

A Galerkin Method for the Simulation of the Transient 2-D/2-D and 3-D/3-D Linear Boltzmann Equation

Matthias K. Gobbert,¹ Samuel G. Webster,^{1,3} and Timothy S. Cale²

Received September 13, 2005; accepted (in revised form) December 8, 2005; Published online February 17, 2006

Many production steps used in the manufacturing of integrated circuits involve the deposition of material from the gas phase onto wafers. Models for these processes should account for gaseous transport in a range of flow regimes, from continuum flow to free molecular or Knudsen flow, and for chemical reactions at the wafer surface. We develop a kinetic transport and reaction model whose mathematical representation is a system of transient linear Boltzmann equations. In addition to time, a deterministic numerical solution of this system of kinetic equations requires the discretization of both position and velocity spaces, each two-dimensional for 2-D/2-D or each three-dimensional for 3-D/3-D simulations. Discretizing the velocity space by a spectral Galerkin method approximates each Boltzmann equation by a system of transient linear hyperbolic conservation laws. The classical choice of basis functions based on Hermite polynomials leads to dense coefficient matrices in this system. We use a collocation basis instead that directly yields diagonal coefficient matrices, allowing for more convenient simulations in higher dimensions. The systems of conservation laws are solved using the discontinuous Galerkin finite element method. First, we simulate chemical vapor deposition in both two and three dimensions in typical micron scale features as application example. Second, stability and convergence of the numerical method are demonstrated numerically in two and three dimensions. Third, we present parallel performance results which indicate that the implementation of the method possesses very good scalability on a distributed-memory cluster with a high-performance Myrinet interconnect.

KEY WORDS: Boltzmann transport equation; spectral Galerkin method; discontinuous Galerkin method; cluster computing; chemical vapor deposition.

¹ Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA. E-mail: gobbert@math.umbc.edu

² Focus Center – New York, Rensselaer: Interconnections for Hyperintegration, Isermann Department of Chemical and Biological Engineering, Rensselaer Polytechnic Institute, CII 6015, 110 8th Street, Troy, NY 12180-3590, USA.

³ Department of Mathematics and Computer Science, Hillsdale College, 33E. College Street, Hillsdale, MI 49242, USA.

1. INTRODUCTION

Several production steps in the manufacturing of integrated circuits (ICs) involve gas flow at pressures that range from very low to atmospheric [16]. The Boltzmann transport equation from gas dynamics [5, 16] is appropriate to model such processes on the micron length scales of patterned features that populate the wafers during IC fabrication. For atomic layer deposition (ALD), we introduced a kinetic transport and reaction model (KTRM) that consists of a linear Boltzmann equation without collision term in two spatial dimensions [10, 14]. We extended it to multiple species in [12, 13] and three spatial dimensions in [24, 25]. Extending the KTRM to include the effect of collisions, for the case where the gas flow is dominated by an inert carrier gas, gives the system of transient linear Boltzmann equations for the n_s reactive species

$$\frac{\partial f^{(i)}}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f^{(i)} = \frac{1}{\text{Kn}} Q_i(f^{(i)}), \quad i = 1, \dots, n_s, \quad (1.1)$$

with the linear collision operators

$$Q_i(f^{(i)}) = \int_{\mathbb{R}^3} \sigma_i(\mathbf{v}, \mathbf{v}') \left[M^{(i)}(\mathbf{v}) f^{(i)}(\mathbf{x}, \mathbf{v}', t) - M^{(i)}(\mathbf{v}') f^{(i)}(\mathbf{x}, \mathbf{v}, t) \right] d\mathbf{v}', \quad (1.2)$$

stated here in dimensionless form. See also [11] for additional details on the derivation of the model and its non-dimensionalization.

The left-hand side of (1.1) describes the convective transport of the gas molecules, and the collision operators (1.2) on the right-hand side of (1.1) describe the effects of molecular collisions. Here, $\sigma_i(\mathbf{v}, \mathbf{v}') = \sigma_i(\mathbf{v}', \mathbf{v}) > 0$ is a given function that treats the collision events and $M^{(i)}(\mathbf{v})$ denotes the Maxwellian distribution of species i . The unknown in this kinetic equation is the scaled probability density $f^{(i)}(\mathbf{x}, \mathbf{v}, t)$, which we call kinetic density in the following to distinguish it from other densities, for the molecules of species i to have a position \mathbf{x} and velocity \mathbf{v} at time t in an infinitesimal volume $d\mathbf{x} d\mathbf{v}$ in phase space.

The model (1.1) is stated in dimensionless form, as indicated by the appearance of the Knudsen number Kn . Kinetic models take both transport and collisions of molecules into account, and these require different reference quantities for their non-dimensionalization. Specifically, the transport involves the characteristic length scale L^* of the spatial domain, while the collision effects are characterized by the mean free path λ (average distance traveled by a molecule between collisions) of the molecules, which is largely determined by the pressure in the chemical reactor. In the non-dimensionalization, the ratio of these characteristic lengths arises and is defined as the Knudsen number $\text{Kn} := \lambda/L^*$ that characterizes the relative

dominance of inter-molecular or molecule–wall collisions in kinetic models [16]. See [12, 14] for the full details of our non-dimensionalization. In the hydrodynamic regime for $\text{Kn} \ll 1.0$, inter-molecular collisions dominate and gas flow is fluid-like. In the transition regime for $\text{Kn} \approx 1.0$, inter-molecular and molecule–wall collisions balance. In the Knudsen regime for $\text{Kn} \gg 1.0$, the effects of inter-molecular collisions are negligible and the flow is dominated by convective transport. Notice that while the set of equations represented by (1.1)–(1.2) above for each species appear decoupled, the solutions for all reactive species will in general be coupled to each other through the boundary conditions at the wafer surface. These boundary conditions are crucial for our applications and involve coupling to general models for how the gas phase species interact with the surface, including chemical reactions that occur at the surface. While we are interested in a method that can handle a wide range of Knudsen numbers including up to $0.01 \leq \text{Kn} \leq \infty$, the focus is on application problems in the transition regime with $\text{Kn} \approx 1.0$, in which the use of kinetic models is most needed [16].

The numerical solution of the linear Boltzmann equation (1.1)–(1.2) presents a challenge for numerical methods due to the large number of independent variables present in a kinetic equation: For 3-D/3-D problems, the three-dimensional spatial domain $\Omega \subset \mathbb{R}^3$ for \mathbf{x} and the three-dimensional velocity vector $\mathbf{v} \in \mathbb{R}^3$ need to be discretized at every time t . Even for 2-D/2-D problems, we need to discretize the four dimensions of the two-dimensional spatial and two-dimensional velocity domains at every time step t . Both two-dimensional and three-dimensional models are of interest in practice; to make the notation concrete, we use three-dimensional notation throughout this paper.

One classical application of the linear Boltzmann equation is neutron transport [5, Chapter IV]. The linearity of this equation is the result of considering the background medium through which the neutrons travel to be significantly more dense and thus neutron–neutron collisions can be neglected. Another classical application of the linear Boltzmann equation is charge transport in semiconductor devices [17]. Here, transport is modeled by the linear Boltzmann equation with forcing term coupled to the Poisson equation, see [4, 22] and the references therein.

Various numerical methods have been designed for these models; see [4] for a detailed review. Two broad categories into which numerical solutions of the Boltzmann equation may fall are stochastic methods and deterministic methods. The most popular of the stochastic approaches are Monte Carlo methods [1]. These methods utilize random sampling and if the number of averaging particles is not chosen large enough, they do not accurately describe transient behavior and equilibrium states. In [4], a deterministic method for the Boltzmann–Poisson system is developed by

transforming the linear Boltzmann equation to a system of conservation laws which is linear when ignoring the coupling with the Poisson equation. This system is solved by a WENO finite difference method with explicit Runge–Kutta time-stepping. Extensive numerical evidence is presented to demonstrate the advantages of an efficient deterministic scheme over existing Monte Carlo methods, whose advantages include “faster speed, noise-free resolution, and easiness for arbitrary moment evaluations” [4]. These conclusions are also supported in [9].

In the spirit of [4, 9], in order to have direct access to the kinetic density $f^{(i)}(\mathbf{x}, \mathbf{v}, t)$ as functions of both \mathbf{x} and \mathbf{v} in a transient model and to avoid the stochastic variability associated with Monte Carlo methods, we develop here a deterministic numerical method for the system of linear Boltzmann equations in (1.1). We choose to use (a modification of) the numerical method found in [22] for the semiconductor Boltzmann equation, because the KTRM in (1.1)–(1.2) can be formulated in a similar way. However, the numerical challenges associated with our KTRM lie in different aspects of the problem: The fundamental challenge of the semiconductor Boltzmann equation results from its coupling with a Poisson equation that is driven by an applied voltage whose values vary over a huge range. But the spatial domain of a transistor channel is often reasonably approximated by a one-dimensional domain, and symmetry considerations can reduce the dimension of the velocity space. In our KTRM, we have a simpler transport term and no coupling to another equation (in the gas phase), but we are keenly interested in developing a model and numerical method that can handle fully three-dimensional spatial domains of irregular shape with associated three-dimensional velocity discretizations. The crucial coupling for our applications involves the reaction models at the wafer surface and affects the solution of (1.1) via the boundary conditions there. Moreover, we need models for several reactive species in practice, giving us the *system* of transient linear Boltzmann equations in (1.1).

Our interest in models in higher dimensions is the reason why we use a modification of classical Galerkin methods for the linear Boltzmann equation: To discretize the velocity space, Galerkin methods approximate the kinetic density $f^{(i)}(\mathbf{x}, \mathbf{v}, t)$ for each species i by an expansion in basis functions in velocity space, whose expansion coefficients depend on position and time. The Galerkin ansatz of inserting the expansion for $f^{(i)}(\mathbf{x}, \mathbf{v}, t)$ in (1.1) and testing against the basis functions approximates each linear Boltzmann equation by a system of transient linear hyperbolic conservation laws. The classical choice of basis functions uses Hermite polynomials [7, 15, 21–23], which result in dense coefficient matrices in the hyperbolic systems. We use here an alternative choice of basis functions, a collocation basis, that results in *diagonal* coefficient matrices. This

makes simulations in higher dimensions substantially more convenient in practice, as diagonalization of the coefficient matrices is no longer necessary and because the inflow part of the boundary for each equation in the system can be directly identified. Our method can be related to the classical method and analytic results from [22] still apply to guarantee stability and convergence of the velocity discretization. For the solution of the systems of linear hyperbolic conservation laws, we choose to use the discontinuous Galerkin method (DGM), implemented in the code DG [20], because the method is perfectly suited for the resolution of irregular spatial domains. The DGM was first introduced by Reed and Hill [19] for solving similar sets of equations. See, for instance, [8] for more information about the method. In the studies here, we use linear basis functions on each element and first-order explicit time-stepping (Euler); this presents only the initial step designed to focus on the assessment of the code's parallel performance and obtain reference solutions, before using automatic mesh refinement and coarsening already available in DG and higher-order finite elements.

The focus of this paper is to explain our numerical method in detail and to demonstrate (i) that it can solve realistic application problems in two and three dimensions, (ii) that it exhibits convergence for reasonable resolutions in two and three dimensions, in extension of one-dimensional studies in [22, 23], and (iii) that its parallel implementation scales well on a distributed-memory cluster with high-performance interconnect. First, we present a brief outline of our model extension to the process of chemical vapor deposition (CVD) in Section 2, already in dimensionless form. Second, Section 3 introduces our numerical method in detail, contrasts it with existing series expansion methods, and discusses its convergence. Third, our numerical studies cover three aspects: Section 4.1 presents transient studies for single-species CVD that demonstrate that problems of interest can be solved and that it is useful to be able to access the kinetic density $f^{(i)}(\mathbf{x}, \mathbf{v}, t)$ itself, focusing on the transition regime with $\text{Kn} = 1.0$. Section 4.2 contains the numerical demonstrations of stability and convergence in two and three dimensions for a wide range of Kn from 0.01 to ∞ . Finally, Section 4.3 provides performance studies of our implementation on a parallel computer with 64 processors to show the speedup possible for our method on such a platform.

2. THE APPLICATION AND ITS MODEL

2.1. The Application: Microelectronics Manufacturing

The starting point for the manufacturing of ICs is most commonly a silicon wafer. Repeatedly during the manufacturing process, the surface of

the wafer has a microstructure of millions of trenches, via (round holes), and/or other structures, generically called ‘features,’ that were etched into the surface in a previous production step. The class of processes of interest here involves the deposition of films that either partly or completely fill these features. Two types of processes are particularly relevant: atomic layer deposition (ALD) and chemical vapor deposition (CVD). Depending upon the application, the deposited material, which forms due to reactions of gas phase species on the surface, may be either an insulating layer of, say, silicon dioxide SiO_2 , or conductive metal, say, copper Cu.

Figure 1 shows prototypical examples of two- and three-dimensional spatial domains for the processes under consideration, both of which are of interest for practical simulations. Specifically, Fig. 1(a) shows the two-dimensional cross-section of a trench that is modeled as infinite in the third dimension. The mathematical domain $\Omega \subset \mathbb{R}^2$ of our model is the region filled by gas inside and just above the feature. The domain boundary is comprised of three distinct sections: $\partial\Omega = \Gamma_w \cup \Gamma_t \cup \Gamma_s$. Here, Γ_w denotes the solid wafer surface indicated by the hash marks in Fig. 1(a), Γ_t is the top boundary of the domain that forms the interface with the bulk gas in the reactor chamber, and Γ_s denotes the union of the portion of the boundary on the left and right sides of the domain above the wafer surface; by construction, all segments that comprise Γ_s are orthogonal to one of the coordinate axes. The length L denotes the width of the feature mouth and can be $0.25 \mu\text{m}$ or less in typical applications today. The number A specifies the feature aspect ratio (depth over width of the feature).

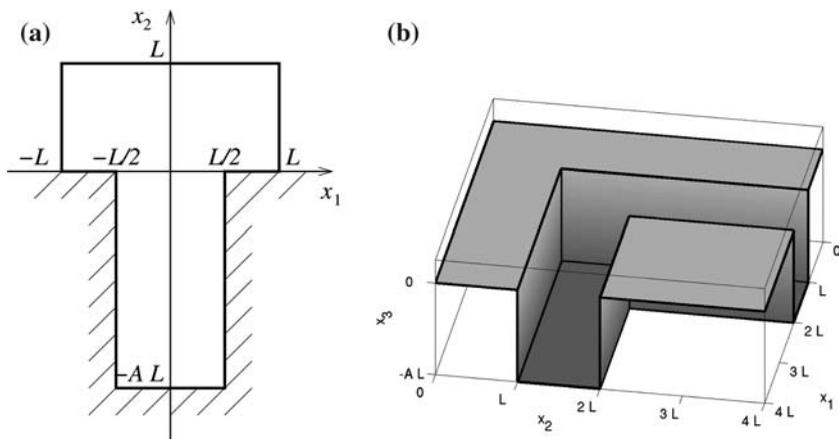


Fig. 1. Schematics of (a) two-dimensional and (b) three-dimensional domains defining the feature width L and aspect ratio A .

Figure 1(b) sketches the corner of a trench designed to analyze the three-dimensional effects of a sharp corner on the behavior of the manufacturing process. Each trench is modeled as semi-infinite in the direction away from the corner. L and A denote again the width and aspect ratio of the feature. The domain $\Omega \subset \mathbb{R}^3$ of our model comprises the gas-filled area inside and just above the surface. Again, the boundary of Ω is comprised of three distinct sections: $\partial\Omega = \Gamma_w \cup \Gamma_t \cup \Gamma_s$. Here, Γ_w denotes the solid wafer surface indicated by the gray shading in Fig. 1(b), Γ_t is the top of the boundary of the domain that forms the interface with the bulk gas in the reactor chamber (at the top of the figure), and Γ_s denotes the union of the remaining portions of the boundary; by construction, the segments of Γ_s are orthogonal to one of the coordinate axes.

Figure 2 shows a schematic to demonstrate the evolution of the deposition of a thin initial layer during chemical vapor deposition. The figure shows a single-species model with the (generic) reactive gas A fed from the bulk gas above the wafer surface. Upon reacting at the wafer surface, a layer of solid material deposits. Starting with an empty trench in Fig. 2(a), deposition starts in areas that ‘see’ most of the gaseous reactant, the flat wafer areas to the left and right of the trench and the bottom of the trench as well as its side walls in (b). As deposition continues in (c), it becomes clear that the corners at the bottom of the trench take the longest time to cover with solid, so that by that time already additional layers of solid have deposited on the areas to the left and right of the trench in (d).

The operating conditions of a typical reactor are high temperature $T = 500$ K (approximately 400°F) and total pressures P_{total} ranging over a wide range of values. For our simulations, we assume a fixed temperature T and vary the total pressure P_{total} . A low total pressure of $P_{\text{total}} = 1$ Torr (atmospheric pressure is 760 Torr) corresponds to a mean free path (the average distance that a molecule of gas travels between collisions) of about $\lambda = 100 \mu\text{m} = 10^{-4}$ m. Higher total pressures such as

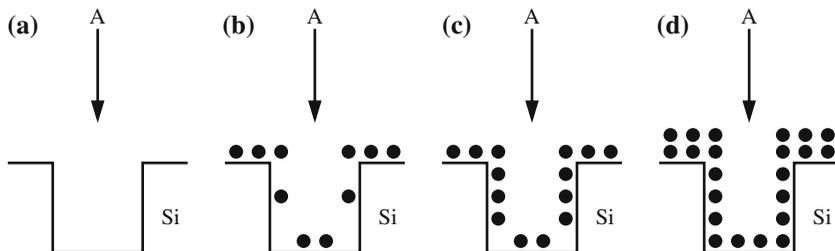


Fig. 2. Schematic of the evolution of deposition of a thin initial layer during chemical vapor deposition.

$P_{\text{total}} = 10^2$ and 10^4 Torr correspond to smaller mean free paths of about $\lambda = 1 \mu\text{m} = 10^{-6}$ m and $\lambda = 10^{-2} \mu\text{m} = 10^{-8}$ m, respectively. Models of interest range from feature scale models designed to represent individual features as sketched in Fig. 1 with a typical length scale of $L^* = 1 \mu\text{m} = 10^{-4}$ cm to mesoscopic scale models of several hundred or several thousand features with typical length scales of $L^* = 0.01$ to 0.1 cm. Here, we focus on the feature scale and thus fix the length scale of interest L^* . Therefore, with L^* fixed, varying the total pressure in the reactor changes λ and hence the Knudsen number Kn . We directly use Kn as the parameter that controls the transport regime represented by the dimensionless model (1.1)–(1.2), from the near-hydrodynamic regime ($\text{Kn} = 0.01$) through the transition regime ($\text{Kn} = 1.0$) to the Knudsen regime ($\text{Kn} = \infty$).

2.2. The Model Equations

As discussed in the Introduction, we use the Boltzmann equation of gas dynamics as our starting point for modeling processes such as atomic layer deposition or chemical vapor deposition, for Knudsen numbers above 0.01. More precisely, we formulate a system of Boltzmann transport equations for all gaseous species present in the reactor. In addition to the reactive species $i = 1, \dots, n_s$, this includes an inert carrier gas, denoted by $i = 0$, that is present in the processes under consideration. Then the appropriate model is a system of equations for all of the distribution functions $f^{(i)}(\mathbf{x}, \mathbf{v}, t)$, $i = 0, 1, \dots, n_s$. Following the approach suggested in [5, Chapter II], we obtain a system of Boltzmann transport equations for all gaseous species

$$\frac{\partial f^{(i)}}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f^{(i)} = \frac{1}{\text{Kn}} \sum_{j=0}^{n_s} Q_{ij}(f^{(i)}, f^{(j)}), \quad i = 0, 1, \dots, n_s, \quad (2.1)$$

with the collision operators given by

$$\begin{aligned} & Q_{ij}(f^{(i)}, f^{(j)}) \\ &= \int_{\mathbb{R}^3} \int_0^{2\pi} \int_0^{\pi/2} \left[f^{(i)}(\mathbf{v}') f^{(j)}(\mathbf{v}'_*) - f^{(i)}(\mathbf{v}) f^{(j)}(\mathbf{v}_*) \right] B_{ij}(\vartheta, |\mathbf{V}|) d\vartheta d\varepsilon d\mathbf{v}_*. \end{aligned} \quad (2.2)$$

We state the model already in dimensionless form here, as indicated by the factor $1/\text{Kn}$. Here, $B_{ij}(\vartheta, |\mathbf{V}|)$ models the details of the collisions between molecules of species i and j . The short-hand notation under the integral implies the same position \mathbf{x} and time t for all $f^{(i)}$ and $f^{(j)}$.

As stated, (2.1)–(2.2) is a system of coupled, non-linear Boltzmann transport equations for all gaseous species and is challenging to solve

numerically, because of the high dimensionality of the space $(\mathbf{x}, \mathbf{v}) \in \Omega \times \mathbb{R}^3$, because of the fivefold integral in the collision operators, and because of the non-linearity in the integral. Using properties of the applications considered here, which results in a system of linear Boltzmann equations for the reactive species only, it is possible to alleviate the latter two challenges significantly.

We make four realistic assumptions about the class of processes considered: We observe that (i) the reactive chemical species present in the reactor are typically an order of magnitude less dense than the inert carrier gas. This is common operating practice, in order to keep the reactant species from interacting in the gas phase, which could cause gas phase reactions. Such gas phase reactions in general are detrimental. Generalizing [5, Chapter IV] to multiple species, the dominant collisions are then given by those involving the carrier gas $j = 0$ on the right-hand side of (2.1), and it is therefore justified for all equations $i = 0, 1, \dots, n_s$ to consider only the collision operators Q_{i0} and to neglect the collision operators Q_{ij} , $j = 1, \dots, n_s$ [11, 12, 14]. In other words, only collisions between molecules of each reactive species with molecules of the carrier gas are considered. We also observe that (ii) since the carrier gas is inert, i.e., it does not react with the reactive species, its equation $i = 0$ decouples completely from the remaining equations for $i = 1, \dots, n_s$; the inertness is materially needed here to justify the decoupling in the reaction boundary condition. Additionally, we make the reasonable model assumptions that the carrier gas (iii) is itself in steady-state ($\partial f^{(0)}/\partial t = 0$) and (iv) is uniformly distributed in space ($\nabla_{\mathbf{x}} f^{(0)} = 0$) [5]. Then the equation for $i = 0$ in (2.1) reduces to $Q_{00}(f^{(0)}, f^{(0)}) = 0$ only, which has as its solution a Maxwellian $f^{(0)}(\mathbf{x}, \mathbf{v}, t) = M^{(0)}(\mathbf{v})$ [5, 6]. We note that our formulation of the KTRM also tacitly assumes that reactions among the reactive species *in the gas phase* are even less likely to occur than collisions among them; this assumption is appropriate for processes of interest here [11, 16].

Retaining only the collision operators Q_{i0} and using the solution for $f^{(0)}$ in the equations for the reactive species $i = 1, \dots, n_s$ in (2.1) yields the problem (1.1) with the linear collision operator $Q_i(f^{(i)}) := Q_{i0}(f^{(i)}, f^{(0)})$. Following again [5, Chapter IV], we can rewrite the linear collision operator in the form given in (1.2) that contains the known solution $f^{(0)}(\mathbf{v})$ of species $i = 0$ inside $\sigma_i(\mathbf{v}, \mathbf{v}')$ that can be precomputed or modeled directly [11]. The scattering term $\sigma_i(\mathbf{v}, \mathbf{v}')$ is a positive function satisfying the symmetry relation $\sigma_i(\mathbf{v}, \mathbf{v}') = \sigma_i(\mathbf{v}', \mathbf{v})$ due to the principle of detailed balance [5, Chapter IV]. Besides being a traditional formulation of the linear collision operator in this context, we are interested in writing it in this form, because it facilitates the application of the numerical method below.

2.3. Boundary Conditions and Initial Condition

In this section, we discuss the boundary and initial conditions for our application. For simplicity, we use a scalar $f(\mathbf{x}, \mathbf{v}, t)$ in this section, appropriate for a single-species example. Recall from Section 2.1 that the boundary $\partial\Omega$ is composed of three distinct sections, $\partial\Omega = \Gamma_w \cup \Gamma_t \cup \Gamma_s$, where a different boundary condition will be prescribed for each of the three portions. In the following, $\mathbf{n} = \mathbf{n}(\mathbf{x})$ denotes the unit *outward* normal vector at a surface point $\mathbf{x} \in \partial\Omega$.

At the inflow boundary along the interface to the bulk of the gas phase at the top of the domain Γ_t , we assume that the distribution of f is known; specifically, the inflow of the reactants is prescribed by a Maxwellian distribution:

$$f(\mathbf{x}, \mathbf{v}, t) = \frac{c^{\text{top}}}{[2\pi(v^*)^2]^{3/2}} \exp\left(-\frac{|\mathbf{v}|^2}{2(v^*)^2}\right), \quad \mathbf{x} \in \Gamma_t, \quad \mathbf{n} \cdot \mathbf{v} < 0. \quad (2.3)$$

Here, v^* is chosen to be the thermodynamic average speed and c^{top} represents the (dimensionless) concentration at the top of the domain [12, 14].

At each portion of the vertical boundaries above the mean wafer surface Γ_s , specular reflection is used

$$f(\mathbf{x}, \mathbf{v}, t) = f(\mathbf{x}, \mathbf{v}', t), \quad \mathbf{x} \in \Gamma_s, \quad \mathbf{n} \cdot \mathbf{v} < 0, \quad \mathbf{n} \cdot \mathbf{v}' > 0, \quad (2.4)$$

with $\mathbf{v} = \mathbf{v}' - 2\mathbf{n}(\mathbf{n} \cdot \mathbf{v}')$.

The crucial boundary condition for the two applications under consideration is the one at the wafer surface Γ_w that results in the deposition of the solid film. The general form is given as

$$f(\mathbf{x}, \mathbf{v}, t) = \alpha(\mathbf{x}, t) M(\mathbf{v}), \quad \mathbf{x} \in \Gamma_w \quad \mathbf{n} \cdot \mathbf{v} < 0. \quad (2.5)$$

Observe that the separation of variables indicates that the molecules will re-emit from the wafer surface with a Maxwellian distribution, $M(\mathbf{v})$. The term $\alpha(\mathbf{x}, t)$ contains all details of any conserved quantities, reaction rates, etc., and depends on the application. For atomic layer deposition, we refer to [12, 14] and references therein for specific details of this boundary condition. For chemical vapor deposition, we pose the following single-species model; see [11] for more details and a generalization to multi-species models. A model for the deposition rate $R(x, t)$ at the wafer surface for chemical vapor deposition can be written as $R(\mathbf{x}, t) = \gamma_0 \eta(\mathbf{x}, t)$, where γ_0 is the constant, dimensionless sticking factor representing the percentage of molecules that “stick” to the wafer surface and $\eta(\mathbf{x}, t)$ is the flux of the

gaseous species to the surface

$$\eta(\mathbf{x}, t) = \int_{\mathbf{n} \cdot \mathbf{v}' > 0} |\mathbf{n} \cdot \mathbf{v}'| f(\mathbf{x}, \mathbf{v}', t) d\mathbf{v}', \quad \mathbf{x} \in \Gamma_w. \quad (2.6)$$

At the wafer surface, then, we use the boundary condition

$$f(\mathbf{x}, \mathbf{v}, t) = C [\eta(\mathbf{x}, t) - R(\mathbf{x}, t)] M(\mathbf{v}), \quad \mathbf{x} \in \Gamma_w, \quad \mathbf{n} \cdot \mathbf{v} < 0. \quad (2.7)$$

With $R(\mathbf{x}, t) = \gamma_0 \eta(\mathbf{x}, t)$, we obtain

$$f(\mathbf{x}, \mathbf{v}, t) = C [1 - \gamma_0] \eta(\mathbf{x}, t) M(\mathbf{v}), \quad \mathbf{x} \in \Gamma_w, \quad \mathbf{n} \cdot \mathbf{v} < 0, \quad (2.8)$$

with the scaling factor C chosen such that, in the absence of reactions ($R(\mathbf{x}, t) = 0$), mass conservation is guaranteed. In other words, we demand that influx must equal outflux for $R = 0$:

$$\int_{\mathbf{n} \cdot \mathbf{v} < 0} |\mathbf{n} \cdot \mathbf{v}| f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} = \int_{\mathbf{n} \cdot \mathbf{v} > 0} |\mathbf{n} \cdot \mathbf{v}| f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}. \quad (2.9)$$

Finally, at the initial time $t = 0$, the distribution of the gaseous species is again assumed known with a Maxwellian distribution

$$f(\mathbf{x}, \mathbf{v}, 0) = f_{\text{ini}} := \frac{c^{\text{ini}}}{[2\pi(v^*)^2]^{3/2}} \exp\left(-\frac{|\mathbf{v}|^2}{2(v^*)^2}\right) \quad \mathbf{x} \in \Omega, \quad \mathbf{v} \in \mathbb{R}^3, \quad \text{at } t = 0, \quad (2.10)$$

where c^{ini} denotes a constant. For instance, $c^{\text{ini}} = 0$ models the situation in which no reactive gas is present at the beginning of the simulation.

Our boundary conditions are a special case of those in [22] and allow us to use the analytical results of that work later. Specifically, using the notation from [22], we have $\partial\Omega_{\text{in}} = \Gamma_i$ with boundary condition

$$f(\mathbf{x}, \mathbf{v}, t) = f_{\text{in}}(\mathbf{x}, \mathbf{v}, t), \quad \mathbf{x} \in \partial\Omega_{\text{in}}, \quad \mathbf{n} \cdot \mathbf{v} < 0, \quad (2.11)$$

and $\partial\Omega_{\text{re}} = \Gamma_w \cup \Gamma_s$ with

$$f(\mathbf{x}, \mathbf{v}, t) = \int_{\mathbf{n} \cdot \mathbf{v}' > 0} r(\mathbf{x}, \mathbf{v}, \mathbf{v}') f(\mathbf{x}, \mathbf{v}', t) d\mathbf{v}', \quad \mathbf{x} \in \partial\Omega_{\text{re}}, \quad \mathbf{n} \cdot \mathbf{v} < 0. \quad (2.12)$$

As in [22], we make the following assumptions about $r(\mathbf{x}, \mathbf{v}, \mathbf{v}')$ on the reflecting boundary $\partial\Omega_{\text{re}}$:

$$M(\mathbf{v}) = \int_{\mathbf{n} \cdot \mathbf{v}' > 0} r(\mathbf{x}, \mathbf{v}, \mathbf{v}') M(\mathbf{v}') d\mathbf{v}', \quad \mathbf{x} \in \partial\Omega_{\text{re}}, \quad \mathbf{n} \cdot \mathbf{v} < 0, \quad (2.13)$$

and

$$|\mathbf{n} \cdot \mathbf{v}| r(\mathbf{x}, \mathbf{v}, \mathbf{v}') M(\mathbf{v}') = |\mathbf{n} \cdot \mathbf{v}'| r(\mathbf{x}, -\mathbf{v}', -\mathbf{v}) M(\mathbf{v}), \quad \mathbf{x} \in \partial\Omega_{\text{re}}, \quad \mathbf{n} \cdot \mathbf{v} < 0, \quad \mathbf{n} \cdot \mathbf{v}' > 0. \quad (2.14)$$

Equation (2.13) implies the preservation of the steady-state distribution. Equation (2.14) implies that the number of molecules reflected from the wall from a velocity range of $(\mathbf{v}', \mathbf{v}' + d\mathbf{v}')$ to a velocity range of $(\mathbf{v}, \mathbf{v} + d\mathbf{v})$ is equal to the number of molecules reflected from the wall from a velocity range of $(-\mathbf{v}, -\mathbf{v} - d\mathbf{v})$ to a velocity range of $(-\mathbf{v}', -\mathbf{v}' - d\mathbf{v}')$ [5], i.e., the microscopic behavior at the boundary is time reversible. Additionally, we require that for the initial and inflow data

$$\int_{\Omega} \int_{\mathbb{R}^3} f_{\text{ini}}^2 \frac{d\mathbf{v}}{M(\mathbf{v})} d\mathbf{x} \leq B_{\text{ini}} < \infty, \quad (2.15)$$

$$\int_{\partial\Omega_{\text{in}}} \int_{\mathbf{n} \cdot \mathbf{v} < 0} |\mathbf{n} \cdot \mathbf{v}| f_{\text{in}}^2 \frac{d\mathbf{v}}{M(\mathbf{v})} dS \leq B_{\text{in}} < \infty, \quad (2.16)$$

for constants $B_{\text{in}}, B_{\text{ini}}$ for all t . These assumptions are reasonable for the application under consideration.

3. THE NUMERICAL METHOD

For clarity of the presentation, we explain and analyze the numerical method for a single-species model ($n_s = 1$) and drop the species superscripts. The generalization to several reactive species is straightforward, since we plan on using explicit time-stepping to evaluate the boundary conditions and right-hand side at the old time step. Hence, the concrete equation to be solved for $f(\mathbf{x}, \mathbf{v}, t)$ with $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ reads

$$\frac{\partial f}{\partial t} + v^{(1)} \frac{\partial f}{\partial x_1} + v^{(2)} \frac{\partial f}{\partial x_2} + v^{(3)} \frac{\partial f}{\partial x_3} = \frac{1}{\text{Kn}} Q(f), \quad (3.1)$$

where $v^{(1)}, v^{(2)},$ and $v^{(3)}$ are the components of the velocity vector \mathbf{v} in the x_1 -, x_2 -, and x_3 -directions, respectively, and $Q(f)(\mathbf{x}, \mathbf{v}, t)$ is the linear collision operator for a single-species model. In this section only, we use superscripts for the components of the velocity vector $\mathbf{v} = (v^{(1)}, v^{(2)}, v^{(3)})$ to have the subscripts available for the indexing of the velocity mesh.

3.1. Series Expansion Methods Based on Hermite Polynomials

In this section, we detail classical series expansions and the corresponding choice of basis functions following [22, 23]. However, our

notation is different in a number of ways. In particular, we introduce a mapping from the indices in three dimensions to a one-dimensional counting scheme instead of using a vector-valued index in 3-D; this choice makes the implementation of the velocity discretization in computer code more convenient.

An approximation to the kinetic solution is obtained through a series expansion of the form

$$f(\mathbf{x}, \mathbf{v}, t) \approx f_K(\mathbf{x}, \mathbf{v}, t) := \sum_{k=0}^{K-1} \tilde{f}_k(\mathbf{x}, t) \psi_k(\mathbf{v}), \quad (3.2)$$

where $\psi_k(\mathbf{v}), k = 0, 1, \dots, K - 1$, form an orthonormal set of basis functions in velocity space with respect to the weighted L^2 -inner product

$$\langle f, g \rangle_G := \int_{\mathbb{R}^3} f(\mathbf{v}) g(\mathbf{v}) \frac{d\mathbf{v}}{M(\mathbf{v})}, \quad (3.3)$$

whose weight function $1/M(\mathbf{v})$ is motivated by theoretical considerations for the linear Boltzmann equation [18, 23]. This choice of weight function leads naturally to a family of orthonormal basis functions defined by $\psi_k(\mathbf{v}) = M(\mathbf{v}) H_{k_1}^{(1)}(v^{(1)}) H_{k_2}^{(2)}(v^{(2)}) H_{k_3}^{(3)}(v^{(3)})$ which uses products of a Maxwellian with one-dimensional (modified) Hermite polynomials $H_{k_\delta}^{(\delta)}(v^{(\delta)}), k_\delta = 0, \dots, K_\delta - 1, \delta = 1, 2, 3$, as underlying orthonormal basis functions. This approach was already proposed by Grad [15] and used in early work on this problem by Chorin [7].

To construct these basis functions, the weight function $e^{-(v^{(\delta)})^2}$ of the conventional Hermite polynomials is transformed to the one-dimensional Maxwellian $M^{(\delta)}(v) = \exp(-(v^{(\delta)})^2/2)/\sqrt{2\pi}$ for $v^{(\delta)} \in \mathbb{R}$ in each of the dimensions $\delta = 1, 2, 3$. Then the set $\{H_i^{(\delta)}(v^{(\delta)}), i = 0, 1, 2, \dots\}$ forms a family of orthonormal polynomials, where $H_i^{(\delta)}(v^{(\delta)})$ is a polynomial of degree i and $H_i^{(\delta)}(v^{(\delta)})$ and $H_j^{(\delta)}(v^{(\delta)})$ satisfy

$$\int_{-\infty}^{+\infty} M^{(\delta)}(v^{(\delta)}) H_i^{(\delta)}(v^{(\delta)}) H_j^{(\delta)}(v^{(\delta)}) dv^{(\delta)} = \delta_{ij} \quad \text{for all } i, j \in \{0, 1, 2, \dots\}.$$

Here, δ_{ij} denotes the Kronecker delta function, which is 1 for $i = j$ and 0 for $i \neq j$. We use these modified Hermite polynomials to define the following Gaussian quadrature rule for a function $p(v^{(\delta)})$

$$\int_{-\infty}^{+\infty} M^{(\delta)}(v^{(\delta)}) p(v^{(\delta)}) dv^{(\delta)} \approx \sum_{\ell=0}^{K_\delta-1} q_\ell^{(\delta)} \left(M^{(\delta)}(v_\ell^{(\delta)}) \right)^2 p(v_\ell^{(\delta)}) \quad (3.4)$$

with (modified, factored) quadrature weights $q_\ell^{(\delta)}$. By choosing the roots of the K_δ th degree Hermite polynomial $H_{K_\delta}^{(\delta)}(\mathbf{v})$ as the collocation points $v_\ell^{(\delta)}$, $\ell = 0, \dots, K_\delta - 1$, and the quadrature weights $q_\ell^{(\delta)}$, $\ell = 0, \dots, K_\delta - 1$, appropriately, this approximation will be a unique Gaussian quadrature and exact for polynomials up to degree $2K_\delta - 1$. The collocation points $v_\ell^{(\delta)}$ will be symmetric about 0, that is, $v_\ell^{(\delta)} = -v_{K_\delta-1-\ell}^{(\delta)}$ for all $\ell = 0, \dots, K_\delta - 1$.

We would now like to write a three-dimensional quadrature rule based on a Cartesian product of three of the one-dimensional quadrature rules (3.4). Denote by $v_{\mu_\delta}^{(\delta)}$ and $q_{\mu_\delta}^{(\delta)}$, $\mu_\delta = 0, \dots, K_\delta - 1$, the nodes and weights in the x_δ -direction in (3.4) for $\delta = 1, 2, 3$. We can now write the quadrature rule for a function $p(\mathbf{v}) = p(v^{(1)}, v^{(2)}, v^{(3)})$ as

$$\int_{\mathbb{R}} \int_{\mathbb{R}} \int_{\mathbb{R}} M^{(1)}(v^{(1)})M^{(2)}(v^{(2)})M^{(3)}(v^{(3)})p(v^{(1)}, v^{(2)}, v^{(3)})dv^{(1)}dv^{(2)}dv^{(3)} \\ \approx \sum_{\mu_1=0}^{K_1-1} \sum_{\mu_2=0}^{K_2-1} \sum_{\mu_3=0}^{K_3-1} q_{\mu_1}^{(1)}q_{\mu_2}^{(2)}q_{\mu_3}^{(3)}(M^{(1)}(v_{\mu_1}^{(1)}))^2(M^{(2)}(v_{\mu_2}^{(2)}))^2(M^{(3)}(v_{\mu_3}^{(3)}))^2p(v_{\mu_1}^{(1)}, v_{\mu_2}^{(2)}, v_{\mu_3}^{(3)}). \tag{3.5}$$

We would like to define a more compact notation for both the nodes and weights of this quadrature rule. Therefore, define the nodes $\mathbf{v}_k \in \mathbb{R}^3$ as $\mathbf{v}_k = (v_{k_1}^{(1)}, v_{k_2}^{(2)}, v_{k_3}^{(3)})$ for $k_\delta = 0, \dots, K_\delta - 1$, $\delta = 1, 2, 3$. The index k then must range from 0 to $K - 1$ with $K := K_1K_2K_3$. As notation for this transformation, introduce a mapping $\pi(\cdot)$ that maps the set $\{0, \dots, K - 1\}$ to $\{0, \dots, K_1 - 1\} \times \{0, \dots, K_2 - 1\} \times \{0, \dots, K_3 - 1\}$ such that $(k_1, k_2, k_3) = \pi(k)$. We will also write component-wise $k_1 = \pi_1(k)$, $k_2 = \pi_2(k)$, and $k_3 = \pi_3(k)$, i.e., $\pi = (\pi_1, \pi_2, \pi_3)$. Since the mapping is one-to-one, an inverse mapping $\pi^{-1}(\cdot, \cdot, \cdot)$ exists, and k can be computed from (k_1, k_2, k_3) as $k = \pi^{-1}(k_1, k_2, k_3)$. The mapping π^{-1} is explicitly defined as $k = \pi^{-1}(k_1, k_2, k_3) = k_1 + K_1k_2 + K_1K_2k_3$.

We can now write the quadrature rule (3.5) for a function $p(\mathbf{v})$ more compactly as

$$\int_{\mathbb{R}^3} M(\mathbf{v})p(\mathbf{v})d\mathbf{v} \approx \sum_{\mu=0}^{K-1} q_\mu (M(\mathbf{v}_\mu))^2 p(\mathbf{v}_\mu) \tag{3.6}$$

if we also define $q_\mu = q_{\mu_1}^{(1)}q_{\mu_2}^{(2)}q_{\mu_3}^{(3)}$ with $(\mu_1, \mu_2, \mu_3) = \pi(\mu)$. This quadrature is exact for functions $p(\mathbf{v})$ that are products of polynomials in each component $v^{(\delta)}$ up to degree $2K_\delta - 1$ for $\delta = 1, 2, 3$. Similarly, we can now write the basis functions $\psi_k(\mathbf{v}) := M(\mathbf{v})H_k(\mathbf{v})$ for $k = 0, \dots, K - 1$ with $M(\mathbf{v}) = M^{(1)}(v^{(1)})M^{(2)}(v^{(2)})M^{(3)}(v^{(3)})$ and $H_k(\mathbf{v}) := H_{k_1}^{(1)}(v^{(1)})H_{k_2}^{(2)}(v^{(2)})H_{k_3}^{(3)}(v^{(3)})$

for $k_\delta = 0, \dots, K_{\delta-1}$, $\delta = 1, 2, 3$. By construction, the basis functions are orthonormal with respect to the inner product $\langle \cdot, \cdot \rangle_G$.

To obtain an approximation of (3.1) by the spectral Galerkin ansatz, replace $f(\mathbf{x}, \mathbf{v}, t)$ by $f_K(\mathbf{x}, \mathbf{v}, t) = \sum_{\ell=0}^{K-1} \tilde{f}_\ell(\mathbf{x}, t) \psi_\ell(\mathbf{v})$ and test against $\psi_k(\mathbf{v})$ with respect to the inner product $\langle \cdot, \cdot \rangle_G$ to obtain

$$\begin{aligned} \frac{\partial \tilde{f}_k}{\partial t} + \sum_{\ell=0}^{K-1} \langle v^{(1)} \psi_\ell, \psi_k \rangle_G \frac{\partial \tilde{f}_\ell}{\partial x_1} + \sum_{\ell=0}^{K-1} \langle v^{(2)} \psi_\ell, \psi_k \rangle_G \frac{\partial \tilde{f}_\ell}{\partial x_2} \\ + \sum_{\ell=0}^{K-1} \langle v^{(3)} \psi_\ell, \psi_k \rangle_G \frac{\partial \tilde{f}_\ell}{\partial x_3} = \frac{1}{\text{Kn}} \sum_{\ell=0}^{K-1} \langle Q(\psi_\ell), \psi_k \rangle_G \tilde{f}_\ell, \end{aligned}$$

where the orthonormality of the basis functions has already been used. Introducing the coefficient matrices $\tilde{A}^{(\delta)}, \tilde{B} \in \mathbb{R}^{K \times K}$ for $\delta = 1, 2, 3$ with components of the coefficient matrices $\tilde{A}_{k\ell}^{(\delta)} := \langle v^{(\delta)} \psi_\ell, \psi_k \rangle_G$ and $\tilde{B}_{k\ell} := \langle Q(\psi_\ell), \psi_k \rangle_G$, we obtain the following system of conservation laws

$$\frac{\partial \tilde{F}}{\partial t} + \tilde{A}^{(1)} \frac{\partial \tilde{F}}{\partial x_1} + \tilde{A}^{(2)} \frac{\partial \tilde{F}}{\partial x_2} + \tilde{A}^{(3)} \frac{\partial \tilde{F}}{\partial x_3} = \frac{1}{\text{Kn}} \tilde{B} \tilde{F}, \quad \mathbf{x} \in \Omega, \quad t > 0 \quad (3.7)$$

for the vector of coefficient functions $\tilde{F}(\mathbf{x}, t) := (\tilde{f}_0(\mathbf{x}, t), \dots, \tilde{f}_{K-1}(\mathbf{x}, t))^T$. Observe that (3.7) is a first-order hyperbolic system of conservation laws where one must prescribe boundary conditions on that part of $\partial\Omega$ that comprises the inflow boundary. The fact that the constant coefficient matrices $\tilde{A}^{(\delta)}$ are not diagonal makes it difficult to identify which part of $\partial\Omega$ constitutes the inflow boundary for each \tilde{f}_k .

In [22], the choice of a transformation matrix is based on the insight that the summation in the quadrature rule $\sum_{\mu=0}^{K-1} q_\mu \psi_k(\mathbf{v}_\mu) \psi_\ell(\mathbf{v}_\mu)$ can be split into “equal” parts to define an orthogonal matrix $P = (P_{k\ell})$ that is shown to simultaneously transform *all* coefficient matrices $\tilde{A}^{(\delta)}$, $\delta = 1, 2, 3$, to diagonal ones.

Lemma 1. Let $A_{k\ell}^{(\delta)} = v_{k_\delta}^{(\delta)} \delta_{k\ell}$, $P_{k\ell} = \psi_k(\mathbf{v}_\ell) \sqrt{q_\ell}$, $(P^T)_{k\ell} = \psi_\ell(\mathbf{v}_k) \sqrt{q_k}$ for $k, \ell = 0, \dots, K-1$. Then $PP^T = I$ and $\tilde{A}^{(\delta)}$ can be simultaneously diagonalized as $\tilde{A}^{(\delta)} = PA^{(\delta)}P^T$ for all $\delta = 1, 2, 3$.

Applying this transformation P to (3.7) results in the transformation of the system for the transformed unknown $\tilde{\tilde{F}}(\mathbf{x}, t) := P^T \tilde{F}(\mathbf{x}, t)$. A straightforward computation shows that

$$\tilde{f}_k(\mathbf{x}, t) = \sum_{\mu=0}^{K-1} (P^T)_{k\mu} \tilde{f}_\mu(\mathbf{x}, t) = \sum_{\mu=0}^{K-1} \sqrt{q_k} \psi_\mu(\mathbf{v}_k) \tilde{f}_\mu(\mathbf{x}, t) = \sqrt{q_k} f_K(\mathbf{x}, \mathbf{v}_k, t), \tag{3.8}$$

where $\tilde{f}_k(\mathbf{x}, t)$ is the k th element of the transformed vector of coefficient functions given by $\tilde{F}(\mathbf{x}, t) = (\tilde{f}_0(\mathbf{x}, t), \dots, \tilde{f}_{K-1}(\mathbf{x}, t))^T$. The fact that direct evaluation of the approximation $f_K(\mathbf{x}, \mathbf{v}_k, t)$ at one of the discrete velocities $\mathbf{v} = \mathbf{v}_k$ does not give us $\tilde{f}_k(\mathbf{x}, t)$ demonstrates that the application of the initial and boundary conditions could be made simpler. In the next section, we introduce a transformation which allows us to obtain the coefficient functions of a suitable expansion directly from the evaluation of $f_K(\mathbf{x}, \mathbf{v}, t)$ at the velocity vector \mathbf{v}_k without any extraneous factor.

Stability and asymptotic convergence of the numerical solution f_K obtained by the transformation described in Lemma 1 can be shown, which we quote here for later reference [22]. First, define the norm $\|f\|_G^2 := \langle f, f \rangle_G$ induced by the weighted L^2 -inner product (3.3) that depends on \mathbf{x} and t if $f = f(\mathbf{x}, \mathbf{v}, t)$. Then, also define the following $G(t)$ -norm

$$\|f\|_{G(t)}^2 := \int_{\Omega} \|f\|_G^2 d\mathbf{x} = \int_{\Omega} \int_{\mathbb{R}^3} f^2 \frac{d\mathbf{v}}{M(\mathbf{v})} d\mathbf{x}, \quad t \geq 0, \tag{3.9}$$

which is solely a function of time t if $f = f(\mathbf{x}, \mathbf{v}, t)$ [22].

Theorem 2. The solution of the system obtained by transforming (3.7) with initial condition and boundary conditions with (2.15)–(2.16) satisfies

$$\|f_K\|_{G(t)}^2 \leq B_{\text{ini}} + \int_0^t B_{\text{in}} d\tau \quad \text{for all } t \in [0, t_{\text{fin}}]. \tag{3.10}$$

The proof in [22] follows in the spirit of Boltzmann’s H -Theorem by bounding the discretized $\|f_K\|_{G(t)}$. A convergence result is also available [22].

Theorem 3. Let $f(\mathbf{x}, \mathbf{v}, t)$ denote the solution of (3.1) and $f_K(\mathbf{x}, \mathbf{v}, t)$ the solution of the system (3.7). If f is assumed to be sufficiently smooth, then the error between the two solutions satisfies

$$\|f - f_K\|_{G(t)} \rightarrow 0 \quad \text{as } K \rightarrow \infty. \tag{3.11}$$

The proof of this asymptotic convergence result is only available in the case when the reflecting boundaries are assumed orthogonal to a coordinate axis with specular reflection [22].

3.2. A Series Expansion Method Based on a Collocation Basis

In this section, we show that it is possible to choose alternative basis functions $\varphi_k(\mathbf{v})$ in a series expansion

$$f(\mathbf{x}, \mathbf{v}, t) \approx f_K(\mathbf{x}, \mathbf{v}, t) = \sum_{k=0}^{K-1} f_k(\mathbf{x}, t) \varphi_k(\mathbf{v}) \quad (3.12)$$

that lead directly to a diagonal system of conservation laws and explicit identification of the boundary conditions. To demonstrate that such a choice is feasible, consider again the system of conservation laws with non-diagonal coefficient matrices (3.7). It was shown in Lemma 1 that the non-diagonal coefficient matrices $\tilde{A}^{(\delta)}$ could be simultaneously diagonalized; however, the transformed unknowns $\tilde{f}_k(\mathbf{x}, t)$ in (3.8) could not be computed by direct evaluation of $f_K(\mathbf{x}, \mathbf{v}, t)$ at a particular \mathbf{v}_k . The following lemma holds, which shows that another choice of the transformation matrix P leads to the same diagonal matrices $A^{(\delta)}$ and the computation of the unknown coefficient functions $f_k(\mathbf{x}, t)$ in (3.12) by direct evaluation of $f_K(\mathbf{x}, \mathbf{v}, t)$ at the discrete velocity \mathbf{v}_k .

Lemma 4. Let $A_{k\ell}^{(\delta)} = v_{k\delta}^{(\delta)} \delta_{k\ell}$, $P_{k\ell} = q_\ell \psi_k(\mathbf{v}_\ell)$, $(P^{-1})_{k\ell} = \psi_\ell(\mathbf{v}_k)$ for $k, \ell = 0, \dots, K-1$. Then $PP^{-1} = I$ and $\tilde{A}^{(\delta)}$ can be simultaneously diagonalized as $\tilde{A}^{(\delta)} = PA^{(\delta)}P^{-1}$ for all $\delta = 1, 2, 3$.

Proof. Since the polynomial degree of the product $v^{(\delta)} H_{\ell\delta}^{(\delta)} (v^{(\delta)}) H_{k\delta}^{(\delta)} (v^{(\delta)})$ is never larger than $2K_\delta - 1$, the quadrature rule (3.6) is exact and the components of $\tilde{A}^{(\delta)}$ can be computed as

$$\begin{aligned} (PA^{(\delta)}P^{-1})_{k\ell} &= \sum_{\mu=0}^{K-1} \sum_{\nu=0}^{K-1} (q_\mu \psi_k(\mathbf{v}_\mu)) \left(v_{\mu\delta}^{(\delta)} \delta_{\mu\nu} \right) (\psi_\ell(\mathbf{v}_\nu)) \\ &= \sum_{\mu=0}^{K-1} q_\mu v_{\mu\delta}^{(\delta)} \psi_k(\mathbf{v}_\mu) \psi_\ell(\mathbf{v}_\mu) \\ &= \sum_{\mu=0}^{K-1} q_\mu (M(\mathbf{v}_\mu))^2 v_{\mu\delta}^{(\delta)} H_\ell(\mathbf{v}_\mu) H_k(\mathbf{v}_\mu) \\ &= \int_{\mathbb{R}^3} M(\mathbf{v}) v^{(\delta)} H_\ell(\mathbf{v}) H_k(\mathbf{v}) d\mathbf{v} \\ &= \int_{\mathbb{R}^3} v^{(\delta)} \psi_\ell(\mathbf{v}) \psi_k(\mathbf{v}) \frac{d\mathbf{v}}{M(\mathbf{v})} = \left\langle v^{(\delta)} \psi_\ell, \psi_k \right\rangle_G = \tilde{A}_{k\ell}^{(\delta)}. \end{aligned}$$

To verify that $PP^{-1} = I$, we compute, using (3.6),

$$\begin{aligned} (PP^{-1})_{k\ell} &= \sum_{\mu=0}^{K-1} q_{\mu} \psi_k(\mathbf{v}_{\mu}) \psi_{\ell}(\mathbf{v}_{\mu}) = \sum_{\mu=0}^{K-1} q_{\mu} (M(\mathbf{v}_{\mu}))^2 H_{\ell}(\mathbf{v}_{\mu}) H_k(\mathbf{v}_{\mu}) \\ &= \int_{\mathbb{R}^3} M(\mathbf{v}) H_{\ell}(\mathbf{v}) H_k(\mathbf{v}) d\mathbf{v} = \int_{\mathbb{R}^3} \psi_{\ell}(\mathbf{v}) \psi_k(\mathbf{v}) \frac{d\mathbf{v}}{M(\mathbf{v})} \\ &= \langle \psi_k, \psi_{\ell} \rangle_G = \delta_{k\ell}. \end{aligned}$$

□

If we now consider the transformed unknown $F(\mathbf{x}, t) := P^{-1} \tilde{F}(\mathbf{x}, t)$, direct computation shows

$$f_k(\mathbf{x}, t) = \sum_{\mu=0}^{K-1} (P^{-1})_{k\mu} \tilde{f}_{\mu}(\mathbf{x}, t) = \sum_{\mu=0}^{K-1} \tilde{f}_{\mu}(\mathbf{x}, t) \psi_{\mu}(\mathbf{v}_k) = f_K(\mathbf{x}, \mathbf{v}_k, t). \tag{3.13}$$

Observe that this is the analogous result to (3.8) without the scaling factor $\sqrt{q_k}$. Thus, the coefficient functions $f_k(\mathbf{x}, t)$ can be computed by evaluating $f_K(\mathbf{x}, \mathbf{v}_k, t)$ and the unknowns $f_k(\mathbf{x}, t)$ are exactly the unknowns in (3.12) we would like to solve for.

The fact that $f_k(\mathbf{x}, t) = f_K(\mathbf{x}, \mathbf{v}_k, t)$ dictates that we must construct basis functions with the property $\varphi_k(\mathbf{v}_{\mu}) = \delta_{k\mu}$. This condition implies that the basis functions $\varphi_k(\mathbf{v})$ are the collocation basis functions based on the Gaussian quadrature roots with respect to the Maxwellian weight function defined above. We construct the basis functions $\varphi_k(\mathbf{v})$ again as a product of a Maxwellian $M(\mathbf{v})$ and polynomials based on these roots, which ensures that $\text{span}\{\psi_k\} = \text{span}\{\varphi_k\}$. This guarantees that both methods produce the same numerical solution $f_K(\mathbf{x}, \mathbf{v}, t)$, that is, we have

$$f_K(\mathbf{x}, \mathbf{v}, t) = \sum_{\mu=0}^{K-1} \tilde{f}(\mathbf{x}, t) \psi_{\mu}(\mathbf{v}) = \sum_{\mu=0}^{K-1} f(\mathbf{x}, t) \varphi_{\mu}(\mathbf{v}). \tag{3.14}$$

This equivalence immediately guarantees stability and convergence from Theorems 2 and 3, respectively. Additionally, since the Gaussian quadrature is exact for all needed integrals, the basis functions have the following properties.

Lemma 5. For all $k, \ell = 0, \dots, K - 1$, the basis functions $\varphi_k(\mathbf{v})$ satisfy the properties $\langle \varphi_{\ell}, \varphi_k \rangle_G = q_k \delta_{k\ell}$ and $\langle v^{(\delta)} \varphi_{\ell}, \varphi_k \rangle_G = q_k v_{k\delta}^{(\delta)} \delta_{k\ell}$ for $\delta = 1, 2, 3$.

To obtain an approximation of (3.1) by the spectral Galerkin ansatz, insert the approximation $f_K(\mathbf{x}, \mathbf{v}, t) = \sum_{\ell=0}^{K-1} f_\ell(\mathbf{x}, t)\varphi_\ell(\mathbf{v})$ into (3.1) and test against $\varphi_k(\mathbf{v})$ with respect to the inner product $\langle \cdot, \cdot \rangle_G$ to obtain

$$\begin{aligned} \langle \varphi_k, \varphi_k \rangle_G \frac{\partial f_k}{\partial t} + \sum_{\ell=0}^{K-1} \langle v^{(1)}\varphi_\ell, \varphi_k \rangle_G \frac{\partial f_\ell}{\partial x_1} + \sum_{\ell=0}^{K-1} \langle v^{(2)}\varphi_\ell, \varphi_k \rangle_G \frac{\partial f_\ell}{\partial x_2} \\ + \sum_{\ell=0}^{K-1} \langle v^{(3)}\varphi_\ell, \varphi_k \rangle_G \frac{\partial f_\ell}{\partial x_3} = \frac{1}{\text{Kn}} \sum_{\ell=0}^{K-1} \langle Q(\varphi_\ell), \varphi_k \rangle_G f_\ell. \end{aligned}$$

Introducing the coefficient matrices $A^{(\delta)} \in \mathbb{R}^{K \times K}$, $\delta = 1, 2, 3$, and $B \in \mathbb{R}^{K \times K}$ with components $A_{k\ell}^{(\delta)} := \langle v^{(\delta)}\varphi_\ell, \varphi_k \rangle_G / q_k = v_{k\delta}^{(\delta)} \delta_{k\ell}$, $\delta = 1, 2, 3$, and $B_{k\ell} := \langle Q(\varphi_\ell), \varphi_k \rangle_G / q_k$, we obtain the following system of linear hyperbolic partial differential equations

$$\frac{\partial F}{\partial t} + A^{(1)} \frac{\partial F}{\partial x_1} + A^{(2)} \frac{\partial F}{\partial x_2} + A^{(3)} \frac{\partial F}{\partial x_3} = \frac{1}{\text{Kn}} BF, \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (3.15)$$

for the vector of coefficient functions $F(\mathbf{x}, t) := (f_0(\mathbf{x}, t), \dots, f_{K-1}(\mathbf{x}, t))^T$. Observe that the matrices $A^{(\delta)}$ are exactly those that we would have obtained through the diagonalization procedure in Lemma 1. Since the coefficient matrices in (3.15) are diagonal and constant allows us in turn to rewrite the system in conservation form

$$\frac{\partial f_k}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{a}_k f_k) = \frac{1}{\text{Kn}} b_k, \quad k = 0, \dots, K - 1, \quad (3.16)$$

with constant vectors $\mathbf{a}_k := (A_{kk}^{(1)}, A_{kk}^{(2)}, A_{kk}^{(3)})^T = (v_{k1}^{(1)}, v_{k2}^{(2)}, v_{k3}^{(3)})^T = \mathbf{v}_k$ for all $k = 0, \dots, K - 1$ and the right-hand side functions $b_k(\mathbf{x}, t) := \sum_{\ell=0}^{K-1} B_{k\ell}(\mathbf{x}, t) f_\ell(\mathbf{x}, t)$.

Another advantage of $\varphi_k(\mathbf{v})$ is that they lead directly to the appropriate treatment of the boundary conditions. Each hyperbolic partial differential equation for $f_k(\mathbf{x}, t)$ in (3.15) or (3.16) needs to be supplied with boundary conditions on that part of $\partial\Omega$ that constitutes its inflow boundary $\Gamma_k^- := \{\mathbf{x} \in \partial\Omega : \mathbf{n}(\mathbf{x}) \cdot \mathbf{a}_k < 0\}$ with \mathbf{a}_k in (3.16) and where $\mathbf{n}(\mathbf{x})$ denotes the unit outward normal vector at $\mathbf{x} \in \partial\Omega$. Consider a generic kinetic boundary condition at $\mathbf{x} \in \partial\Omega$ for the inflow part of the density function $f(\mathbf{x}, \mathbf{v}, t)$, that is,

$$f(\mathbf{x}, \mathbf{v}, t) = f^-(\mathbf{x}, \mathbf{v}, t), \quad \mathbf{x} \in \partial\Omega, \quad \mathbf{n} \cdot \mathbf{v} < 0, \quad (3.17)$$

with a given function $f^-(\mathbf{x}, \mathbf{v}, t)$. Using the fact that $\varphi_\ell(\mathbf{v}_k) = \delta_{k\ell}$, we have $f(\mathbf{x}, \mathbf{v}_k, t) = f_k(\mathbf{x}, t)$, and we obtain by letting $\mathbf{v} = \mathbf{v}_k$ in (3.17)

$$f_k(\mathbf{x}, t) = f^-(\mathbf{x}, \mathbf{v}_k, t), \quad \mathbf{x} \in \partial\Omega, \quad \mathbf{n} \cdot \mathbf{v}_k < 0. \quad (3.18)$$

Since $\mathbf{a}_k = \mathbf{v}_k$ for all $k = 0, \dots, K - 1$, this condition is exactly the desired condition at the inflow boundary Γ_k^- . That is, for each equation in (3.16), we have the boundary condition

$$f_k(\mathbf{x}, t) = f^-(\mathbf{x}, \mathbf{v}_k, t), \quad \mathbf{x} \in \Gamma_k^-, \quad (3.19)$$

given exactly on the inflow part of $\Gamma_k^- \subset \partial\Omega$ defined above.

3.3. Spatial Discretization and Parallel Implementation

Considering either the system (3.15) or the conservative form (3.16), the problem is now posed in standard form amenable for numerical computations. However, due to the large size K of each system, the irregular shape of the spatial domain $\Omega \subset \mathbb{R}^3$, and the requirement to compute for long times in transient simulations for realistic application examples, they still pose a formidable challenge. We choose to use the discontinuous Galerkin method (DGM) [8], implemented in the code DG [20], applied to the conservation form (3.16) to solve each system, because it is perfectly suited for the resolution of problems in conservation form on domains with potentially very irregular shape. In the studies here, we use linear polynomials on each element. This work does not focus on the time discretization and explicit Euler time-stepping is used at present.

For the parallel computations, the domain Ω is partitioned in a pre-processing step, and each parallel processor is assigned one subdomain. Certain pairs of disjoint subdomains will share edges for two-dimensional domains and faces for three-dimensional domains, and the processors on which they reside will have to communicate with each other at every time step to pass information on fluxes across their shared element edges or faces, respectively. Since we wanted to focus on the performance of the parallel implementation, we do not use automatic mesh refinement and coarsening at present, which would incur additional work and result in additional communications to redistribute elements for load balancing.

The domains in Fig. 1 are in fact relatively regular in shape, and we use quadrilateral elements in 2-D and brick elements in 3-D for them; however, DG also implements, for instance, tetragonal elements in 3-D; see [11] for an example. For our convergence studies and parallel performance studies, the shape of the elements is of minor relevance. In 2-D, the degrees of freedom are the values of the K solution components $f_k^{(i)}(x, t)$, $k = 0, \dots, K - 1$, for each reactive species $i = 1, \dots, n_s$ on all four vertices of each of the N_e quadrilaterals; hence the complexity of the computational problem is given by $4N_e n_s K$ degrees of freedom. In 3-D, we use brick elements with eight vertices, resulting in $8N_e n_s K$ degrees of freedom.

Hence, the complexity of the computational problem is proportional to the number of elements N_e , to the number of species n_s , and to the system size K .

4. NUMERICAL RESULTS

All computational results presented are for single-species simulations ($n_s = 1$) of chemical vapor deposition (CVD) in both two and three spatial dimensions on the computational domains of Fig. 1 with the width of the feature mouth $L = 0.25 \mu\text{m}$ and an aspect ratio $A = 3$. The problem is given by (1.1)–(1.2) with $i = 1$, and we drop the species index in all variables $f^{(i)}$, $M^{(i)}$, σ_i , etc. We use a relaxation time approximation for the collision operator by choosing $\sigma = 1/\tau$ with relaxation time $\tau = 1$ in a dimensionless single-species model [11, 21, 23, 24]. Section 4.1 demonstrates the capabilities of the KTRM and its implementation for an application example both in two and three dimensions. For both choices of the spatial dimension, the evolution over time of two quantities is shown: the dimensionless concentration $c(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}$ as a function of $\mathbf{x} \in \Omega$ and the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ as a function of \mathbf{v} at a selected point $\mathbf{x} \in \Omega$. The application example has operating conditions in the transition regime with $\text{Kn} = 1.0$ that justifies the use of a kinetic model. Additional results for different Kn are available [24]. Section 4.2 demonstrates the numerical stability and convergence for the velocity discretization of our method for a wide range of Kn from 0.01 to ∞ in two and three dimensions. These studies are important, because they demonstrate that the asymptotic convergence predicted by the analytical results is attainable for reasonable resolutions. Finally, Section 4.3 contains parallel performance studies for the implementation of our numerical method. The studies use $\text{Kn} = \infty$, which is the most conservative case, as no calculations are required for the right-hand side in (1.1).

4.1. Application Results

The purpose of CVD is to deposit material in features: either a thin film (liner or layer), or to fill them. This is accomplished by feeding a reactive gas stream over the surface of the silicon wafer. Specifically, we simulate just the initial, transient layer of deposition, as opposed to feature fill of the entire trench. Such a simulation is particularly relevant to processes designed to deposit thin films of material (a few tens of nanometers thick in practice).

We assume that no reactive gas is present in the feature initially: $f(\mathbf{x}, \mathbf{v}, 0) = 0$. The reactive gas is then fed into the top of the domain

with a Maxwellian distribution $M(\mathbf{v})$. When the gas reaches the wafer surface, a fraction of the molecules “stick” to the surface and react with the silicon surface. The fraction of the molecules that stick to the wafer surface is given by the sticking factor γ_0 , which can range from 0 (all particles re-emit) to 1 (all particles stick to the wafer surface). We model the re-emission of the remaining molecules as a Maxwellian distribution, as given by the wafer surface boundary condition (2.8). The sticking factor is chosen as $\gamma_0 = 0.01$, which is a typical values for processes under consideration [3]. This choice indicates that approximately 1% of the gas molecules stick and react with the wafer surface; the remaining molecules are re-emitted with a Maxwellian velocity distribution.

4.1.1. Two-Dimensional Application Studies

In this subsection, we report results for the evolution of dimensionless concentration and kinetic density for the 2-D/2-D simulations in the domain sketched in Fig. 1(a). Figure 3 shows the dimensionless concentration of the reactive species for $\text{Kn} = 1.0$ at six (re-dimensionalized) points in time. In Fig. 3(a), the dimensionless concentration is shown for the initial state $t = 0$ ns. Since none of the reactive gas is present, $c = 0$ everywhere. Figure 3(b, c) shows that after $t = 1$ ns the gas has already reached the feature and after $t = 5$ ns the gas has reached the bottom of the feature, respectively. We also see an increase in concentration just above the flat areas of the wafer surface, which is due to re-emission of molecules from the surface. The feature fill is relatively rapid in this regime, but the concentration profile is not very smooth; in particular, the jumps at the sharp corners in Figure 3 are expected in this regime [2]. Figure 3(d, e) shows the further increase in concentration through $t = 20$ ns. Finally, Figure 3(f) shows that at approximately $t = 40$ ns the feature is nearly filled with the reactive species.

One of the advantages of deterministically solving the Boltzmann equation is the direct access to the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$. In addition to being able to compute $c(\mathbf{x}, t)$ as a function of \mathbf{x} , this also allows for the ability to analyze $f(\mathbf{x}, \mathbf{v}, t)$ as a function of \mathbf{v} at a chosen point $\mathbf{x} \in \Omega$. We choose the point $\mathbf{x} = (0.0, 0.0)$ at the center of the mouth of the feature to avoid the effect of re-emissions from the *flat areas* of the wafer surface, since we are particularly interested in understanding the directionality of the flow observed in the concentration results.

Figure 4 shows the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ for $\text{Kn} = 1.0$ as a function of $\mathbf{v} \in \mathbb{R}^2$ at the same times as the concentration plots in Fig. 3 with $\mathbf{x} = (0.0, 0.0)$ at the mouth of the feature. The plots in this figure are oriented analogously to the concentration plots in Fig. 3, such that flow

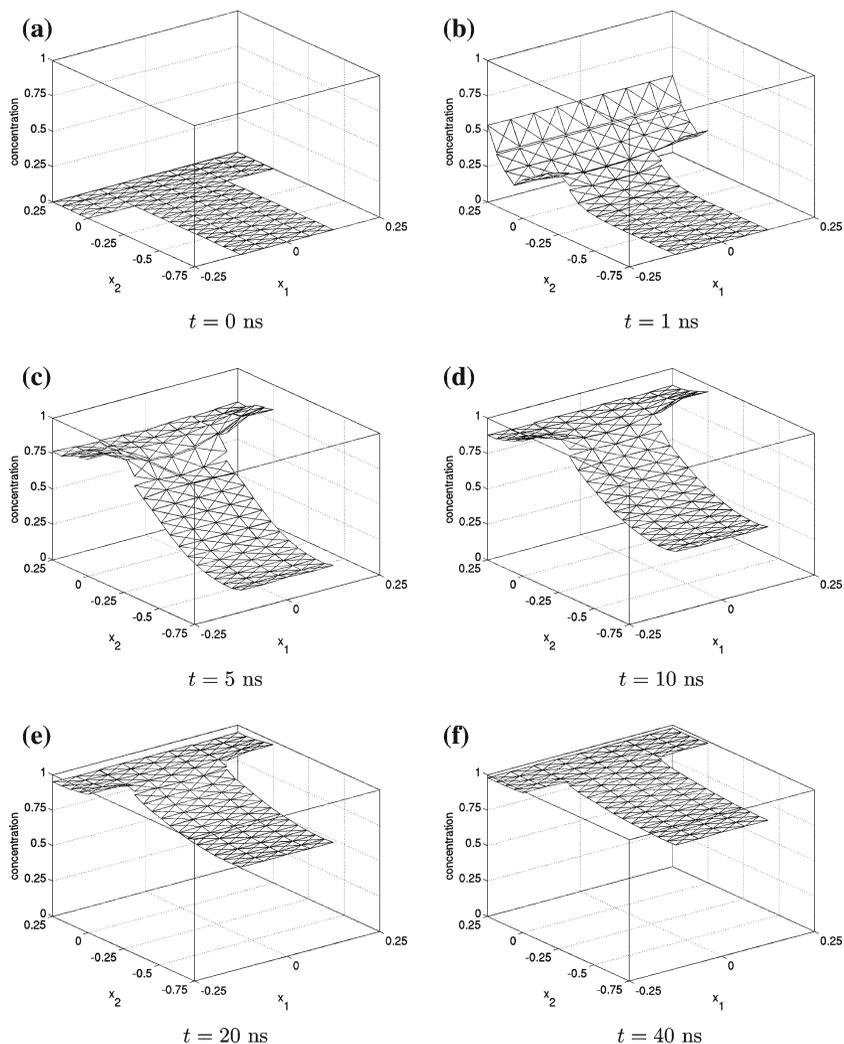


Fig. 3. Dimensionless concentration $c(\mathbf{x}, t)$ as function of $\mathbf{x} \in \Omega$ for $\text{Kn} = 1.0$ at selected (re-dimensionalized) times.

downwards into the feature, to the right in Fig. 3, corresponds to a kinetic density with larger values also on the right of a plot. Figure 4(a) confirms the initial condition $f = 0$. As Fig. 4(b) shows, molecules have already reached the position \mathbf{x} by $t = 1$ ns in this regime, but the kinetic density distribution is skewed in the negative v_2 -direction (the right side of the

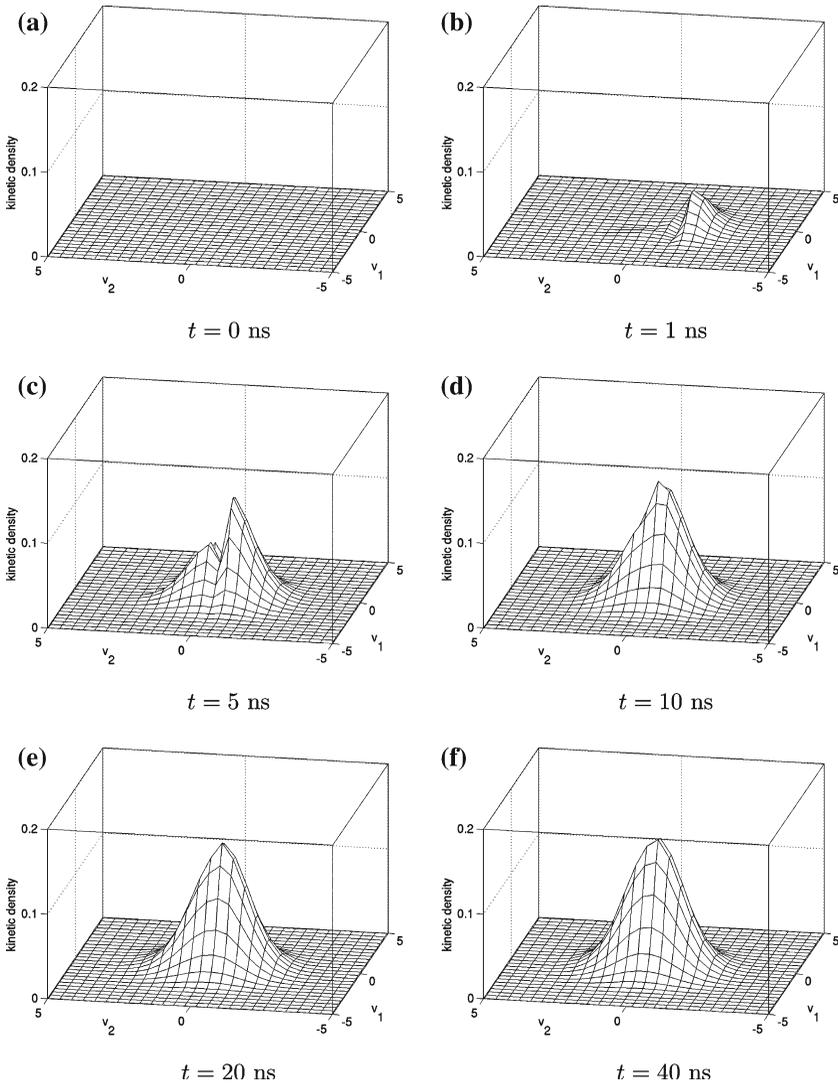


Fig. 4. Kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ as function of $\mathbf{v} \in \mathbb{R}^2$ for $\text{Kn} = 1.0$ at selected (re-dimensionalized) times. Coordinates of spatial point at the feature mouth: $\mathbf{x} = (0.0, 0.0)$.

plot). This indicates that the flow is directional. This is due to the fact that the molecules have not been re-emitted from the wafer surface and have not been randomized by collisions yet in this regime. In Figure 4(c), one sees a continued increase in the kinetic density for $v_2 < 0$ (the right side

of the plot), however, one also sees an increase in density in the $v_2 > 0$ direction (the left side) due to re-emission from the surface. An examination of the concentration in Fig. 3 shows that the gas reaches the mean wafer surface after only a few nanoseconds. Thus, re-emission in the positive v_2 -direction is seen and, hence, an increase in the corresponding components of the kinetic density. Observe in Fig. 4(d–f) that the density becomes *apparently* uniform about the origin due to collisions and re-emission with a Maxwellian velocity distribution; actually, the density is *not* uniform, but an inspection of the individual numbers reveals that its upward velocity components are slightly smaller than the downward components due to the deposition of material at the wafer surface.

In summary, the flow for $\text{Kn} = 1.0$ gives rapid feature fill, yet the gas profile is relatively rough. This corresponds to a less collisional flow in the transition regime. The directionality of the flow is clearly visible in the plots of the kinetic density f and explains our interest in directly accessing $f(\mathbf{x}, \mathbf{v}, t)$ as function of \mathbf{v} .

4.1.2. Three-Dimensional Application Studies

In this subsection, we report results for the evolution of dimensionless concentration and kinetic density for the 3-D/3-D simulations in the domain sketched in Fig. 1(b). Figure 5 shows slice plots of the dimensionless concentration of the reactive species for $\text{Kn} = 1.0$ at several (re-dimensionalized) points in time. The slices are four horizontal cross-sections of the domain Ω at heights $x_3 = -0.75, -0.50, -0.25,$ and 0.00 . The shapes of the three lower slices indicate the shape of the trench corner domain, while the top slice at $x_3 = 0.0$ includes the flat areas of the wafer surface. The different shades of gray indicate the value of the dimensionless concentration $0 \leq c(\mathbf{x}, t) \leq 1$. The lightest shade of gray corresponds to $c = 0$ and the darkest shade indicates $c = 1$. We do not show the initial concentration $c = 0$ at $t = 0$ ns. Figure 5(a) shows that the concentration has started reaching the flat parts of the wafer surface at $x_3 = 0.0$ already after 1 ns. Examining Fig. 5(b), one sees that a high concentration on the flat parts of the wafer surface has already been attained after only 2 ns. Less concentration is still seen above the mouth of the L-shaped trench, as the molecules continue to travel down into the feature there. But after just 5 ns, some molecules have reached the bottom of the feature in Fig. 5(c). Figure 5(d, f) shows the evolution of the concentration up to $t = 40$ ns, at which point all slices show significant levels of concentration. The short transport time is a function of the less collisional flow in this regime.

We now study the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ in three dimensions. Figure 6 shows the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ in the transition regime for

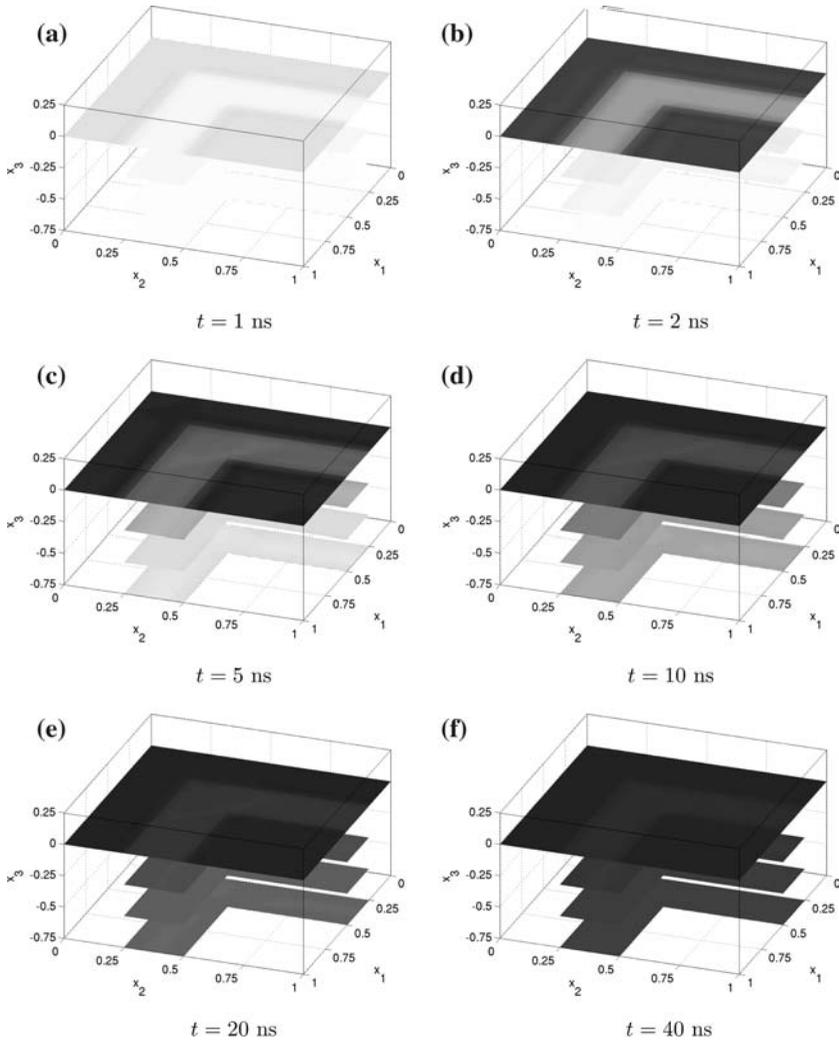


Fig. 5. Slice plots of the dimensionless concentration $c(\mathbf{x}, t)$ as function of $\mathbf{x} \in \Omega$ for $\text{Kn} = 1.0$ at selected (re-dimensionalized) times. The horizontal slices are at heights $x_3 = -0.75, -0.50, -0.25,$ and 0.00 . Gray scale from light color for $c=0$ to dark color for $c=1$.

$\text{Kn}=1.0$ as a function of $\mathbf{v} \in \mathbb{R}^3$ with \mathbf{x} at the mouth of the feature in the corner with the physical point $\mathbf{x} = (0.375, 0.375, 0.0)$ at the same times as in Fig. 5. Figure 6 shows isosurface plots of the kinetic density at $f^* = 0.005$ at several (re-dimensionalized) points in time. That is, the shape in each plot shows all points of the velocity domain with $f(\mathbf{x}, \mathbf{v}, t) = f^*$, up

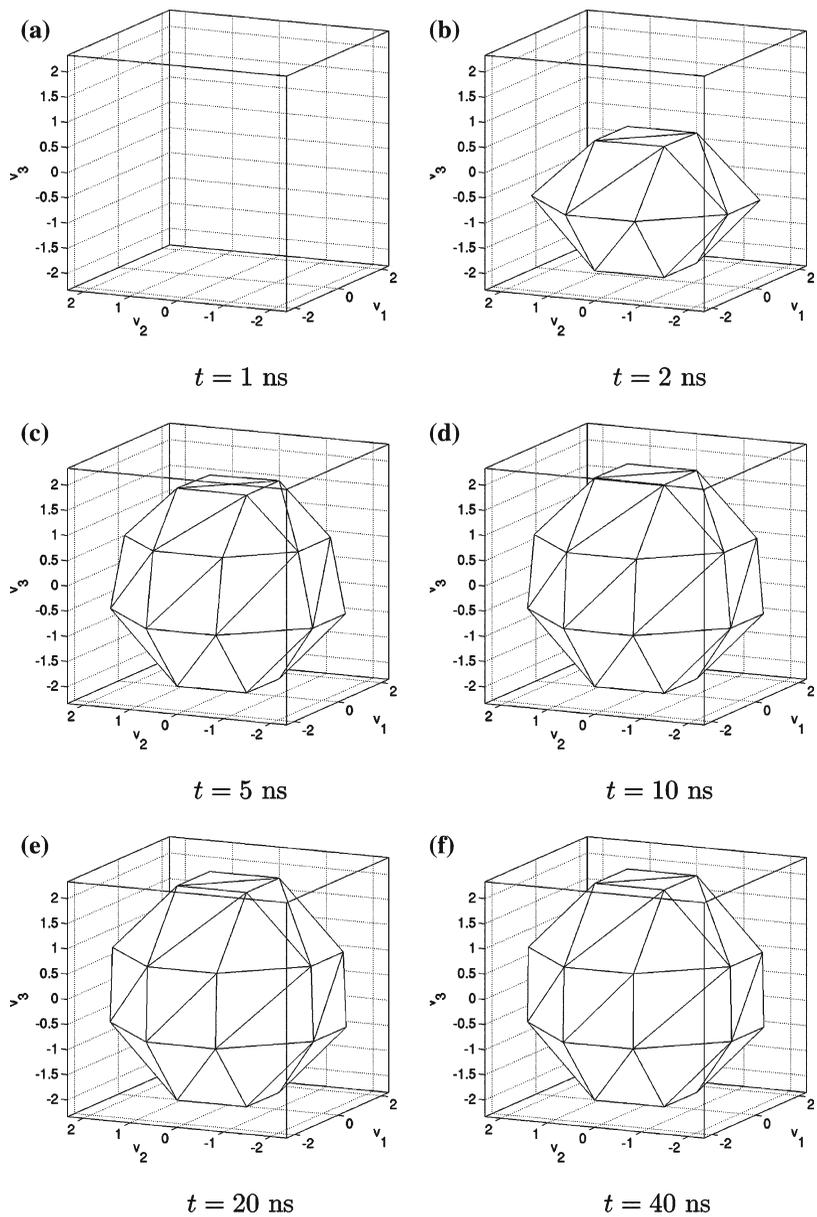


Fig. 6. Isosurface plots of the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ as function of $\mathbf{v} \in \mathbb{R}^3$ for $\text{Kn}=1.0$ at selected (re-dimensionalized) times. Coordinates of spatial point above the corner trench: $\mathbf{x} = (0.375, 0.375, 0.0)$.

to the resolution of the velocity discretization. The reference value $f^* = 0.005$ is selected such that a Maxwellian distribution results in an isosurface that fits in the axis limits in the plots in Fig. 6 given by the discrete velocities. Figure 6(a) shows an empty plot, which indicates that by $t = 1$ ns, the kinetic density $f(\mathbf{x}, \mathbf{v}, t)$ at this position \mathbf{x} is still strictly less than f^* in all velocity components. This agrees with Fig. 5(a) that indicates a very low concentration at the mouth of the feature. Figure 6(b) shows that by $t = 2$ ns molecules with downward velocity components have reached the point \mathbf{x} , indicated by $f > f^*$ for $v_3 < 0$ components. Observe that the density is skewed heavily in the negative v_3 -direction. This is due to the fact that the flow is downward into the feature. In Figure 6(c) at $t = 5$ ns, f has now attained values above f^* also in the components for $v_3 > 0$, but the flow continues to show directionality indicated by the slightly angled sides of the isosurface shape. As Fig. 6(c)–(f) shows, this directionality becomes less pronounced over time, but does persist in this regime up to $t = 40$ ns.

These 3-D/3-D results confirm that the flow is less collisional for Knudsen numbers in the transition regime. The plots of the kinetic density show that significant fractions of the incoming reactive species get to the bottom of the feature relatively rapidly. While this is clearly visible in the plots of the kinetic density in Fig. 6, notice that the concentration plots in Fig. 5 do not reveal this. This again explains our interest in directly accessing the kinetic density as a function of velocity.

4.2. Numerical Convergence of the Spectral Galerkin Method

In this section, we present two- and three-dimensional numerical stability and convergence demonstrations for our numerical method. Its asymptotic stability and convergence are guaranteed by theorems in [22] as the number of expansion coefficients $K \rightarrow \infty$. But the computational studies in [22] only cover one spatial dimension; our studies extend the results to higher dimensions. This is important to establish that reasonable values of finite K give acceptable results.

As the true solution f in Theorem 3 is unknown, we use instead $f := f_K(\mathbf{x}, \mathbf{v}, t)$, where $K = 32 \times 32 = 1024$ is the finest solution in velocity space that we obtained for the two-dimensional case. The spatial mesh with mesh size $h = 0.03125$ and the constant time step $\Delta t = 10^{-4}$ were chosen such that the velocity error dominates [24]. The simulations were run for 10 ns starting from an initial condition f_{ini} with $c^{\text{ini}} = 0.9$ in (2.10) for all $\mathbf{x} \in \Omega$.

Table I displays numerical stability and convergence for our method with $\text{Kn} = 0.01, 1.0, 100.0$, and ∞ . Column 1 shows the level of velocity

Table I. Demonstration of Numerical Stability and Convergence for the Spectral Galerkin Discretization of 2-D Velocity Space for $\text{Kn}=0.01, 1.0, 100.0,$ and ∞

K	$\ f_K\ _{G(t)}$	$\ f\ _{G(t)} - \ f_K\ _{G(t)}$	$\frac{\ f\ _{G(t)} - \ f_{K/4}\ _{G(t)}}{\ f\ _{G(t)} - \ f_K\ _{G(t)}}$	$\ f - f_K\ _{G(t)}$	$\frac{\ f - f_{K/4}\ _{G(t)}}{\ f - f_K\ _{G(t)}}$
Kn=0.01					
4	0.47861	6.86e-03	N/A	8.73e-03	N/A
16	0.48252	2.95e-03	2.3254	3.69e-03	2.3676
64	0.48425	1.22e-03	2.4180	1.51e-03	2.4364
256	0.48508	3.90e-04	3.1282	4.94e-04	3.0640
Kn=1.0					
4	0.52021	5.28e-03	N/A	9.31e-03	N/A
16	0.52299	2.51e-03	2.1036	4.70e-03	1.9813
64	0.52455	9.20e-04	2.7283	4.63e-03	1.0144
256	0.52522	2.50e-04	3.6801	2.88e-03	1.6052
Kn=100.0					
4	0.52303	6.20e-03	N/A	1.11e-02	N/A
16	0.52615	3.10e-03	2.0162	5.73e-03	1.9470
64	0.52801	1.20e-03	2.5371	5.63e-03	1.0177
256	0.52887	3.50e-04	3.4571	4.33e-03	1.3026
Kn= ∞					
4	0.52306	6.20e-03	N/A	1.12e-02	N/A
16	0.52599	3.27e-03	1.8960	8.78e-03	1.2738
64	0.52805	1.21e-03	2.7025	6.63e-03	1.3227
256	0.52891	3.50e-04	3.4571	4.34e-03	1.5261

discretization ranging from $K = 2 \times 2 = 4$ to $K = 16 \times 16 = 256$. The second column contains the $G(t)$ -norm of the numerical solution $\|f_K\|_{G(t)}$, which converges to the $G(t)$ -norm of f , as predicted by Theorem 2, where $\|f\|_{G(t)} = 0.48547, 0.52547, 0.52922,$ and 0.52926 for $\text{Kn}=0.01, 1.0, 100.0,$ and ∞ , respectively. Column 3 contains the error between the norm of f and the norms of f_K . The decreasing errors indicate that the norms of the numerical solution f_K are converging to the norm of f . The fourth column contains the ratio of successive errors between these norms. The fifth column shows the error between f and the numerical solution f_K in the $G(t)$ -norm. As K increases, the error $\|f - f_K\|_{G(t)}$ decreases and demonstrates numerical convergence of our method in all regimes, as predicted by Theorem 3. Notice for completeness that the theorem is only proved for specular reflection on $\partial\Omega_{\text{re}}$ [22], so our numerical studies using the reaction condition on the wafer surface $\Gamma_w \subset \partial\Omega_{\text{re}}$ constitute an extension beyond the theory. The last column contains the observed convergence rates for our method. A comparison of the values in column 6 indicates that the convergence is better in the near-hydrodynamic regime than in the other regimes.

We now present three-dimensional numerical stability and convergence results. As the true solution f is unknown, we define again $f := f_K(\mathbf{x}, \mathbf{v}, t)$, where $K = 8 \times 8 \times 8 = 512$ is the finest solution in velocity space that we obtained in three dimensions. The spatial mesh with mesh size $h = 0.0625$ and constant time step $\Delta t = 10^{-4}$ were chosen such that the velocity error dominates [24]. The simulations were run for 10 ns starting from an initial condition f_{ini} with $c^{\text{ini}} = 0.9$ in (2.10) for all $\mathbf{x} \in \Omega$.

Table II summarizes the stability and convergence results for the 3-D/3-D case. Column 1 shows the level of velocity discretization for $K = 2 \times 2 \times 2 = 8$ and $K = 4 \times 4 \times 4 = 64$. The second column contains the $G(t)$ -norm of the discrete solution f_K that approaches the $G(t)$ -norm of f , as predicted by Theorem 2, where $\|f\|_{G(t)} = 0.62035, 0.65907, 0.66249$, and 0.66314 for $\text{Kn} = 0.01, 1.0, 100.0$, and ∞ , respectively. Column 3 contains the error between the norms of f and f_K . The decreasing errors indicate that the norms of the numerical solution f_K are converging to the norm of f . The fourth column contains the ratio of successive errors between these norms. The fifth column shows the error between f and the numerical solution f_K in the $G(t)$ -norm. As K increases, the error $\|f - f_K\|_{G(t)}$ decreases and demonstrates convergence in all regimes, as predicted by Theorem 3. The last column contains the observed convergence rate for our method. A comparison suggests again that the order of convergence may be higher in the near-hydrodynamic regime.

These studies in two and three dimensions show that asymptotic convergence is achieved as $K \rightarrow \infty$ in agreement with Theorem 3 for a range

Table II. Demonstration of Numerical Stability and Convergence for the Spectral Galerkin Discretization of 3-D Velocity space for $\text{Kn} = 0.01, 1.0, 100.0$, and ∞

K	$\ f_K\ _{G(t)}$	$\ f\ _{G(t)} - \ f_K\ _{G(t)}$	$\frac{\ f\ _{G(t)} - \ f_{K/8}\ _{G(t)}}{\ f\ _{G(t)} - \ f_K\ _{G(t)}}$	$\ f - f_K\ _{G(t)}$	$\frac{\ f - f_{K/8}\ _{G(t)}}{\ f - f_K\ _{G(t)}}$
Kn = 0.01					
8	0.61551	4.84e-03	N/A	5.89e-03	N/A
64	0.61848	1.87e-03	2.5882	1.79e-03	3.2905
Kn = 1.0					
8	0.65713	1.94e-03	N/A	8.43e-03	N/A
64	0.65869	3.80e-04	5.1052	5.09e-03	1.6561
Kn = 100.0					
8	0.65953	2.96e-03	N/A	1.01e-02	N/A
64	0.66137	1.12e-03	2.6429	6.73e-03	1.5007
Kn = ∞					
8	0.65956	3.58e-03	N/A	1.01e-02	N/A
64	0.66142	1.72e-03	2.0814	6.75e-03	1.4963

of Knudsen numbers $0.01 \leq \text{Kn} \leq \infty$ and that results obtained for finite resolutions K that are attainable in practice are reliable.

4.3. Parallel Performance Studies

In this section, we present results from parallel performance studies for the KTRM using the DGM code DG on a 64-processor Beowulf cluster, in extension of earlier demonstrations on a cluster with 8 processors [26]. Specifically, the machine is an IBM 1350 cluster arranged as 32 dual-processor nodes with 2.0 GHz Intel Xeon (512 kB L2 cache) chips and 1 GB of memory per node. The nodes are connected by a high-performance Myrinet interconnect, and files are served by a fast ethernet. Communication among nodes is accomplished using the Message-Passing Interface (MPI) standard.

The motivation for parallel computing is that several processors working on a problem (with fixed size) should be able to solve the problem faster than a single processor, because the calculation operations in the algorithm are distributed across the processors and executed in parallel. That is, ideally, a computation using p processors should be p times as fast as a computation on 1 processor. But for algorithms, such as DGM, that require information exchange between the processors (fluxes between finite elements at boundaries of subdomains; see Section 3.3), using more processors necessitates additional communication operations. Therefore, using more processors simultaneously causes decreasing calculation time and increasing communication time, leading to a point of diminishing returns; discovering this point is the purpose of parallel performance studies in practice.

Since both calculations and communications are an inherent part of parallel computing, both types of operations must be included in timings of parallel programs, which are therefore based on wall clock time (and not CPU time or similar). We compute the wall clock time of a simulation as the difference between time stamps assigned by the operating system to the first and last output file created during the simulation; this is the most *pessimistic* timing measure possible, because in addition to calculations and communications of our code, it includes any other delays associated with the operating system and, for instance, writing of output files. Using this pessimistic measure, we cannot rely on the resolution of simulation times of less than 1 min; such timings will be reported as < 1 min.

The parallel performance results presented involve simulations with $\text{Kn} = \infty$. A Knudsen number of $\text{Kn} = \infty$ corresponds to a Boltzmann equation with zero right-hand side. Since there are no arithmetic operations needed to compute the right-hand side, these simulations contain

fewer calculations than those with $Kn < \infty$. Thus, speedup results based on simulations with $Kn = \infty$ are again the most pessimistic ones possible, because cases with $Kn < \infty$ involve more calculations – without incurring additional communications – and hence may exhibit better parallel performance.

In 2-D, we choose a mesh with $N_e = 320$ elements and vary K to control the computational complexity of the problem. We expect to see an increase in observed wall clock times for larger K , but also better speedup. Table III contains the observed wall clock times T_p in minutes for computations using p processors for three cases of progressively finer velocity resolution; on $p = 1$ processor, a serial implementation of the same algorithm is used. An examination of each column for a fixed p indicates that, as expected, an increase in run time is seen for larger K . For each fixed resolution though, as we increase the number of processors, a decrease is seen in the simulation times; some of the simulations for $K = 64$ are so short that we cannot resolve their timings with our resolution of 1 min accuracy.

Figure 7 contains speedup and efficiency plots for the 2-D/2-D simulations. The first plot in the figure contains speedup $S_p := T_1/T_p$, defined as ratio of the wall clock time T_1 using 1 processor over the time T_p using

Table III. Observed Wall Clock Time T_p in Minutes for p Processors for 2-D/2-D Simulations for $Kn = \infty$

N_e	K	DOF	$p=1$	$p=2$	$p=4$	$p=8$	$p=16$	$p=32$	$p=64$
320	64	81,920	16	8	5	3	1	< 1	< 1
320	256	327,680	125	73	43	22	10	6	3
320	1024	1,310,720	1785	887	450	227	122	67	35

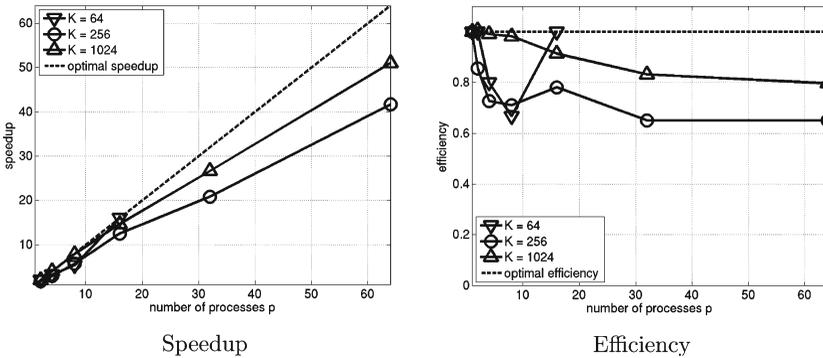


Fig. 7. Speedup and efficiency for 2-D/2-D simulations with $N_e = 320$ for $Kn = \infty$.

p processors, for the three resolutions with the horizontal axis denoting the number of processors p and the vertical axis denoting speedup. The optimal value is $S_p = p$ and shown as a dashed line for visual guidance. The speedup is very good up to 16 processors in all cases; we cannot plot any speedup for $p = 32$ and 64 for $K = 64$. For the more complex cases, speedup tapers off a little for $p > 16$. The best speedup is seen for the most computationally complex case $K = 1024$. The second plot in Fig. 7 shows the efficiency $E_p := S_p/p$ for the three test cases. The optimal value is $E_p = 1$ and shown as a dashed line for visual guidance. Since the efficiency is a function of speedup, we see the same results as in the speedup plot in principle. But an advantage of the efficiency plot is that it allows us to determine that there is a drop-off in efficiency from 1 to 2 as well as to 4 and 8 processors in some cases; this may be a result of initial startup associated with a parallel code on the cluster combined with possibly less than optimal splitting of the spatial domain Ω into subdomains for certain values of p . But this initial drop-off in efficiency does not continue as p grows, giving very good results for $p \geq 16$ in fact. The best efficiency is seen for the most complex case $K = 1024$, which stays above 80% all the way up to 64 processes.

We now conduct analogous 3-D/3-D parallel performance studies. We choose a mesh with $N_e = 1984$ elements and vary K . Table IV contains the observed wall clock time for computations using p processors for the three different velocity resolutions. As expected, we see longer simulation times for the more complex problems and a decrease in computation time with the use of additional processors.

Figure 8 contains speedup and efficiency results for the 3-D/3-D studies. The first plot in the figure shows the parallel speedup. We see again that the more complex cases lead to better performance. The efficiency plot in Fig. 8 allows us again to see in more detail that there is a drop-off in efficiency from 1 to 2 processors. But both plots show that speedup and efficiency remains excellent all the way up to 64 processors for the most complex case with $K = 512$.

Table IV. Observe Wall Clock Time T_p in Minutes for p Processors for 3-D/3-D Simulations for $\text{Kn} = \infty$

N_e	K	DOF	$p=1$	$p=2$	$p=4$	$p=8$	$p=16$	$p=32$	$p=64$
1984	8	126,976	19	11	6	4	2	1	<1
1984	64	1,015,808	120	59	30	16	12	5	3
1984	512	8,126,464	3307	1663	902	480	218	124	60

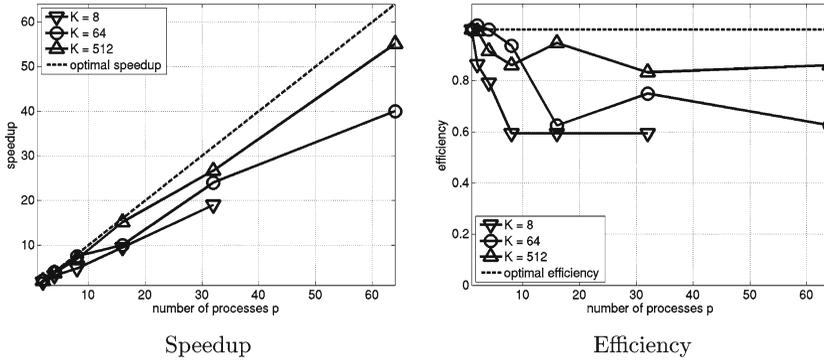


Fig. 8. Speedup and efficiency for 3-D/3-D simulations with $N_e = 1984$ for $Kn = \infty$.

The performance results demonstrate that time is saved by parallel computations in all cases, while the improvement with each doubling of process numbers p is not very uniform in some cases. This non-uniformity may be the result of the particular partitioning of the domain Ω into subdomains; notice that the partitioning is independently chosen for every p , so doubling the number of processes may or may not introduce additional communications. But all studies demonstrate that for the most complex cases with the largest numbers of degrees of freedom, parallel performance is excellent, with efficiency remaining above at least 80% all the way up to 64 processes. Recall that the choice of $Kn = \infty$ reduced the complexity of the cases shown as no right-hand side is computed and recall that our measure of wall clock time as a difference in time stamps of output files make these results *pessimistic* predictors of performance. These results are also consistent with additional performance studies with K fixed and N_e varying [24].

5. Conclusions

In Section 2, we sketched the application problem of chemical vapor deposition and the extension of the kinetic transport and reaction model (KTRM) to include the effect of collisions. The KTRM is given by a system of transient linear Boltzmann equations for the reactive species.

In Section 3, we presented a spectral Galerkin method used to discretize velocity space that approximates each linear Boltzmann equation by a system of linear conservation laws. We show how our choice of basis functions leads directly to diagonal coefficient matrices in the system of conservation laws and also allows for the explicit identification of the inflow

boundary condition for each equation in the system. Since the expansion of the Galerkin method is equal to the one obtained classically, Theorems 2 and 3 also guarantee stability and convergence of our method.

Section 4 presented examples of application results in two and three dimensions that show the capability of the model and its numerical method. The results indicate that the application for Knudsen numbers in the transition regime admits solutions that are different from Maxwellians at some points in time, justifying our interest in accessing the kinetic density directly. Meanwhile, the solutions are still close enough to Maxwellians to allow for the use of a moment method of our type. We have numerically demonstrated convergence and stability for 2-D/2-D and 3-D/3-D simulations. While these results are guaranteed from previous analysis in the limit as the number of expansion terms K tends to infinity, these demonstrations are important to convince ourselves that convergence can indeed be achieved with the finite values for K that are realistically possible for high-dimensional simulations. Finally, we demonstrate that the parallel implementation of our method provides an efficient tool for large numbers of processors on a distributed-memory cluster with high-performance interconnect.

ACKNOWLEDGMENTS

The hardware used in the computational studies was partially supported by the SCREMS grant DMS-0215373 from the U.S. National Science Foundation with additional support from the University of Maryland, Baltimore County. See www.math.umbc.edu/~gobbert/kali for more information on the machine and on projects performed using it. The first author also wishes to thank the Institute for Mathematics and its Applications (IMA) at the University of Minnesota for its hospitality during Fall 2004. The IMA is supported by funds provided by the U.S. National Science Foundation. The second author wishes to acknowledge the University of Maryland, Baltimore County for its support through a DRIF RAS during Spring 2003. The third author acknowledges support from MARCO, DARPA, and NYSTAR through the Interconnect Focus Center.

REFERENCES

1. Bird, G. A. (1976). *Molecular Gas Dynamics*, Oxford University Press.
2. Cale, T. S., Bloomfield, M. O., Richards, D. F., Jansen, K. E., and Gobbert, M. K. (2002). Integrated multiscale process simulation. *Comp. Mater. Sci.* **23**, 3–14.
3. Cale, T. S., Merchant, T. P., Borucki, L. J., and Labun, A. H. (2000). Topography simulation for the virtual wafer fab. *Thin Solid Films* **365**(2), 152–175.

4. Carrillo, J. A., Gamba, I. M., Majorana, A., and Shu, C.-W. (2003). A WENO-solver for the transients of Boltzmann–Poisson system for semiconductor devices: Performance and comparisons with Monte Carlo methods. *J. Comput. Phys.* **184**, 498–525.
5. Cercignani, C. (1988). *The Boltzmann Equation and Its Applications*, Vol. 67 of *Applied Mathematical Sciences*. Springer-Verlag.
6. Cercignani, C. (2000). *Rarefied Gas Dynamics: From Basic Concepts to Actual Calculations*, Cambridge Texts in Applied Mathematics. Cambridge University Press.
7. Chorin, A. J. (1972). Numerical solution of Boltzmann's equation. *Comm. Pure Appl. Math.* **25**, 171–186.
8. Cockburn, B., Karniadakis, G. E., and Shu, C.-W. (eds.) (2000). *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Vol. 11 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag.
9. Fatemi, E., and Odeh, F. (1993). Upwind finite difference solution of Boltzmann equation applied to electron transport in semiconductor devices. *J. Comput. Phys.* **108**, 209–217.
10. Gobbert, M. K., and Cale, T. S. (2001). A feature scale transport and reaction model for atomic layer deposition. In Swihart, M. T., Allendorf, M. D., and Meyyappan, M. (eds.), *Fundamental Gas-Phase and Surface Chemistry of Vapor-Phase Deposition II*, volume 2001–13, The Electrochemical Society Proceedings Series pp. 316–323.
11. Gobbert, M. K., and Cale, T. S. (2006). A kinetic transport and reaction model and simulator for rarefied gas flow in the transition regime. *J. Comput. Phys.* (in press).
12. Gobbert, M. K., Prasad, V., and Cale, T. S. (2002). Modeling and simulation of atomic layer deposition at the feature scale. *J. Vac. Sci. Technol. B* **20**(3), 1031–1043.
13. Gobbert, M. K., Prasad, V., and Cale, T. S. (2002). Predictive modeling of atomic layer deposition on the feature scale. *Thin Solid Films* **410**, 129–141.
14. Gobbert, M. K., Webster, S. G., and Cale, T. S. (2002). Transient adsorption and desorption in micrometer scale features. *J. Electrochem. Soc.* **149**(8), G461–G473.
15. Grad, H. (1949). On the kinetic theory of rarefied gases. *Commun. Pure Appl. Math.* **2**, 331–407.
16. Kersch, A., and Morokoff, W. J. (1995). *Transport Simulation in Microelectronics*, Vol. 3 of *Progress in Numerical Simulation for Microelectronics*. Birkhäuser Verlag, Basel.
17. Markovich, P. A., Ringhofer, C. A., and Schmeiser, C. (1990). *Semiconductor Equations*, Springer-Verlag.
18. Poupaud, F. (1991). Diffusion approximation of the linear semiconductor Boltzmann equation: Analysis of boundary layers. *Asymptotic Anal.* **4**, 293–317.
19. Reed, W. H., and Hill, T. R. (1973). Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM.
20. Remacle, J.-F., Flaherty, J. E., and Shephard, M. S. (2003). An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev.* **45**(1), 53–72.
21. Ringhofer, C. (2000). Space–time discretization of series expansion methods for the Boltzmann transport equation. *SIAM J. Numer. Anal.* **38**(2), 442–465.
22. Ringhofer, C., Schmeiser, C., and Zwirchmayr, A. (2001). Moment methods for the semiconductor Boltzmann equation on bounded position domains. *SIAM J. Numer. Anal.* **39**(3), 1078–1095.
23. Schmeiser, C., and Zwirchmayr, A. (1998). Convergence of moment methods for linear kinetic equations. *SIAM J. Numer. Anal.* **36**(1), 74–88.
24. Webster, S. G. (2004). Stability and convergence of a spectral Galerkin method for the linear Boltzmann equation. Ph.D. thesis, University of Maryland, Baltimore County.

25. Webster, S. G., Gobbert, M. K., and Cale, T. S. (2002). Transient 3-D/3-D transport and reactant–wafer interactions: Adsorption and desorption. In Timans, P., Gusev, E., Roozeboom, F., Ozturk, M., and Kwong, D. L. (eds.), *Rapid Thermal and Other Short-Time Processing Technologies III*, volume 2002–11, The Electrochemical Society Proceedings Series, pp. 81–88.
26. Webster, S. G., Gobbert, M. K., Remacle, J.-F., and Cale, T. S. (2002). Parallel numerical solution of the Boltzmann equation for atomic layer deposition. In Monien, B., and Feldmann, R. (eds.), *Euro-Par 2002 Parallel Processing*, Vol. 2400 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 452–456.