_{-1} & t_{-2} & t_{-3} & \dots & t_{-(N-2)} & t_{-(N-1)} & t_0 \end{pmatrix} \quad (8)$$

Considering the diagonalization of a circulant matrix, it can be observed that the discrete Fourier matrix F_N as well as its complex conjugate F_N^* represent unitary operations. However, the diagonal matrix consisting of the eigenvalues will in general not be unitary. In order to embed it in a unitary matrix, Mahasinghe and Wang[1] make use of the principle of *unitary dilation*.

1.4 Quantum circuit implementation based on unitary dilation

As discussed in previous sections, the first step involves discretization of the linear advection equation using an upwind-biased discretization method. With periodic boundary conditions, we limit the approach to cases where this creates an $N \times N$ circulant matrix \mathcal{A} . Then similar to Equation (7) we embed this matrix into a $2N \times 2N$ matrix \mathcal{C}_T

$$\mathcal{C}_T = \begin{pmatrix} \mathcal{A} & 0 \\ 0 & \mathcal{A} \end{pmatrix} ; \quad \mathcal{C}_T \begin{pmatrix} \psi \\ 0 \end{pmatrix} = \begin{pmatrix} \mathcal{A}\psi \\ 0 \end{pmatrix} \quad (9)$$

where matrices B_T as in Equation (7) are not required since \mathcal{A} is already circulant. This step is introduced so that adding the extra matrices B_T as in Equation (7) would be a step towards extending the current approach to discretized problems resulting in non-circulant Toeplitz matrices. Also, embedding the state vector ψ into a length $2N$ vector with N zero-amplitudes in the 'second' half of the vector facilitates the construction of the quantum implementation. The next step in the proposed algorithm involves the diagonalization of the circulant matrix \mathcal{A} , so that we find,

$$\mathcal{C}_T \begin{pmatrix} \psi \\ 0 \end{pmatrix} = \begin{pmatrix} F_N^* \Lambda_A F_N & 0 \\ 0 & F_N^* \Lambda_A F_N \end{pmatrix} \begin{pmatrix} \psi \\ 0 \end{pmatrix} \quad (10)$$

where Λ_A is the $N \times N$ diagonal matrix whose diagonal elements λ_m correspond to the eigenvalues of \mathcal{A} ,

$$\lambda_m = \sum_{k=0}^{N-1} c_k \omega^k = \sum_{k=0}^{N-1} c_k e^{2\pi i m k / N}, \quad m = 0, 1, \dots, N-1 \quad (11)$$

with c_0, \dots, c_{N-1} the N elements on the first row of matrix \mathcal{A} and $\omega = e^{2\pi i / N}$. In constructing the quantum-circuit implementation for the matrix-vector multiplication defined in Equation (10), it is important to observe that the discrete Fourier transform and its inverse defined by F_N and F_N^* , respectively, are unitary operations. However, the diagonal matrix Λ_A is in general non-unitary. The operations acting on $(\psi, 0)^T$ defined in Equation (10) can be split in three steps, i.e. after applying the discrete Fourier transform, the product of the non-unitary diagonal matrix and the state vector needs to be computed (to be followed by the inverse discrete Fourier transform). Based on the augmented solution vector $(\psi, 0)^T$ of length $2N$, unitary dilation can be used to replace matrix-vector product involving the original non-unitary diagonal matrix defined in Equation (10) with the following product,

$$\begin{pmatrix} \frac{1}{k} \Lambda_A & \sqrt{\mathcal{I} - \frac{1}{k^2} \Lambda_A \Lambda_A^*} \\ \sqrt{\mathcal{I} - \frac{1}{k^2} \Lambda_A^* \Lambda_A} & -\frac{1}{k} \Lambda_A^* \end{pmatrix} \begin{pmatrix} \psi \\ 0 \end{pmatrix} \quad (12)$$

where \mathcal{I} is the $N \times N$ identity matrix and k is the magnitude of the largest eigenvalue in $[\lambda_0, \dots, \lambda_{N-1}]$ (in general complex numbers). After applying the multiplication in Equation (12), followed by the inverse discrete Fourier transform, the augmented solution vector then contains the state,

$$\begin{pmatrix} \frac{1}{k} F_N^* \Lambda_A F_N \psi \\ F_N^* \sqrt{\mathcal{I} - \frac{1}{k^2} \Lambda_A^* \Lambda_A} F_N \psi \end{pmatrix} \quad (13)$$

so that up to a scaling factor k , the 'first' part (first N elements) of the state vector contains the required result.

Key aspect for quantum-circuit implementation:

- Starting from an augmented solution vector of length $2N$ $(\psi, 0)^T$, it follows that after the application of the matrix-vector product as outlined above will create non-zero amplitudes in the augmented part of the state vector initialized with zeros;
- In a time-marching approach with repeated application of the approach sketched above this 'noise' in the added part of the state vector causes errors in results after the 2nd application of the method - therefore the circuit can only be used for a single time-update;
- In a quantum-circuit implementation there is no (known) method to re-initialize the augmented part of the state vector with zeros without changing the state of the first N elements of the solution vector - extension to multiple time-step updates is therefore non-trivial;
- For the first N elements to obtain the correct state $\frac{1}{k}F_N^*\Lambda_A F_N\psi$, the quantum algorithm does not need to correctly implement the product involving $\sqrt{\mathcal{I} - \frac{1}{k^2}\Lambda_A^*\Lambda_A}$ correctly when applied to $(\psi, 0)^T$.

In the following the quantum algorithm is developed further, using the simplification referred to in the last item.

For a one-dimensional problem with N degrees of freedom, the solution vector $\psi = (\psi_0, \psi_1, \dots, \psi_{N-1})^T$ can be stored in a qubit register $|\tilde{\psi}\rangle$ with $\log_2(N)$ qubits. In the quantum algorithm developed here, we need an augmented solution vector (as outlined previously). For the one-dimensional problem with N degrees of freedom, an $\log_2(N) + 1$ qubit register $|\psi\rangle = |0\rangle|\tilde{\psi}\rangle$ is used instead. As illustration, for $N = 4$, and initial solution $\psi^{init} = (\psi_0^{init}, \psi_1^{init}, \psi_2^{init}, \psi_3^{init})^T$, qubit register $|\psi\rangle$ would be initialized as,

$$|\psi\rangle = \begin{bmatrix} \psi_{000} \\ \psi_{001} \\ \psi_{010} \\ \psi_{011} \\ \psi_{100} \\ \psi_{101} \\ \psi_{110} \\ \psi_{111} \end{bmatrix} = \begin{bmatrix} \psi_0^{init} \\ \psi_1^{init} \\ \psi_2^{init} \\ \psi_3^{init} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

Applying the diagonal matrix product, using unitary dilation, then involves the following unitary,

$$\left(\begin{array}{cccc|cccc} \frac{\lambda_0}{k} & 0 & 0 & 0 & \sqrt{1 - \frac{\lambda_0\lambda_0^*}{k^2}} & 0 & 0 & 0 \\ 0 & \frac{\lambda_1}{k} & 0 & 0 & 0 & \sqrt{1 - \frac{\lambda_1\lambda_1^*}{k^2}} & 0 & 0 \\ 0 & 0 & \frac{\lambda_2}{k} & 0 & 0 & 0 & \sqrt{1 - \frac{\lambda_2\lambda_2^*}{k^2}} & 0 \\ 0 & 0 & 0 & \frac{\lambda_3}{k} & 0 & 0 & 0 & \sqrt{1 - \frac{\lambda_3\lambda_3^*}{k^2}} \\ \hline \sqrt{1 - \frac{\lambda_0\lambda_0^*}{k^2}} & 0 & 0 & 0 & -\frac{\lambda_0^*}{k} & 0 & 0 & 0 \\ 0 & \sqrt{1 - \frac{\lambda_1\lambda_1^*}{k^2}} & 0 & 0 & 0 & -\frac{\lambda_1^*}{k} & 0 & 0 \\ 0 & 0 & \sqrt{1 - \frac{\lambda_2\lambda_2^*}{k^2}} & 0 & 0 & 0 & -\frac{\lambda_2^*}{k} & 0 \\ 0 & 0 & 0 & \sqrt{1 - \frac{\lambda_3\lambda_3^*}{k^2}} & 0 & 0 & 0 & -\frac{\lambda_3^*}{k} \end{array} \right) \quad (15)$$

Using qubits q_1 and q_2 within the 3-qubit register as control-qubits, the unitary operation defined in the previous equation can be split into 4 controlled operations on qubit q_0 (i.e. with qubits q_1 and q_2 acting as control qubits):

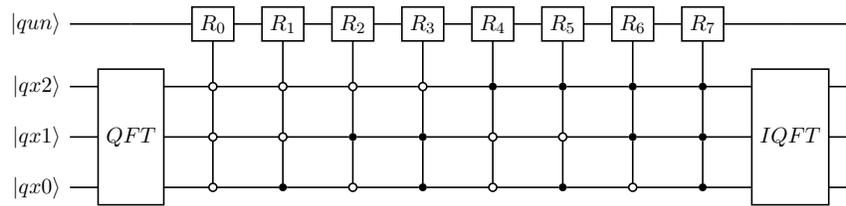
$ q_1 q_2\rangle$	controlled single-qubit operation	
$ 00\rangle$	$\begin{pmatrix} \frac{\lambda_0}{k} & \sqrt{1 - \frac{\lambda_0 \lambda_0^*}{k^2}} \\ \sqrt{1 - \frac{\lambda_0 \lambda_0^*}{k^2}} & -\frac{\lambda_0^*}{k} \end{pmatrix}$	$\begin{pmatrix} \psi_{000} \\ \psi_{100} \end{pmatrix}$
$ 01\rangle$	$\begin{pmatrix} \frac{\lambda_1}{k} & \sqrt{1 - \frac{\lambda_1 \lambda_1^*}{k^2}} \\ \sqrt{1 - \frac{\lambda_1 \lambda_1^*}{k^2}} & -\frac{\lambda_1^*}{k} \end{pmatrix}$	$\begin{pmatrix} \psi_{001} \\ \psi_{101} \end{pmatrix}$
$ 10\rangle$	$\begin{pmatrix} \frac{\lambda_2}{k} & \sqrt{1 - \frac{\lambda_2 \lambda_2^*}{k^2}} \\ \sqrt{1 - \frac{\lambda_2 \lambda_2^*}{k^2}} & -\frac{\lambda_2^*}{k} \end{pmatrix}$	$\begin{pmatrix} \psi_{010} \\ \psi_{110} \end{pmatrix}$
$ 11\rangle$	$\begin{pmatrix} \frac{\lambda_3}{k} & \sqrt{1 - \frac{\lambda_3 \lambda_3^*}{k^2}} \\ \sqrt{1 - \frac{\lambda_3 \lambda_3^*}{k^2}} & -\frac{\lambda_3^*}{k} \end{pmatrix}$	$\begin{pmatrix} \psi_{011} \\ \psi_{111} \end{pmatrix}$

(16)

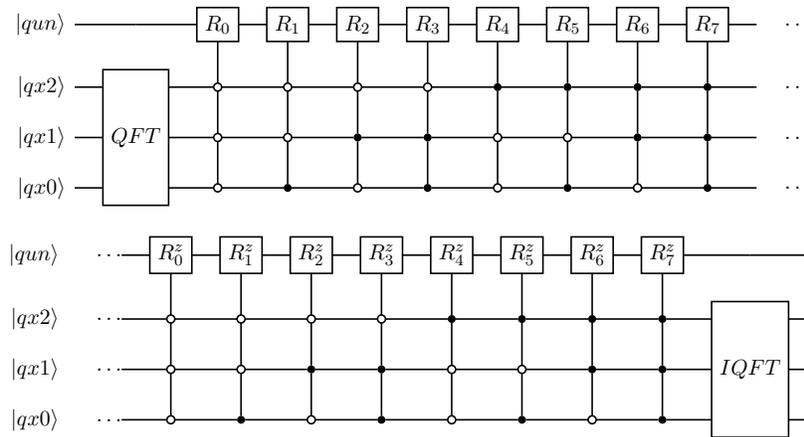
Here it is important to stress that the eigenvalues are in general complex numbers. The 'simplification' referred to earlier facilitated by having the 'second' half of the solution vector initialized with zeros in the quantum algorithm context enables a significant reduction in quantum circuit design. Specifically, it means that in the controlled single-qubit operations only the λ_m/k for $m = 0, \dots, 3$ need to be represented exactly to obtain the required output from the quantum algorithm. In the present work, the unitary operations as illustrated in Equation (16) are implemented using controlled versions of the elementary rotations $R_y(\beta)$ and $R_z(\gamma)$, with β and γ defining the respective angles over which rotation takes place. Then,

$$R_y(\beta)R_z(\gamma) = \begin{pmatrix} \cos(\beta/2)\cos(\gamma/2) - i\cos(\beta/2)\sin(\gamma/2) & -\sin(\beta/2)\cos(\gamma/2) - i\sin(\beta/2)\sin(\gamma/2) \\ \sin(\beta/2)\cos(\gamma/2) - i\sin(\beta/2)\sin(\gamma/2) & \cos(\beta/2)\cos(\gamma/2) + i\cos(\beta/2)\sin(\gamma/2) \end{pmatrix} \quad (17)$$

represents the effect of applying $R_z(\gamma)$ and $R_y(\beta)$ in succession. Here, γ and β are functions of the eigenvalues of the discretization operator. In the interest of brevity, this is not detailed here further. For more detail we refer to the Mahasinghe and Wang[1]. Figure 1 illustrates the quantum-circuit implementation for the linear advection problem for a periodic 8-cell one-dimensional domain.



(a) Quantum circuit with combined rotation gates $R = R^z R^y$



(b) Quantum circuit with separated rotation gates R^z and R^y

Figure 1: Quantum circuit implementation single-time step update for linear advection equation. $N_x = 8$, periodic boundary conditions.

2.0 SOLUTION METHODS FOR SYSTEMS OF LINEAR PDES

The previous sections showed quantum-circuit construction methods for linear PDEs, in particular the linear advection equation on a periodic one-dimensional domain. The presented techniques can be extended to systems of linear PDEs. However, motivated by our current research into quantum algorithms for lattice-based models of fluid dynamics and gas dynamics, the final part of these notes will introduce time-integration methods for systems of linear PDEs based on the 'reservoir' technique[2].

2.1 Reservoir technique as basis for quantum algorithms for linear PDEs

The reservoir technique was analyzed and applied to Godunov-type schemes for gas dynamics with the aim of achieving zero or very low numerical diffusion by Alouges et al.[2]. In their work it was applied to the Collela-Glaz solver, showing that for the Sod tube test problem impressive accuracy can be achieved when compared to results from finite-volume methods involving higher-order reconstructions (MUSCL, ENO WENO), despite the first-order accuracy of the stencil used in the advection step. In the context of quantum algorithms, we refer to the work on reservoir-technique based methods by Fillion-Gourdeau and co-workers[3, 4].

The motivation to investigate the reservoir technique in the context of quantum-algorithm designs for linear PDEs can be summarized as follows:

- Quantum-circuit implementations of discretization methods that are 1st-order accurate in space are appealing since such methods will not involve complex (non-linear) limiters (e.g. as used in MUSCL-type methods) or smoothness sensors (ENO, WENO method), etc. The reservoir technique offers the potential for reduction of numerical dissipation even when using these 1st-order accurate method;
- The 'CFL=1'-like condition used for the different 'waves' associated different eigenvalues of the considered system of equations offers the potential to use 'streaming'-type methods where values move between lattice/mesh points during a time-step.

As a step toward quantum algorithms for the Discrete-Velocity Method (DVM) for the kinetic Boltzmann equations, the finite-volume discretization of the collisionless Boltzmann equation is presented here. The collisionless Boltzmann equation and quantum-circuit implementations for the DVM will be discussed in more detail in a following set of lecture notes in the current Lecture Series.

The collisionless Boltzmann equation defines the single-particle distribution in a three-dimensional phase (velocity-space) for each point in three-dimensional space and therefore involves a seven-dimensional solution space (including time) for a free-molecular gas consisting of a single monatomic species[5]. In the free-molecular flow regime, collisions between molecules is neglected, while collisions of particles with solid walls do need to be included. The collisionless Boltzmann equation can be written for a single-species flow as,

$$\frac{\partial F(\vec{x}, \vec{c}; t)}{\partial t} + \vec{c} \cdot \frac{\partial F(\vec{x}, \vec{c}; t)}{\partial \vec{x}} = 0 \quad (18)$$

$$F_{initial} = \frac{\rho}{(2\pi RT)^{3/2}} \exp \left[-\frac{(\vec{c} - \vec{u}_0)^2}{2RT} \right] \quad (19)$$

where $F(\vec{x}, \vec{c}; t)$ is the single-particle distribution function, and $\vec{x} = (x, y, z)^T$ and $\vec{c} = (c_x, c_y, c_z)^T$ represent three-dimensional space and three-dimensional phase (velocity) space, respectively. $F_{initial}$ defines the

Maxwell-Boltzmann equilibrium distribution, for a local gas mass density ρ , temperature T and mean gas velocity \vec{u}_0 , typically used in simulations as initial condition. The particle number density n and gas mass density ρ are related as $\rho = nm$, for molecular mass m . For the Maxwell-Boltzmann equilibrium distribution it follows that the most probable (thermal) speed of a particle depends on temperature $V_{mp} = \sqrt{2RT}$ with R the specific gas constant for the gas considered. For a molecular mass m this gas constant $R = k_b/m$, with k_b the Boltzmann constant. In the following set of lectures notes, the dimensional reduction of the collisionless Boltzmann equation will be detailed. Here it suffices to say that **for 1D and 2D flows**, a system of two linear PDEs of the type shown in Equation (18) governing **two reduced distribution function** results.

2.2 Finite-Volume method for one-dimensional collisionless Boltzmann equation

For illustration purposes, a one-dimensional uniformly spaced finite-volume domain is considered with cell (centre) index j . The cell interface between cells $j - 1$ and j is denoted with index $j - 1/2$ and similarly data related to right cell interface of cell j (connecting cells j and $j + 1$) are indexed as $j + 1/2$. The uniform cell spacing is Δx . A discrete-velocity method (DVM) with uniformly spaced segments based on the trapezoidal integration rule is employed with n_{DV} discrete velocities to discretize the phase space of the one-dimensional collisionless Boltzmann equation. The discrete velocities are defined as $c_k \in [c_{min}, \dots, c_{max}]$, for $k = 0, \dots, n_{DV} - 1$ and a uniform step size in velocity space $\Delta c = (c_{max} - c_{min})/n_{DV}$. Furthermore, $c_{min} = -c_{max}$. It is assumed that all discrete velocities are non-zero, making the system of equations strictly hyperbolic. The reduced particle distribution function $f(x_j, c_k; t^n)$ in cell j for discrete-velocity k at time level t^n is denoted here as $f_{k;j}^n$. Equivalently $g_{k;j}^n$ for reduced particle distribution function $g(x_j, c_k; t^n)$. Using up-wind fluxes in velocity space, the discretized one-dimensional collisionless Boltzmann equation then becomes,

$$\begin{pmatrix} f_{k;j} \\ g_{k;j} \end{pmatrix}^{n+1} = \begin{pmatrix} f_{k;j} \\ g_{k;j} \end{pmatrix}^n - c_k \frac{\Delta t_n}{\Delta x} \begin{cases} \begin{pmatrix} f_{k;j} \\ g_{k;j} \end{pmatrix}^n - \begin{pmatrix} f_{k;j-1} \\ g_{k;j-1} \end{pmatrix}^n & \text{for } c_k > 0 \\ \begin{pmatrix} f_{k;j+1} \\ g_{k;j+1} \end{pmatrix}^n - \begin{pmatrix} f_{k;j} \\ g_{k;j} \end{pmatrix}^n & \text{for } c_k < 0 \end{cases} \quad (20)$$

The Euler forward-in-time integration as used in this equation clearly limits the admissible time-step according to the CFL criterion to $\Delta t_n \leq \frac{\Delta x}{c_{max}}$ where c_{max} represents the largest discrete velocity in absolute value.

Considering Equation (20) we can observe that if we use a first-order accurate method in space (i.e. distribution functions are assumed constant within each cell), time-integration with CFL=1 leads to an exact propagation for the distribution function(s) $f_{k;j}$ and $g_{k;j}$ corresponding to the largest discrete velocity in absolute value, i.e. k is such that $|c_k| = c_{max}$. For all other indices k , the distribution functions will move less than a cell width Δx during the time step Δt_n , leading to the need to interpolate the discretized solution within each cell and therefore introducing numerical dissipation. **The reservoir technique as applied here to collisionless Boltzmann equation is aimed at avoiding this and involves 'exact' propagation of each of the distribution functions from one cell-center to next neighbours during the time-integration process.** A helpful characteristic of the considered reduced collisionless Boltzmann system is that both reduced distribution functions convect at the same discrete velocities, i.e. the convection operator in both equations have the same eigenvalues.

2.3 Reservoirs, CFL counters and variable time-step

For each cell face, CFL counters are introduced as $c_{k;j\pm 1/2}^n$ for $k = 0, \dots, n_{DV} - 1$. At the start of the time-integration these counters are all initialized to zero. These counters will be updated during each time step by

$|c_k|\Delta t_n/\Delta x$. For convenience, the following temporary variables are introduced,

$$C_{k;j\pm 1/2}^{n+1} = c_{k;j\pm 1/2}^n + |c_k|\frac{\Delta t_n}{\Delta x} \quad (21)$$

Since for the considered system, the eigenvalues (discrete-velocities) in the upwind discretization are identical for each cell face, the CFL counters are identical for each cell face as well. This greatly simplifies the following integration method since only a single set of counters for all discrete velocities have to be considered rather than a set for each cell face.

The time-step Δt_n is limited such that $C_{k;j\pm 1/2}^{n+1} \leq 1$. Here, the time-step Δt_n is selected by finding the minimum among all j and k ,

$$\Delta t_n = \min_{j,k} \left(\left[1 - c_{k;j\pm 1/2}^n \right] \frac{\Delta x}{|c_k|} \right) \quad (22)$$

This choice of time step will result in at least one of the CFL counters to reach 1 in the considered time step, i.e. $C_{k;j\pm 1/2}^{n+1} = 1$ for one or more discrete velocities k (typically only for one eigenvalue during each time step), while never exceeding the value of 1. The underlying idea of the reservoir technique is to introduce reservoirs for each cell face $R_{k;j\pm 1/2}^n$ for $k = 0, \dots, n_{DV} - 1$ which are initially set to zero at the start of the time integration. At each time step we fill up the reservoirs $R_{k;j\pm 1/2}$ with the current numerical flux difference upwinding depending on the sign of c_k . As long as the CFL counter for the considered discrete velocity remains less than 1, this process continues, i.e. with the CFL counters gradually updated according to

$$C_{k;j\pm 1/2}^{n+1} = c_{k;j\pm 1/2}^n + |c_k|\frac{\Delta t_n}{\Delta x} \quad (23)$$

Furthermore, temporary variables $\tilde{f}_{k;j\pm 1}$ and $\tilde{g}_{k;j\pm 1}$ are introduced to facilitate the update due to the numerical flux difference upwinding for both reduced distribution functions. The idea is that the temporary variables will be updated when the CFL counter hits the value 1, while the update will go into the reservoirs for CFL counters below 1. The temporary variables $\tilde{f}_{k;j\pm 1}$ and $\tilde{g}_{k;j\pm 1}$ are used to update the solution to the new time level $n + 1$, therefore the only non-zero updates will occur for the discrete velocity (or velocities) for which the CFL counter reached 1. Once a CFL counter $C_{k;j\pm 1/2}^n$ for a discrete velocity k has reached 1, this counter as well as the reservoir associated with this discrete velocity will be set to 0 before the start of the next time-step.

Table 1: Reservoir method for velocity-space boundaries ± 8 reference velocity units ($\sqrt{2RT_r}$) and $\Delta x = 1$.

n_{DV}	n_{cycle}	Δc	$c_{k,min}$	$c_{k,max}$	T_{cycle}	ave. Δt
16	49	1.000	0.5000	7.5000	2.0	0.04167
32	213	0.500	0.2500	7.7500	4.0	0.01887
64	825	0.250	0.1250	7.8750	8.0	0.00971
128	3327	0.125	0.0625	7.9375	16.0	0.00481

To facilitate the development of quantum algorithms based on the reservoir technique, a set of modifications was introduced by Todorova and Steijl[6].

3.0 SUMMARY AND FURTHER DEVELOPMENTS

Finite-Volume discretization methods for linear PDEs that lead to circulant matrices were discussed. The quantum-circuit implementation of these sparse matrices was detailed. The quantum circuits represent the time-update by a single time-step in time-integration of a linear PDE. A key feature of the sparse-matrix resulting from the finite-volume discretization is that in general it is a non-unitary matrix. It was shown how such a non-unitary can be embedded in a matrix of double the original size. For the linear advection equation with periodic boundary conditions, the discretization as well as the quantum-circuit construction were detailed. Finally, time-integration for systems of linear PDEs were discussed, in particular method based on the 'reservoir' technique. This latter part can be seen as a first step toward the quantum-circuit implementation of the Discrete-Velocity Method (DVM) for the kinetic Boltzmann equation, as discussed in a following set of lectures notes in the current Lecture Series.

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support provided by the University of Glasgow.

REFERENCES

- [1] Mahasinghe, A. and Wang, J., "Efficient quantum circuits for Toeplitz and Hankel matrices," *Journal of Physics A: Mathematical and Theoretical*, Vol. 49, No. 27, 2016, pp. 275301.
- [2] Alouges, F., De Vuyst, F., Le Coq, G., and Lorin, E., "The reservoir technique: a way to make Godunov-type schemes zero or very low diffuse. Application to Colella–Glaz solver," *European Journal of Mechanics - B/Fluids*, Vol. 27, No. 6, 2008, pp. 643 – 664.
- [3] Fillion-Gourdeau, F., MacLean, S., and Laflamme, R., "Algorithm for the solution of the Dirac equation on digital quantum computers," *Phys. Rev. A*, Vol. 95, Apr 2017, pp. 042343.
- [4] Fillion-Gourdeau, F. and Lorin, E., "Simple digital quantum algorithm for symmetric first-order linear hyperbolic systems," *Numerical Algorithms*, Vol. 82, 2019, pp. 1009–1045.
- [5] Cercignani, C., *The Boltzmann Equations and its Applications*, Springer-Verlag, New York, 2nd ed., 1987.
- [6] Todorova, B. and Steijl, R., "Quantum Algorithm for the collisionless Boltzmann equation," *J. Comp. Phys.*, Vol. 409, 2020, pp. 109347.

