

A high order cell-centered semi-Lagrangian scheme for multi-dimensional kinetic simulations of neutral gas flows

Y. Güçlü^{1,*}, W. N. G. Hitchon

Department of Electrical and Computer Engineering, University of Wisconsin, 1415 Engineering Dr., Madison, WI 53706, USA

Abstract

The term ‘Convected Scheme’ (CS) refers to a family of algorithms, most usually applied to the solution of Boltzmann’s equation, which uses a method of characteristics in an integral form to project an initial cell forward to a group of final cells. As such the CS is a ‘forward-trajectory’ semi-Lagrangian scheme. For multi-dimensional simulations of neutral gas flows, the cell-centered version of this semi-Lagrangian (CCSL) scheme has advantages over other options due to its implementation simplicity, low memory requirements, and easier treatment of boundary conditions. The main drawback of the CCSL-CS to date has been its high numerical diffusion in physical space, because of the 2nd order remapping that takes place at the end of each time step. By means of a Modified Equation Analysis, it is shown that a high order estimate of the remapping error can be obtained a priori, and a small correction to the final position of the cells can be applied upon remapping, in order to achieve full compensation of this error. The resulting scheme is 4th order accurate in space while retaining the desirable properties of the CS: it is conservative and positivity-preserving, and the overall algorithm complexity is not appreciably increased. Two monotone (i.e. non-oscillating) versions of the fourth order CCSL-CS are also presented: one uses a common flux-limiter approach; the other uses a non-polynomial reconstruction to evaluate the derivatives of the density function. The method is illustrated in simple one- and two-dimensional examples, and a fully 3D solution of the Boltzmann equation describing expansion of a gas into vacuum through a cylindrical tube.

Keywords: Boltzmann equation, Advection, Convected Scheme, Neutral gas kinetics

1. Introduction

In this paper we describe a straightforward and easily implemented scheme for solution of the Boltzmann equation (or similar kinetic equations) with a number of desirable

*Corresponding author.

Email addresses: yguclu@msu.edu (Y. Güçlü), hitchon@engr.wisc.edu (W. N. G. Hitchon)

¹Present address: Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA.

properties (straightforward conservation, positivity preservation and simplicity of implementation). The starting point is a solution of the kinetic equation in integral form, which uses the method of characteristics to take an initial cell and propagate it forward in time until it is remapped onto the phase space mesh. We refer to the scheme as a ‘Convected Scheme’ (CS) to reflect the fact that it focusses on describing the convective motion of the fluid [1]. More recently a terminology has arisen that would call this a ‘forward semi-Lagrangian’ scheme.

A version of this scheme follows initial cells until they are almost totally depleted by collisions [2; 3]. In this ‘Long Lived Moving Cell’ (LLMC) scheme, collisions remove particles from the cell periodically and the scheme ‘remaps’ those particles to other locations in velocity space. The earlier versions of the scheme limited the duration of the ballistic move to a small fraction of a collision time, after which the entire contents of the cell were remapped [1], the remapping reflecting the effects of the ballistic move (i.e. employing the method of characteristics) and for a minority of the population, reflecting in addition the effects of collisions.

The benefit of moving the initial cell is essentially that it allows us to focus on the conservation of various quantities associated with a single initial cell (number, energy, momentum) when the contents of that cell are remapped to the phase space mesh. Particular attention was paid in plasma simulations to handling the ballistic motion near the points at which particles bounce, and in the vicinity of the zero energy cell, which turns out to be a little difficult to handle while conserving energy [4; 5].

In the earlier work, the order of the scheme was not considered to be of great concern, since the methods were integral methods and the number of remappings per collision time (which is often the critical physical time period) or even during the entire simulation was small. The remapping errors were thus believed to be modest. Nevertheless, if the desirable properties of the scheme can be retained in a higher order version, this would represent an improvement.

It is the purpose of this paper to report a simple set of modifications (essentially a small correction to the final position of the MC upon remapping) which changes the order of the spatial move from second to fourth. The basic CS is unchanged by these modifications, in that conservation laws can be built into the scheme as before; the ease of implementation is similarly basically unaffected. (While a similar procedure can certainly be applied to the energy/velocity move, its effects on energy conservation mean that its desirability is less clear cut. In this work, we focus on neutral particle kinetics, where the velocity is constant.)

Section 2 briefly reviews the Convected Scheme, with particular emphasis on the ballistic operator and the remapping rule it employs. Section 3 discusses the qualitative and quantitative behavior of the numerical errors arising in the semi-Lagrangian solution of a transport problem, and it gives the motivations for the development of a high-order ballistic operator. In Section 4, a closed-form solution for the leading error of the Cell-Centered Semi-Lagrangian CS is obtained, which is amenable to be directly compensated by the use of ‘small corrections’ to the final position of the MCs, as described in Section 5. When a uniform velocity-field is considered, the 1D results generalize naturally to a 2D or 3D domain, by simply applying the corrections independently for each of the Cartesian axes. Section 6 presents a simplified treatment for the solution of the 3D continuity equation with non-uniform velocity field. Section 7 describes two different strategies for suppressing the oscillating behavior in the numerical solution which arises when the

initial conditions are not sufficiently smooth (for smoothness, a continuous second order derivative would be required): the resulting scheme correctly preserves the monotonicity of the solution for generic initial conditions. Section 8 describes applications to solution of the 1D advection equation with uniform velocity, 2D continuity equation with non-uniform velocity, and 3D-3V Boltzmann equation describing the gas expansion from a nozzle through a tube into vacuum. This 3D kinetic test was chosen for its applicability to space thruster simulations.

2. The Convected Scheme remapping rule

The Convected Scheme (CS) is a mesh-based method for the solution of Boltzmann's equation, which describes the time evolution of the distribution function $f(t, \mathbf{x}, \mathbf{v})$ of a set of identical particles. The scalar field $f(t, \mathbf{x}, \mathbf{v})$ can be interpreted as the particle number density in phase space and it has dimensions of $[m^{-6}s^3]$. The differential form of Boltzmann's equation is

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = \left[\frac{\partial f}{\partial t} \right]_{coll} \quad (1)$$

where $\mathbf{F}(t, \mathbf{x}, \mathbf{v})$ is the force field acting on each particle, and the term on the right-hand side is the *Boltzmann collision integral*, which describes the effect of collisions among particles. Equation (1) is expressed in an Eulerian form, and 7 scalar independent variables are present. Equivalently, one can focus on a Lagrangian trajectory $(\mathbf{x}(t), \mathbf{v}(t))$ in phase space, defined as:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{v}(t) \\ \frac{d\mathbf{v}}{dt} = \frac{1}{m} \mathbf{F}(t, \mathbf{x}, \mathbf{v}) \end{cases} \quad (2)$$

with initial conditions $(\mathbf{x}_0, \mathbf{v}_0)$ at $t = t_0$. Substituting (2) into (1), the *l.h.s.* can be identified as the total derivative of f , and (1) is rewritten as

$$\frac{Df}{Dt} = \left[\frac{\partial f}{\partial t} \right]_{coll}, \quad (3)$$

which states that the time rate of change of the distribution function along the trajectory (2) is only determined by the collision operator. Hence, in the absence of collisions f is constant along each phase-space trajectory. Following naturally from these considerations, the Convected Scheme solves (3) using a time-splitting procedure:

1. first, the distribution function is evolved for one time step according to the collision operator only

$$\frac{\partial f}{\partial t} = \left[\frac{\partial f}{\partial t} \right]_{coll} \quad (4a)$$

2. then, the updated distribution function is advected for one time step along the characteristic trajectories, according to the collisionless equation

$$\frac{Df}{Dt} = 0. \quad (4b)$$

Whether to execute the steps above in this order or not is really an arbitrary choice, since it only depends on the definition of time-step. The collision operator invariably involves the flux of particles in velocity space, but it is usually totally local in physical space. As such, step 1 is naturally solved on the phase-space mesh. As concerns step 2, the Convected Scheme really uses the integral form of (4b), with a single phase-space cell as the control-volume at $t = t_0$:

$$\iint_{C(t)} f(t, \mathbf{x}, \mathbf{v}) \, d\mathbf{x} \, d\mathbf{v} = \iint_{C_0} f(t_0, \mathbf{x}_0, \mathbf{v}_0) \, d\mathbf{x}_0 \, d\mathbf{v}_0 \quad (5)$$

where the *r.h.s.* is just the total number of particles contained in cell C_0 at instant t_0 , and $C(t)$ is the control-volume that evolves from C_0 by means of the map defined by (2). For obvious reasons, $C(t)$ is called a Moving Cell (MC). The meaning of (5) is clear: it is merely a statement of mass conservation for a MC, during the ballistic motion. But to understand the practical utility of (5), a suitable discretization of the variables must be introduced. The total number of particles in the initial phase-space cell C_0 is

$$N_0 = f_{ij}(t_0) \Gamma_{ij} = N_{ij}(t_0)$$

where $f_{ij}(t)$ is the average value of the distribution function in the phase-space cell centered at point $(\mathbf{x}_i, \mathbf{v}_j)$, and Γ_{ij} is the corresponding volume in phase-space of the cell. It is evident from the last expression that the Eulerian phase-space mesh is needed not only to solve the collisional step, but also to give proper initial conditions to the ballistic operator:

$$N_{ij}^{MC}(t) \equiv N_{ij}(t_0) \quad t \geq t_0 .$$

At the end of the time step, the particles in the MC are remapped back onto the Eulerian mesh, where the collision operator is applied next. Alternatively, it is possible to evaluate the fraction of particles in the MC that have undergone a collision, and this permits one to remap to the mesh only those particles. In such cases, one can let the MC move for more than one time step, until the number of particles contained is a tiny fraction of the initial number, or until the geometrical distortion of the MC will require remapping to preserve accuracy. The application of the aforementioned ideas led to the Long-Lived Moving Cell (LLMC) version of the CS [2; 3].

In any case, the exact time evolution of the MC can be evaluated analytically only in a few situations, and important approximations must be introduced regarding the following points:

1. parametrization of the profile of f in the MC,
2. parametrization of the shape of the MC in phase-space,
3. definition of the Lagrangian trajectories that are to be traced for each MC.

Evidently, the choices made about the points above must be compatible with each other. As concerns point 1, while the MC is always assumed to have a uniform density profile in physical space, it is usually considered to be a Dirac delta in velocity space.

As concerns point 2, the MC is usually obtained from the Cartesian product of a convex polyhedron in physical space and a convex polyhedron in velocity space. This is always the case when dealing with more than one dimension in physical space. Moreover, the MC is usually not deformable in velocity space, but it can usually undergo

expansion/contraction and distortion in physical space.

As concerns point 3, there are three ways of handling the motion of the cell, that we have employed: first, we can move the cell center and assume the cell shape does not change; second, we can move the centers of cell faces and assume that each face does not change its orientation; third, we can move vertices of cells. In any case, a single initial velocity, equal to the central velocity \mathbf{v}_j , is given to the trajectories.

At the end of the ballistic move, particles must be placed back on the phase-space mesh according to the Convective Scheme remapping rule; for a domain in \mathbb{R}^m , this consists in depositing the particles from the Moving Cell onto the overlapped fixed cells C_i , according to the m-dimensional integral over C_i :

$$N_i += \oint_{C_i} n_{MC}(\mathbf{x}) d\mathbf{x} = \oint_{C_i \cap MC} n_{MC}(\mathbf{x}) d\mathbf{x} \quad (6)$$

where $n_{MC}(\mathbf{x})$ is the m-dimensional density in the Moving Cell (MC). The symbol $+=$ is the *increment assignment operator*, meaning that the quantity on the *l.h.s.* is increased by the amount on the *r.h.s.*; hence, (6) denotes the contribution to N_i due to a given MC. Since we are assuming that n_{MC} is constant in the MC, the above rule can be simplified to

$$\begin{aligned} N_i += & n_{MC} \oint_{C_i \cap MC} d\mathbf{x} = n_{MC} V_{C_i \cap MC} = \\ & = n_{MC} V_{MC} \frac{V_{C_i \cap MC}}{V_{MC}} = N_{MC} F_i \end{aligned} \quad (7)$$

where the symbol V is used to denote the volumes in \mathbb{R}^m , and F_i is called the "overlapping fraction". Obviously, the sum of all the overlapping fractions is one:

$$\sum_i F_i = \sum_i \left(\frac{V_{C_i \cap MC}}{V_{MC}} \right) = \frac{V_{MC}}{V_{MC}} = 1$$

In the simplest version of a Convected Scheme, the MCs are rigidly translated according to the motion of their center, and hence the density in each MC is uniform and constant during the time step. In the semi-Lagrangian version at hand, the MCs are remapped onto the mesh at the end of each time-step. This scheme will be referred to as the *Cell-Centered Semi-Lagrangian Convected Scheme* (CCSL-CS).

As we shall see, the CCSL-CS has many desirable properties: it is conservative, it preserves positivity, and it has no phase-error. On the other hand, the accuracy of the scheme is only 2nd order in space, the leading error being a strong numerical diffusion term which has a non-linear dependence on the Courant parameter. In order to increase the accuracy of the scheme, two different approaches can be used:

1. allowing for a non-uniform density in the MC;
2. given an a priori estimate of the remapping error, applying a small correction to the final position of the MC to compensate for it.

The former option complicates significantly the evaluation of the integral in (6), since (7) is not applicable. Seeking a more numerically efficient method, the latter option is investigated hereafter.

3. Numerical errors in a semi-Lagrangian scheme

In this section we discuss the errors which arise in a semi-Lagrangian (SL) solution of a transport problem. We point out that the order of the errors in a SL scheme without a Courant limit is different from the order of the same scheme if a Courant limit were assumed to hold.

The ballistic operator of the Convected Scheme is a discrete approximation of (4b), and it is effectively decoupled into two different phases:

1. the integration of the characteristic trajectories forward in time;
2. the remapping of the transported field from the Lagrangian mesh to the fixed (Eulerian) mesh.

As such, the CS ballistic operator falls into the category of ‘forward-trajectory semi-Lagrangian methods’. Semi-Lagrangian (SL) methods were first introduced by Robert at the beginning of the eighties [6], and extensively applied to weather forecasting, oceanography, and fluid dynamics in general. Most early SL methods traced the characteristic trajectories backward in time, from time t to $t - \Delta t$, so that the interpolation (or the remapping) was done from an Eulerian to a Lagrangian mesh, at time step $t - \Delta t$. SL schemes that project information forward along the trajectory have been in existence for some time (see the earlier versions of the CS, for example) but the terminology ‘forward-trajectory’ scheme seems to have originated with Purser in the early nineties [7].

As typical of SL schemes [8], there is no numerical stability requirement that defines a maximum allowed time step Δt_{\max} for a given mesh spacing Δx , i.e. no Courant-Friedrichs-Lewy (CFL) [9] limit applies to this explicit scheme. (Nevertheless, a ‘deformational CFL limit’ may still apply [7], which is usually much less restrictive.) As a consequence, provided that the integration of the characteristics matches its own stability requirements, the time discretization Δt and the space discretization Δx can be chosen independently, based on accuracy considerations.

To assess the accuracy of the scheme, there are two complementary approaches:

1. a-priori estimates of the Local Truncation Error (LTE), i.e. the error introduced by one time-step of the algorithm, as a function of Δt and Δx ;
2. a-posteriori analysis of the Global Error (GE), i.e. the error in the solution after a certain integration time T .

If it can be shown that the $\text{LTE} \leq \alpha \Delta t$ as $\Delta t, \Delta x \rightarrow 0$ (in which case the scheme is said to be *consistent*), and provided the scheme is also *stable*, then *convergence* is obtained, i.e. the $\text{GE}(t = T) \rightarrow 0$ for $\Delta t, \Delta x \rightarrow 0$. It is therefore natural to ask ‘how fast’ the scheme *converges* to the exact solution as $\Delta t, \Delta x \rightarrow 0$.

In a typical explicit Eulerian scheme the CFL limit requires $\Delta t = O(\Delta x)$, which makes it easy to write the Local Truncation Error in a form like $\text{LTE} = O(\Delta x)^s$. For sufficiently smooth initial conditions, and ‘well-behaved’ boundary-conditions, one can show that $\text{GE} = O(\Delta x)^s$ as well, and for this reason s is called the *order of convergence* of the scheme.

The form of the LTE is more complicated for a SL scheme, especially if the trajectories are obtained with a high-order ODE integrator; here, no relation is introduced between Δt and Δx , as they can be chosen independently. To lowest order approximation, one can decompose the LTE into two separate contributions: if the trajectory integrator

has order r , it will introduce an error $O(\Delta t^{r+1})$ at each time step; similarly, if the remapping algorithm has order p , it will introduce an error $O(\Delta x^p)$ at each time step. At time instant $t = T$ the Global Error in the solution will be approximately $N = T/\Delta t$ times the error at each time step, which yields an estimate for the rate of convergence of the form

$$\|f(x, T) - f_{ex}(x, T)\| = A \Delta t^r + B \frac{\Delta x^p}{\Delta t},$$

where $f(x, T)$ is the computed (numerical) solution at time T , and $f_{ex}(x, T)$ is the *exact* solution at the same time instant; a proper norm $\|\cdot\|$ should be chosen (usually L_1 , L_2 or L_∞) for the above relation to be meaningful. The last result, although not obtained by a rigorous derivation, is fully consistent with some accurate analyses for a broad class of SL schemes [10].

For a kinetic simulation of a neutral gas, if the physical coordinates are fixed in an inertial reference frame and the gravitational force is neglected, then $\mathbf{F} = 0$ in Boltzmann's equation (1), and the particles move in straight trajectories between two successive collisions. In this situation, the Convected Scheme is a so-called 'discrete velocity method' [11]: at each instant in time and position in space, the velocity of the particles can only assume one of the prescribed values defined by the mesh in velocity space. As a consequence, the velocity of the MC does not change during the ballistic phase, and no remapping is needed in velocity space; for each value \mathbf{v}_α of the velocity vector, the ballistic operator solves one passive advection equation with uniform constant velocity \mathbf{v}_α over the whole physical domain $\mathcal{D} \subset \mathbb{R}^3$:

$$\frac{\partial}{\partial t} f(\mathbf{x}, \mathbf{v}_\alpha) + \mathbf{v}_\alpha \cdot \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{v}_\alpha) = 0, \quad \mathbf{x} \in \mathcal{D}.$$

Figure 1 shows the typical behavior of the ballistic operator of the CSSL-CS in a 1D case, with uniform mesh spacing Δx , for a single velocity u_α . In the very simple case at hand,

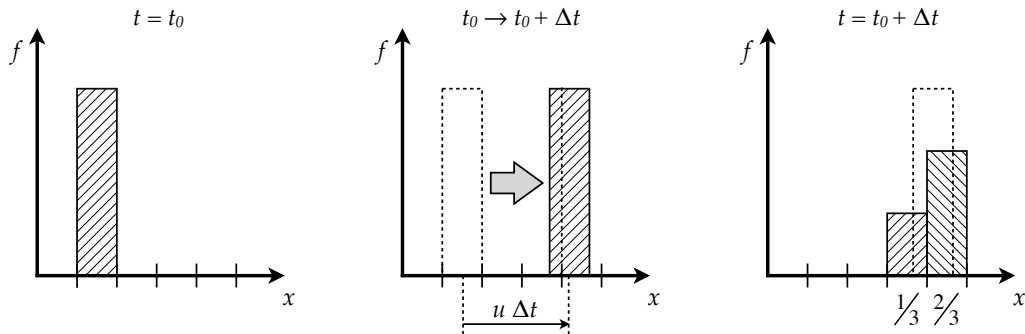


Figure 1: Ballistic operator of the 1D CSSL-CS, for a uniform mesh spacing. The middle frame shows the Moving Cell (MC) after a time step. Assuming that two thirds of the MC falls into the right hand cell which the MC overlaps, that is, the fraction of the MC's area which is in the right cell is two thirds, we would put two thirds of the particles in that cell.

a single MC is advected by a distance $u_\alpha \Delta t$, and it is then remapped on the 2 overlapped cells, the fraction being simply proportional to the overlap integral as given by (6) (in 1D it is a length, in 2D an area, and in 3D a volume). Already, the characteristic features of the scheme are clearly visible: positive attributes are

- since the advected quantity f is inherently positive, and since the integrals are positive as well, the scheme is *positivity preserving*;
- the centroid of the displaced density lies at its exact position, which means that the kinetic fluxes evaluated by the scheme are exact;

and negative attributes are:

- the density profile has flattened (i.e. two cells contain what was previously contained in one cell only) and one can easily imagine this process continuing until the density is uniform: *numerical diffusion*;
- the density profile has become asymmetric: this is a higher-order error, the *numerical dispersion*, which will be quickly masked by the numerical diffusion.

Intuitively, if the MC had moved by an exact multiple of Δx , its whole content would be remapped to one cell only, and there would be neither numerical diffusion nor numerical dispersion. In the interesting case that the MC had moved by exactly $(m + 0.5) \Delta x$ (m integer), we would have a situation with the maximum possible numerical diffusion (as the flattening would be maximum), but no numerical dispersion (since the final density profile would still be perfectly symmetric). In general, the error depends on the fractional part U_α of the displacement, written as

$$u_\alpha \Delta t = (m_\alpha + U_\alpha) \Delta x, \quad m_\alpha \in \mathbb{I}, \quad U_\alpha \in (-1, 1) \subset \mathbb{R},$$

and of course, the error will depend on the density profile itself (e.g., there is no numerical error for a uniform density profile).

If the following notation is introduced:

$$U_\alpha^+ = \frac{1}{2} (U_\alpha + |U_\alpha|) \geq 0$$

$$U_\alpha^- = \frac{1}{2} (U_\alpha - |U_\alpha|) \leq 0$$

the 1D CCSL-CS algorithm can be recast as a simple finite-difference single-step scheme with a 3-point stencil:

$$f_i^{k+1} = f_{i+m_\alpha-1}^k U_\alpha^+ + f_{i+m_\alpha}^k (1 - |U_\alpha|) - f_{i+m_\alpha+1}^k U_\alpha^-$$

where i is the index over space, and k is the index over time. Apart from the integer shift of m_α cells, the last expression is equivalent to the 1D Donor-cell scheme; it is important to note that this equivalence only holds in the case of uniform velocity, while for a generic velocity field there is an important difference: *the velocity in the CCSL-CS is not defined at the cell edges, but at the cell centers*. It should be pointed out that the analogy between the CCSL-CS and the Donor-cell scheme only holds for 1D domains, as the two schemes generalize differently to higher dimensions.

Based on the aforementioned considerations, one can easily adapt the results for the 1D Donor-cell scheme, obtained in [12] by a rigorous Modified Equation Analysis (MEA)

in the hypothesis of ‘sufficiently smooth’ density profile, to get a two-term expression for the *Modified Equation* of the 1D CCSL-CS:

$$\frac{\partial f}{\partial t} + u_\alpha \frac{\partial f}{\partial x} = \underbrace{\frac{\Delta x^2}{2\Delta t} (|U_\alpha| - U_\alpha^2) \frac{\partial^2 f}{\partial x^2}}_{\text{numerical diffusion}} + \underbrace{\frac{\Delta x^3}{6\Delta t} (-U_\alpha + 3|U_\alpha|U_\alpha - 2U_\alpha^3) \frac{\partial^3 f}{\partial x^3}}_{\text{numerical dispersion}} + H.O.T. , \quad (8)$$

where *H.O.T.* stands for ‘higher order terms’. The first term on the right hand side (*r.h.s.*) of the modified equation (8) is the leading error of the CCSL-CS, which scales with $\Delta x^2/\Delta t$. Such an error term is linearly proportional to the second spatial derivative of $f(x)$, and it has the form of a Numerical Diffusion:

$$\varepsilon_1 = \frac{\partial}{\partial x} \left(D_\alpha \frac{\partial f}{\partial x} \right) \quad D_\alpha = \frac{\Delta x^2}{2\Delta t} (|U_\alpha| - U_\alpha^2)$$

Such a numerical diffusion error, which may seem relatively innocuous, is actually quite pernicious to the numerical results, not only for the magnitude of the error, but also for the non-uniform distribution of the error in phase-space. Hence, it is worth analyzing the behavior of this term in some detail.

For a kinetic gas simulation, the velocities u_α are defined by the velocity mesh, and the time-step Δt is chosen based on the accuracy of the time-splitting procedure in (4) (a reasonable value of Δt is between 1/10 and 1/5 of the average collision time); as a consequence, Δx remains the only free parameter that influences the numerical diffusion coefficient, through the factor Δx^2 and the parameter U_α .

Figure 2 shows the non-monotonic decrease of the leading numerical diffusion error for decreasing values of the mesh spacing Δx , for a fixed velocity u_α and fixed time-step size Δt . The information provided by this picture is that the decrease of Δx reduces the error on average, but may in fact increase the error for certain velocities u_α which were positioned on a monotonically increasing region of the curve. The non-linear relation

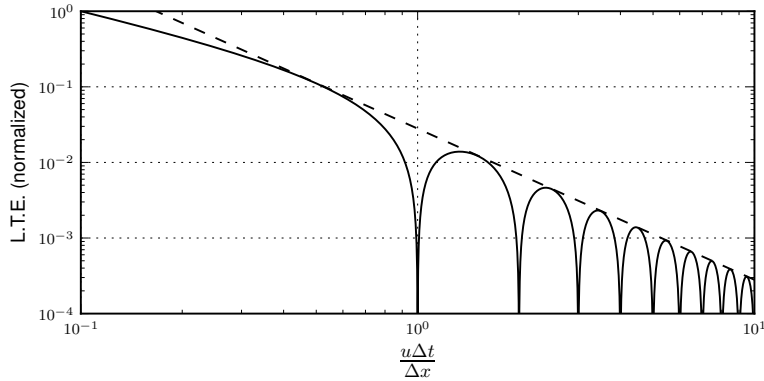


Figure 2: Numerical diffusion of the CCSL-CS in the solution of the 1D advection equation with uniform velocity u . The Local Truncation Error (normalized to 1) is plotted against the ‘Courant parameter’ $C_p = u\Delta t/\Delta x$ (solid line). The graph is obtained by keeping $u\Delta t$ constant (as typical in a kinetic simulation of a neutral gas), and reducing progressively the mesh-size Δx . The reference dashed line is multiply tangent to the curve of the LTE, and decreases as Δx^2 .

between the diffusion error and the velocity u_α produces strong anisotropy in the diffusion term when 2D and 3D spatial domains are considered. For example, the numerical diffusion error in 2D is

$$\mathcal{E}_1^{(2D)} = D_x \frac{\partial^2 f}{\partial x^2} + D_y \frac{\partial^2 f}{\partial y^2}, \quad \text{with} \quad \begin{cases} D_x = \frac{\Delta x^2}{2\Delta t} (|U_\alpha| - U_\alpha^2) \\ D_y = \frac{\Delta y^2}{2\Delta t} (|V_\alpha| - V_\alpha^2) \end{cases} \quad (9)$$

where U_α and V_α are obtained from the fractional part of the displacements $u_\alpha \Delta t$ and $v_\alpha \Delta t$, along the x and y directions respectively.

Assuming $\Delta x = \Delta y$, the diffusion coefficients in the two directions are most likely to be widely different from each other, i.e. *the numerical diffusion of the 2D CCSL-CS is strongly anisotropic*. To better appreciate this effect, one can fix the magnitude $\|\mathbf{v}\| = S$ of the velocity vector, and write $u = S \cos \theta$ and $v = S \sin \theta$, to obtain $D_x(\theta)$ and $D_y(\theta)$; here $\theta = \mathbf{v} \cdot \hat{\mathbf{x}} / \|\mathbf{v}\|$ is the angle between the velocity vector and the X axis of the grid. Figure 3 presents polar plots of $D_x(\theta)$ (solid line) and $D_y(\theta)$ (dashed line), for a few different values of the velocity magnitude: the numerical diffusion is isotropic only for those values of θ where the two curves cross each other; for all other values, there is a preferential direction for diffusion.

Intuitively, in a 3D geometry the situation gets even worse, as the numerical diffusion has more degrees of freedom for causing distortion in the numerical solution. As there is no way to avoid such a non-uniform distribution of the numerical error in phase-space, the problem can be mitigated only by globally reducing the magnitude of the error, a result that can be achieved by decreasing the mesh-size Δx .

In 3D, the solution converges with the same asymptotic behavior as in 1D and 2D, i.e. $\|\mathcal{E}\| = O(\Delta x^2)$, but the number N of cells in the domain increases as Δx^{-3} . Hence, $N \propto \|\mathcal{E}\|^{-3/2}$, which means that to reduce the error by a factor of 2, the number of cells in the domain must be increased by ≈ 2.83 times; accordingly, the memory requirement and the simulation time will increase by (at least) the same factor. $N \propto \|\mathcal{E}\|^{-3/2}$ is a quite unfavorable scaling for a multi-dimensional kinetic simulation, as the computational limitations may easily prevent us from reaching the required accuracy in the solution.

The aforementioned considerations are the main reasons that motivated the development of a higher-order ballistic operator, which will be proved to be 4th order accurate in space, that is $\|\mathcal{E}\| = O(\Delta x^4)$. The error of the new scheme still behaves ‘erratically’, similarly to the original scheme, but it is already much smaller in magnitude when a coarse mesh spacing is employed, and it has a much better scaling when Δx is reduced. In fact, the high-order scheme has $N \propto \|\mathcal{E}\|^{-3/4}$, so that to reduce the error by a factor of 2, the number of cells in the domain must be increased by ≈ 1.68 times only.

The novelty of the implementation is in the fact that, even though it is the remapping stage which causes the numerical error, this stage is left unaltered; instead, the remapping error is compensated a-priori by applying ‘small corrections’ to the final position of the MC prior to remapping. As another way to put it, an error is deliberately introduced in the trajectory integrator (which was exact, since the particles move in straight trajectories), such that it *exactly compensates for the diffusion and dispersion errors*, at least for sufficiently smooth density functions.

The resulting ballistic operator maintains the most desirable properties of the original CCSL-CS, (which are inherited from the fact that the remapping method is left unchanged): exact mass conservation, preservation of positivity, and simple and effi-

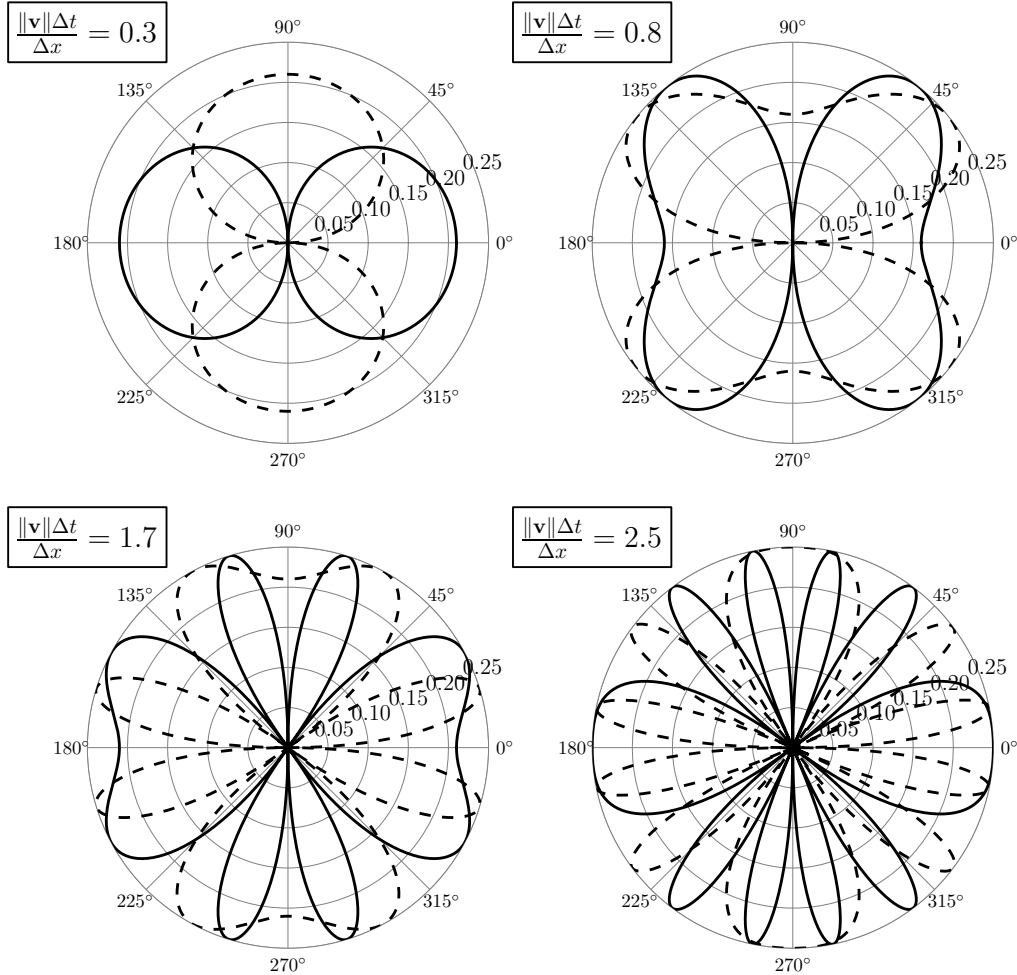


Figure 3: Numerical diffusion of the CCSL-CS scheme in the solution of the 2D advection equation with uniform velocity. For four different values of the speed $\|\mathbf{v}\|$, the diffusion coefficients along the directions x and y are plotted as functions of the angle $\theta = \mathbf{v} \cdot \hat{\mathbf{x}} / \|\mathbf{v}\|$: the solid lines represent $D_x(\theta)$, while the dashed lines represent $D_y(\theta)$. The value of a diffusion coefficient is obtained from the local radius $r(\theta) \in [0, 0.25]$ of the corresponding curve in the polar plot, multiplied by $\Delta x^2 / \Delta t$ (see (9)).

cient implementation. Moreover, the resulting scheme can be slightly modified to ensure monotonicity preservation: two different strategies will be presented to deal with this.

In the next Section, the LTE of the scheme is assessed when a non-uniform velocity field is considered. This is necessary in order to account for the small modifications to the final position of the MC: in fact, once divided by the constant time-step Δt , a modification to the position can be equivalently regarded as a modification to the velocity. Such corrections to the velocity field may be interpreted as ‘anti-diffusive’ and ‘anti-dispersive’ velocities: in this sense, our approach here is similar to the one used in

the MPDATA algorithm [13; 14], which is a high-order method based on the Donor-cell scheme. The main advantage of the high-order version of the CCSL-CS over MPDATA is that the corrections are evaluated a-priori and in a single step, without the need for a multi-pass slowly converging procedure.

As an aside, the general treatment in the next Section, which does not require the velocity field to be uniform, opens the CCSL-CS to a range of applications it was not originally designed for.

4. Modified equation analysis for the ‘low-order’ CCSL-CS

This section aims at the assessment of the error introduced by the 1D CCSL-CS, by means of a modified equation analysis (MEA) [15]. This is a standard procedure commonly applied to finite-difference schemes, based on a Taylor expansion of the solution around the point of interest (t, x) ; as such, we will assume the transported function to be ‘sufficiently smooth’.

In order to make the following treatment more manageable in the presence of a non-uniform velocity field, it will be assumed that the Courant parameter $U = u\Delta t/\Delta x$ is less than 1. As a consequence, $\Delta t = O(\Delta x)$ and the order of accuracy in space is equivalent to the order of accuracy in time; moreover, the schemes effectively ‘lose’ one order of accuracy: the original CCSL-CS will be said to be ‘1st order accurate’ (although in fact, since the time step is not constrained by the Courant criterion, it is actually second order accurate in space), and the high-order CCSL-CS will be regarded as being ‘3rd order accurate’ (again, the spatial order of accuracy is in fact fourth, since the time step is not in reality constrained to be small).

As a starting point, the equation to be solved is now the 1D continuity equation

$$\partial_t n + \partial_x (u n) = 0 \quad (10)$$

where the velocity $u(x, t)$ is not constrained to be constant in time or uniform in space. Nevertheless, for the sake of clarity, it will be assumed that $u > 0$ unless otherwise stated. In this situation, the 1D CCSL-CS is

$$n_i^{k+1} = U_{i-1}^k n_{i-1}^k + (1 - U_i^k) n_i^k \quad (11)$$

which is, after substitution of the 3rd order Taylor expansions about the point (x_i, t^k) ,

$$\begin{aligned} & \left(\Delta t \partial_t + \frac{\Delta t^2}{2} \partial_{tt}^2 + \frac{\Delta t^3}{6} \partial_{ttt}^3 \right) n + O(\Delta t^4) = \\ & - \left(\Delta x \partial_x - \frac{\Delta x^2}{2} \partial_{xx}^2 + \frac{\Delta x^3}{6} \partial_{xxx}^3 \right) (U n) + O(\Delta x^4) \end{aligned} \quad (12)$$

where $n \equiv n_i^k$ and $U \equiv U_i^k$. As a first step, the time derivatives of n are eliminated by means of the substitution:

$$n_t \Delta t = - (U n)_x \Delta x + \mathcal{E} \Delta t \quad (13)$$

where \mathcal{E} is a (1st order) spurious term introduced by the numerical scheme. Henceforth \mathcal{E} will be referred to as the ‘modified equation error’ (MEE), in order to distinguish it

from the ‘local truncation error’ (LTE), which was defined in Section 3 as the error *in the solution* introduced by the scheme after a single time-step. When the solution is ‘sufficiently smooth’, the LTE can be obtained from the integral of \mathcal{E} over the time step, so that $\text{LTE} \approx \mathcal{E} \Delta t$.

An expression for n_{tt} can be obtained by differentiating (13) with respect to time, multiplying by Δt , and substituting the *r.h.s.* of (13) wherever the n_t term appears:

$$n_{tt} \Delta t^2 = U (Un)_{xx} \Delta x^2 + U_x (Un)_x \Delta x^2 - (U_t n)_x \Delta x \Delta t - (U \mathcal{E})_x \Delta x \Delta t + \mathcal{E}_t \Delta t^2 \quad (14)$$

where the last two terms on the *r.h.s.* are third order since they contain the MEE itself. The same procedure can be repeated to get an expression for the third time derivative:

$$\begin{aligned} n_{ttt} \Delta t^3 = & -U^2 (Un)_{xxx} \Delta x^3 + \left(2U_t \frac{\Delta t}{\Delta x} - 3UU_x \right) (Un)_{xx} \Delta x^3 \\ & + \left(2U_{xt} \frac{\Delta t}{\Delta x} - U_x^2 - UU_{xx} \right) (Un)_x \Delta x^3 \\ & + U (U_t n)_{xx} \Delta x^2 \Delta t + U_x (U_t n)_x \Delta x^2 \Delta t - (U_{tt} n)_x \Delta x \Delta t^2 + H.O.T. \end{aligned} \quad (15)$$

where the *H.O.T.* come from the terms containing the MEE, which are now fourth order. Substituting (13), (14) and (15) into (12), one gets:

$$\begin{aligned} n_t + (un)_x = & \frac{\Delta x^2}{2 \Delta t} \left[(1 - U) (Un)_{xx} - U_x (Un)_x + (U_t n)_x \frac{\Delta t}{\Delta x} \right] \\ & + \frac{\Delta x^3}{6 \Delta t} \left[(U^2 - 1) (Un)_{xxx} + \left(3UU_x - 2U_t \frac{\Delta t}{\Delta x} \right) (Un)_{xx} \right. \\ & \quad \left. + \left(U_x^2 + 2U_{xx} - 2U_{xt} \frac{\Delta t}{\Delta x} \right) (Un)_x - U (U_t n)_{xx} \frac{\Delta t}{\Delta x} \right. \\ & \quad \left. - U_x (U_t n)_x \frac{\Delta t}{\Delta x} + (U_{tt} n)_x \left(\frac{\Delta t}{\Delta x} \right)^2 \right] \\ & + \frac{1}{2} \left[(U \mathcal{E})_x \Delta x - \mathcal{E}_t \Delta t \right] + O(\Delta x^3) \end{aligned} \quad (16)$$

where the *r.h.s.* is the truncation error. The fact that the expression for the MEE is formally implicit does not mean the problem is insoluble. In fact, since the MEE is first order, the last term in brackets on the *r.h.s.* is second order. This implies that the leading-term (first order) of the MEE is already known at this stage, and it is just the first term on the *r.h.s.* of (16):

$$\mathcal{E}_1 = \frac{\Delta x^2}{2 \Delta t} \left[(1 - U) (Un)_{xx} - U_x (Un)_x + (U_t n)_x \frac{\Delta t}{\Delta x} \right]. \quad (17)$$

The second order term of the MEE can be obtained by substituting $\mathcal{E} \approx \mathcal{E}_1$ into (16). Accordingly, the space and time derivatives of the MEE are approximated as

$$\mathcal{E}_x \approx \frac{\Delta x^2}{2 \Delta t} \left[(1 - U) (Un)_{xxx} - 2U_x (Un)_{xx} - U_{xx} (Un)_x + (U_t n)_{xx} \frac{\Delta t}{\Delta x} \right] \quad (18)$$

and

$$\begin{aligned} \mathcal{E}_t \approx \frac{\Delta x^3}{2\Delta t^2} & \left[(U^2 - U) (Un)_{xxx} + \left((3U - 2)U_x - 2U_t \frac{\Delta t}{\Delta x} \right) (Un)_{xx} \right. \\ & + \left((U - 1)U_{xx} + U_x^2 - 2U_{xt} \frac{\Delta t}{\Delta x} \right) (Un)_x \\ & \left. + (1 - U) (U_t n)_{xx} \frac{\Delta t}{\Delta x} - U_x (U_t n)_x \frac{\Delta t}{\Delta x} + (U_{tt} n)_x \left(\frac{\Delta t}{\Delta x} \right)^2 \right] \end{aligned} \quad (19)$$

where the approximation $n_t \Delta t \approx -(Un)_x \Delta x$ was used instead of the full (13). The modified equation of the 1D CCSL-CS, subject to the hypothesis $u > 0$, is obtained by substituting (17) for \mathcal{E} , (18) for \mathcal{E}_x , and (19) for \mathcal{E}_t into (16),

$$\begin{aligned} n_t + (un)_x = \frac{\Delta x^2}{2\Delta t} & \left[(1 - U) (Un)_{xx} - U_x (Un)_x + (U_t n)_x \frac{\Delta t}{\Delta x} \right] \\ & + \frac{\Delta x^3}{6\Delta t} \left[(3U - 2U^2 - 1) (Un)_{xxx} + \left(\frac{9}{2}U_x - 6UU_x + U_t \frac{\Delta t}{\Delta x} \right) (Un)_{xx} \right. \\ & + \left(\frac{3}{2}U_{xx} - 2UU_x - 2U_x^2 + U_{xt} \frac{\Delta t}{\Delta x} \right) (Un)_x + \left(2U - \frac{3}{2} \right) (U_t n)_{xx} \frac{\Delta t}{\Delta x} \\ & \left. + 2U_x (U_t n)_x \frac{\Delta t}{\Delta x} - \frac{1}{2} (U_{tt} n)_x \left(\frac{\Delta t}{\Delta x} \right)^2 \right] + O(\Delta x^3) , \end{aligned} \quad (20)$$

which shows that, for a given velocity field $u(x, t)$, the PDE that is really solved by the scheme (11) approximates the continuity equation $n_t + (un)_x = 0$ only to first order. In fact, (20) contains the spurious terms $\mathcal{E}_1 = O(\Delta x)$ and $\mathcal{E}_2 = O(\Delta x^2)$, which are the first two terms of the ‘modified equation error’ (MEE).

5. High-order version of the CCSL-CS

In the following treatment a novel approach is used in order to devise a third order version of the CCSL-CS. From a Lagrangian view-point, the basic idea is to *apply a correction to the position of the Moving-Cell before it is remapped*. Since a correction to the position is $\delta x = \delta u \Delta t$, one can think of a correction to the velocity as well.

It is important to consider that the velocity field $u(x, t)$ need not be the physical velocity field, which we call $u_0(x, t)$ from now on. In fact, we can assume that

$$u = u_0 + u_1 + u_2 \quad \text{or} \quad U = U_0 + U_1 + U_2 \quad (21)$$

where $U_1 \propto \Delta x$ and $U_2 \propto \Delta x^2$ will be evaluated by requiring (20) be a third order approximation to the ‘physical’ continuity equation:

$$n_t + (u_0 n)_x = O(\Delta x^3) . \quad (22)$$

In other words, without changing the remapping rule on which the CS is based, a third order scheme can be obtained by applying first and second order corrections (u_1 and u_2) to the physical velocity field u_0 . Such a method is direct (i.e. non iterative), and guarantees the compensation of both \mathcal{E}_1 and \mathcal{E}_2 for sufficiently smooth density profiles. After some algebra, the substitution of (21) into (20) gives (22) only if

$$U_1 = \frac{\Delta x}{2} \left[(1 - U_0) \frac{1}{n} (U_0 n)_x + U_{0t} \frac{\Delta t}{\Delta x} \right], \quad (23)$$

and

$$U_2 = \frac{\Delta x^2}{12} \left[(1 - 3U_0 + 2U_0^2) \frac{1}{n} (U_0 n)_{xx} + \left(2U_0 U_{0x} - 3U_{0x} - 4U_{0t} \frac{\Delta t}{\Delta x} \right) \frac{1}{n} (U_0 n)_x + (6 - 5U_0) \frac{1}{n} (U_{0t} n)_x \frac{\Delta t}{\Delta x} + 2U_{0tt} \left(\frac{\Delta t}{\Delta x} \right)^2 \right]. \quad (24)$$

The expressions (23) and (24) formally contain the *exact* space and time derivatives of U_0 and n , but for any practical situation, such derivatives will be evaluated numerically. For (22) to hold, both U_1 and U_2 must be estimated with third order accuracy; accordingly, the finite difference schemes must be second order accurate in (23), and first order in (24). Examples of the numerical evaluation of the derivatives are given in Section 8.

With the proper approximations for U_1 and U_2 , the scheme that results from using the velocity corrections (21) in the 1D CCSL-CS (11) is third order accurate, conservative, and very simple to implement. Most important, even if the high-order scheme does not ensure monotonicity preservation, it does preserve the positivity of the solution at all times, thanks to the fact that the remapping phase (inherently sign-preserving) is left unchanged. If the initial conditions are ‘sufficiently smooth’ (or, equivalently, the gradients are ‘well resolved’), the high-order scheme just formulated performs very well, as it also preserves the monotonicity in the solution. When sharp gradients are present instead, spurious oscillations may arise in the solution: in such a situation, a ‘non-oscillatory’ version of the present scheme is necessary, which will be discussed in Section 7.

When the velocity u_0 is constant and uniform, formulas (23) and (24) are further simplified: in such a situation, Cartesian 2D and 3D versions of the algorithm are obtained straightforwardly by applying independent corrections to each component of the velocity vector, resulting in a very compact and numerically efficient method. In fact, all these properties make this approach particularly suitable for application to neutral gas kinetic simulations.

6. 3D analysis

One might wonder whether the above procedure could be used to obtain a third order scheme for the advection equation in three dimensions, for a generic flow field. In fact, that would open the possibility to apply the CCSL-CS to the multi-dimensional solution of generic hyperbolic conservation equations (for example, the Euler equations

that govern inviscid flow dynamics). Such an application is beyond the scope of this work, and it is opinion of the authors that a complete third order analysis in 3D can be carried out only with a considerable amount of algebra. Nevertheless, for completeness, a brief outline of the procedure to obtain a second-order scheme in 3D is given hereafter. The equation to be solved is now the 3D continuity equation

$$\partial_t n + \partial_x (u n) + \partial_y (v n) + \partial_z (w n) = 0,$$

where u , v , and w are respectively the x , y , and z components of the velocity vector. For the sake of clarity, it will be assumed that u , v , and w are all > 0 , and that

$$U = \frac{u\Delta t}{\Delta x} < 1, \quad V = \frac{v\Delta t}{\Delta y} < 1, \quad W = \frac{w\Delta t}{\Delta z} < 1.$$

Under the hypothesis above, if a Cartesian grid with uniform spacing over each direction is considered, the 3D CCSL-CS can be written as a finite-difference scheme,

$$\begin{aligned} n_{i,j,k}^{m+1} &= n_{i,j,k}^m (1 - U_{i,j,k}^m) (1 - V_{i,j,k}^m) (1 - W_{i,j,k}^m) \\ &+ n_{i-1,j,k}^m U_{i-1,j,k}^m (1 - V_{i-1,j,k}^m) (1 - W_{i-1,j,k}^m) \\ &+ n_{i,j-1,k}^m (1 - U_{i,j-1,k}^m) V_{i,j-1,k}^m (1 - W_{i,j-1,k}^m) \\ &+ n_{i,j,k-1}^m (1 - U_{i,j,k-1}^m) (1 - V_{i,j,k-1}^m) W_{i,j,k-1}^m \\ &+ n_{i-1,j-1,k}^m U_{i-1,j-1,k}^m V_{i-1,j-1,k}^m (1 - W_{i-1,j-1,k}^m) \\ &+ n_{i,j-1,k-1}^m (1 - U_{i,j-1,k-1}^m) V_{i,j-1,k-1}^m W_{i,j-1,k-1}^m \\ &+ n_{i-1,j,k-1}^m U_{i-1,j,k-1}^m (1 - V_{i-1,j,k-1}^m) W_{i-1,j,k-1}^m \\ &+ n_{i-1,j-1,k-1}^m U_{i-1,j-1,k-1}^m V_{i-1,j-1,k-1}^m W_{i-1,j-1,k-1}^m, \end{aligned}$$

where i , j , and k are the indices over the three spatial directions, and m is the index over time. A 2^{nd} order Taylor series expansion about the point (x_i, y_j, z_k, t^m) gives:

$$\begin{aligned} n_t \Delta t + (nU)_x \Delta x + (nV)_y \Delta y + (nW)_z \Delta z &= \\ - n_{tt} \frac{\Delta t^2}{2} + O(\Delta t^3) + (nU)_{xx} \frac{\Delta x^2}{2} + (nV)_{yy} \frac{\Delta y^2}{2} + (nW)_{zz} \frac{\Delta z^2}{2} & \quad (25) \\ + (nUV)_{xy} \Delta x \Delta y + (nVW)_{yz} \Delta y \Delta z + (nWU)_{zx} \Delta z \Delta x + O(\Delta s^3), \end{aligned}$$

where $\Delta s = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$.

As usual, we can get rid of n_{tt} by differentiating the equality

$$n_t \Delta t = - (nU)_x \Delta x - (nV)_y \Delta y - (nW)_z \Delta z + \mathcal{E} \Delta t \quad (26)$$

with respect to time. Seeking a second order scheme, the term containing the modified equation error \mathcal{E} can be neglected, since it contributes to higher order terms only. After some algebraic manipulations, this gives:

$$\begin{aligned} n_{tt} \Delta t^2 &\simeq \left[U (nU)_x \Delta x + nVU_y \Delta y + nWU_z \Delta z - nU_t \Delta t \right]_x \Delta x \\ &+ \left[V (nV)_y \Delta y + nWV_z \Delta z + nUV_x \Delta x - nV_t \Delta t \right]_y \Delta y \\ &+ \left[W (nW)_z \Delta z + nUV_x \Delta x + nVW_y \Delta y - nW_t \Delta t \right]_z \Delta z \\ &+ 2 (nUV)_{xy} \Delta x \Delta y + 2 (nVW)_{yz} \Delta y \Delta z + 2 (nWU)_{zx} \Delta z \Delta x . \end{aligned} \quad (27)$$

The substitution of (27) into (25) leads to the modified equation (to 2^{nd} order) for the 3D CSSL-CS:

$$\begin{aligned}
n_t + (un)_x + (vn)_y + (wn)_z &= \frac{\Delta x}{2\Delta t} [(1-U)(nU)_x \Delta x + nVU_y \Delta y + nWU_z \Delta z + nU_t \Delta t]_x \\
&+ \frac{\Delta y}{2\Delta t} [(1-V)(nV)_y \Delta y + nWV_z \Delta z + nUV_x \Delta x + nV_t \Delta t]_y \\
&+ \frac{\Delta z}{2\Delta t} [(1-W)(nW)_z \Delta z + nUW_x \Delta x + nVW_y \Delta y + nW_t \Delta t]_z \\
&+ H.O.T.
\end{aligned} \tag{28}$$

At this point we can assume that u , v and w are obtained by a first order perturbation to the physical velocity:

$$\begin{cases} u = u_0 + u_1 \\ v = v_0 + v_1 \\ w = w_0 + w_1 \end{cases} \quad \text{or} \quad \begin{cases} U = U_0 + U_1 \\ V = V_0 + V_1 \\ W = W_0 + W_1 \end{cases}$$

so that, after substitution into (28), the corrections are obtained by imposing the compensation of the first order errors. For the general case with no restriction on the sign of the velocity components, one gets:

$$U_1 = \left[(|U_0| - U_0^2) \frac{(nU_0)_x}{nU_0} \right] \frac{\Delta x}{2} + [V_0 U_{0y}] \frac{\Delta y}{2} + [W_0 U_{0z}] \frac{\Delta z}{2} + [U_{0t}] \frac{\Delta t}{2} \tag{29a}$$

$$V_1 = \left[(|V_0| - V_0^2) \frac{(nV_0)_y}{nV_0} \right] \frac{\Delta y}{2} + [W_0 V_{0z}] \frac{\Delta z}{2} + [U_0 V_{0x}] \frac{\Delta x}{2} + [V_{0t}] \frac{\Delta t}{2} \tag{29b}$$

$$W_1 = \left[(|W_0| - W_0^2) \frac{(nW_0)_z}{nW_0} \right] \frac{\Delta z}{2} + [U_0 W_{0x}] \frac{\Delta x}{2} + [V_0 W_{0y}] \frac{\Delta y}{2} + [W_{0t}] \frac{\Delta t}{2} \tag{29c}$$

and the correction to be applied to the final position of the MC is simply

$$\delta \mathbf{r} = \begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \begin{pmatrix} U_1 \Delta x \\ V_1 \Delta y \\ W_1 \Delta z \end{pmatrix}.$$

When the velocity field and the density are smooth functions of space and time, the corrections above lead to a simple and efficient 2^{nd} order version of the 3D CSSL-CS, for a generic velocity field. As usual, the scheme is conservative and positivity preserving.

7. Monotonic variations

For a constant velocity field u_0 , the first order version of the 1D CSSL-CS guarantees the solution $n(x, t)$ will conserve the monotonicity of the initial conditions. Unfortunately, this is not true for the third order version. In fact, wherever the advected function $n(x)$ happens not to be differentiable in space, the finite difference approximations in (23) and (24) become meaningless and the estimates for U_1 and U_2 are incorrect. At best, the accuracy of the scheme drops to first order; more likely, U_1 and U_2 are overestimated and

$O(1)$. As a consequence, Gibbs' oscillations would appear, with an amplitude that does not decrease on refining the mesh. In such a situation the scheme would probably retain a first-order accuracy in the L_2 norm, but *it cannot converge to the correct solution* in the L_∞ norm.

Limiters

In order to make the third order CCSL-CS monotonic, i.e. non-oscillating, it should be noticed that the spurious oscillations are due to the corrections U_1 and U_2 . Hence, a natural approach is to reduce the magnitude of these corrections with a *limiter function* wherever the conditions are favorable for spurious oscillations to arise. Basically, in order to preserve monotonicity, the limiter function is supposed to reduce the scheme to first order in the proximity of a discontinuity in n_x . In practice, it happens that U_1 and U_2 are reduced also in regions where no such limiting is needed: a typical long-time effect is the artificial flattening of smooth local maxima and minima. Several limiter functions were obtained by adapting the flux limiters commonly used in high order Total Variation Diminishing (TVD) schemes [16; 17; 18; 19]. Here, the general implementation is based on the slight modification of (21) into

$$U = U_0 + \phi(r) [U_1 + U_2]$$

where $\phi(r)$ is the limiter function and r is the ratio of successive gradients on the solution mesh, with a slightly modified definition in order to assure that $-1 \leq r \leq 1$:

$$r = \min \left[\frac{n_i - n_{i-1}}{n_{i+1} - n_i}, \frac{n_{i+1} - n_i}{n_i - n_{i-1}} \right].$$

In the tests performed so far, the best results were obtained using the Superbee limiter designed by Roe [20] in 1986

$$\phi(r) = \max [0, \min (2r, 1)],$$

where the original simple formula is further simplified here because $r \leq 1$.

Non-polynomial reconstruction

Taking a slightly different point of view, the finite difference approximations in (23) and (24) are evaluated from a local quadratic polynomial interpolation of the discrete function $\{n_i\}$ between three equally spaced grid points: once the interpolating polynomial is obtained, it is analytically differentiated in order to give an approximation for the first and second derivatives of $n(x)$ at the center point. In general, it is well-known that polynomial interpolations are not suited to reconstruct 'non-smooth' functions on a uniform grid, and hence it is no surprise that spurious oscillations are triggered whenever there is a discontinuity in the solution.

Following naturally from these considerations is the idea of using non-polynomial functions to reconstruct the density function inside a cell. In [21] a third order non-oscillating reconstruction technique is outlined, which uses only three grid points (resulting in an ideally compact stencil), and that does not need limiters. Such a method is based on a *Local Double Logarithmic Reconstruction* (LDLR) of the form

$$r_0(x) \simeq A + B \log(x + C) + D \log(x + E),$$

within cell C_0 . As described in more detail later on, the five parameters in last equation are obtained by imposing conservation, formal third order accuracy, symmetry, and Local Variation Boundedness (LVB). The complete form of the reconstructing function is

$$r_0(x) = \bar{n}_0 + \phi_0(x) - \frac{1}{\Delta x} \int_{C_0} \phi_0(\xi) d\xi \quad (30)$$

where \bar{n}_0 is the average value of $n(x)$ within cell C_0 , and $\frac{1}{\Delta x} \int_{C_0} \phi_0(\xi) d\xi$ represents the mean value of the function $\phi_0(x)$, which is defined as:

$$\phi_0(x) = -\frac{c \Delta x}{a} \log \left[(x - x_0) - \frac{\Delta x}{2} \left(\frac{2}{a} - 1 \right) \right] - \frac{d \Delta x}{b} \log \left[(x - x_0) - \frac{\Delta x}{2} \left(\frac{2}{b} - 1 \right) \right]. \quad (31)$$

For any choice of the coefficients a , b , c and d , the reconstruction (30) is conservative, since

$$\frac{1}{\Delta x} \int_{C_0} r_0(x) dx \equiv \bar{n}_0. \quad (32)$$

The requirement of third order accuracy requires the reconstruction to match the second order finite difference approximations to the derivative $u'(x)$ at the cell faces:

$$\begin{aligned} r'_0 \left(x_0 - \frac{\Delta x}{2} \right) &= \frac{\bar{u}_0 - \bar{u}_{-1}}{\Delta x} = \delta_1, \\ r'_0 \left(x_0 + \frac{\Delta x}{2} \right) &= \frac{\bar{u}_1 - \bar{u}_0}{\Delta x} = \delta_2, \end{aligned}$$

which permits one to solve for the parameters c and d as functions of a , b , δ_1 and δ_2 :

$$c = \frac{(a-1)[\delta_2(1-b) - \delta_1]}{b-a}, \quad (33)$$

$$d = \delta_1 - c. \quad (34)$$

The reconstruction should be symmetric, in the sense that if $r'_0(x_0 - \Delta x/2) = -r'_0(x_0 + \Delta x/2)$, then $r'_0(x_0)$ should be zero. This permits one to find b as a function of a ,

$$b = \frac{a}{a-1}, \quad (35)$$

and hence all the 3 parameters b , c and d that show up in (31) are univocally determined once that a is known. To construct a reconstruction procedure of sufficiently low variation Marquina's concept of *Local Variation Boundedness* is used:

Definition (Local Variation Bounded [22]). The local variation of a function $f(x)$ in a cell C_i is given by $LV(f_i) = TV(f)|_{C_i}$. The function is Local Variation Bounded (LVB) in C_i if $LV(f_i) = O(\Delta x)$, where Δx is the cell size.

An expression for $a(\delta_1, \delta_2)$ which ensures that the reconstruction (31) be a well defined smooth LVB function is [21]

$$a(\delta_1, \delta_2) = (1 - \text{TOL}) \left(1 + \text{TOL} - \frac{2|\delta_1|^q |\delta_2|^q + \text{TOL}}{|\delta_1|^{2q} + |\delta_2|^{2q} + \text{TOL}} \right), \quad (36)$$

where $\text{TOL} = 0.1\Delta x^q$ and $q = 1.4$. Once that a is determined from (36), b , c and d are obtained from (35), (33) and (34), respectively. For the purposes of the third-order CSSL-CS, the first and second derivatives of the reconstruction (30) are needed, which only need to be evaluated for $x = x_0$, giving the following compact formulas:

$$\begin{aligned} r'_0(x_0) &= \frac{2c}{2-a} + \frac{2d}{2-b}, \\ r''_0(x_0) &= -\frac{1}{\Delta x} \left[\frac{4ac}{(2-a)^2} + \frac{4bd}{(2-b)^2} \right]. \end{aligned}$$

The most demanding part of the algorithm is the evaluation of the coefficient a by means of (36), since it requires the exponentiation of $|\delta_1|$ and $|\delta_2|$ by the real power q , operation which is usually much slower than taking the natural logarithm or exponential. The number of exponentiations can be reduced to one if the *smoothness parameter* θ is used,

$$\theta_0 = \frac{\Delta x \min(|\delta_1|, |\delta_2|) + \text{TOL}}{\Delta x \max(|\delta_1|, |\delta_2|) + \text{TOL}}$$

where TOL is in this case a small number that depends on the machine precision ϵ ; robustness and accuracy can be achieved by choosing $\text{TOL} = \epsilon^{3/4}$. The expression for a can be reformulated as a function of θ_0 alone,

$$a(\theta_0) = (1 - \text{TOL}) \left(1 + \text{TOL} - \frac{2(\theta_0)^q}{1 + (\theta_0)^{2q}} \right),$$

so that only one exponentiation to a real power is required. In the numerical tests performed, a considerable speedup was obtained by creating a look-up table for $a(\theta_0)$ in preprocessing.

8. Numerical tests

The order-of-convergence estimates that were carried out in the preceding sections were based exclusively on what we called the ‘modified equation error’ (MEE), which is the difference between the model equation and the modified equation of the scheme. Accordingly, we labeled a scheme as ‘first order’ or ‘third order’ depending on the form of the MEE. This can be misleading, because we are interested in approximating to high order not just the model equation, but the solution itself. In fact, the Modified Equation that is solved by the numerical scheme is a *singular perturbation* to the model equation, and as such it may lead to ‘spurious’ features that were not present before. Whether or not such features are triggered depends mainly on the initial density profile and on the boundary conditions.

Following from the above considerations, the primary purpose of this section is to assess the Global Error (GE) of the scheme, which is the actual difference between the *exact solution* and the *approximated solution*. The GE is quantified a-posteriori in the L_∞ , L_2 or L_1 norm, depending on what is more appropriate, and convergence analyses are performed by successively decreasing the mesh size. A limited number of test-cases are considered, both in 1D and 2D.

In order to show the capabilities of the scheme in a ‘real-world’ application, the final part of this section is devoted to a 3D-3V kinetic simulation: the expansion of a rarefied neutral gas injected through a cylinder into vacuum will be shown, for different Knudsen regimes.

1D advection equation

As the simplest test case, the third order CCSL-CS is applied to the solution of the 1D continuity equation (10) in the hypothesis of constant and uniform velocity. The solution $n(x, t) > 0$ is sought over the domain $(x, t) \in [0, L] \times [0, T]$ with periodic boundary conditions in space $n(0, t) \equiv n(L, t)$, and initial conditions $n(x, 0) = n_0(x)$. The independent variables are normalized by introducing $\xi = x/L$ and $\tau = t/T_p$, where $T_p = L/u$ is the characteristic flow time. Under these assumptions, we will numerically solve

$$n_\tau + n_\xi = 0 \quad (38)$$

on the domain $(\xi, \tau) \in [0, 1] \times [0, 1]$ (i.e., $T = T_p$ is chosen). The analytic solution to (38) is $n(\xi, \tau) = n_0(\xi - \tau)$; due to the periodic boundary conditions, the final solution coincides with the initial conditions: $n(\xi, 1) = n_0(\xi)$. The application of the third order CCSL-CS to (38) comes from the usual CS remapping rule, which is used after moving the cell a distance $U\Delta\xi$, where $U = U_0 + U_1 + U_2$. We have $U_0 = u\Delta t/\Delta x = \Delta\tau/\Delta\xi$, and it will be imposed here that $|U_0| < 1$. From (23) and (24), the first and second order corrections in the case of uniform and constant (but not necessarily positive) U_0 are:

$$U_1 = \frac{\Delta\xi}{2} (|U_0| - U_0^2) \frac{n_\xi}{n}$$

$$U_2 = \frac{\Delta\xi^2}{12} (U_0 - 3|U_0|U_0 + 2U_0^3) \frac{n_{\xi\xi}}{n}$$

which will be approximated using the finite difference expressions

$$\frac{1}{n} \frac{\partial n}{\partial \xi} \Big|_{\xi=\xi_i} = \frac{\partial (\log n)}{\partial \xi} \Big|_{\xi=\xi_i} = \frac{1}{\Delta\xi} \log \left(\frac{n_{i+1}}{n_{i-1}} \right) + O(\Delta\xi^2) \quad (39a)$$

$$\frac{1}{n} \frac{\partial^2 n}{\partial \xi^2} \Big|_{\xi=\xi_i} = \frac{4}{\Delta\xi^2} \frac{n_{i+1} - 2n_i + n_{i-1}}{n_{i+1} + 2n_i + n_{i-1}} + O(\Delta\xi^2) \quad (39b)$$

but different choices can be made, as long as a second order approximation is used for (39a) and a first order approximation is used for (39b).

As a first test case, a Gaussian profile centered at $\xi = 0.5$ is given as initial condition,

$$n(\xi, 0) = 0.1 + \exp \left[- \left(\frac{\xi - 0.5}{0.1} \right)^2 \right]$$

and then evolved in time for $\tau \in [0, 1]$. A Courant parameter $U_0 = 0.3$ is chosen. The final solution obtained with the third order CCSL-CS is compared qualitatively with the original first order version in Fig. 4, for different numbers N of subdivisions in space ($\Delta\xi = 1/N$). The L_2 and L_∞ norms of the error for the third order scheme are also computed by doubling the number of subdivisions up to 3200 cells. The results are

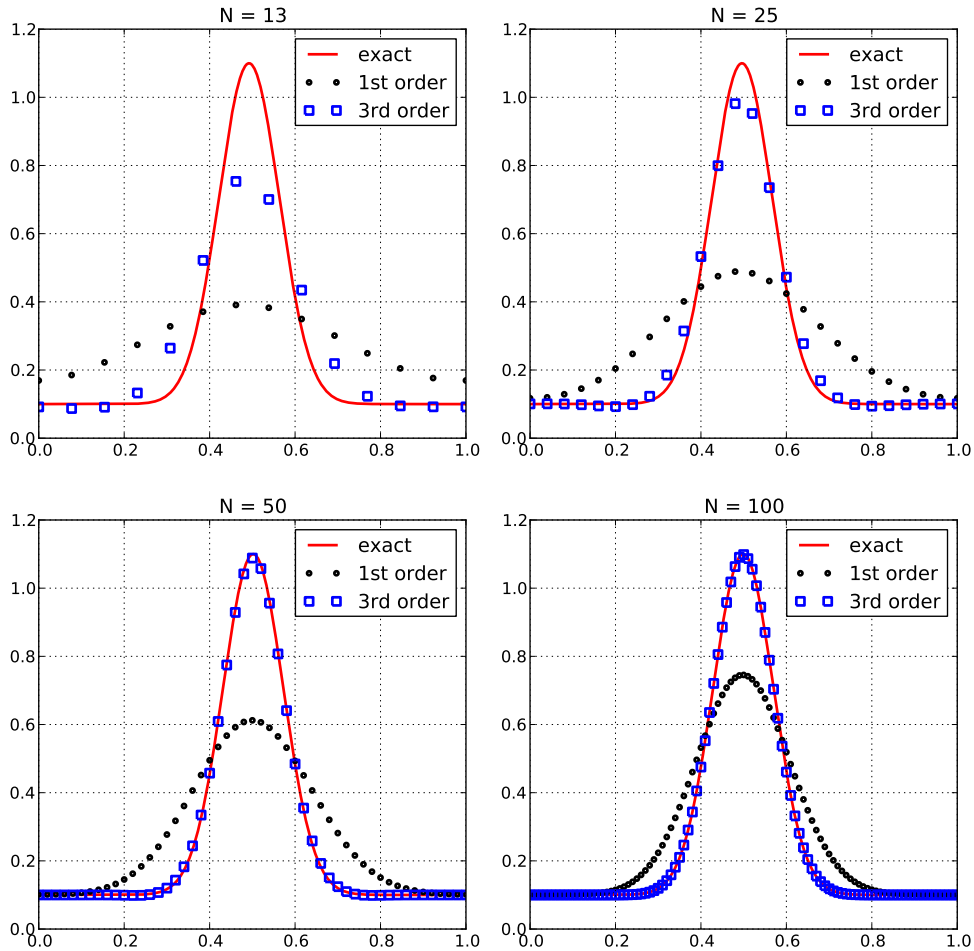


Figure 4: 1D advection with Gaussian initial profile: final solution at time $\tau = 1.0$ for increasing number N of spatial subdivisions.

summarized by Fig. 5, which confirms that the error of the scheme decreases as $(\Delta\xi)^3$ in both the norms; it is most noticeable that the asymptotic convergence behavior of the scheme is reached very soon, starting from $N \approx 30$.

As a second test case, a rectangular profile is given as initial condition:

$$n(\xi, 0) = \begin{cases} 1.1 & \text{if } 0.25 < \xi < 0.75, \\ 0.1 & \text{otherwise.} \end{cases}$$

As before, $\tau \in [0, 1]$ and $U_0 = 0.3$, and a series of identical simulations is performed using an increasingly fine mesh. Since the initial density profile is not ‘smooth’, the 3rd order CCSL-CS undergoes non-physical oscillations in the regions close to the discontinuities.

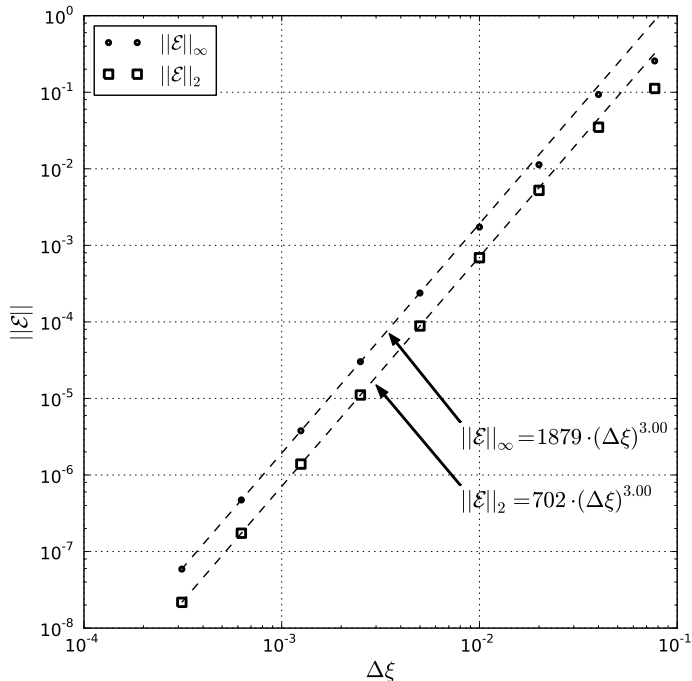


Figure 5: Convergence analysis for 1D advection with Gaussian initial profile. The L_∞ and L_2 norms of the error at the end of the simulation ($\tau = 1.0$) are shown for decreasing mesh-size $\Delta \xi$.

This effect does not compromise the positive-definiteness of the scheme, which is inherited from the CS remapping rule, but it considerably deteriorates the solution. Hence, the non-oscillatory option described in Section 7 is implemented here, and Fig. 6 gives a qualitative picture of the improvement upon the standard 1st order scheme. As is common practice in the case of discontinuous solutions, a convergence analysis based on the L_1 norm of the error is presented in Fig. 7: as expected, both schemes exhibit sub-linear convergence [23].

The GE of the (nominally) 1st order scheme goes as $(\Delta \xi)^{0.5}$, a result obtained a-priori in [24] (pg. 121). The GE of the (nominally) 3rd order scheme is proportional to $(\Delta \xi)^{0.8}$, which is slightly better than the expected $(\Delta \xi)^{0.75}$ for the non-limited case: in fact, in [23] the order of convergence for unlimited stable schemes is established as $(\Delta \xi)^{p/(p+1)}$ where p is the order of the approximation for smooth flows.

One may want to investigate the behavior of the monotonic version of the high-order CCSL-CS when ‘smooth’ initial conditions are given, in order to assess how the performance degrades with respect to the non-monotonic version. For this purpose, the first test case (with a Gaussian profile as initial conditions) is repeated using the Local Double Logarithmic Reconstruction described in Section 7. The results are shown in Fig. 8, for an increasing number of subdivisions in space: as typical of monotonicity-preserving

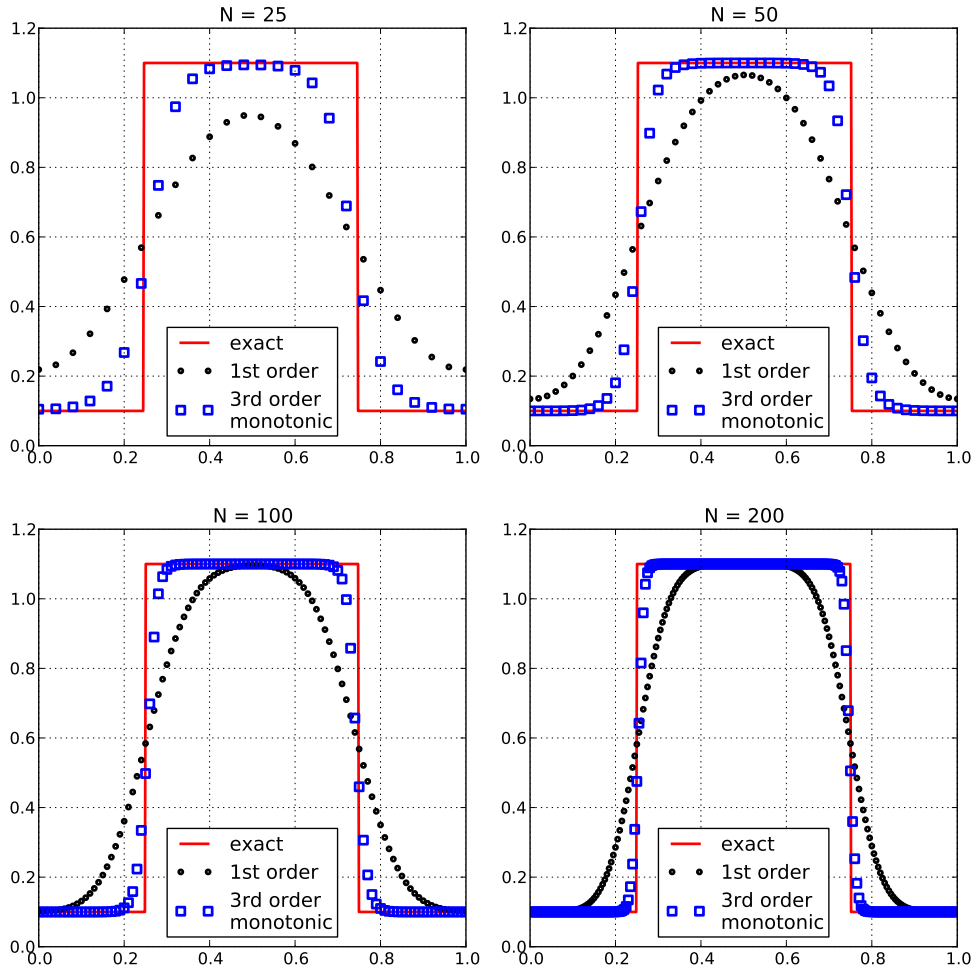


Figure 6: 1D advection with rectangular initial profile: final solution at time $\tau = 1.0$ for increasing number N of spatial subdivisions.

schemes, the peak of the Gaussian profile is flattened by the numerical diffusion that is introduced in the non-monotonic regions of the density profile. Accordingly, the accuracy of the scheme is significantly affected: instead of the cubic convergence shown in Fig. 5, the monotonic version of the high-order CCSL-CS achieves quadratic convergence in the L_2 norm, and only sub-quadratic convergence in the L_∞ norm (see Fig. 9).

Given the higher accuracy of the non-monotonic scheme, and considered the fact that it is inherently positivity-preserving, the monotonic version is only preferred in those situations where spurious oscillations in the solution are expected.

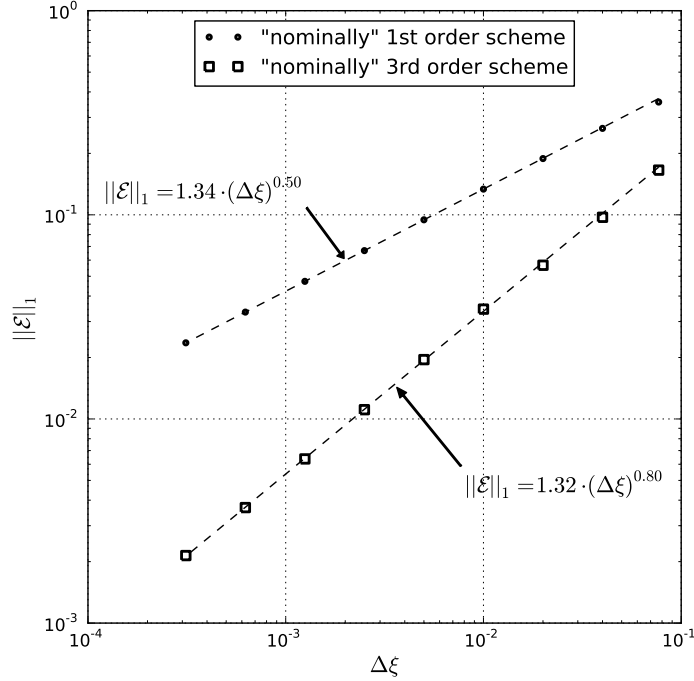


Figure 7: Convergence analysis for 1D advection with rectangular initial profile. The L_1 norm of the error at the end of the simulation ($\tau = 1.0$) is shown for decreasing mesh-size $\Delta\xi$.

2D continuity equation with rotating velocity field

As a test case with non-uniform velocity, the continuity equation is solved on a square domain with a given rotating velocity field. The equation to be solved is:

$$\partial_t n + \partial_x (u n) + \partial_y (v n) = 0$$

on the square domain $(x, y) \in [0, L] \times [0, L]$ and in the time interval $t \in [0, T]$. The two components of the velocity vector are

$$\begin{aligned} u &= -\omega (y - L/2) \\ v &= +\omega (x - L/2) \end{aligned} ,$$

where $\omega = 2\pi/T$ [rad/s] is the angular frequency of the rotating field. After normalization of the spatial and temporal variables with respect to L and T , one gets:

$$\partial_\tau n + \partial_\xi (\bar{u} n) + \partial_\eta (\bar{v} n) = 0 \quad (40)$$

where $(\xi, \eta) \in [0, 1] \times [0, 1]$, $\tau \in [0, 1]$, and the normalized velocities were introduced:

$$\bar{u} = uT/L = -2\pi (\eta - 0.5) \quad (41a)$$

$$\bar{v} = vT/L = +2\pi (\xi - 0.5) \quad (41b)$$

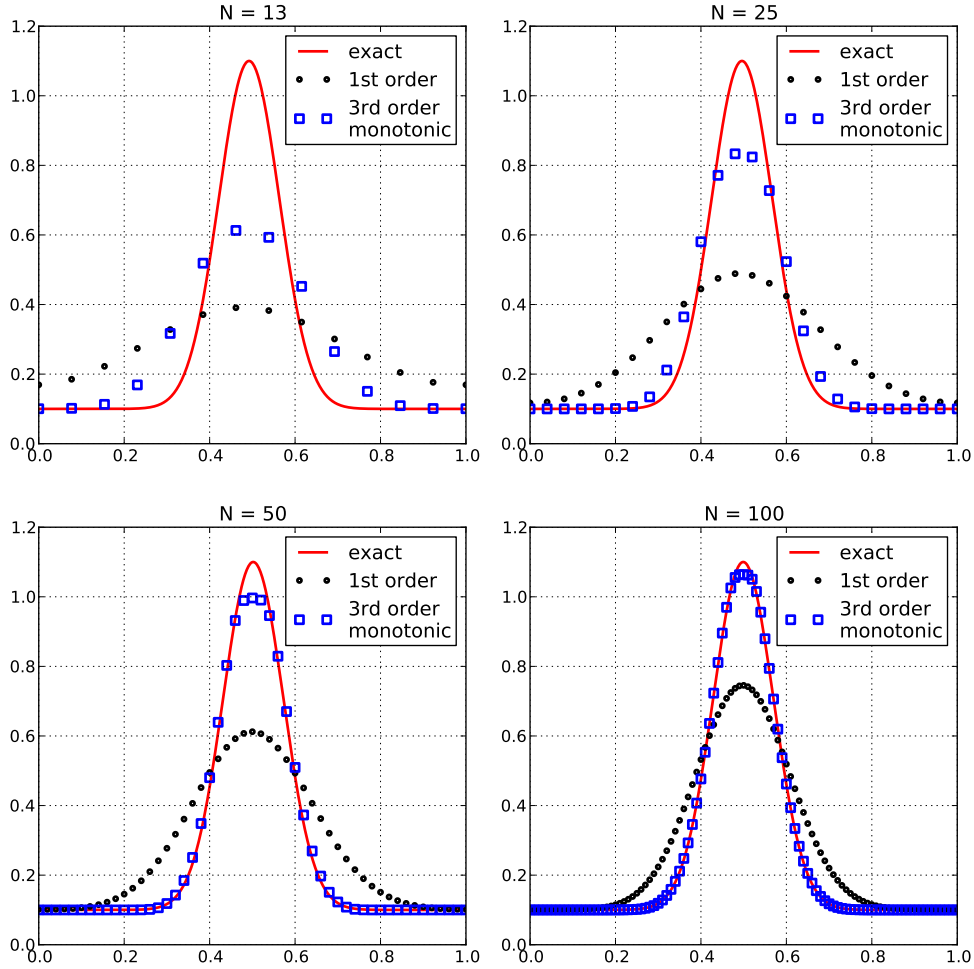


Figure 8: 1D advection with Gaussian initial profile: final solution at time $\tau = 1.0$ for increasing number N of spatial subdivisions. A non-oscillating version of the high-order CCSL-CS is employed, which uses a Local Double Logarithmic Reconstruction (LDLR) for evaluating the spatial derivatives of the density.

The application of the second order CCSL-CS to (40) comes from the usual CS remapping rule, which is used after moving the cell a distance $\Delta \mathbf{r} = [U\Delta\xi, V\Delta\eta]$, where $U = U_0 + U_1$ and $V = V_0 + V_1$. With the new variables we have $U_0 = u \Delta t / \Delta x = \bar{u} \Delta\tau / \Delta\xi$ and $V_0 = v \Delta t / \Delta y = \bar{v} \Delta\tau / \Delta\eta$. The same number of subdivisions is employed along both the x and y axes, so that $\Delta\eta = \Delta\xi$. The corrections U_1 and V_1 are evaluated using second

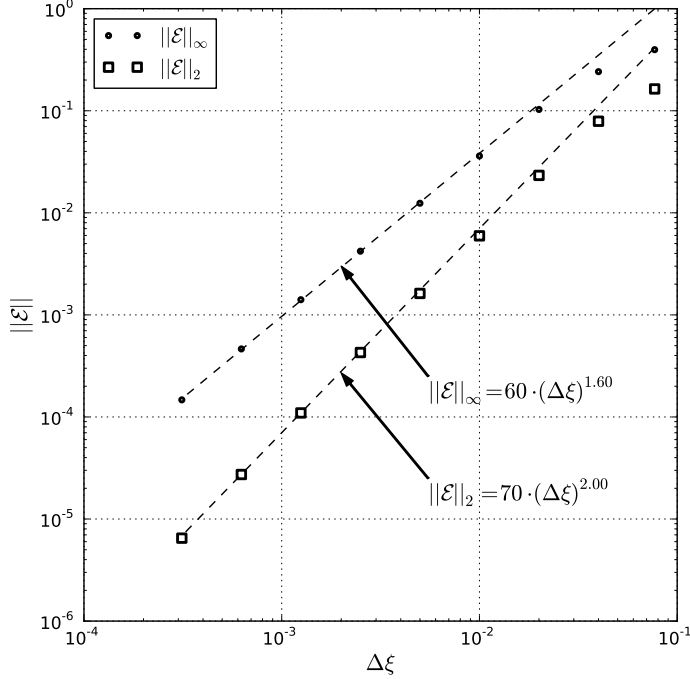


Figure 9: Convergence analysis for 1D advection with Gaussian initial profile, for the non-oscillating version of the high-order CCSL-CS, which uses a Local Double Logarithmic Reconstruction (LDLR) of the derivatives. The L_∞ and L_2 norms of the error at the end of the simulation ($\tau = 1.0$) are shown for decreasing mesh-size $\Delta\xi$.

order finite difference approximations to (29):

$$(U_1)_{i,j} = \frac{1}{2} (|U_0| - U_0^2)_{i,j} \frac{(n U_0)_{i+1,j} - (n U_0)_{i-1,j}}{(n U_0)_{i+1,j} + (n U_0)_{i-1,j}} + \frac{1}{4} (V_0)_{i,j} [(U_0)_{i,j+1} - (U_0)_{i,j-1}]$$

$$(V_1)_{i,j} = \frac{1}{2} (|V_0| - V_0^2)_{i,j} \frac{(n V_0)_{i,j+1} - (n V_0)_{i,j-1}}{(n V_0)_{i,j+1} + (n V_0)_{i,j-1}} + \frac{1}{4} (U_0)_{i,j} [(V_0)_{i+1,j} - (V_0)_{i-1,j}]$$

The maximum Courant parameter in the domain is reached at the boundaries and it is fixed at 1, i.e. $|U_0|_{\max} = |\bar{u}_{\max} \Delta\tau / \Delta\xi| = 1$, and $|V_0|_{\max} = |\bar{v}_{\max} \Delta\tau / \Delta\eta| = 1$. According to (41), the time step is also fixed:

$$\Delta\tau = \frac{\Delta\xi}{\pi} .$$

The system is evolved for a time $\tau \in [0, 0.25]$, which means that the bell performs one quarter ($\pi/2$) of a full rotation (2π), during the whole simulation. A cosine bell profile

is given as initial condition:

$$n(\xi, \eta, 0) = \begin{cases} 0.1 + \frac{1}{2} \left[1.0 + \cos\left(\frac{\pi r}{R}\right) \right] & \text{if } r < R \\ 0.1 & \text{otherwise} \end{cases}$$

where $r = \sqrt{(\xi - \xi_c)^2 + (\eta - \eta_c)^2}$ is the distance from the center (ξ_c, η_c) of the bell, and R is the ‘radius’ of the bell. In the following simulations, the center is chosen at the point $(\xi_c, \eta_c) = (0.5, 0.75)$, and a radius $R = 0.2$ is used.

The final solution and the final error are shown in Figs. 10 and 11, using respectively 50 and 100 subdivisions along each axis; the use of the 2nd order scheme considerably improves the solution over the original 1st order method. To complete the analysis, Fig. 12 shows the asymptotic behavior of the error of the two schemes for decreasing mesh-size: as already noticed in the 1D test-cases, the high-order CCSL-CS quickly reaches its theoretical order of convergence.

3D-3V Boltzmann’s equation for neutrals

As a ‘real world’ application, a 3D kinetic simulation of neutral gas expansion is presented. Kinetic in this context means we are describing the system at a level where we obtain the distribution of the particles with respect to independent variables including both space and velocity: we solve for the scalar field $f(t, \mathbf{x}, \mathbf{v})$ [$m^{-6}s^3$], which is the particle number density in phase space. The kinetic description is necessary because the hypothesis of local thermal equilibrium is not applicable, and hence the form of $f(t, \mathbf{x}, \mathbf{v})$ for any point in space cannot be obtained from local integral properties of the fluid, like density, mean velocity and temperature. Local thermal equilibrium is not reached because the mean free path is comparable with the system dimensions; or, as another way to put it, the collision time is comparable with the time needed to traverse the system.

Using the kinetic formalism, the time evolution of a rarefied single species neutral gas is described by a simplified form of Boltzmann’s equation,

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} = \left[\frac{\partial f}{\partial t} \right]_{\text{coll}}, \quad (42)$$

where all force fields are neglected: the domain is small enough to permit neglect of the gravitational force, and the frame of reference is considered to be inertial. Two-body short-range interactions are taken into account by means of the *Boltzmann collision integral* on the right hand side of the equation. As described in Section 2, the integral form of (42) is solved using a time-splitting procedure, i.e. by successively applying a *ballistic operator* and a *collision operator*. Such a mesh-based procedure has important advantages over particle approaches like the Direct Simulation Monte-Carlo [25], e.g. the absence of statistical noise, a more uniform resolution in velocity space, and reduced memory requirements for a given accuracy level. As a downside, the application of the Convective Scheme to neutral simulations poses a number of challenges:

1. reducing the numerical diffusion in space introduced by the ballistic operator,
2. approximating the Boltzmann collision integral with an efficient collision operator,
3. treating generic boundary conditions in a semi-Lagrangian framework,

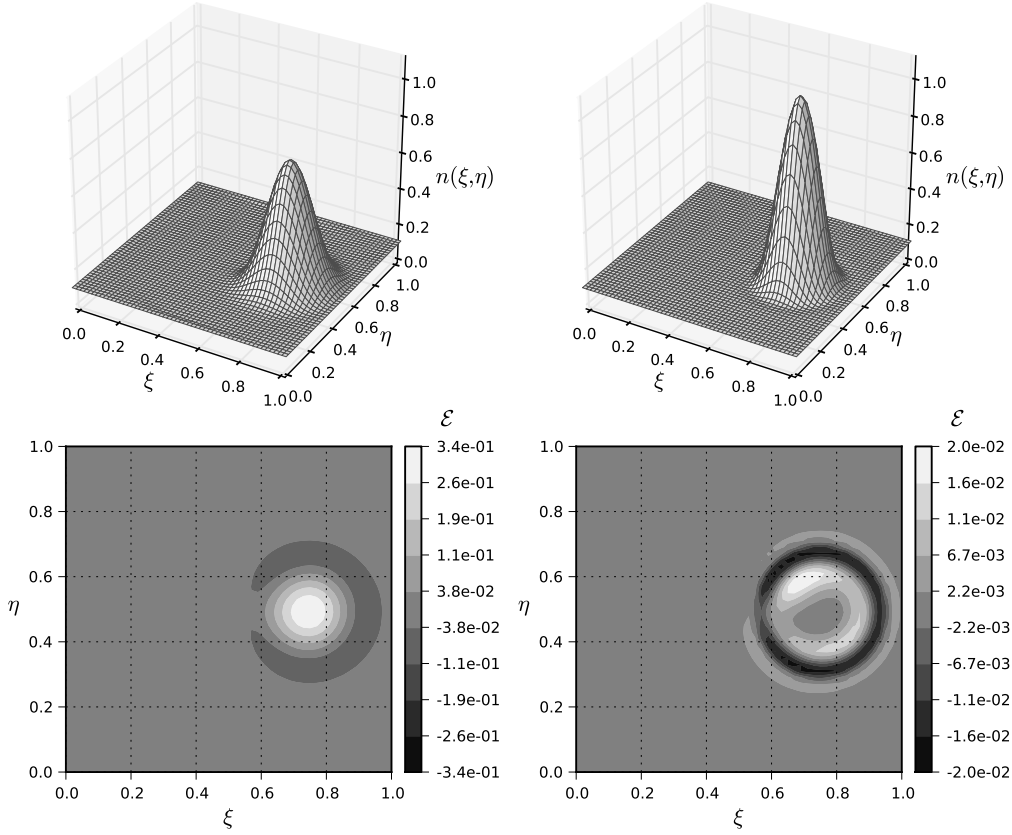


Figure 10: Results for 2D advection with rotating velocity field, for a number of subdivisions $N_x = N_y = 50$. A cosine bell is given as the initial density profile. On the top row: final solution evaluated by the 1st order scheme (left) and by the 2nd order scheme (right). On the bottom row: error for the 1st (left) and the 2nd order scheme (right).

4. reducing spurious coupling effects between spatial and velocity meshes (e.g. the ‘ray effect’ in the presence of sharp gradients).

Brief outlines of methods for addressing points 2 and 3, which are strictly related to the velocity mesh set-up, are given in the appendix. Point 4 has not been thoroughly investigated by the authors yet, and it is a major topic for future research (but see also [26] where particles were allowed to travel within rays of finite angular extent). Instead, the reduction of numerical diffusion (point 1) is the principal motivation of this work: a 3D version of the third-order CCSL-CS is used as the ballistic operator of the scheme, and the results are compared to the standard first-order version.

The test case that is shown hereafter is part of the hybrid simulation of a Helicon plasma thruster. Argon gas at ambient temperature is injected through a small injector into a slender quartz tube, which is surrounded by a helical RF antenna and magnets, which create an axial static magnetic field. On the other end, the tube is open to vacuum. Since it is not uncommon for Helicon plasma thrusters to obtain ionization

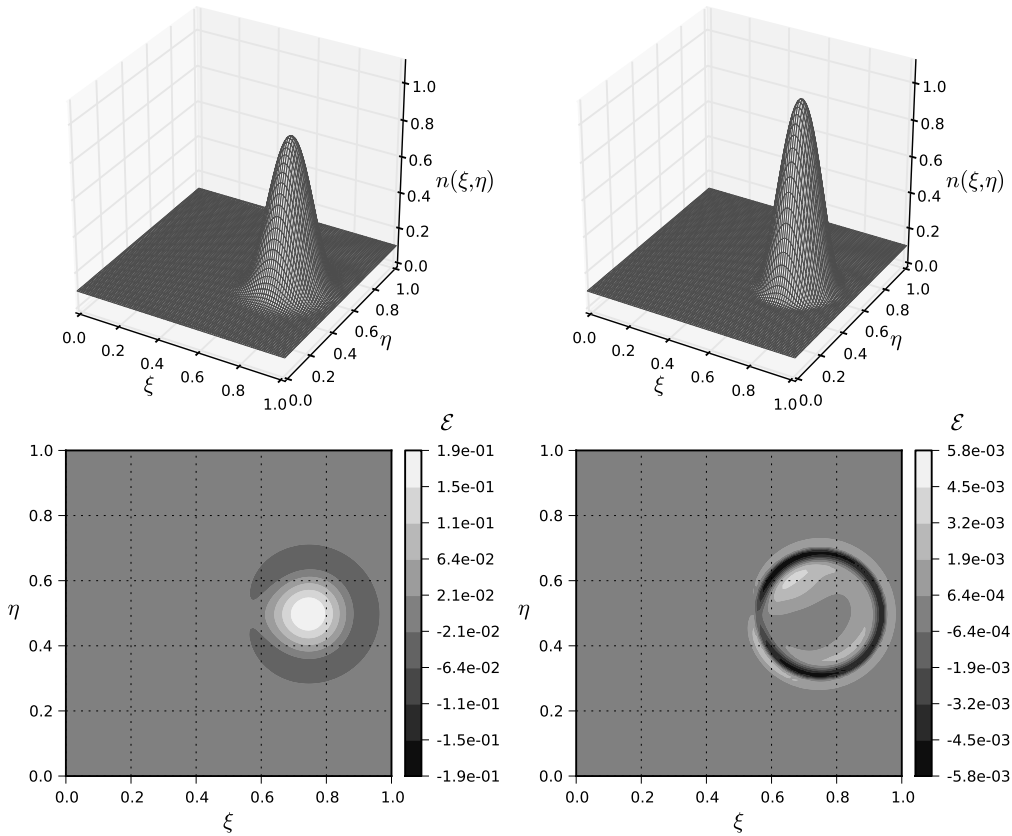


Figure 11: Results for 2D advection with rotating velocity field, for a number of subdivisions $N_x = N_y = 100$. A cosine bell is given as the initial density profile. On the top row: final solution evaluated by the 1st order scheme (left) and by the 2nd order scheme (right). On the bottom row: error for the 1st (left) and the 2nd order scheme (right).

efficiencies higher than 90%, the density of neutral gas decreases steeply along the axis of the source: the regime is collisional (low Knudsen number) in proximity to the injector, and it is practically non-collisional (high Kn) toward the exit section.

In the following test-cases no plasma is present, and only Ar-Ar hard-sphere elastic collisions are considered. Accordingly, there is no negative source term due to ionization, and hence the gas density along the axis of the cylinder decreases less steeply than it does in the nominal operational regime of the thruster. In order to test the behavior of the scheme for high Knudsen numbers ($\text{Kn} > 1$), an additional simulation with reduced injected flow-rate is run.

The simulation domain is a half cylinder (cut on the meridian plane) of diameter $d = 20$ mm and length $L = 80$ mm, open on one end to vacuum (perfectly absorbing boundary), and closed on the other end by a plane wall with a small orifice (the injector) at the center. All material walls are assumed adiabatic. The spatial mesh is a uniform Cartesian grid with the z coordinate aligned along the symmetry axis of the cylinder.

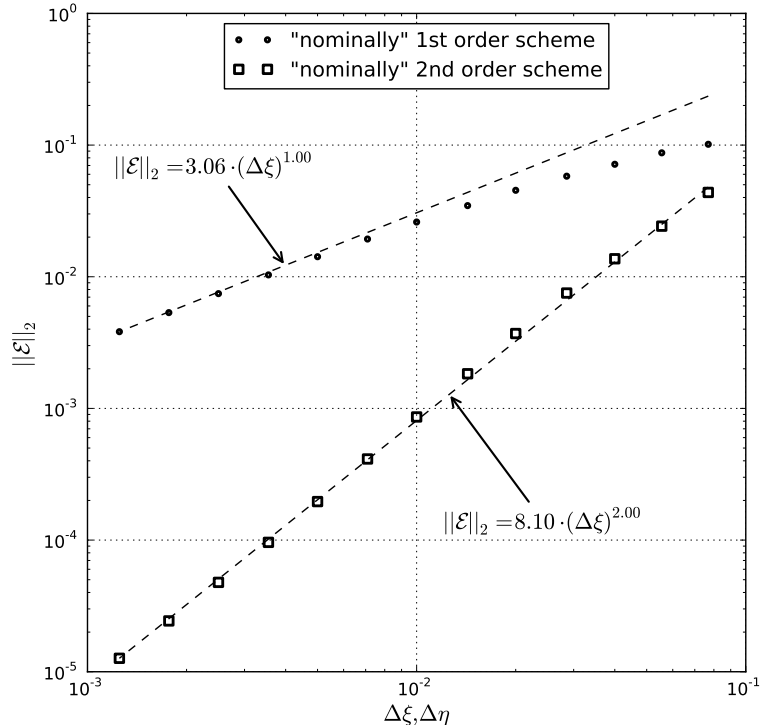


Figure 12: Convergence analysis for 2D advection with rotating velocity field. A cosine bell is given as the initial density profile. The L_2 norm of the error at the end of the simulation ($\tau = 0.25$) is shown for decreasing mesh-size $\Delta\xi \equiv \Delta\eta$.

The cell size is $\Delta x = \Delta y = 0.833$ mm and $\Delta z = 1.000$ mm; a total of 17920 spatial cells is used. The curvature of the external wall of the cylinder is approximated with a ‘staircase’ approach. The velocity mesh is obtained by the independent discretization of the magnitude and the direction of the velocity vector: the speed axis $\|\mathbf{v}\|$ extends from 0 to 1200 m/s, with 15 subdivisions; the unit vector $\mathbf{v}/\|\mathbf{v}\|$ can assume 642 different directions, which are obtained by a hierarchical subdivision of the unit sphere (see Appendix A for more details). Accordingly, the whole velocity space is discretized into 9630 cells. The phase-space mesh is obtained by the Cartesian product of the spatial mesh and the velocity mesh, for a total of approximately 173 million phase-space cells. Since two double-precision values of the distribution function are stored for each phase-space cell, the simulation requires about 2.6 GB of RAM.

Different kinds of boundary conditions are employed in the model:

1. at the material walls, *adiabatic diffuse reflection* is implemented: the particles that collide with the walls are reflected back isotropically into the domain, but their kinetic energy is unchanged. In order to deal efficaciously with the staircase wall, an integral formulation is employed (see Appendix C);
2. at the meridian plane, *periodic boundary conditions* are implemented: particles

that travel through such a plane undergo a 180° rigid-body rotation about the z axis. This method permits us to simulate only half of the cylinder, exploiting the axial symmetry of the geometry. Unfortunately, the velocity mesh employed does not have a 90° symmetry, which would have allowed us to simulate only one quarter of the cylinder;

3. at the open end of the cylinder, *perfect absorption* is implemented: all particles that cross the exit plane are simply absorbed. This approximation is believed to give quite reasonable results, based on geometrical considerations: because of the high Knudsen number, it is quite improbable that a particle leaving from the exit section will undergo a series of collisions that will direct it back into the domain;
4. at the injector, the flow-rate and the gas properties are imposed, based on the hypothesis of critical (sonic) flow. This integral formulation is equivalent to imposing *Dirichlet boundary conditions* on the distribution function at the inlet.

The distribution function at the inlet section is assumed to be a drifting Maxwellian. As such, its free parameters are the density n , the mean velocity \mathbf{u} and the temperature T , which are uniquely determined from the total temperature $T_0 = 300$ K, flow-rate $\dot{n} = 3 \cdot 10^{18}$ part/s, section area $A \simeq 2.78$ mm², and Mach number $Ma = 1$.

The collision operator is based on a simplified Bhatnagar-Gross-Krook (BGK) model [27], and its efficient and conservative implementation is described briefly in Appendix B.

Figs. 13-16 show the steady-state results for the simulations that use the nominal inlet flow-rate $\dot{n} = 3 \cdot 10^{18}$ part/s: due to the low values of the Knudsen number realized in the domain (ranging from ≈ 0.02 in front of the injector to ≈ 2 at the exit section), such a situation is labeled as a ‘low-Kn’ test case. Figs. 17-20 show the steady-state results for the simulations that use a reduced inlet flow-rate $\dot{n} = 1 \cdot 10^{18}$ part/s: due to the reduced density in the domain, the Knudsen number ranges from ≈ 0.1 to ≈ 7 (‘high-Kn’ test case). Two different simulations are run for each test case: one uses the standard first-order ballistic operator, and the other uses the new third-order algorithm. The results are compared in each figure and, as expected, the third order algorithm gives the sharpest results.

Most noticeable are the differences in the region close to the injector, where the high density gradients enhance the effects of numerical diffusion: in both regimes most of the density drop between the injector and the exit section is concentrated just in front of the injector, and this effect is better captured by the third order algorithm (see Figs. 13 and 17). As a result of the higher average density obtained in both regimes, the first-order scheme appears to overestimate the temperature by 10-20 K over the whole domain (Figs. 14 and 18). At the same time, the first-order scheme appreciably underestimates the peak value of the Mach number in front of the injector (Figs. 15 and 19). Moreover, in the low-Kn regime, an annular recirculation region appears just next to the injector, and such a phenomenon is much better resolved by the third order scheme (see Figs. 16).

In the high-Kn test case the recirculation region vanishes as expected (Fig. 20), and the low numerical diffusion of the high order scheme makes it possible to observe the spurious ‘ray-effect’ due to the combination of highly divergent flow field, Dirac-delta velocity mesh, and low collisionality. Such a mesh-based artifact is typical of the so-called ‘discrete-velocity’ models [11] often used in rarefied gas-dynamics: since the velocity can only assume a discrete number of directions, particles which should propagate isotropi-

cally from a certain spatial cell can only propagate along the allowed ‘rays’. Generally this effect is mitigated by the contribution from the surrounding cells, because the particles coming from those cells tend to ‘fill-in’ the gaps between the previous rays; in fact, there is no ray-effect if the density is uniform. On the other hand, when the density gradient is large, and specifically when the mean velocity field is highly divergent, the contribution from the surrounding cells may be insufficient and the ray-effect becomes apparent.

When the low-order CCSL-CS is employed (top of Fig. 20), its high numerical diffusion makes the rays ‘smear out’ in space, so that the gaps between the rays are filled in and the solution looks deceptively smooth, even for a highly divergent mean velocity field. As explained in Section 3, such numerical diffusion is strongly anisotropic and it depends non-linearly on the speed and direction of the velocity vector; as a consequence, the numerical results depart from the exact solution in a quite unpredictable way.

When the high-order version of the CCSL-CS is employed instead (bottom of Fig. 20), with very little numerical diffusion in space, only collisions can mitigate the ray-effect: some of the particles change speed and direction by effect of collisions, and in the next time step they end up filling in the gaps between the rays. Since collisions act on a length scale equal to the mean free path λ , and rays are only visible when they are separated by more than Δx , a simple relation should be satisfied to avoid the ray-effect,

$$\Delta\theta < \frac{\Delta x}{\lambda}, \quad (43)$$

which holds for $\Delta x \ll \lambda$. The inequality (43) relates the maximum angular spacing $\Delta\theta$ between two adjacent discrete velocities to the amount of *physical diffusion in velocity space* due to particle collisions. In the high-Kn test-case presented, λ is very large and the ‘no-ray’ condition (43) is in fact not satisfied, as memory requirements made it prohibitive for such a run.

While it may be argued that the low-order scheme ‘performs better’ than the high-order one, because it did not show the unphysical ray-effect in the numerical tests presented, our concern is that the low-order scheme is hiding problems intrinsic in the simulation setup. Nevertheless, when just a qualitative description of the physical system is sought, a low-order solution affected by numerical diffusion is likely to be preferable over a high-order solution affected by the ray-effect. Our point here is that we have developed a simple high order scheme, the choice of whether to use it or not in a given instance is a separate matter.

9. Conclusions

A high order scheme for solution of ‘kinetic equations’ was derived, which is 4th order accurate in space. The scheme is based on the ‘Convected Scheme’ (CS), which is a method of characteristics applied to the evolution of particle density in phase space. The CS moves initial cells and maps them back onto the phase space mesh, in such a way as to build in local conservation laws (for density, energy etc.) in a form (discussed in detail elsewhere [1]) such that, for instance, the mean energy is conserved for each final spatial cell, for the given initial phase space cell. A Modified Equation Analysis applied to the CS was used to show that 4th order accuracy in space may be obtained by simply

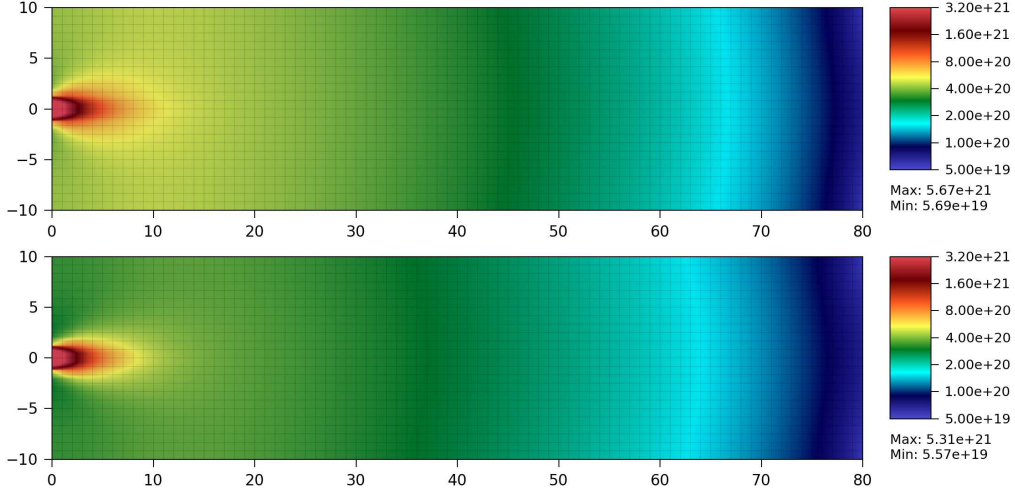


Figure 13: Steady state density distribution for the low-Kn test case: comparison between the 1^{st} order result (top), and the 3^{rd} order result (bottom). Axis scales are in [mm], density is in [m^{-3}]. Model grid underlies the plot.

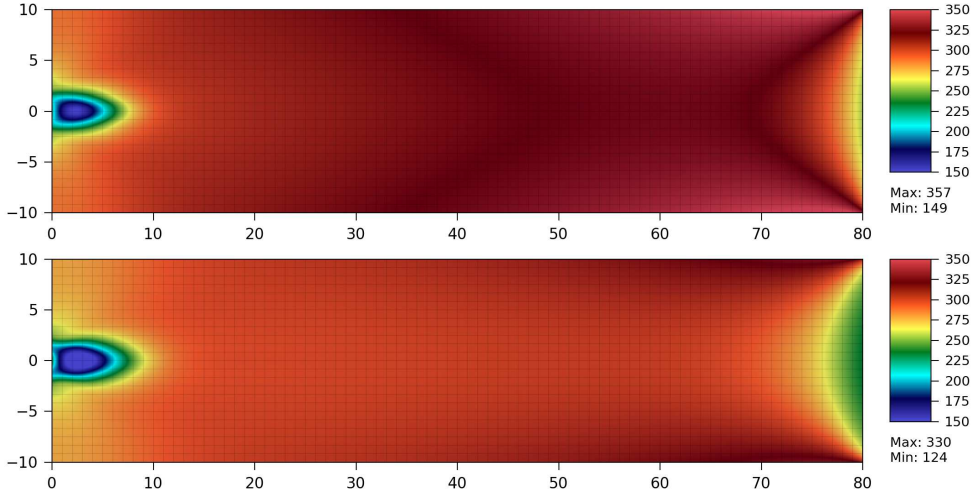


Figure 14: Steady state temperature distribution for the low-Kn test case: comparison between the 1^{st} order result (top), and the 3^{rd} order result (bottom). Axis scales are in [mm], temperature is in [K]. Model grid underlies the plot.

applying a small correction to the position of the cell which is to be mapped back to the mesh.

Since the remapping phase was left unchanged, the high-order scheme retains the most desirable properties of the basic CS: it is conservative, it preserves positivity, and it is easy to implement. In particular, positivity preservation was achieved naturally without

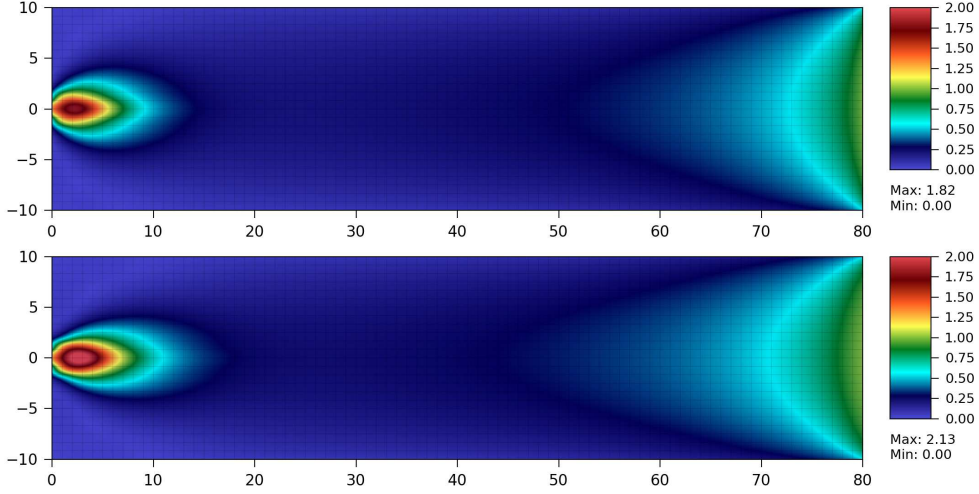


Figure 15: Steady state Mach number distribution for the low-Kn test case: comparison between the 1st order result (top), and the 3rd order result (bottom). Axis scales are in [mm], Ma is dimensionless. Model grid underlies the plot.

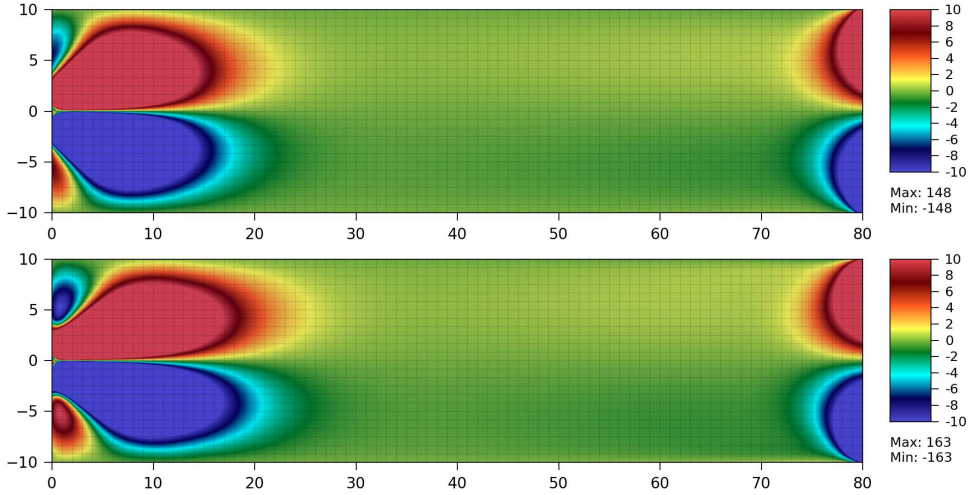


Figure 16: Steady state distribution of the y component of the velocity vector for the low-Kn test case: comparison between the 1st order result (top), and the 3rd order result (bottom). Axis scales are in [mm], velocity is in [m/s]. Model grid underlies the plot.

the need to strictly enforce monotonicity; this is an advantage over those schemes that use high-order reconstructions to achieve higher-order convergence. Nevertheless, there exist situations where the numerical solution shows unphysical oscillations (e.g. in the presence of shocks); two different ‘monotonicity-preserving’ versions of the CCSL-CS were derived for use in such applications.

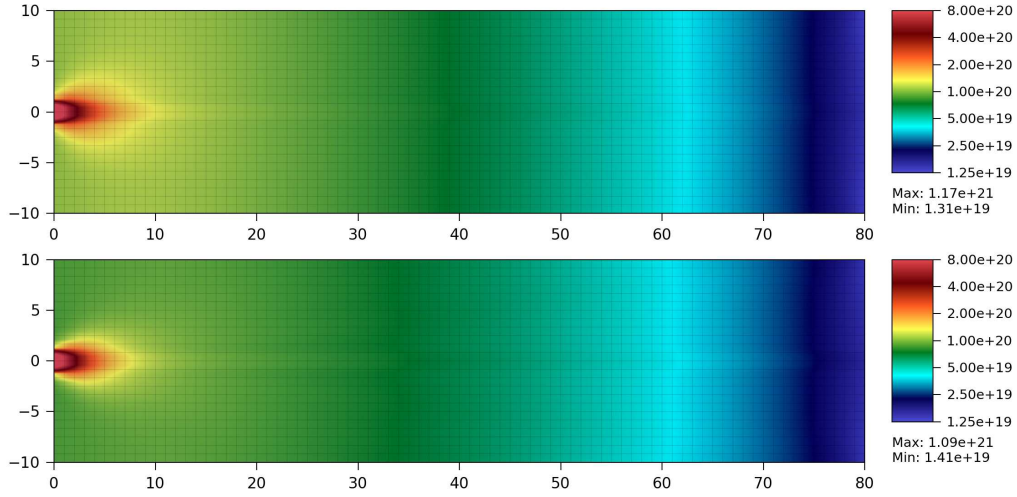


Figure 17: Steady state density distribution for the high-Kn test case: comparison between the 1st order result (top), and the 3rd order result (bottom). Axis scales are in [mm], density is in [m⁻³]. Model grid underlies the plot.

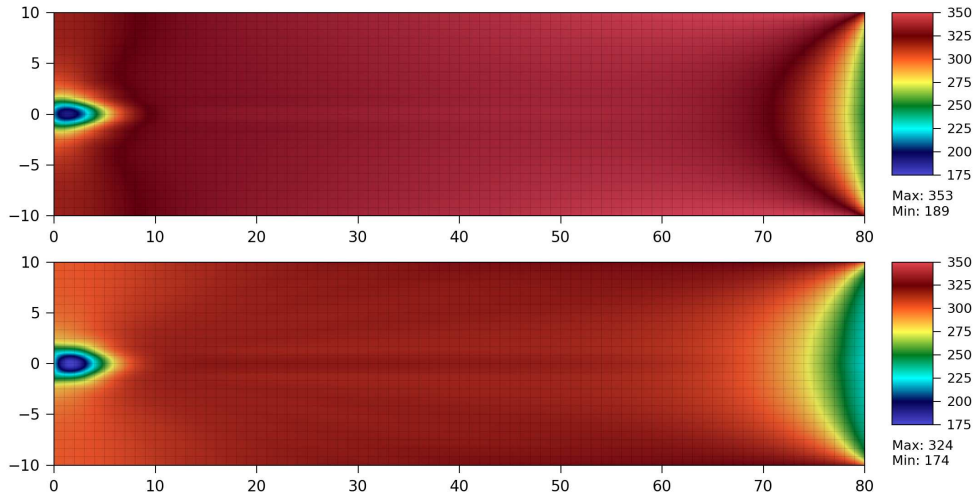


Figure 18: Steady state temperature distribution for the high-Kn test case: comparison between the 1st order result (top), and the 3rd order result (bottom). Axis scales are in [mm], temperature is in [K]. Model grid underlies the plot.

Numerical results were illustrated in application to 1, 2 and 3D examples. The order of the scheme appears to be confirmed by these examples. The 3D example consisted of gas flow from a nozzle through a cylindrical tube into vacuum, and this calculation was performed in Cartesian coordinates and the corresponding velocities. Results were shown at different Knudsen's numbers, illustrating apparent changes in the flow patterns

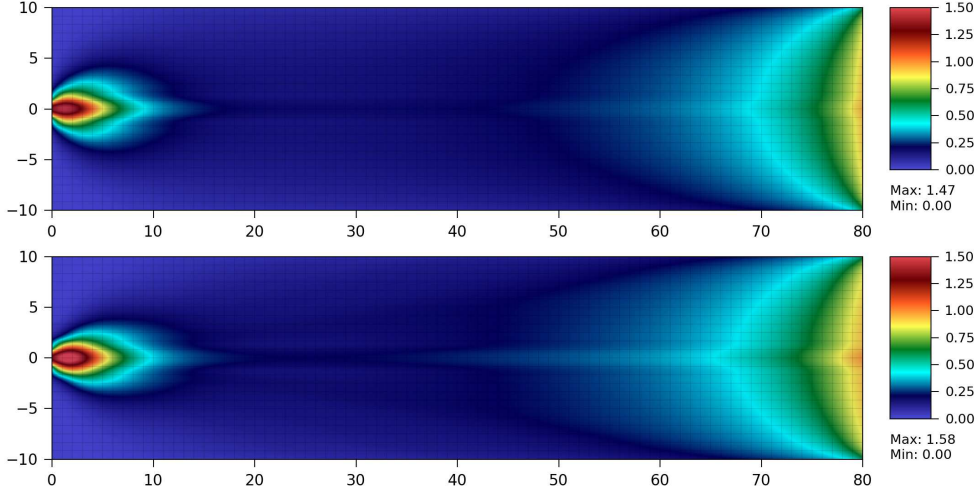


Figure 19: Steady state Mach number distribution for the high-Kn test case: comparison between the 1st order result (top), and the 3rd order result (bottom). Axis scales are in [mm], Ma is dimensionless. Model grid underlies the plot.

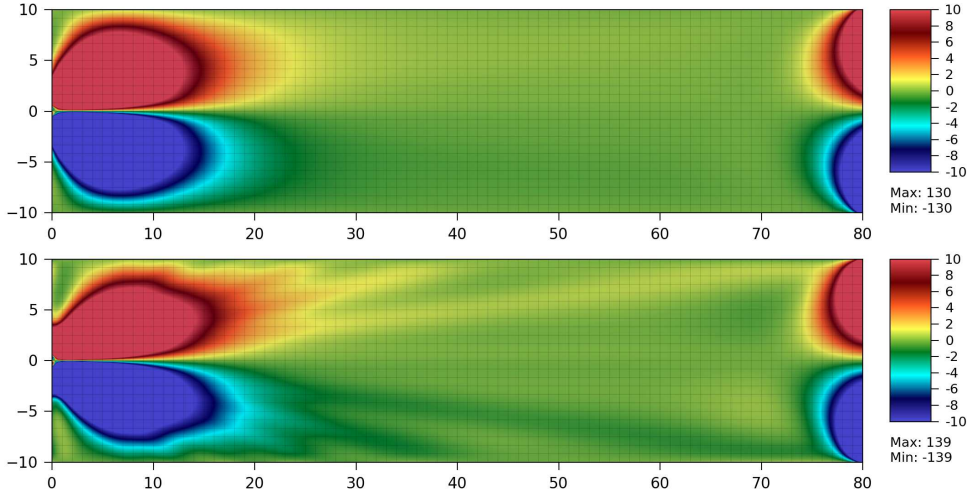


Figure 20: Steady state distribution of the y component of the velocity vector for the high-Kn test case: comparison between the 1st order result (top), and the 3rd order result (bottom). Axis scales are in [mm], velocity is in [m/s]. Model grid underlies the plot.

as Kn was varied across the ‘transitional regime’.

Since the multi-dimensional kinetic simulation of a neutral gas flow is the main application of the present scheme, one of the important conclusions to this work is that we should think carefully about the ray-effect and ways to mitigate it, other than by numerical diffusion. This will be the subject of future work.

10. Acknowledgments

One of us (YG) is grateful to the University of Padua's Doctoral School in Sciences, Technologies, and Measures for Space for support of foreign studies.

Appendix A. Velocity Mesh

The velocity grid is a double-indexed list of discrete velocity vectors, $\mathbf{v}_{ij} = \hat{\mathbf{n}}_i v_j$, where i is the index labeling directions, and j is the index labeling speeds. The unit vectors $\hat{\mathbf{n}}_i$ must be chosen carefully in order to maximize the uniformity of their angular spacing; in order to accomplish this, the surface of a unit sphere is first subdivided into 20 identical triangles (spherical icosahedron), then each triangle is refined into 4 smaller triangles by connecting the midpoints of their sides by great circles. This refinement is repeated recursively, until the required angular resolution is obtained: each vertex on the unit sphere identifies a unit vector $\hat{\mathbf{n}}_i$, with a solid angle $\Delta\Omega_i$ associated with it. The calculation of $\Delta\Omega_i$ requires the creation of the dual mesh of the given triangular mesh: such a dual mesh is composed of hexagons, plus 20 pentagons, and the areas of those spherical polygons numerically coincide with the solid angles required. The speed axis v is discretized in concentric spherical shells of uniform thickness Δv_j , and we let v_j be the center radius of the shell. The volume of the cell (i, j) in velocity space [m^3s^{-3}] can be evaluated as

$$V_{ij} = \Delta\Omega_i \Delta v_j \left(v_j^2 + \frac{\Delta v_j^2}{12} \right) .$$

Appendix B. Collision Operator

In an ‘operator split’ environment, the Collision Operator evolves the distribution function for one time-step Δt according to (4a). The effect of the Boltzmann Collision Integral on the R.H.S. is approximated using an integral formulation of the Bhatnagar-Gross-Krook (BGK) model [27], which is implemented in two steps, both local in space:

1. the number of particles that scatter out of each phase-space cell during the time-step Δt , due to collisions against the background gas, is:

$$\Delta N_{\text{out}} = N_0 [1 - \exp(-\nu \Delta t)] ,$$

where N_0 is the number of particles inside the phase-space cell at the beginning of the time-step, and $\nu(\mathbf{v}_c)$ is the total collision frequency;

2. all the particles that have undergone a collision in the same spatial cell are put into a drift-Maxwellian distribution $\mathcal{M}(\mathbf{v})$, which gives the number of particles that scatter into each phase-space cell:

$$\Delta N_{\text{in}} = \mathcal{M}(\mathbf{v}_c) \cdot V_c , \tag{B.1}$$

where V_c is the volume of the cell in phase-space [m^6s^{-3}].

Both the aforementioned steps give rise to numerical problems. As a first issue, the formula for $\nu(\mathbf{v}_c)$ is a three-dimensional integral over velocity-space:

$$\nu(\mathbf{v}_c) = \int_{\mathbb{R}^3} \|\mathbf{v}_c - \mathbf{v}\| \sigma(\|\mathbf{v}_c - \mathbf{v}\|) f(\mathbf{v}) d^3\mathbf{v}, \quad (\text{B.2})$$

where $\sigma(\|\mathbf{v}_c - \mathbf{v}\|)$ is the cross-section [m^2] for elastic collisions between like particles. In the context of the BGK approximation, (B.2) is a waste of computational effort. In order to approximate the integral on the RHS, an average value $\langle\sigma\rangle$ is used (justified because the cross section for elastic collisions of non-excited Ar does not change appreciably in the range of velocities of interest). Moreover, the real distribution $f(\mathbf{v})$ is approximated by a suitable analytic distribution function $f^*(\mathbf{v})$:

$$f(\mathbf{v}) \approx f^*(\mathbf{v}) = \frac{n}{4\pi v_T^2} \delta_D(\|\mathbf{v} - \mathbf{u}\| - v_T), \quad (\text{B.3})$$

where n , \mathbf{u} and v_T are the density, mean velocity and thermal velocity of the original $f(\mathbf{v})$, respectively, and δ_D is the Dirac delta function. Substituting (B.3) into (B.2), introducing the random velocity $\mathbf{v}' = \mathbf{v} - \mathbf{u}$, and using the identity $d^3\mathbf{v}' = (v')^2 d\Omega dv'$, one gets:

$$\nu(\mathbf{v}_c) \approx \frac{n\langle\sigma\rangle}{4\pi} \oint_{\mathbb{S}} \sqrt{(v'_c)^2 + v_T^2 - 2v'_c v_T \cos\theta} d\Omega, \quad (\text{B.4})$$

which is solvable using $d\Omega = \cos\theta d\theta d\phi$. Consistently with the approximations already introduced, (B.4) can be further simplified by neglecting the $\cos\theta$ dependence inside the square root, leading to a very easily manageable formula for the total collision frequency:

$$\nu(\mathbf{v}_c) \approx n\langle\sigma\rangle\sqrt{(v'_c)^2 + v_T^2}.$$

A second issue is that (B.1) does not ensure mass, momentum and energy conservation on a finite grid: to satisfy such a requirement, one should compute *exactly*

$$\Delta N_{\text{in}} = \int_c \mathcal{M}(\mathbf{v}) d^3\mathbf{x} d^3\mathbf{v} = \langle\mathcal{M}\rangle_c V_c \neq \mathcal{M}(\mathbf{v}_c) V_c,$$

but this is extremely difficult, due to the complicated shape of the domain in velocity space. As an efficient alternative to computing the exact integral of the exact Nominal Maxwellian (NM) distribution function $\mathcal{M}(\mathbf{v})$, one can still use (B.1), but substituting the NM with what we refer to as a ‘‘Discrete Conservation-Corrected Maxwellian’’ (DCCM), $\mathcal{M}^*(\mathbf{v})$. The DCCM is constructed by applying small corrections to the NM that ensure explicitly all the aforementioned conservation rules on the grid. Among other possible choices, we write $\Delta f(\mathbf{v})$ as a first-order correction to the original Maxwellian: each value of the sampled Nominal Maxwellian is written as a function of the parameters to be conserved,

$$f_\alpha(n, \mathbf{u}, \varepsilon') = n \left(\frac{3}{4\pi\varepsilon'} \right)^{\frac{3}{2}} \exp \left[-\frac{\|\mathbf{v}_\alpha - \mathbf{u}\|^2}{4\varepsilon'/3} \right],$$

and Δf_α is the first order term of the Taylor series of f_α ,

$$\Delta f_\alpha = \frac{\partial f_\alpha}{\partial n} \Delta n + \frac{\partial f_\alpha}{\partial u_x} \Delta u_x + \frac{\partial f_\alpha}{\partial u_y} \Delta u_y + \frac{\partial f_\alpha}{\partial u_z} \Delta u_z + \frac{\partial f_\alpha}{\partial \varepsilon'} \Delta \varepsilon'.$$

What we have obtained is a 5-term expansion in the basis functions $\frac{\partial f_\alpha}{\partial(\cdot)}$, with unknown coefficients $[\Delta n, \Delta u_x, \Delta u_y, \Delta u_z, \Delta \varepsilon']$. These 5 coefficients are uniquely determined by requiring the combination of all the Δf_α s to satisfy the 5 scalar conservation equations.

Appendix C. Boundary Conditions

For a semi-Lagrangian scheme such as the Convected Scheme, boundary conditions (BCs) are applied to the *characteristics* that come off the wall, and no modification is made due to the wall on the characteristics that go towards the wall. As a consequence, BCs must be handled in a way that satisfies causality, and this is a considerable advantage when compared to Eulerian schemes. On the other hand, the outgoing characteristics must be correlated with the incoming characteristics, explicitly for each case, and this may be a non-trivial problem where quantities like mass or energy are to be exactly conserved.

The use of periodic planes enables the simulation of an axially-symmetric geometry using only half (or one quarter) of the computational domain. Implementing periodic planes is straightforward: when a part of a MC crosses the plane, its position and velocity are transformed by a rotation about the symmetry axis. Issues concerning this procedure take place wherever the transformed velocity does not coincide with one of the discrete values of the velocity mesh: in such cases, it is necessary to remap the velocity vector onto two or more vectors, and this fact introduces numerical diffusion in velocity space (although only in direction, and not in energy). Another possible side-effect is the late-time creation of non-physical density bumps in the region close to the axis, which take place essentially because of the asymmetries in the particle fluxes. The angular mesh used in the current version of the code is symmetric with respect to each of the three principal planes, and this property permits one to simulate only one half of the cylinder without incurring the above drawbacks.

Since in the CS no random numbers are generated, diffuse adiabatic reflection is implemented in an *integral* fashion, which permits us to model any geometry of the wall (a staircase approach is used here) by using a thick layer of ghost-cells:

1. the MCs coming from the domain pass seamlessly through the boundary wall, and they remap onto the ghost-cells by applying the same *volume-rule* that is used inside the domain;
2. the ghost-cells gather the incoming particles on an energy basis, so that the information about the angular direction is lost, but mass and kinetic energy can be exactly conserved;
3. for every ghost cell and for each speed, a list of the spatial cells in the domain which are ‘visible’ from that cell is constructed. Equal densities are launched into equal solid angles, for those cells in the list. This rule guarantees that an initial uniform isotropic distribution retains those properties after bouncing off the wall.

For any finite mesh-size, the implementation above guarantees that mass and total kinetic energy of the colliding particles are exactly conserved, while their average parallel momentum is transferred to the wall. Moreover, such an algorithm isotropizes the distribution function without introducing any diffusion in energy space, and it has the overall

effect of converting directional kinetic energy into random kinetic energy: as a consequence of the friction against the wall (no-slip condition), the temperature of the gas locally increases in an irreversible thermodynamic process.

The injector effectively imposes Dirichlet BCs on the distribution function. In a semi-Lagrangian framework, this must be implemented with particular care to ensure mass and energy conservation. The solution employed is to load the ghost cells immediately behind the inlet section with a drift-Maxwellian distribution function (having density n , mean velocity $\|\mathbf{u}\|$ and temperature T imposed a priori), and to launch the particles into the domain using the usual ballistic mover. This procedure does not guarantee that the mass flux per unit area [$\text{m}^{-2}\text{s}^{-1}$] equals the expected value $n\|\mathbf{u}\|$; in fact, the *net mass flux* through any given plane surface $\mathcal{S}_{\mathbf{k}}$ with normal \mathbf{k} is the sum of a positive contribution Γ^+ due to particles that cross the plane in the direction of \mathbf{k} (i.e. $\mathbf{v} \cdot \mathbf{k} > 0$), plus a negative contribution Γ^- due to particles that cross the plane in the opposite direction ($\mathbf{v} \cdot \mathbf{k} < 0$). If the same distribution function is loaded on both sides of $\mathcal{S}_{\mathbf{k}}$, then $\Gamma^+ + \Gamma^- = n\|\mathbf{u}\|$ as expected; but in our model the distribution function is loaded only on one side of $\mathcal{S}_{\mathbf{k}}$, and Γ^- is not imposed: noticing that in a sonic condition we have $\|\Gamma^+\| \gg \|\Gamma^-\|$, a satisfactory solution is to scale the density of the imposed distribution by a correction factor close to one, and not to let particles exit from the injector, so that $\Gamma^+ = n\|\mathbf{u}\|$ and $\Gamma^- = 0$.

We stress here that, due to the uniform Cartesian spatial mesh employed in the simulations, the injector is modeled in a very simplified way, using just a few cells. (A detailed analysis of the region close to the injector would require a high local resolution, using an unstructured grid as in [28], or an adaptive multi-grid approach).

References

- [1] W. N. G. Hitchon, Plasma Processes for Semiconductor Fabrication, Cambridge University Press, 1999.
- [2] J. Feng, W. N. G. Hitchon, Self-consistent kinetic simulation of plasmas, Phys. Rev. E 61 (2000) 3160–3173.
- [3] A. J. Christlieb, W. N. G. Hitchon, E. R. Keiter, A computational investigation of the effects of varying discharge geometry for an inductively coupled plasma, IEEE Trans. Plasma Sci. 28 (2000) 2214–2231.
- [4] W. N. G. Hitchon, G. J. Parker, J. E. Lawler, Physical and numerical verification of discharge calculations, IEEE Trans. Plasma Sci. 21 (1993) 228–238.
- [5] W. N. G. Hitchon, G. J. Parker, J. E. Lawler, Accurate models of collisions in glow discharge simulations, IEEE Trans. Plasma Sci. 22 (1994) 267–274.
- [6] A. Robert, A stable numerical integration scheme for the primitive meteorological equations, Atmos. Ocean 19 (1981) 35–46.
- [7] R. J. Purser, L. M. Leslie, An efficient interpolation procedure for high-order three-dimensional semi-Lagrangian models, Mon. Weather Rev. 119 (1991) 2492–2498.
- [8] A. Staniforth, J. Côté, Semi-Lagrangian integration schemes for atmospheric models – a review, Mon. Weather Rev. 119 (1991) 2206–2223.
- [9] R. Courant, K. O. Friedrichs, H. Lewy, Über die partiellen Differenzgleichungen der mathematische Physik, Math. Ann. 100 (1928) 32–74.
- [10] M. Falcone, R. Ferretti, Convergence analysis for a class of high-order semi-Lagrangian advection schemes, SIAM J. Numer. Anal. 35 (1998) 909–940.
- [11] J. E. Broadwell, Study of rarefied shear flow by the discrete velocity method, J. Fluid Mech. 19 (1964) 401–414.
- [12] L. Margolin, P. K. Smolarkiewicz, Antidiffusive velocities for multipass Donor Cell advection, SIAM J. Sci. Comput. 20 (1998) 907–929.

- [13] P. K. Smolarkiewicz, A fully multidimensional positive definite advection transport algorithm with small implicit diffusion, *J. Comput. Phys.* 54 (1984) 325–362.
- [14] P. K. Smolarkiewicz, W. W. Grabowski, The multidimensional positive definite advection transport algorithm: nonoscillatory option, *J. Comput. Phys.* 86 (1990) 355–375.
- [15] R. F. Warming, B. J. Hyett, The modified equation approach to the stability and accuracy analysis of finite-difference methods, *J. Comput. Phys.* 14 (1974) 159–179.
- [16] J. P. Boris, D. L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, *J. Comput. Phys.* 11 (1973) 38–69.
- [17] B. van Leer, Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme, *J. Comput. Phys.* 14 (1974) 361–370.
- [18] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* 49 (1983) 357–393.
- [19] P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. Numer. Anal.* 21 (1984) 995–1011.
- [20] P. L. Roe, Characteristic-based schemes for the Euler equations, *Annu. Rev. Fluid Mech.* 18 (1986) 337–365.
- [21] R. Artebrant, H. J. Schroll, Limiter-free third order logarithmic reconstruction, *SIAM J. Sci. Comput.* 28 (2006) 359–381.
- [22] A. Marquina, Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws, *SIAM J. Sci. Comput.* 15 (1994) 892–915.
- [23] J. W. Banks, T. Aslam, W. J. Rider, On sub-linear convergence for linearly degenerate waves in capturing schemes, *J. Comput. Phys.* 227 (2008) 6985–7002.
- [24] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser Verlag, 1992.
- [25] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford University Press, 2nd edition, 1994.
- [26] A. J. Christlieb, W. N. G. Hitchon, I. D. Boyd, Q. Sun, Kinetic description of flow past a micro-plate, *J. Comput. Phys.* 195 (2004) 508–527.
- [27] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (1954) 511–525.
- [28] D. A. Fixel, W. N. G. Hitchon, Convective scheme solution of the boltzmann transport equation for nanoscale semiconductor devices, *J. Comput. Phys.* 227 (2007) 1387–1410.