



A fast implicit solver for semiconductor models in one space dimension [☆]

M. Paul Laiu ^{a,*}, Zheng Chen ^b, Cory D. Hauck ^{a,1}

^a Computational and Applied Mathematics Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

^b Department of Mathematics, University of Massachusetts Dartmouth, 285 Old Westport Road, Dartmouth, MA 02747, USA

ARTICLE INFO

Article history:

Received 12 March 2019

Received in revised form 13 April 2020

Accepted 12 May 2020

Available online 19 May 2020

Keywords:

Semiconductor Boltzmann equation

Boltzmann-Poisson systems

Drift-diffusion limit

Domain decomposition

Synthetic acceleration

Preconditioner

ABSTRACT

Several different approaches are proposed for solving fully implicit discretizations of a simplified Boltzmann-Poisson system with a linear relaxation-type collision kernel. This system models the evolution of free electrons in semiconductor devices under a low-density assumption. At each implicit time step, the discretized system is formulated as a fixed-point problem, which can then be solved with a variety of methods. A key algorithmic component in all the approaches considered here is a recently developed sweeping algorithm for Vlasov-Poisson systems. A synthetic acceleration scheme has been implemented to accelerate the convergence of iterative solvers by using the solution to a drift-diffusion equation as a preconditioner. The performance of four iterative solvers and their accelerated variants has been compared on problems modeling semiconductor devices with various electron mean-free-path.

Published by Elsevier Inc.

1. Introduction

The Boltzmann-Poisson system is considered an accurate kinetic model of electron transport in semiconductor devices [1]. This system describes the evolution of an electron distribution function using a semi-classical Boltzmann kinetic equation and generates a self-consistent electric field by coupling the Boltzmann equation to a Poisson equation that is driven by the electron density. Numerical simulation of the Boltzmann-Poisson system is known to be difficult for several reasons, including the nonlinear coupling between equations, the nonlinear collision operator that describes electron-electron and electron-background interactions, and the dimension of the computational domain. Indeed, simulating a three-dimensional device requires the solution to a six-dimensional Boltzmann equation.

Under a low-density assumption, electron-electron interactions become negligible and electrons can be treated as classical particles interacting with a material background. In such cases, the nonlinear collision operator can be replaced by a

[☆] This manuscript has been authored, in part, by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

* Corresponding author.

E-mail addresses: laiump@ornl.gov (M.P. Laiu), zchen2@umassd.edu (Z. Chen), hauckc@ornl.gov (C.D. Hauck).

¹ This author's research was sponsored by the Office of Advanced Scientific Computing Research and performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

linear, relaxation time approximation [1–3] when the steady-state solution to the Boltzmann-Poisson system is of interest. In the case that the electron transport is through channels that are parallel to the electric field, the semiconductor device is effectively one-dimensional [1,3] and can therefore be simulated using the approximated Boltzmann-Poisson system in one space dimension.

In addition to traditional Direct Simulation Monte Carlo methods [4], many deterministic numerical schemes have been developed for solving the Boltzmann-Poisson system and its simplified variants. The deterministic schemes considered in previous works discretize the position-velocity phase space of the Boltzmann equation and its simplified variants using the weighted essentially non-oscillatory (WENO) finite difference method [5–7], the discontinuous Galerkin (DG) method [8–10], and the spectral-difference method [11]. These schemes either consider a steady-state Boltzmann-Poisson system or use explicit time-stepping schemes to capture transient behavior. To guarantee stability, explicit time-stepping schemes usually require the size of the time steps to be proportional to the mean-free-path of particles in the system. Such a restriction can be computational prohibitive for highly collisional problems, where the mean-free-path is small. An explicit asymptotic preserving scheme was introduced to address this issue in [12], where the stability is guaranteed under a parabolic time-step restriction (independent of the mean-free-path) for highly collisional problems. Later, an implicit-explicit (IMEX) asymptotic preserving scheme was developed in [13] to relax the parabolic time-step restriction. However, these and similar approaches do not allow for large-scale local variations in the mean-free-path that are common in multiscale problems.

In this paper, we consider a fully implicit numerical scheme for solving the simplified Boltzmann-Poisson system under the low density assumption in one space dimension. The fully implicit time-stepping method allows for larger time steps that are independent of the mean-free-path, regardless of the collisionality of the problem. Such stability comes at the cost of solving large, possibly ill-conditioned, linear and nonlinear algebraic equations. Hence efficient numerical solvers are needed to update the numerical solution at each time step. In [14], a fast, fully implicit solver was proposed for the nonlinear Vlasov-Poisson system, which is the collisionless variant of the simplified Boltzmann-Poisson system considered in this paper. At each time step, this solver applies a special decomposition of the phase space to allow for the use of the sweeping technique that are commonly used to accelerate the solution of radiation transport problems [15–17]. To utilize the fast solver [14] in the collisional case, we consider the scattering term as a source and formulate the simplified Boltzmann-Poisson system as a nonlinear fixed-point problem at each implicit time step. At each fixed-point iteration, a collisionless problem with source is solved and the electron distribution that solves the collisionless problem is used to update the collision term, which becomes the source at the next fixed-point iteration. The fixed-point problem reaches a solution when the collisionless problem gives an electron distribution that is consistent with the collision term.

Two types of fixed-point formulations for the simplified Boltzmann-Poisson systems are considered and compared in this paper. The main difference between the two formulations is in the treatment of the scattering source in the relaxation operator: in the first case, both the electric field and the scattering source are lagged; in the second, only the electric field is lagged. As a result, problems in the first formulation are solved in a single iteration loop, while the ones in the second formulation require two nested loops. We apply various iterative solvers on these problems and compare their performance. To solve problems in the first formulation, we consider Picard iteration (see, e.g., [18, Section I.8]) and also Anderson acceleration [19,20]. For problems in the second formulation, we solve the nonlinear outer loop via Anderson acceleration and solve the linear inner loop using either Picard iteration or the generalized minimal residual method (GMRES) [21]. We do not apply Picard iteration on the outer loop since preliminary numerical results suggest that this approach is not competitive.

We also consider accelerated/preconditioned variants of the solvers described above, based on the idea of synthetic acceleration (SA) [15,17,22,23], an approach that accelerates convergence of iterative solvers by applying a correction term in between each iteration. This correction term is obtained by solving coarse, cheap, or low-order approximate equations to the error equation of the base iterative solver. Thus, many of the SA schemes can be viewed as two-level multigrid algorithms [24] or preconditioned iterative solvers [15,25,26]. For neutron transport problems, the correction terms can be computed by solving the transport equation on a coarse mesh [26,27] or solving a diffusion equation [15,28,29] that is a low-order approximation to the transport equation near the collision limit where the mean-free-path is small. In this paper, we compute the correction term by solving a drift-diffusion equation that approximates the simplified Boltzmann-Poisson system in the highly collisional, low-field regime.

The various strategies are tested on a one-dimensional silicon n^+-n^- diode problem [3,5,8,30,31]. The algebraic equations to be solved are derived via a backward Euler discretization in time and a discontinuous Galerkin discretization in the position-velocity phase space. The low-order time discretization is chosen primary to simplify the presentation; however the DG discretization is important for capturing the drift-diffusion limit. While other discretizations are possible, our focus here is on the efficiency of the solver strategy. Thus for various levels of collisionality, the computation time and iteration count of each fixed-point formulation and each iterative solver are compared. These results provide a guideline on the selection of iterative solvers for problems with different material profiles.

The remainder of the paper is organized as follows. In Section 2, the simplified Boltzmann-Poisson system, the drift-diffusion equation, and the time, space, and velocity discretization for solving them are described. Section 3 provides details of fixed-point formulations for the discretized equations as well as the iterative solvers for these fixed-point problems. In Section 4, the implementation details and numerical results for the various iterative solvers are reported for the n^+-n^- diode problem. Conclusions and discussion are given in Section 5.

2. Preliminaries

2.1. Semiconductor models

We consider the kinetic model

$$\partial_t f + v \partial_x f + \frac{q_e}{m} E \partial_v f = \omega (\rho M_\Theta - f), \quad (1a)$$

$$E = \partial_x \Phi, \quad \partial_x^2 \Phi = \frac{q_e}{\varepsilon_p} (\rho - D). \quad (1b)$$

Here (1a) describes the evolution of the electron distribution $f = f(t, x, v)$, which is a function of position $x \in [0, L]$, velocity $v \in \mathbb{R}$, and time $t \geq 0$; the electric field $E = E(t, x)$ in (1b) is the spatial gradient² of a potential $\Phi = \Phi(t, x)$ that satisfies a Poisson equation with a source due to the balance between a given doping profile $D = D(x)$ and the particle (electron) concentration $\rho = \rho(t, x) = \int_{\mathbb{R}} f(t, x, v) dv$. The constants q_e , m , and ε_p denote, respectively, the magnitude of the electron charge, the effective electron mass, and the electric permittivity of the material. The collision frequency $\omega = \omega(x)$ takes the form $\omega = \frac{q_e}{m\mu}$ with electron mobility $\mu = \mu(x)$, and the absolute Maxwellian

$$M_\Theta(v) := (2\pi\Theta)^{-\frac{1}{2}} e^{-v^2/2\Theta}, \quad (2)$$

where the background temperature $\Theta := \frac{k_B}{m} T$ with k_B the Boltzmann constant and T the lattice temperature. For the detail derivations of this model, we refer to the reader to [1,32].

2.1.1. Scaled semiconductor models

Since the qualitative behavior of solutions to (1) largely depends on the scales of the system, we introduce non-dimensional variables $\hat{t} = \frac{t}{t_0}$, $\hat{x} = \frac{x}{x_0}$, $\hat{v} = \frac{v}{v_0}$ and express (1) in terms of the scaled variables

$$\hat{f}(\hat{t}, \hat{x}, \hat{v}) = \frac{f(t, x, v)}{f_0}, \quad \hat{\omega}(\hat{x}) = \frac{\omega(x)}{\omega_0}, \quad \hat{D}(\hat{x}) = \frac{D(x)}{D_0}, \quad \hat{\rho}(\hat{t}, \hat{x}) = \frac{\rho(t, x)}{\rho_0}. \quad (3)$$

Let Φ_0 be a nominal value of the potential Φ and let $\Delta\Phi(t, x) := \Phi(t, x) - \Phi_0$. Because the solution of (1) is independent of Φ_0 , we consider the scaled potential $\hat{\Phi} = \Phi_0 + \frac{\Delta\Phi}{[\Phi]}$. By assuming $\hat{\rho} = \int_{\mathbb{R}} \hat{f} d\hat{v}$, the scaled system takes the form

$$\delta \partial_{\hat{t}} \hat{f} + \hat{v} \partial_{\hat{x}} \hat{f} + \beta^2 E \partial_{\hat{v}} \hat{f} = \frac{\omega}{\epsilon} (\rho M_{\alpha^2} - \hat{f}), \quad (4a)$$

$$E = \partial_{\hat{x}} \hat{\Phi}, \quad \partial_{\hat{x}}^2 \hat{\Phi} = \frac{\gamma^2}{\beta^2} (\zeta \rho - D), \quad (4b)$$

where the hats on the variables dropped for simplicity. In (4), the kinetic Strouhal and Knudsen numbers [33,34] are given by $\delta = \frac{x_0}{v_0 t_0}$ and $\epsilon = \frac{v_0}{x_0 \omega_0}$, respectively, and the ratios are defined as $\zeta := \frac{\rho_0}{D_0}$, $\alpha := \frac{\Theta^{1/2}}{v_0}$, $\beta := \frac{B_0}{v_0}$, and $\gamma := \frac{C_0}{v_0}$ with

$$\Theta^{1/2} = \left(\frac{k_B}{m} T \right)^{1/2}, \quad B_0 := \left(\frac{q_e [\Phi]}{m} \right)^{1/2}, \quad \text{and} \quad C_0 := x_0 \omega_{pe} := x_0 \left(\frac{q_e^2 D_0}{\varepsilon_p m} \right)^{1/2}. \quad (5)$$

Here $\Theta^{1/2}$ is the thermal velocity, B_0 is the ballistic velocity, and in plasma physics, ω_{pe} is known as the plasma frequency [35,36].

2.1.2. The drift-diffusion limit

To describe semiconductors with different characteristics, there exist various scaling of the semiconductor model (4), such as the low-field scaling [1,37], the high-field scaling [3,38,39], and the ballistic scaling [5]. In this paper, we use the solution of a drift-diffusion equation as a preconditioner to accelerate the solution procedure for (4) in the low-field, highly collisional regime. While this preconditioner is expected to work well only in this regime, the discretizations of (4) and the formulation of solvers for the resulting algebraic equations do not rely on any particular scaling.

The low-field scaling of (4) assumes that the ratio β is an $O(1)$ quantity and that the ratio $\zeta = 1$, i.e., the scaling of the particle concentration ρ and the doping profile D is identical. Under these assumptions, when ϵ is small, i.e., when the electron mean-free-path $\lambda := \frac{v_0}{\omega_0}$ is much smaller than the spatial scale x_0 , the collision term on the right-hand side of (4a) is much larger than the drift term $\beta^2 E \partial_{\hat{v}} \hat{f}$; it thus becomes the dominant term. In this situation, it is necessary to

² There is a sign difference between the electric field defined in (1) and the usual physics definition convention. We make this choice to match the definition used in the fast sweeping algorithm in [14] for solving Vlasov-Poisson systems. In particular, the sign choice in (1) implies that sign of E determines the direction of flow in the velocity variable. The sweeping algorithm is used extensively in this paper. See Section 2.2.3 for details.

choose $\delta \approx \epsilon$ in order to observe the nontrivial dynamics in the long time scale, in which case (4) can be approximated by a drift-diffusion-Poisson model.

It is shown in [37] that when the potential Φ is known and sufficiently smooth, a standard drift-diffusion model can be derived from (4a) in the low-field, collision limit ($\delta \approx \epsilon \rightarrow 0$) by expanding the distribution function f via the Hilbert expansion as

$$f(t, x, v) = \varrho(t, x) M_{\alpha^2}(v) + O(\epsilon). \quad (6)$$

This result has been extended in [40] and [41] respectively to the one-dimensional and multi-dimensional Boltzmann-Poisson systems with a self-consistent potential via Poisson coupling as in (4b). The resulting drift-diffusion-Poisson model takes the form

$$\xi \partial_t \varrho - \partial_x (\omega^{-1} \partial_x \varrho) + \beta^2 \partial_x (\omega^{-1} E \varrho) = 0, \quad (7a)$$

$$E = \partial_x \Phi, \quad \partial_x^2 \Phi = \frac{\gamma^2}{\beta^2} (\varrho - D), \quad (7b)$$

where (7a) is a drift-diffusion equation coupled with the Poisson system (7b), and $\xi > 0$ is the ratio between δ and ϵ , i.e., $\delta = \xi \epsilon$. We refer to this low-field, collision limit as the “drift-diffusion limit”.

In this paper, we use the solution to the drift-diffusion equation as a preconditioner for solving the scaled semiconductor kinetic equation (4a). The numerical method we used to solve the drift-diffusion equation is discussed in Section 2.4, and the drift-diffusion preconditioning approach is introduced in Section 3.3.

2.2. Solving the kinetic equation

In this paper, implicit time discretization of the kinetic equation (4a) is considered. In the simplified case when the electric field E and the particle concentration ρ are known a priori, implicit time discretization of (4a) leads to linear systems that can be solved efficiently via the fast sweeping algorithm proposed in [14]. The implicit time discretization, the position-velocity phase space discretization, and the fast sweeping approach for solving (4a) in the simplified case are discussed in Sections 2.2.1, 2.2.2, and 2.2.3, respectively. Based on the method presented in this section, we propose several iterative solvers in Section 3 for solving (4a) in the self-consistent case that E and ρ are coupled via the Poisson equation (4b).

2.2.1. Time discretization

In the temporal domain $[0, t_{\text{final}}]$, we apply a uniform discretization with time step size Δt and denote $f^n \approx f(t^n, \cdot, \cdot)$, where $t^n = n \Delta t$. To simplify the presentation, we consider the backward Euler scheme in this paper. Although this scheme is only first-order accurate, it can be used as a building block for higher-order implicit schemes, such as the singly diagonally implicit Runge-Kutta method (SDIRK) [18,42]. Applying the backward Euler scheme to (4a) leads to

$$v \partial_x f^{n+1} + \beta^2 E^{n+1} \partial_v f^{n+1} + \left(\frac{\delta}{\Delta t} + \frac{\omega}{\epsilon} \right) f^{n+1} = \frac{\omega}{\epsilon} \rho^{n+1} M_{\alpha^2} + \frac{\delta}{\Delta t} f^n, \quad (8)$$

where $\rho^{n+1} = \int_{\mathbb{R}} f^{n+1} dv$ and E^{n+1} is coupled via the Poisson equation (4b) with $\rho = \rho^{n+1}$. For the remainder of Section 2.2, we assume that ρ^{n+1} and E^{n+1} are known a priori at time t^n . In this simplified case, (8) becomes a linear steady-state Vlasov problem with a source:

$$v \partial_x f + \eta E \partial_v f + \sigma f = q, \quad (9)$$

where, in an abuse of notation, $f = f^{n+1}(x, v)$ denotes the steady-state unknown, $E = E^{n+1}(x)$ denotes a given electric field, η and σ are positive constants, and $q = q(x, v)$ denotes a general source. This simplified steady-state problem (9) can be solved efficiently using the fast sweeping algorithm proposed in [14]. We give the details of this algorithm in Section 2.2.3.

2.2.2. Phase space discretization

For the position-velocity (x - v) phase space discretization, we apply the discontinuous Galerkin method with \mathbb{P}^1 elements and upwind numerical flux. This choice is motivated by the fact that DG methods with upwind flux allows for the application of the fast sweeping technique introduced in Section 2.2.3, as shown in [14]. Another important advantage of DG methods is that, unlike finite volume and finite difference methods, they are able to capture the diffusion limit for neutral particle transport using a standard upwind flux [43–46]. The drift-diffusion limit is derived under the same scaling, and numerical results suggest that DG methods capture this limit as well. A rigorous analysis of such a result is the subject of future work.

We also note that, although higher-order DG elements can be used without affecting the sweeping technique in Section 2.2.3, the spatial accuracy is still limited by the Poisson solver in Section 2.3. Specifically, the spatial discretization of

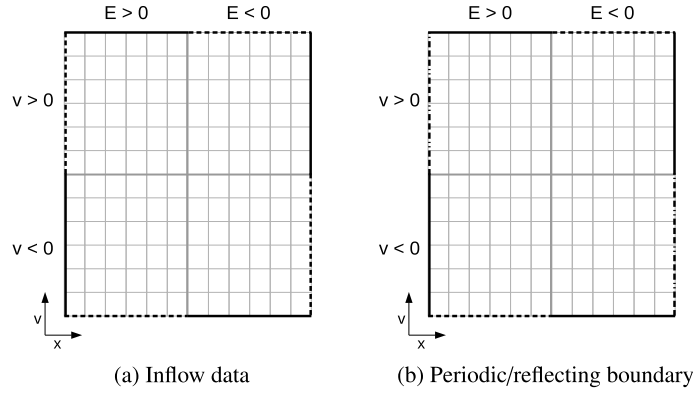


Fig. 1. Examples of boundary conditions: Figs. 1a and 1b show the inflow data and periodic/reflecting boundary in the x -direction, respectively. In both figures, inflow data is given on the v -boundary. Here solid lines denote the outflow boundary ($\partial\mathcal{D}^{\text{out}}$), dashed lines denote the inflow data ($\partial\mathcal{D}^{\text{Data}}$), and dash-dotted lines denote solution-dependent inflow boundary ($\partial\mathcal{D}^{\Lambda}$).

potential Φ is restricted to at most second order, which results in a first-order approximation of the electric field E . We refer the reader to Section 2.3 and [14, Remark 1] for discussions on this restriction.

For the position-velocity (x - v) phase space, we truncate the velocity domain from \mathbb{R} , the entire real line, to a finite interval $[a_v, b_v]$. The position-velocity computational domain is then $\mathcal{D} := [0, L] \times [a_v, b_v]$. Let $\partial\mathcal{D}$ be the boundary of \mathcal{D} and $n(x, v) \in \mathbb{R}^2$ be the outward normal at $(x, v) \in \partial\mathcal{D}$. As usual, we decompose the boundary into two disjoint pieces: $\partial\mathcal{D} = \partial\mathcal{D}^{\text{in}} \cup \partial\mathcal{D}^{\text{out}}$ where

$$\partial\mathcal{D}^{\text{in}} := \{(x, v) : (v, E) \cdot n(x, v) \leq 0\} \quad \text{and} \quad \partial\mathcal{D}^{\text{out}} := \{(x, v) : (v, E) \cdot n(x, v) > 0\}. \quad (10)$$

We further decompose $\partial\mathcal{D}^{\text{in}}$ into pieces: $\partial\mathcal{D}^{\text{in}} = \partial\mathcal{D}^{\Lambda} \cup \partial\mathcal{D}^{\text{Data}}$, where $\partial\mathcal{D}^{\Lambda}$ is the portion of the inflow boundary that depends on the interior solution, e.g., periodic or reflecting boundary, and $\partial\mathcal{D}^{\text{Data}}$ is the portion upon which data is given. With these notations, the steady state equation (9) takes the form

$$\begin{cases} v\partial_x f + \eta E\partial_v f + \sigma f = q \\ f(x, v) = (\Lambda f)(x, v), & (x, v) \in \partial\mathcal{D}^{\Lambda} \\ f(x, v) = w(x, v), & (x, v) \in \partial\mathcal{D}^{\text{Data}} \end{cases} \quad (11)$$

where w is known and the abstract linear operator Λ , which maps functions on $\partial\mathcal{D}^{\text{out}}$ to functions on $\partial\mathcal{D}^{\Lambda}$, can be used to describe periodic or reflecting boundary conditions.

The computational domain \mathcal{D} is discretized into $N_x \times N_v$ rectangular cells of uniform size $\Delta x \times \Delta v$. For $i = 1, \dots, N_x$ and $j = 1, \dots, N_v$, the cell $C_{i,j}$ is centered at $(x_i, v_j) := ((i - \frac{1}{2})\Delta x, a_v + (j - \frac{1}{2})\Delta v)$. We denote the set of all cells by \mathcal{T} and the set of all edges by \mathcal{F} . The set \mathcal{F} is then decomposed into disjoint sets

$$\mathcal{F} = \mathcal{F}^{\Lambda} \cup \mathcal{F}^{\text{Data}} \cup \mathcal{F}_x \cup \mathcal{F}_v, \quad (12)$$

where \mathcal{F}^{Λ} contains cell edges on the boundary component $\partial\mathcal{D}^{\Lambda}$, $\mathcal{F}^{\text{Data}}$ contains cell edges on the boundary component $\partial\mathcal{D}^{\text{Data}}$, and \mathcal{F}_x and \mathcal{F}_v contains the remaining cell edges that are perpendicular to the x and v axes, respectively. We further decompose $\mathcal{F}^{\Lambda} = \mathcal{F}_x^{\Lambda} \cup \mathcal{F}_v^{\Lambda}$ and $\mathcal{F}^{\text{Data}} = \mathcal{F}_x^{\text{Data}} \cup \mathcal{F}_v^{\text{Data}}$, where the subscripts denote the axis to which the edges are perpendicular. Fig. 1 illustrates two types of boundary conditions on \mathcal{D} . Specifically, Fig. 1a shows the inflow boundary that only depends on prescribed data, and Fig. 1b shows the inflow boundary that depends on the interior solution, e.g., periodic or reflecting boundary conditions, in the x direction. Here, solid lines represent the outflow boundary $\partial\mathcal{D}^{\text{out}}$, dashed lines represent the inflow data $\partial\mathcal{D}^{\text{Data}}$, and dash-dotted lines represent $\partial\mathcal{D}^{\Lambda}$, the part of inflow boundary that depends on interior solutions. The sets of edges \mathcal{F}_x^{Λ} , \mathcal{F}_v^{Λ} , $\mathcal{F}_x^{\text{Data}}$, $\mathcal{F}_v^{\text{Data}}$, \mathcal{F}_x , and \mathcal{F}_v then can be easily associated to the edges shown in Fig. 1.

Let $\mathcal{Z} = \{g \in L^2(\mathbb{R} \times \mathbb{R}) : g|_{\mathcal{D}^c} = 0\}$ and $\mathcal{Z}^h := \{g^h \in \mathcal{Z} : g^h|_C \in \mathbb{P}^1(C), \forall C \in \mathcal{T}\}$, where $\mathbb{P}^1(C)$ denotes the space of polynomials up to degree one on the cell C . For $g^h \in \mathcal{Z}^h$, the traces on the two sides of an edge $e \in \mathcal{F}_x \cup \mathcal{F}_v$ are defined as

$$g^{h,\pm}(x, v) = \lim_{\epsilon \rightarrow 0^+} g^h(x \pm \epsilon, v \pm \epsilon), \quad \text{for all } (x, v) \in e. \quad (13)$$

For these edges, the numerical trace of g^h is defined via upwinding. Specifically, let \bar{v}_{e_x} denote the value of v at the center of edge $e_x \in \mathcal{F}_x$, and let \bar{E}_{e_v} denote the value of E at the center of edge $e_v \in \mathcal{F}_v$, the numerical traces on e_x and e_v are respectively defined as

$$\hat{g}^h(x, v) = \begin{cases} g^{h,-}(x, v), & \bar{v}_{e_x} > 0, \\ g^{h,+}(x, v), & \bar{v}_{e_x} < 0, \end{cases} \quad \hat{g}^h(x, v) = \begin{cases} g^{h,-}(x, v), & \bar{E}_{e_v} > 0, \\ g^{h,+}(x, v), & \bar{E}_{e_v} < 0. \end{cases} \quad (14)$$

This definition guarantees a constant upwind direction on each edge. For a test function $g^h \in \mathcal{Z}^h$, the traces on a boundary edge $e \in \mathcal{F}^\Lambda \cup \mathcal{F}^{\text{Data}}$ are defined as

$$g^{h,\partial}(x, v) = \lim_{\epsilon \rightarrow 0^+} g^h(x - \epsilon n_x, v - \epsilon n_v), \quad \text{for all } (x, v) \in e, \quad (15)$$

where n_x and n_v are the first and second components of the outward normal n , respectively.

With these definitions, the discontinuous Galerkin method solves for $f^h \in \mathcal{Z}^h$ that satisfies

$$\mathcal{A}(f^h, g^h) = Q(g^h), \quad \forall g^h \in \mathcal{Z}^h, \quad (16)$$

with the bilinear operator

$$\begin{aligned} \mathcal{A}(f^h, g^h) := & \sum_{C \in \mathcal{T}} \iint_C (-\bar{v}_C f^h \partial_x g^h - \eta \bar{E}_C f^h \partial_v g^h + \sigma f^h g^h) dx dv \\ & - \sum_{e \in \mathcal{F}_x} \bar{v}_e \int_e \hat{f}^h (g^{h,+} - g^{h,-}) dv - \sum_{e \in \mathcal{F}_v} \eta \bar{E}_e \int_e \hat{f}^h (g^{h,+} - g^{h,-}) dx \\ & - \sum_{e \in \mathcal{F}_x^\Lambda} |\bar{v}_e| \int_e (\Lambda f^h) g^{h,\partial} dv - \sum_{e \in \mathcal{F}_v^\Lambda} \eta |\bar{E}_e| \int_e (\Lambda f^h) g^{h,\partial} dx \end{aligned} \quad (17)$$

and the source

$$Q(g^h) := \sum_{C \in \mathcal{T}} \iint_C q g^h dx dv + \sum_{e \in \mathcal{F}_x^{\text{Data}}} |\bar{v}_e| \int_e w g^{h,\partial} dv + \sum_{e \in \mathcal{F}_v^{\text{Data}}} \eta |\bar{E}_e| \int_e w g^{h,\partial} dx, \quad (18)$$

where \bar{v}_C and \bar{E}_C are the values of v and E at the center of cell C , respectively.

In the case of neutral particles ($E = 0$), the upwind definition of the numerical traces in (14) allows (16) to be solved with an explicit sweeping procedure that moves through the computational domain in a direction determined by the sign of v . Such sweeping procedures are commonly used for solving radiation transfer problems [15–17]. However, in the case of charged particles, the procedure no longer applies when both v and E are allowed to change sign over the phase space domain. This is because, in such cases, changes in the upwind direction may create cyclic dependencies in the elements, see, e.g., [14, Figure 3.1]. To address this challenge, a domain decomposition approach was introduced in [14] to break these dependencies. We briefly discuss this approach and the associated sweeping method in the next subsection.

2.2.3. Domain decomposition and fast sweeping

The domain decomposition method separates the phase space into subdomains along the line $\{v = 0\}$. Let $\mathcal{D}_1 := [0, L] \times (0, b_v]$ and $\mathcal{D}_2 := [0, L] \times [a_v, 0)$ be the subdomains, and let $\Gamma := [0, L] \times \{0\}$. We assume that there exists some index j_0 such that $C_{i,j_0} \subseteq \mathcal{D}_2$ and $C_{i,j_0+1} \subseteq \mathcal{D}_1$, and denote the set of cell edges in Γ as \mathcal{F}_0 . We further decompose \mathcal{F}_0 into disjoint sets \mathcal{F}_0^+ and \mathcal{F}_0^- based on the sign of the electric field at the edge center. For $g^h \in \mathcal{Z}^h$, we define

$$g_1^h(x, v) = \begin{cases} g^h(x, v), & (x, v) \in \mathcal{D}_1, \\ 0, & \text{otherwise}, \end{cases} \quad g_2^h(x, v) = \begin{cases} g^h(x, v), & (x, v) \in \mathcal{D}_2, \\ 0, & \text{otherwise}. \end{cases} \quad (19)$$

The bilinear form in (16) then can be expanded as

$$\mathcal{A}(f^h, g^h) = \mathcal{A}(f_1^h, g_1^h) + \mathcal{A}(f_2^h, g_2^h) + \mathcal{A}(f_1^h, g_2^h) + \mathcal{A}(f_2^h, g_1^h). \quad (20)$$

By the definition of \mathcal{A} , it is straightforward to verify that

$$\mathcal{A}(f_1^h, g_2^h) = - \sum_{e \in \mathcal{F}_0^-} \eta |\bar{E}_e| \int_e \hat{f}^h g^{h,-} dx, \quad (21a)$$

$$\mathcal{A}(f_2^h, g_1^h) = - \sum_{e \in \mathcal{F}_0^+} \eta |\bar{E}_e| \int_e \hat{f}^h g^{h,+} dx, \quad (21b)$$

where the edges in \mathcal{F}_0^+ do not appear in (21a) since f_1^h does not contribute to the numerical traces \hat{f}^h on these edges due to upwinding. Similarly, the edges in \mathcal{F}_0^- are not included in (21b). We then define for all $\mathcal{F}_0^\pm \subset \Gamma$, the edge values

$$\hat{f}_1^{h,*} = \mathcal{P}_1(f_1^h) := \begin{cases} \hat{f}_1^h, & \text{on } \mathcal{F}_0^-, \\ 0, & \text{on } \mathcal{F}_0^+, \end{cases} \quad \text{and} \quad \hat{f}_2^{h,*} = \mathcal{P}_2(f_2^h) := \begin{cases} \hat{f}_2^h, & \text{on } \mathcal{F}_0^+, \\ 0, & \text{on } \mathcal{F}_0^-. \end{cases} \quad (22)$$

The system (16) is then equivalent to the coupled system

$$\mathcal{A}(f_1^h, g_1^h) = Q(g_1^h) - \mathcal{B}(\hat{f}_2^{h,*}, g_1^h), \quad (23a)$$

$$\mathcal{A}(f_2^h, g_2^h) = Q(g_2^h) - \mathcal{B}(\hat{f}_1^{h,*}, g_2^h), \quad (23b)$$

$$\hat{f}_1^{h,*} = \mathcal{P}_1(f_1^h), \quad (23c)$$

$$\hat{f}_2^{h,*} = \mathcal{P}_2(f_2^h), \quad (23d)$$

where $\mathcal{B}(\hat{f}_1^{h,*}, g_2^h) := \mathcal{A}(f_1^h, g_2^h)$, $\mathcal{B}(\hat{f}_2^{h,*}, g_1^h) := \mathcal{A}(f_2^h, g_1^h)$, and the equations in (23a) and (23b) are coupled only through the projections in (23c) and (23d). Suppose that $\hat{f}_1^{h,*}$ and $\hat{f}_2^{h,*}$ are known; then (23a) and (23b) are fully decoupled. In each subdomain (\mathcal{D}_1 or \mathcal{D}_2) of the phase space, the sign of v is fixed and only E is allowed to change sign. Thus, there is no cyclic dependency in these subdomains, and the decoupled systems (23a) and (23b) can be solved independently via explicit sweeping approach in \mathcal{D}_1 and \mathcal{D}_2 , respectively.

Let \mathbf{f} be the expansion coefficients of $f^h := f_1^h + f_2^h$ on an orthogonal basis of $\mathbb{P}^1(C)$ for all $C \in \mathcal{T}$, and let $\hat{\mathbf{f}}^*$ be the expansion coefficients of $\hat{f}^{h,*} := \hat{f}_1^{h,*} + \hat{f}_2^{h,*}$ on an orthogonal basis of $\mathbb{P}^1(e)$ for all $e \in \mathcal{F}_0$. Then (23) can be expressed a linear system in \mathbf{f} and $\hat{\mathbf{f}}^*$ as

$$A\mathbf{f} = \mathbf{q} - B\hat{\mathbf{f}}^*, \quad (24a)$$

$$\hat{\mathbf{f}}^* = P\mathbf{f}, \quad (24b)$$

where \mathbf{q} is the vector of expansion coefficients of the source term $Q(g^h)$, and the matrices A , B , and P are defined based on the operators \mathcal{A} , \mathcal{B} , \mathcal{P}_1 , and \mathcal{P}_2 . See [14] for the detailed definitions. Applying A^{-1} from the left on both sides of (24a) and plugging the resulting equation into (24b) leads to a much smaller linear system

$$(I + PA^{-1}B)\hat{\mathbf{f}}^* = PA^{-1}\mathbf{q}, \quad (25)$$

where the operation A^{-1} can be performed efficiently via the sweeping approach. Then $\hat{\mathbf{f}}^*$ can be computed by solving (25) with a Krylov solver, such as the generalized minimal residual method (GMRES) [21]. After obtaining $\hat{\mathbf{f}}^*$, the full expansion coefficient \mathbf{f} is then computed by a final sweeping procedure

$$\mathbf{f} = A^{-1}(\mathbf{q} - B\hat{\mathbf{f}}^*). \quad (26)$$

To summarize, (25)–(26) defines a mapping from the discretized source \mathbf{q} to the vector \mathbf{f} which solves the discretized form of the steady-state equation (11).

2.3. Solving the Poisson equation

We solve the Poisson equation (4b) with a continuous Galerkin method with \mathbb{Q}^1 elements on the same spatial mesh as given in Section 2.2.2. Because the method is standard, we omit the details and refer the reader to, for example, [47,48] for complete presentation. For a given particle concentration ρ , this method maps the Galerkin discretization of ρ to a discretized potential Φ , and, since \mathbb{Q}^1 elements are used, the discretized electric field E can be directly calculated from Φ . Further, this discretization of Φ guarantees that E does not change sign within a spatial cell, which is necessary for the application of the sweeping algorithm introduced in Section 2.2.3. Specifically, as pointed out in [14, Remark 1], the sweeping algorithm requires a single upwind direction assigned on each cell. Therefore, to apply the sweeping technique, the electric field given by the Poisson solver cannot change signs within a spatial cell, which restricts the approximation of E to be first order. The development of higher-order Poisson solvers that allows for the application of the sweeping algorithm is a research topic which warrants further investigation.

2.4. Solving the drift-diffusion equation

In the drift-diffusion limit, numerically solving the scaled semiconductor kinetic equation (4a) is difficult since the system is stiff. As discussed in Section 2.1.2, the drift-diffusion equation (7a) serves as a good approximation to (4a) near the drift-diffusion limit. To accelerate the solution procedure of (4a) near this limit, we apply a synthetic acceleration [15,16] and use the solution to (7a) as a preconditioner when solving (4a). The detailed discussion of this acceleration technique is given in Section 3.3. Here we focus on the discretization of (7a). Discretizing in time with backward Euler gives

$$\frac{\xi}{\Delta t} Q^{n+1} - \partial_x(\omega^{-1} \partial_x Q^{n+1}) + \beta^2 \partial_x(\omega^{-1} E^{n+1} Q^{n+1}) = \frac{\xi}{\Delta t} Q^n. \quad (27)$$

If E^{n+1} is known a priori at time t^n , then (27) takes the steady-state form

$$-\partial_x(\omega^{-1}\partial_x\varrho) + \beta^2\partial_x(\omega^{-1}E\varrho) + \frac{\xi}{\Delta t}\varrho = Q, \quad (28)$$

where E is a given electric field and Q is a general source.

We solve (28) with the direct Discontinuous Galerkin method with interface correction (DDG-IC). The original DDG scheme [49] is derived based on the weak formulation of (28) with numerical fluxes approximating derivatives of the solution at element boundaries. The interface correction for DDG was introduced later in [50] to obtain the optimal $(k+1)$ -th order of accuracy for polynomial approximations of degree k . It is shown in [51] that, with proper choices of numerical flux and limiters, the DDG-IC method satisfies the maximum principle with accuracy up to third order.

Let the spatial domain $[0, L]$ be divided into N_x cells $\{I_j\}_{j=1}^{N_x}$, where $I_j = [x_{j-1}, x_j]$ with $x_j = j\Delta x$, and let $\mathcal{V}^h = \{\varphi^h \in L^2(\mathbb{R}) : \varphi^h|_{[0,L]^c} = 0, \varphi^h|_{I_j} \in \mathbb{P}^1(I_j), \forall j = 1, \dots, N_x\}$ denote the numerical solution space. For $\varphi^h \in \mathcal{V}^h$, we define the numerical trace of φ^h at cell interface x_j as $\varphi_j^{h,\pm} := \lim_{\epsilon \rightarrow 0^+} \varphi^h(x_j \pm \epsilon)$ for $j = 0, \dots, N_x$. The jump and average of φ^h at x_j are defined respectively as

$$[\varphi^h]_j = \varphi_j^{h,+} - \varphi_j^{h,-} \quad \text{and} \quad (\overline{\varphi^h})_j = \frac{1}{2}(\varphi_j^{h,+} + \varphi_j^{h,-}). \quad (29)$$

The DDG-IC scheme then solves (28) by finding the solution $\varrho^h \in \mathcal{V}^h$ such that, for any test function $\varphi^h \in \mathcal{V}^h$ and on any cell I_j ,

$$\begin{aligned} & \int_{I_j} \tau^h \partial_x \varrho^h \partial_x \varphi^h dx - \varphi_j^{h,-} (\widehat{\tau^h \partial_x \varrho^h})_j + \varphi_{j-1}^{h,+} (\widehat{\tau^h \partial_x \varrho^h})_{j-1} + (\overline{\partial_x \varphi^h})_j [\tau^h \partial_x \varrho^h]_j + (\overline{\partial_x \varphi^h})_{j-1} [\tau^h \partial_x \varrho^h]_{j-1} \\ & - \int_{I_j} \tau^h \beta^2 E \varrho^h \partial_x \varphi^h dx + \varphi_j^{h,-} (\widehat{\tau^h E \varrho^h})_j - \varphi_{j-1}^{h,+} (\widehat{\tau^h E \varrho^h})_{j-1} + \frac{\xi}{\Delta t} \int_{I_j} \varrho^h \varphi^h dx = \int_{I_j} Q \varphi^h dx, \end{aligned} \quad (30)$$

where τ^h is the L^2 orthogonal projection of ω^{-1} onto \mathcal{V}^h , and the fourth and fifth terms in (30) are the interface correction terms. Here the numerical flux $\widehat{\tau^h \partial_x \varrho^h}$ at x_j is defined as

$$(\widehat{\tau^h \partial_x \varrho^h})_j = \frac{2}{\Delta x} [\tau^h \varrho^h]_j + (\overline{\tau^h \partial_x \varrho^h})_j = \frac{2}{\Delta x} (\tau_j^{h,+} \varrho_j^{h,+} - \tau_j^{h,-} \varrho_j^{h,-}) + \frac{1}{2} (\tau_j^{h,+} (\partial_x \varrho)_j^{h,+} + \tau_j^{h,-} (\partial_x \varrho)_j^{h,-}), \quad (31)$$

and the Lax-Friedrich flux is used for $\widehat{\tau^h E \varrho^h}$, i.e.,

$$(\widehat{\tau^h E \varrho^h})_j = \frac{1}{2} (E_j \tau_j^{h,-} \varrho_j^{h,-} + E_{j+1} \tau_j^{h,+} \varrho_j^{h,+} - \alpha_j [\varrho^h]_j), \quad \alpha_j := \max\{|E_j \tau_j^{h,-}|, |E_{j+1} \tau_j^{h,+}|\}, \quad (32)$$

where E_j and E_{j+1} are the values of E at the cell centers of I_j and I_{j+1} , respectively. Here the term ω^{-1} is not involved in the numerical fluxes since the collision frequency ω is assumed to be known on the entire spatial domain, including the cell boundaries.

3. Nonlinear solution strategies

In this section, we propose several strategies for solving (4). To simplify the discussion, we first introduce a concise operator notation for the fast sweeping method discussed in Section 2.2.3. Specifically, we write (8) as

$$\mathcal{L}_{E^{n+1}} f^{n+1} = S\rho^{n+1} + s, \quad (33)$$

where

$$\mathcal{L}_E f := v \partial_x f + \beta^2 E \partial_v f + \left(\frac{\delta}{\Delta t} + \frac{\omega}{\epsilon} \right) f, \quad S\rho := \frac{\omega}{\epsilon} \rho M_{\alpha^2}, \quad \text{and} \quad s := \frac{\delta}{\Delta t} f^n. \quad (34)$$

We refer to $S\rho$ and s as the scattering source and the general source, respectively. If $E^{n+1} = \tilde{E}$ and $\rho^{n+1} = \tilde{\rho}$, where \tilde{E} and $\tilde{\rho}$ are known, then f^{n+1} satisfies a steady-state problem of the form (9):

$$\mathcal{L}_{\tilde{E}} f = S\tilde{\rho} + s. \quad (35)$$

We let

$$\mathcal{L}_{\tilde{E}} f = S\tilde{\rho} + s \quad (36)$$

denote the discretization of (35) as described in Section 2.2.2. Here the operators L and S are discretized versions of \mathcal{L} and \mathcal{S} while f , \tilde{E} , $\tilde{\rho}$, and s denote the discretizations of f , \tilde{E} , $\tilde{\rho}$, and s , respectively. The solver discussed in Section 2.2.3 then computes f^{n+1} by solving the linear problem

$$f = L_{\tilde{E}}^{-1}(S\tilde{\rho} + s), \quad (37)$$

where the operation $L_{\tilde{E}}^{-1}$ is performed using the sweeping algorithm from [14] that is summarized in Section 2.2.3.

In the self-consistent setting, $\rho^{n+1} := \int_{\mathbb{R}} f^{n+1} dv$, and E^{n+1} is coupled to ρ via the Poisson equation (4b). Thus, instead of the linear problem (37), we solve

$$f = L_E^{-1}(SPf + s), \quad E = F(Pf), \quad (38)$$

where P denotes integration over the computational velocity domain $[a_v, b_v]$ and F , which maps a given particle concentration to an electric field, denotes the solution procedure of the Poisson equation (4b) using the continuous Galerkin method in Section 2.3. This problem is nonlinear since E depends on f .

The problem (38) can be solved via nonlinear fixed-point iterative solvers. However, the cost of solving (38) could be prohibitive in the multi-dimensional setting due to the high dimensionality of f . To reduce the problem dimension, a common trick (see, e.g., [15,17]) is to integrate the first equation in (38) with respect to v and solve the resulting fixed-point problem for ρ and E :

$$\rho = PL_E^{-1}(S\rho + s), \quad E = F(\rho). \quad (39)$$

The solution to (39) gives ρ^{n+1} and E^{n+1} . Thus f^{n+1} can be computed by a final sweeping procedure by setting $\tilde{E} = E^{n+1}$ and $\tilde{\rho} = \rho^{n+1}$ in (37). In the remainder of this section, we consider two different formulations of (39).

3.1. Type-I formulation

The type-I approach formulates (39) as a nonlinear fixed-point problem on ρ , i.e.,

$$\rho = G_1(\rho) := PL_{F(\rho)}^{-1}(S\rho + s) \quad (40)$$

where G_1 is nonlinear due to the coupling between $E(=F(\rho))$ and ρ . To solve (40), two iterative solvers are considered: standard Picard iteration (see, e.g., [18, Section I.8]) and Anderson acceleration [19,20].

3.1.1. Type-I – Picard iteration

With an initial guess $\rho^{(0)}$, Picard iteration lags both the scattering source and the electric field terms in G_1 and updates the electron concentration by evaluating G_1 . Specifically, the Picard iteration update at iteration $k+1$ is given by

$$\rho^{(k+1)} = G_1(\rho^{(k)}) = PL_{F(\rho^{(k)})}^{-1}(S\rho^{(k)} + s). \quad (41)$$

It is well-known that Picard iteration converges when G_1 is a contraction mapping, and the rate of convergence depends on the spectral radius of the Jacobian of G_1 .

3.1.2. Type-I – Anderson acceleration

Anderson acceleration was first proposed in [19] as an acceleration method based on nonlinear Krylov solvers for fixed-point problems. Here we adopt the variant given in [20] for solving (40). At iteration $k+1$, Anderson acceleration first computes the residual

$$h_1^{(k)} := G_1(\rho^{(k)}) - \rho^{(k)}, \quad (42)$$

then solves the least-squares problem

$$\alpha^* := \operatorname{argmin}_{\alpha \in \mathbb{R}^{m_k+1}} \left\{ \left\| \sum_{i=0}^{m_k} \alpha_i h_1^{(k-i)} \right\|_2^2 : \sum_{i=0}^{m_k} \alpha_i = 1 \right\} \quad (43)$$

with $m_k := \min\{m, k\}$, and finally updates

$$\rho^{(k+1)} = \sum_{i=0}^{m_k} \alpha_i^* G_1(\rho^{(k-i)}). \quad (44)$$

Here the truncation parameter m is a nonnegative integer that indicates the maximum number of residuals maintained in memory. When $m=0$, Anderson acceleration reduces to standard Picard iteration. For $m>0$, Anderson acceleration updates

ρ with a convex combination of the previous m_k iterates that leads to the minimum residual. It is proved in [52] that, on linear problems, Anderson acceleration converges as long as Picard iteration does; for nonlinear problems, local convergence of Anderson acceleration is guaranteed under assumptions similar to the standard ones for local convergence of Newton's method. Since Anderson acceleration utilizes information from the previous iterations, it is expected to converge faster than Picard iteration in practice, but at a cost of additional memory usage. When the problem is linear, Anderson acceleration has been shown to be equivalent to GMRES under some mild assumptions [20].

3.2. Type-II formulation

The type-II fixed-point formulation of (39) aims to reduce the nonlinearity of the type-I formulation (40) by relaxing the coupling between the electric field E and electron concentration ρ in the iterative procedure. Specifically, we write (39) as

$$(I - PL_E^{-1}S)\rho = PL_E^{-1}s, \quad E = F(\rho) \quad (45)$$

with I the identity operator, and then plug in $E = F(\rho)$ to obtain the type-II formulation of (39)

$$\rho = G_2(\rho) := (I - PL_{F(\rho)}^{-1}S)^{-1}PL_{F(\rho)}^{-1}s, \quad (46)$$

where $G_2(\rho)$ depends on ρ only through the electric field $F(\rho)$.

The structure of (46) motivates a nested iterative procedure, which starts with an “outer loop” that solves the type-II fixed-point problem $\rho = G_2(\rho)$ with some iterative solvers, such as Picard iteration. In the outer loop, the evaluation of $G_2(\rho)$ at some ρ is then performed in an “inner loop” in which the linear system

$$(I - PL_{F(\rho)}^{-1}S)x = PL_{F(\rho)}^{-1}s \quad (47)$$

is (iteratively) solved for x at a given electric field $F(\rho)$, and $G_2(\rho)$ is then set to x . An example of the nested procedure that uses Anderson acceleration in the outer loop is given in the following paragraph. The intent is that, for a fixed electric field $F(\rho)$ given by the outer loop, the inner loop will provide $G_2(\rho)$, a fast and accurate update of ρ , which is then fed into the outer loop, thereby improving the overall efficiency.

Anderson acceleration is used in this paper to solve the nonlinear fixed-point problem (46) in the outer loop.³ At iteration $k + 1$ in the outer loop, Anderson acceleration first computes the residual

$$h_2^{(k)} := G_2(\rho^{(k)}) - \rho^{(k)}, \quad (48)$$

then solves the least-squares problem

$$\alpha^* := \operatorname{argmin}_{\alpha \in \mathbb{R}^{m_k+1}} \left\{ \left\| \sum_{i=0}^{m_k} \alpha_i h_2^{(k-i)} \right\|_2^2 : \sum_{i=0}^{m_k} \alpha_i = 1 \right\} \quad (49)$$

with $m_k := \min\{m, k\}$, and finally updates

$$\rho^{(k+1)} = \sum_{i=0}^{m_k} \alpha_i^* G_2(\rho^{(k-i)}). \quad (50)$$

To evaluate $G_2(\rho^{(k)})$ in (48), we need to solve the linear system (47) with ρ replaced by $\rho^{(k)}$, and assign $G_2(\rho^{(k)})$ to be x , the solution of (47). This system is solved using either Picard iteration or GMRES [21], which forms the inner loop of the nested procedure. In the following subsections, we discuss the applications of these two iterative solvers for evaluating $G_2(\rho^{(k)})$ in the inner loop.

3.2.1. Type-II – Anderson acceleration with Picard iteration

Let $\rho^{(k)}$ denote the k -th iterate in the outer loop. To evaluate $G_2(\rho^{(k)})$, we apply Picard iteration on an equivalent fixed-point formulation of (47). Specifically, at iteration $k + 1$ in the outer loop, Picard iteration updates

$$\rho^{(k,\ell+1)} = PL_{F(\rho^{(k)})}^{-1}(S\rho^{(k,\ell)} + s), \quad (51)$$

where $\rho^{(k,0)} = \rho^{(k)}$. Let $\rho^{(k,*)} := \lim_{\ell \rightarrow \infty} \rho^{(k,\ell)}$ denote the limit point of iterates generated by (51), then it follows that $G_2(\rho^{(k)}) = \rho^{(k,*)}$.

³ Picard iteration can serve as the nonlinear solver in the outer loop as well. However, we only consider Anderson acceleration here, since numerical results indicate that Picard iteration is not competitive in terms of computation time on this problem.

3.2.2. Type-II – Anderson acceleration with GMRES

Let $\rho^{(k)}$ still denote the k -th iterate in the outer loop. $G_2(\rho^{(k)})$ can also be evaluated by solving the linear system

$$(I - PL_{F(\rho^{(k)})}^{-1}S)G_2(\rho^{(k)}) = PL_{F(\rho^{(k)})}^{-1}s \quad (52)$$

using GMRES. In general, the GMRES solver is expected to converge in fewer iterations than the Picard iteration in (51). We will verify this in the numerical results reported in Section 4.

3.3. Synthetic acceleration for semiconductor equations

Synthetic acceleration (SA) schemes were first developed in [22,23] to improve efficiency of iterative solvers for transport equations. The basic idea of these schemes is to compute a coarse, cheap, or low-order correction term from residuals of the base iterative solver, and apply this correction to the current iterate to accelerate convergence of the base solver. As noted in [24], many of the synthetic acceleration schemes can be formulated as two-level multigrid algorithms.

In Section 3.3.1, we derive an SA scheme for the semiconductor equation (4a), and then apply this scheme to both type-I and type-II Picard iteration solvers considered in Sections 3.1.1 and 3.2.1. In Section 3.3.2, we follow the same approach as in [15,25,26] to formulate the synthetic acceleration as a preconditioner. We then accelerate the type-I and type-II Krylov solvers in Sections 3.1.2 and 3.2.2 by applying these solvers on the preconditioned problems.

3.3.1. SA scheme on Picard iteration

The derivation of SA schemes for steady-state linear transport equations with neutral particles can be found in, for example, [15, Section II.B.] and [17, Section 2-3]. These equations are well approximated in collisional by diffusion equations, which are used to compute cheap corrections to iterates in a solver, resulting in the diffusion synthetic acceleration (DSA) scheme [15,17,28,29].

For the semiconductor equations, it is known [1,37,41] that drift-diffusion equations serve as proper low-order approximations to the semiconductor equations in the drift-diffusion limit, as discussed in Section 2.1.2. In this section, we derive an SA scheme for semiconductor equations with the correction term computed by solving a drift-diffusion equation. The derivation is mostly a straightforward extension of the derivation for the DSA scheme in [15], and we include it here for completeness.

To derive the SA scheme, we first rewrite the semi-discrete semiconductor equation (33) as a steady-state equation

$$\mathcal{L}_{\mathcal{F}(\mathcal{P}f)}f = \mathcal{S}\mathcal{P}f + s. \quad (53)$$

Here \mathcal{P} denotes integration over the velocity domain and \mathcal{F} denotes the mapping from the particle concentration $\rho = \mathcal{P}f$ to the electric field $E = \mathcal{F}(\mathcal{P}f)$ via the Poisson equation (4b). Applying $\mathcal{L}_{\mathcal{F}(\mathcal{P}f)}^{-1}$ on both sides of (53) and solving the resulting equation with Picard iteration leads to

$$f^{(k+1/2)} = \mathcal{L}_{\mathcal{F}(\mathcal{P}f^{(k)})}^{-1}(\mathcal{S}\mathcal{P}f^{(k)} + s), \quad (54)$$

where index of the update is now $k + \frac{1}{2}$ instead of $k + 1$. Integrating (54) with respect to v gives

$$\rho^{(k+1/2)} = \mathcal{P}\mathcal{L}_{\mathcal{F}(\rho^{(k)})}^{-1}(\mathcal{S}\rho^{(k)} + s), \quad (55)$$

which is a continuous version of the type-I Picard iteration (41). To derive a correction for (55), we write (54) as

$$\mathcal{L}_{\mathcal{F}(\mathcal{P}f^{(k)})}f^{(k+1/2)} = \mathcal{S}\mathcal{P}f^{(k)} + s, \quad (56)$$

and subtract (56) from (53). By adding and subtracting terms in the resulting equation, we have

$$(\mathcal{L}_{\mathcal{F}(\mathcal{P}f)} - \mathcal{L}_{\mathcal{F}(\mathcal{P}f^{(k)})})f + \mathcal{L}_{\mathcal{F}(\mathcal{P}f^{(k)})}\psi - \mathcal{S}\mathcal{P}\psi = \mathcal{S}\mathcal{P}(f^{(k+1/2)} - f^{(k)}), \quad (57)$$

where $\psi := f - f^{(k+1/2)}$ denotes the error. If ψ can be computed by solving (57), then it can be used as a correction to Picard iterates by taking $f^{(k+1)} = f^{(k+1/2)} + \psi$. However, solving (57) is equivalent to finding f .

The SA scheme considered here computes corrections to Picard iterates by solving a reduced-order equation that approximates (57). To obtain the reduced-order equation, we apply \mathcal{P} to both sides of (57), which leads to

$$\mathcal{P}\mathcal{L}_{\mathcal{F}(\rho^{(k)})}\psi - \mathcal{P}\mathcal{S}\phi = \mathcal{P}\mathcal{S}(\rho^{(k+1/2)} - \rho^{(k)}), \quad (58)$$

where $\phi := \mathcal{P}\psi$ is the integral of ψ over the velocity domain. Here the first term in (57) vanishes since, for any electron distribution g and any electric fields E_1 and E_2 , it follows from (34) that $\mathcal{P}(\mathcal{L}_{E_1} - \mathcal{L}_{E_2})g = \beta^2(E_1 - E_2) \int_{\mathbb{R}} \partial_v g dv = 0$, provided that g goes to zero as $v \rightarrow \pm\infty$. Motivated by the drift-diffusion limit, we approximate the operators on the left-hand side of (58) by a drift-diffusion operator \mathcal{D}_E defined as

$$\mathcal{D}_E \phi = -\epsilon \partial_x (\omega^{-1} \partial_x \phi) + \epsilon \beta^2 \partial_x (\omega^{-1} E \phi) + \left(\frac{\delta}{\Delta t} \right) \phi. \quad (59)$$

Thus, (58) is approximated by

$$\mathcal{D}_{\mathcal{F}(\rho^{(k)})} \tilde{\phi} = \mathcal{PS}(\rho^{(k+1/2)} - \rho^{(k)}). \quad (60)$$

We correct the Picard iterate in (55) with $\tilde{\phi}$, a solution to (60), by taking $\rho^{(k+1)} = \rho^{(k+1/2)} + \tilde{\phi}$, which results in the SA scheme

$$\rho^{(k+1/2)} = \mathcal{PL}_{\mathcal{F}(\rho^{(k)})}^{-1}(\mathcal{S}\rho^{(k)} + s), \quad (61a)$$

$$\rho^{(k+1)} = \rho^{(k+1/2)} + \mathcal{D}_{\mathcal{F}(\rho^{(k)})}^{-1} \mathcal{PS}(\rho^{(k+1/2)} - \rho^{(k)}), \quad (61b)$$

where (61b) follows from the definition of $\tilde{\phi}$ in (60). The SA scheme based on the type-I Picard iteration (41) is then given by a discretized version of (61)

$$\rho^{(k+1/2)} = \mathbf{PL}_{\mathbf{F}(\rho^{(k)})}^{-1}(\mathbf{S}\rho^{(k)} + \mathbf{s}), \quad (62a)$$

$$\rho^{(k+1)} = \rho^{(k+1/2)} + \mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{PS}(\rho^{(k+1/2)} - \rho^{(k)}), \quad (62b)$$

where $\mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1}$ denotes the solution procedure of the drift-diffusion equation (60) using the DDG-IC solver presented in Section 2.4.

An analogous SA scheme based on the type-II Picard iteration (51) can be derived by repeating the analysis above with minor modification in the indices in (54). The resulting SA takes the form

$$\rho^{(k,\ell+1/2)} = \mathbf{PL}_{\mathbf{F}(\rho^{(k)})}^{-1}(\mathbf{S}\rho^{(k,\ell)} + \mathbf{s}), \quad (63a)$$

$$\rho^{(k,\ell+1)} = \rho^{(k,\ell+1/2)} + \mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{PS}(\rho^{(k,\ell+1/2)} - \rho^{(k,\ell)}), \quad (63b)$$

where $\mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1}$ is still performed using the DDG-IC solver in Section 2.4.

3.3.2. Preconditioner form of the SA scheme

The SA scheme in Section 3.3.1 is derived specifically for Picard iteration. It is well-known [15,25,26] that many SA schemes can be formulated as preconditioners. Following this approach, we derive the SA scheme in the preconditioner forms for the type-I Anderson acceleration and the type-II GMRES solver considered in Sections 3.1.2 and 3.2.2, respectively.

We first consider the type-I Anderson acceleration in Sections 3.1.2. In this case, the SA scheme applies Anderson acceleration (42)–(44) to a preconditioned version of fixed-point problem (40). This preconditioned problem is derived from the SA scheme based on type-I Picard iteration (62) by first rewriting the correction process (62b) in the residual form as

$$\rho^{(k+1)} - \rho^{(k)} = (\mathbf{I} + \mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{PS})(\rho^{(k+1/2)} - \rho^{(k)}). \quad (64)$$

Plugging (62a) into (64) then gives

$$\rho^{(k+1)} - \rho^{(k)} = -(\mathbf{I} + \mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{PS})(\mathbf{I} - \mathbf{PL}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{S})\rho^{(k)} - \mathbf{PL}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{s}, \quad (65)$$

which is equivalent to a standard Picard iteration update on the (preconditioned) fixed-point problem

$$\rho = \tilde{\mathbf{G}}_1(\rho) := \rho - (\mathbf{I} + \mathbf{D}_{\mathbf{F}(\rho)}^{-1} \mathbf{PS})(\mathbf{I} - \mathbf{PL}_{\mathbf{F}(\rho)}^{-1} \mathbf{S})\rho - \mathbf{PL}_{\mathbf{F}(\rho)}^{-1} \mathbf{s}. \quad (66)$$

Here (66) is a preconditioned version of (40) with preconditioner $(\mathbf{I} + \mathbf{D}_{\mathbf{F}(\rho)}^{-1} \mathbf{PS})$. We obtain the SA scheme based on type-I Anderson acceleration by replacing each \mathbf{G}_1 in (42)–(44) with $\tilde{\mathbf{G}}_1$.

For the type-II GMRES solver in Section 3.2.2, we follow a similar approach and use (63) to derive a preconditioned version of the linear system (52):

$$(\mathbf{I} + \mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{PS})(\mathbf{I} - \mathbf{PL}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{S})\mathbf{G}_2(\rho^{(k)}) = (\mathbf{I} + \mathbf{D}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{PS})\mathbf{PL}_{\mathbf{F}(\rho^{(k)})}^{-1} \mathbf{s}. \quad (67)$$

Therefore, the SA scheme based on type-II GMRES solver computes $\mathbf{G}_2(\rho^{(k)})$ by solving the preconditioned system (67).

4. Numerical results

The iterative solvers from the previous section are tested and compared on the one-dimensional silicon $n^+ - n - n^+$ diode problem [3,5,8,30,31] with different collision frequencies. In Section 4.1, we describe the silicon diode problem and state the implementation details. In Section 4.2, we consider “single-scale” problems where the collision frequency is assumed to be constant throughout the spatial domain. Results from these single-scale tests illustrate characteristics of the different solvers. In Section 4.3, we consider more realistic “multiscale” problems with collision frequencies varying in the spatial domain. We first model the diode using the collision frequency specified in [30] which depends on the doping profile. We then consider a more challenging problem with collision frequency that changes more drastically over the spatial domain.

4.1. Silicon diode problem setup and implementation details

In the silicon (Si) $n^+ - n - n^+$ diode problem, we simulate the electron movement in an one-dimension Si diode of length $L = 0.6 \mu\text{m}$ with a bias voltage $V_{\text{bias}} = 1 \text{ V}$. Here the electron charge, the effective electron mass, and the electric permittivity of the Si material are respectively given by $q_e = 1.602 \times 10^{-19} \text{ C}$, $m = 2.368 \times 10^{-31} \text{ kg}$, $\epsilon_p = 1.034 \times 10^{-10} \text{ F/m}$. The Boltzmann constant $k_B = 1.38 \times 10^{-23} \text{ J/K}$, and the lattice temperature $T = 300 \text{ K}$. As in [30], for $x \in [0, 0.6] (\mu\text{m})$, the doping profile is

$$D(x) = \begin{cases} 2 \times 10^{21} \text{ m}^{-3}, & x \in [0.1, 0.5] (\mu\text{m}) \\ 5 \times 10^{23} \text{ m}^{-3}, & \text{otherwise} \end{cases}. \quad (68)$$

With these data, we apply the following scaling: $x_0 = 10^{-6} \text{ m}$, $[\Phi] = 1 \text{ V}$, and $D_0 = 10^{21} \text{ m}^{-3}$. We set $\zeta = 1$ in (4b) (low-field scaling). The velocities in (5) take values

$$\Theta^{1/2} = 1.322 \times 10^5 \text{ m/s}, \quad B_0 = 8.225 \times 10^5 \text{ m/s}, \quad \text{and} \quad C_0 = 1.024 \times 10^6 \text{ m/s}. \quad (69)$$

We set the reference velocity to be $v_0 = \max\{\Theta^{1/2}, B_0, C_0\} = C_0$. Thus, the ratios in (4) are $\alpha = 0.129$, $\beta = 0.803$, and $\gamma = 1$. The nondimensional doping profile on $x \in [0, 0.6]$ is

$$D(x) = \begin{cases} 2, & x \in [0.1, 0.5] \\ 500, & \text{otherwise} \end{cases}. \quad (70)$$

In the numerical simulations, we impose smooth transitions into the doping profile as in [31]. These smooth transitions are constructed by cubic splines, and the transition regions are of width 0.04, centered at 0.1 and 0.5.

The initial condition and the incoming boundary data for the kinetic equation (4a) are respectively given by $f(t_0, x, v) = D(x)M_{\alpha^2}(v)$ and

$$f(t, 0, v) = D(0)M_{\alpha^2}(v), \quad \forall v > 0, \quad f(t, L, v) = D(L)M_{\alpha^2}(v), \quad \forall v < 0. \quad (71)$$

Without loss of generality, we let $\Phi_0 = \Phi(t, 0) = 0 \text{ V}$ in the Poisson equation (4b). The boundary data for (4b) then become $\Phi(t, 0) = 0 \text{ V}$ and $\Phi(t, L) = V_{\text{bias}}$. Since $[\Phi] = 1 \text{ V}$, the scaled boundary data are $\Phi(t, 0) = 0$ and $\Phi(t, L) = 1$.

The computation is performed on a truncated domain $\mathcal{D} := [0, 0.6] \times [-2, 2]$. The velocity space is truncated from \mathbb{R} to $[-2, 2]$, i.e., $v_{\text{max}} = 2$, since the value of Maxwellian $M_{\alpha^2}(v)$ is smaller than the machine precision when $|v| > 2$. We discretize \mathcal{D} into 200×50 uniform rectangular elements of size $\Delta x \times \Delta v$, and solve the semiconductor model (4) from initial time $t_0 = 0$ to final time $t_f = 0.5$, at which point the system is essentially in steady state. The kinetic equation (4a) is solved using the fast sweeping algorithm detailed in Section 2.2, the Poisson equation (4b) is solved via the continuous Galerkin method discussed in Section 2.3, and the drift-diffusion equation (60), which is involved in the synthetic acceleration procedure, is solved by the direct discontinuous Galerkin method in Section 2.4. The Poisson equation and the drift-diffusion equation are both solved on a uniform mesh with 200 elements on $[0, 0.6]$. We choose the parameter $m = 3$ in Anderson acceleration (42)–(44). The relative tolerances for the type-I and type-II iterative solvers are set to 10^{-8} , while the relative tolerance in the GMRES solver for solving (25) in the fast sweeping algorithm is set to 10^{-10} . The lower tolerance on the fast sweeping GMRES solver is due to the fact that it is used to evaluate L_E^{-1} , which is a fundamental building block in both type-I and type-II solvers. The iterative solvers are terminated once the relative residual is below the set tolerance, or when the solvers reach the maximum allowed number of iterations, which is set to 10,000 for all solvers. This number is set to be large solely for studying the behavior of different iterative solvers. For practical applications, the maximum allowed number of iterations should be set much lower.

In the following sections, we consider the diode problem described in this subsection with various collision frequencies and compare the performance of the eight iterative solvers introduced in Section 3. We also make a formal efficiency comparison of the proposed implicit scheme to standard explicit schemes and standard implicit-explicit (IMEX) asymptotic preserving (AP) schemes. In all tests, the implicit time step is chosen to be $\Delta t = \Delta x$. We note that the CFL condition for standard explicit schemes takes the form $\Delta t \leq \min\{C_1 \delta \Delta x / v_{\text{max}}, C_2 \delta \epsilon / \omega_{\text{max}}\}$ and the CFL condition for standard IMEX-AP

Table 1

Single-scale problem with $\omega(x) = 1$, $\delta = \epsilon = 0.200$, and $\Delta t = \Delta x$. Averaged iteration counts and total computation time (sec) for the compared solvers and their accelerated variants.

Solver	SA	Iteration		Total Time	Solver	SA	Iteration			Total Time
		FP	SW				FP	LS	SW	
Type-I	N	4.6	1.3	8.8	Type-II	N	3.2	3.0	1.3	21.4
PI	Y	4.5	1.3	9.7	AA/PI	Y	3.2	2.9	1.3	22.5
Type-I	N	3.0	1.4	7.9	Type-II	N	3.3	1.4	3.2	43.7
AA	Y	3.3	1.3	9.2	AA/GMRES	Y	7.3	2.1	3.3	97.8

schemes takes the form $\Delta t \leq \max\{C_3\delta\Delta x/v_{\max}, C_4\delta\omega_{\min}\Delta x^2/\epsilon\}$ [53], where v_{\max} , v_{\min} , ω_{\max} and ω_{\min} are the maximum and minimum values of the velocity and collision frequency over the spatial domain, respectively. For simplicity, we assume that the $O(1)$ constants C_1 , C_2 , C_3 , and C_4 are all equal to one. Here we do not consider the time step restrictions associated to Δv , since Δx is much smaller than Δv . Also, the condition $\Delta t \leq \delta\epsilon/\omega_{\max}$ for standard explicit schemes is never active in problems tested in this section due to the small value of Δx . In the formal comparison, we assume that an explicit or IMEX step roughly requires the same computation time as one sweeping iteration in an implicit time steps. Thus for each iterative solver, we compute the ratio

$$\chi_{\text{EX/IMEX}} := \frac{\frac{\Delta t_{\text{IM}}}{\Delta t_{\text{EX/IMEX}}}}{\text{total number of sweeping iterations per implicit time step}} \quad (72)$$

as an efficiency indicator of the proposed scheme. We also note that parallelization of the explicit or IMEX updates is often possible, while the sweeping procedure in the implicit scheme requires serial implementation. We do not take this fact into account in the comparison.

4.2. Single-scale test

In this section, we test the iterative solvers on problems with constant collision frequency on the spatial domain. We first consider the low collision case, where the electron mobility is approximated (see [30]) by

$$\mu_{\text{Si}}(x) = 0.0088 + \frac{0.1793 \times 10^{23}}{1.4320 \times 10^{23} + D(x)} \quad (73)$$

with $D(x) = 2 \times 10^{21} \text{ m}^{-3}$, which determines the collision frequency $\omega(x) = \frac{q_e}{m\mu_{\text{Si}}(x)}$. We scale the collision frequency by $\omega_0 = 5.114 \times 10^{12} \text{ s}^{-1}$, and the nondimensional collision frequency is $\omega(x) = 1$. The Knudsen number is $\epsilon = \frac{v_0}{x_0\omega_0} = 0.200$ and we choose $\delta = \epsilon = 0.200$.

Table 1 reports the averaged iteration counts and total computation time for each iterative solver. For type-I methods, column “FP” reports the averaged iteration counts per time step for solving the fixed-point problem (40), and column “SW” gives the averaged GMRES iteration counts per FP iteration for solving (25) from the fast sweeping algorithm when computing \mathbf{L}_E^{-1} in the FP iteration. For type-II methods, column “FP” reports the averaged outer-loop iteration counts per time step for solving the fixed-point problem (46), column “LS” shows the averaged inner-loop iteration counts per outer-loop iteration for solving the linear system (47) when evaluating \mathbf{G}_2 in (46), and column “SW” gives the averaged GMRES iteration counts for solving (25) in each inner-loop iteration. As for the solvers, “PI” and “AA” stand for Picard iteration and Anderson acceleration, respectively.

From the results reported in Table 1, we first observe that the accelerated solvers are slower than the unaccelerated ones. This result is to be expected, since the system is away from the drift-diffusion limit due to the relatively small collision frequency. We also observe that the type-I solvers are faster than the type-II solvers. We conclude that for the type-II solvers, the additional computation cost of solving the linear system (47) when evaluating \mathbf{G}_2 outweighs the benefit of the more accurate updates for the fixed-point problem. As we expected, the Krylov-type solvers (Anderson acceleration and GMRES) converge in fewer iterations than Picard iteration does. For type-I solvers, this results in less computation time, while for type-II solvers, the higher computation time per iteration of GMRES makes it slower than Picard iteration.

For this problem, the explicit and IMEX-AP CFL conditions both take the form $\Delta t \leq \delta\Delta x/v_{\max}$, which results in time steps that are 10x smaller than the implicit time step $\Delta t = \Delta x$ used in the test. For each tested iterative solver, the value of the efficient indicator χ defined in (72) is reported in Table 2. Here Iter_{SW} denotes the total number of sweeping iterations required in one implicit step. From Table 2, the proposed scheme with type-I solvers results in $\chi > 1$ when comparing to both the explicit and IMEX-AP schemes. When using type-II solvers, $\chi < 1$ as the higher iteration counts outweigh the benefit of larger implicit time steps.

We next test the iterative solvers on problems with large collision frequency that is 100x of the one in the previous problem, and we scale the collision frequency by $\omega_0 = 5.114 \times 10^{14} \text{ s}^{-1}$ so that the nondimensional collision frequency is

Table 2

Single-scale problem with $\omega(x) = 1$, $\delta = \epsilon = 0.200$, and $\Delta t = \Delta x$. The values of the efficiency indicator χ (72) and the total number of sweeping iterations in an implicit step (Iter_{SW}) are reported. This serves as a formal efficiency comparison between the proposed implicit scheme and the standard explicit and IMEX-AP schemes.

1 implicit step \sim 10 IMEX-AP steps \sim 10 explicit steps									
Solver	SA	Iter_{SW}	χ_{IMEX}	χ_{EX}	Solver	SA	Iter_{SW}	χ_{IMEX}	χ_{EX}
Type-I PI	N	6.0	1.67	1.67	Type-II AA/PI	N	12.5	0.80	0.80
	Y	5.9	1.69	1.69		Y	12.1	0.83	0.83
Type-I AA	N	4.2	2.38	2.38	Type-II AA/GMRES	N	14.8	0.68	0.68
	Y	4.3	2.33	2.33		Y	50.6	0.20	0.20

Table 3

Single-scale problem with $\omega(x) = 1$, $\delta = \epsilon = 0.002$, and $\Delta t = \Delta x$. Averaged iteration counts and total computation time (sec) for the compared solvers and their accelerated variants.

Solver	SA	Iteration		Total Time	Solver	SA	Iteration			Total Time
		FP	SW				FP	LS	SW	
Type-I PI	N	1399.3	1.0	2011.6	Type-II AA/PI	N	2.5	440.1	1.0	2165.2
	Y	131.9	1.0	232.9		Y	6.1	84.4	1.0	918.0
Type-I AA	N	49.7	1.0	99.9	Type-II AA/GMRES	N	6.1	22.7	2.1	392.2
	Y	29.0	1.1	67.8		Y	7.6	9.2	2.1	218.3

Table 4

Single-scale problem with $\omega(x) = 1$, $\delta = \epsilon = 0.002$, and $\Delta t = \Delta x$. The values of the efficiency indicator χ (72) and the total number of sweeping iterations in an implicit step (Iter_{SW}) are reported. This serves as a formal efficiency comparison between the proposed implicit scheme and the standard explicit and IMEX-AP schemes.

1 implicit step \sim 333 IMEX-AP steps \sim 1000 explicit steps									
Solver	SA	Iter_{SW}	χ_{IMEX}	χ_{EX}	Solver	SA	Iter_{SW}	χ_{IMEX}	χ_{EX}
Type-I PI	N	1399.3	0.24	0.71	Type-II AA/PI	N	1100.3	0.30	0.91
	Y	131.9	2.53	7.58		Y	514.8	0.65	1.94
Type-I AA	N	49.7	6.71	20.12	Type-II AA/GMRES	N	290.8	1.15	3.44
	Y	32.0	10.42	31.25		Y	146.8	2.27	6.81

still $\omega(x) = 1$. The Knudsen number is then $\epsilon = \frac{v_0}{\chi_0 \omega_0} = 0.002$ and again we choose $\delta = \epsilon = 0.002$. We still choose the time step to be $\Delta t = \Delta x$. For this problem, the system is close to the drift-diffusion limit due to the large collision frequency. Thus we expect that the drift-diffusion based synthetic acceleration would provide sufficient accurate corrections and result in faster convergence for the iterative solvers. The iteration counts and computation time for this problem are reported in Table 3. From these results, we observe that SA schemes indeed require fewer number of iteration and speed up the base iterative solvers from 1.5x to 8.7x. We also note that the computation time per iteration remains roughly the same, which implies that the time spent on computing the drift-diffusion correction term is essentially negligible.

For this problem, the standard explicit time step satisfies $\Delta t \leq \delta \Delta x / v_{\text{max}}$, which is 1000x smaller than the implicit time step $\Delta t = \Delta x$. The standard IMEX-AP time step satisfies $\Delta t \leq \delta \omega_{\text{min}} \Delta x^2 / \epsilon$, which is 333x smaller than the implicit time step. From Table 4, in most cases, the proposed scheme results in $\chi > 1$ when comparing to both the explicit and IMEX-AP schemes. The exceptions are the unaccelerated type-I PI solver and the type-II AA/PI solvers.

4.3. Multiscale test

In this section, we test the solvers on multiscale problems with collision frequencies varying in the spatial domain. The first multiscale problem is the “standard” silicon diode problem from [30,31]. Here the collision frequency is determined by the approximated electron mobility based on the doping profile. Specifically, with the approximate formula $\mu_{\text{Si}}(x) = 0.0088 + \frac{0.1793 \times 10^{23}}{1.4320 \times 10^{23} + D(x)}$ at $T = 300$ K and the doping profile $D(x)$ in (68), the electron mobility is given by

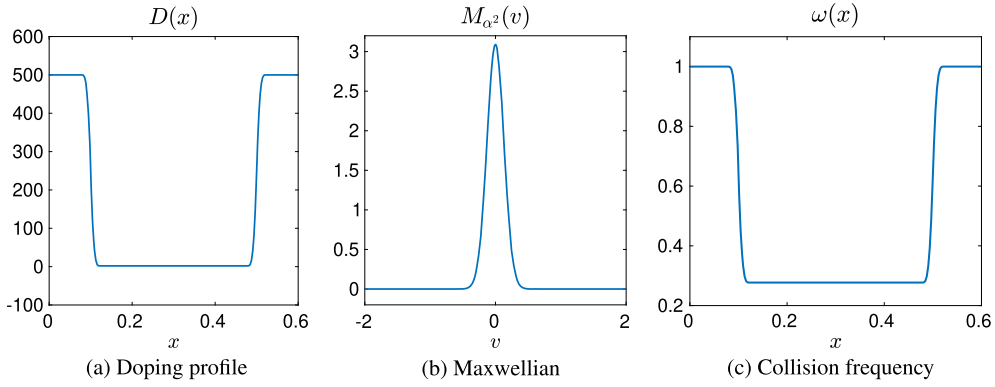


Fig. 2. The scaled doping profile, Maxwellian, and collision frequency for the standard $n^+ - n - n^+$ diode problem.

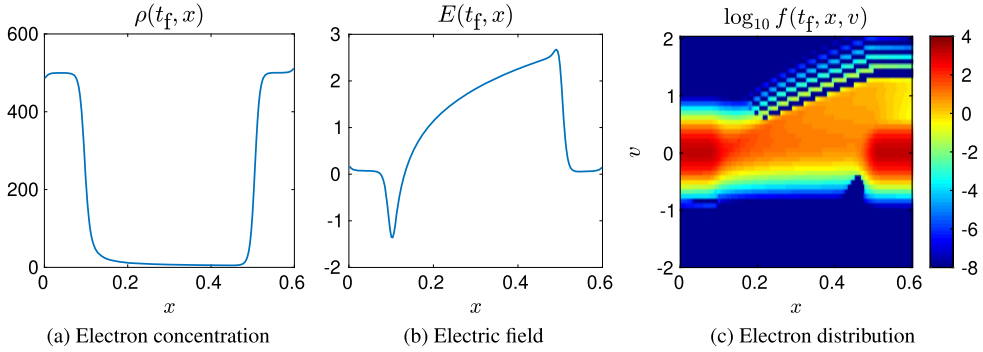


Fig. 3. The electron concentration, electric field, and electron distribution at the final time $t_f = 0.5$ for the standard $n^+ - n - n^+$ diode problem with scaled collision frequency given in (75). Here the electron distribution f is plotted in Fig. 3c in logarithmic scale, and the magnitude of the observed oscillations at positive velocities is smaller than 0.1% of the maximum value of f . These oscillations are stable artifacts that can be mitigated or removed by refining the velocity discretization. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\mu_{\text{Si}}(x) = \begin{cases} 0.1323 \frac{\text{m}^2}{\text{Vs}}, & x \in [0.1, 0.5] \text{ (}\mu\text{m)} \\ 0.0367 \frac{\text{m}^2}{\text{Vs}}, & \text{otherwise} \end{cases}. \quad (74)$$

The nondimensional collision frequency on $x \in [0, 0.6]$ is then

$$\omega(x) = \begin{cases} 0.277, & x \in [0.1, 0.5] \\ 1, & \text{otherwise} \end{cases} \quad (75)$$

with the scaling $\omega_0 = 1.843 \times 10^{13} \text{ s}^{-1}$. Here the Knudsen number $\epsilon = \frac{v_0}{x_0 \omega_0} = 0.056$, and we choose $\delta = \epsilon = 0.056$. As mentioned in Section 4.1, the doping profile used in the numerical tests includes the smooth transitions as in [31]. The collision frequency at these transition regions are computed from the smooth doping profile using the approximate formula given above.

Fig. 2 shows the scaled doping profile $D(x)$ with smooth transitions, Maxwellian $M_{\alpha^2}(v)$, and collision frequency $\omega(x)$ in this silicon diode problem. Fig. 3 illustrate the electron concentration ρ , the electric field E , and the full electron distribution f at the final, steady-state time $t_f = 0.5$. As expected, all tested solvers give identical results up to the set numerical tolerance, and these results agree with the those reported in other places in the literatures, such as [5,30].

Table 5 reports the iteration counts and computation time for each of the iterative solvers. The system is away from the drift-diffusion limit with the given collision frequency. Hence the results in Table 5 are similar to the ones in Table 1 and the drift-diffusion synthetic acceleration does not accelerate the convergence of the iterative solvers. The type-I solvers still outperform the type-II solvers for this test.

Here the implicit time step $\Delta t = \Delta x$ is roughly 35x larger than the standard explicit and IMEX-AP time steps $\Delta t \leq \delta \Delta x / v_{\text{max}}$. We observe in Table 6 that the implicit scheme generally gives $\mathcal{X} > 1$, except when the accelerated type-II AA/GMRES solver is used.

We next consider another multiscale problem with much stronger variation in the collision frequency. Specifically, the scaled collision frequency considered in this problem is

Table 5

Standard $n^+ - n - n^+$ diode problem with $\omega(x)$ given in (75), $\delta = \epsilon = 0.056$, and $\Delta t = \Delta x$. Averaged iteration counts and total computation time (sec) for the compared solvers and their accelerated variants.

Solver	SA	Iteration		Total Time	Solver	SA	Iteration			Total Time
		FP	SW				FP	LS	SW	
Type-I	N	3.8	1.4	7.6	Type-II	N	2.6	4.8	1.4	28.6
PI	Y	4.6	1.6	10.1	AA/PI	Y	2.6	4.5	1.4	29.6
Type-I	N	2.6	1.4	7.3	Type-II	N	2.8	2.2	3.7	49.8
AA	Y	2.8	1.4	8.2	AA/GMRES	Y	6.0	3.5	3.6	110.7

Table 6

Standard $n^+ - n - n^+$ diode problem with $\omega(x)$ given in (75), $\delta = \epsilon = 0.056$, and $\Delta t = \Delta x$. The values of the efficiency indicator χ (72) and the total number of sweeping iterations in an implicit step (Iter_{SW}) are reported. This serves as a formal efficiency comparison between the proposed implicit scheme and the standard explicit and IMEX-AP schemes.

1 implicit step \sim 35 IMEX-AP steps \sim 35 explicit steps									
Solver	SA	Iter_{SW}	χ_{IMEX}	χ_{EX}	Solver	SA	Iter_{SW}	χ_{IMEX}	χ_{EX}
Type-I	N	5.3	6.58	6.58	Type-II	N	17.5	2.00	2.00
PI	Y	7.4	4.76	4.76	AA/PI	Y	16.4	2.14	2.14
Type-I	N	3.6	9.62	9.62	Type-II	N	22.8	1.54	1.54
AA	Y	3.9	8.93	8.93	AA/GMRES	Y	75.6	0.46	0.46

$$\omega(x) = \begin{cases} 1 \times 10^{-5}, & x \in [0.1, 0.5] \\ 1, & \text{otherwise} \end{cases} \quad (76)$$

with $\omega_0 = 2.048 \times 10^{15} \text{ s}^{-1}$ and $\delta = \epsilon = \frac{v_0}{\chi_0 \omega_0} = 0.0005$. The maximum collision frequency in this problem is higher than the one in the first multiscale problem by two orders of magnitude. Similarly, the minimum collision frequency here is two orders of magnitude smaller than the one in the first multiscale problem.

The iteration counts and computation time for this problem are reported in Table 7. These results show that for this multiscale problem, synthetic acceleration with the drift-diffusion model still speeds up the convergence of convergent iterative solvers, but the difference is not as significant as in the single-scale case with large collision frequency. We suspect that the multiscale nature of this problem makes the drift-diffusion acceleration ineffective, especially in regions with relatively low collision frequency. We also observe that three of the tested solvers (accelerated type-I PI, standard and accelerated type-II AA/GMRES) result in diverging residuals (denoted as ∞). The divergent result of the accelerated type-I PI scheme indicates that applying the drift-diffusion based synthetic acceleration on multiscale problems may lead to unstable schemes. This observation is related to a known deficiency of diffusion synthetic acceleration (DSA). Specifically, it has been reported in [54] and analyzed in [55] that DSA becomes unstable and gives divergent results when applied to highly collisional problems. To guarantee convergence, the spatial discretization of the diffusion equation has to be consistent to the discretization of the original transport equation. We refer the reader to [15] and [16] for a complete discussion. As for the type-II AA/GMRES schemes, we suspect that the failure of convergence is due to the fact that the linear system (47) becomes ill-conditioned in this multiscale setup.

Finally, we note that on this problem, the standard explicit time step needs to satisfy $\Delta t \leq \delta \epsilon / \omega_{\text{max}}$, which is 12000x smaller than the implicit time step $\Delta t = \Delta x$, while the standard IMEX-AP time step satisfies $\Delta t \leq \delta \Delta x / v_{\text{max}}$, which is 4000x smaller than the implicit time step. As shown in Table 8, the efficient indicator $\chi > 1$ for all iterative solvers that lead to a convergent implicit scheme. Among these iterative solvers, the most efficient one is the accelerated type-I AA solver.

5. Conclusions and discussion

We have proposed a fully implicit numerical scheme for solving Boltzmann-Poisson systems with an approximate collision operator that describes linear relaxation. At each implicit time step, the updated solution comes from a nonlinear fixed-point problem. We have formulated Boltzmann-Poisson systems as two types of fixed-point problems: the type-I problems that are solved in a single strongly coupled iterative loop and the type-II problems that require two nested iterative loops. We have applied Picard iteration and Anderson acceleration to solve the type-I problems. For the type-II problems, the outer loop of the nested iterative procedure is performed by Anderson acceleration while the inner loop uses either Picard iteration or GMRES. The performance of these iterative solvers and their synthetically accelerated (SA) variants is compared on several scaled versions of a standard silicon diode problem. Numerical results show that (i) solving the type-I fixed-point problems requires fewer iterations than solving the type-II problems and (ii) Anderson acceleration is more efficient than

Table 7

Multiscale problem with $\omega(x)$ given in (76), $\delta = \epsilon = 0.0005$, and $\Delta t = \Delta x$. Averaged iteration counts and total computation time (sec) for the compared solvers and their accelerated variants. Here ∞ denotes the case that diverging residual is observed.

Solver	SA	Iteration		Total Time	Solver	SA	Iteration			Total Time
		FP	SW				FP	LS	SW	
Type-I PI	N	520.4	1.0	736.2	Type-II AA/PI	N	1.6	143.0	1.0	508.0
	Y	∞	∞	∞		Y	2.0	73.5	1.0	317.6
Type-I AA	N	17.1	1.0	27.4	Type-II AA/GMRES	N	∞	∞	∞	∞
	Y	10.0	1.0	19.1		Y	∞	∞	∞	∞

Table 8

Multiscale problem with $\omega(x)$ given in (76), $\delta = \epsilon = 0.0005$, and $\Delta t = \Delta x$. The values of the efficiency indicator X (72) and the total number of sweeping iterations in an implicit step (Iter_{SW}) are reported. This serves as a formal efficiency comparison between the proposed implicit scheme and the standard explicit and IMEX-AP schemes.

1 implicit step \sim 4000 IMEX-AP steps \sim 12000 explicit steps									
Solver	SA	Iter_{SW}	X_{IMEX}	X_{EX}	Solver	SA	Iter_{SW}	X_{IMEX}	X_{EX}
Type-I PI	N	520.4	7.69	23.06	Type-II AA/PI	N	229.0	17.47	52.40
	Y	∞	—	—		Y	93.7	42.69	128.07
Type-I AA	N	17.1	233.92	701.75	Type-II AA/GMRES	N	∞	—	—
	Y	10.0	402.01	1206.03		Y	∞	—	—

Picard iteration on type-I problems, in terms of both iteration counts and computation time. These results also confirm that SA schemes with a drift-diffusion model converge faster than the standard iterative solvers when the system is near the drift-diffusion limit (when collision frequency is large). For systems away from this limit, there is no observed advantage in using these accelerated schemes. To address this issue, some potential approaches to be investigated in the future include (i) modifying the drift-diffusion SA schemes to incorporate the boundary conditions as proposed in [56] for a neutron transport problem and (ii) deriving SA schemes based on low-order/low-cost approximations other than the drift-diffusion equation. For (ii), possible candidates of such approximation include the S_2 -SA scheme considered in [26,27] and the more general, two-level multigrid algorithms as the one found in [57, Section 6.2].

Another potential future work is to apply the hybrid schemes [58,59] proposed for linear transport equations. These schemes decompose the transport solution into collisional and non-collisional components. At each time step, the hybrid schemes use cheap, low-resolution approximations for the collisional component and reserve expensive, high resolution approximations for the non-collisional component. Since the primary difficulty on solving the Boltzmann-Poisson system is the nonlinear coupling between the electric field and collision term, we expect that the hybrid approach would significantly lower the computation cost while producing quality solutions that are comparable to the ones given by a uniformly high-resolution solver.

There are several open problems that require further investigation. One of them is to extend the proposed method to higher order. High-order time discretization, e.g., SDIRK method, can be obtained by cascading the backward Euler scheme in Section 2.2.1. High-order phase-space discretization can be achieved by using the DG method in Section 2.2.2 with high-order elements. However, it is not trivial to expand the Poisson solver in Section 2.3 to high order without affecting the sweeping technique in Section 2.2.3, due to the sign change of electric field within a spatial cell, as discussed in Section 2.3 and [14, Remark 1]. Another important open problem is to perform a comprehensive efficiency comparison between the proposed fully implicit scheme and standard explicit/IMEX schemes. As opposed to the formal comparison results reported in Section 4, the comprehensive comparison should take both the computation time and consistency errors into account. In particular, while the larger time steps used in the fully implicit scheme speed up the solution procedure, they could lead to larger consistency errors in time, which may become the dominant error in the solution. A comprehensive efficiency comparison is needed to verify that, at least on some class of problems, the fully implicit solution indeed achieves some given precision in less computation time when comparing to the explicit/IMEX solutions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] P.A. Markowich, C.A. Ringhofer, C. Schmeiser, *Semiconductor Equations*, Springer-Verlag, New York, 1990.

- [2] D. Ferry, R. Grondin, *Physics of Submicron Devices*, Microdevices Series, Plenum Press, 1991.
- [3] C. Cercignani, I.M. Gamba, C.D. Levermore, A drift-collision balance for a Boltzmann-Poisson system in bounded domains, *SIAM J. Appl. Math.* 61 (2001) 1932–1958.
- [4] C. Jacoboni, P. Lugli, *The Monte Carlo Method for Semiconductor Device Simulation*, Computational Microelectronics, Springer, Vienna, 1989.
- [5] J.A. Carrillo, I.M. Gamba, C.-W. Shu, Computational macroscopic approximations to the one-dimensional relaxation-time kinetic system for semiconductors, *Physica D* 146 (2000) 289–306.
- [6] J.A. Carrillo, I.M. Gamba, A. Majorana, C.-W. Shu, A direct solver for 2D non-stationary Boltzmann-Poisson systems for semiconductor devices: a MESFET simulation by WENO-Boltzmann schemes, *J. Comput. Electron.* 2 (2003) 375–380.
- [7] J.A. Carrillo, I.M. Gamba, A. Majorana, C.-W. Shu, 2D semiconductor device simulations by WENO-Boltzmann schemes: efficiency, boundary conditions and comparison to Monte Carlo methods, *J. Comput. Phys.* 214 (2006) 55–80.
- [8] Y. Cheng, I.M. Gamba, A. Majorana, C.-W. Shu, A discontinuous Galerkin solver for Boltzmann-Poisson systems in nano devices, *Comput. Methods Appl. Mech. Eng.* 198 (2009) 3130–3150.
- [9] Y. Cheng, I.M. Gamba, A. Majorana, C.-W. Shu, Discontinuous Galerkin solver for Boltzmann-Poisson transients, *J. Comput. Electron.* 7 (2008) 119–123.
- [10] Y. Cheng, I.M. Gamba, A. Majorana, C.-W. Shu, A brief survey of the discontinuous Galerkin method for the Boltzmann-Poisson equations, *SeMA J.* 54 (2011) 47–64.
- [11] C. Ringhofer, A mixed spectral-difference method for the steady state Boltzmann-Poisson system, *SIAM J. Numer. Anal.* 41 (2003) 64–89.
- [12] S. Jin, L. Pareschi, Discretization of the multiscale semiconductor Boltzmann equation by diffusive relaxation schemes, *J. Comput. Phys.* 161 (2000) 312–330.
- [13] G. Dimarco, L. Pareschi, V. Rispoli, Implicit-explicit Runge-Kutta schemes for the Boltzmann-Poisson system for semiconductors, *Commun. Comput. Phys.* 15 (2014) 1291–1319.
- [14] C. Garrett, C. Hauck, A fast solver for implicit integration of the Vlasov-Poisson system in the Eulerian framework, *SIAM J. Sci. Comput.* 40 (2018) B483–B506.
- [15] M.L. Adams, E.W. Larsen, Fast iterative methods for discrete-ordinates particle transport calculations, *Prog. Nucl. Energy* 40 (2002) 3–159.
- [16] E.W. Larsen, J.E. Morel, *Advances in Discrete-Ordinates Methodology*, Springer Netherlands Dordrecht, 2010, pp. 1–84.
- [17] E.E. Lewis, J.W.F. Miller, *Computational Methods in Neutron Transport*, John Wiley and Sons, New York, 1984.
- [18] E. Hairer, S. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 1993.
- [19] D.G. Anderson, Iterative procedures for nonlinear integral equations, *J. ACM* 12 (1965) 547–560.
- [20] H. Walker, P. Ni, Anderson acceleration for fixed-point iterations, *SIAM J. Numer. Anal.* 49 (2011) 1715–1735.
- [21] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (1986) 856–869.
- [22] V. Lebedev, The iterative KP method for the kinetic equation, in: *Proc. Conf. on Mathematical Methods for Solution of Nuclear Physics Problems*, vol. 93, Nov. 17–20, 1964 (in Russian).
- [23] H. Kopp, Synthetic method solution of the transport equation, *Nucl. Sci. Eng.* 17 (1963) 65–74.
- [24] E.W. Larsen, *Transport Acceleration Methods as Two-Level Multigrid Algorithms*, Birkhäuser Basel, Basel, 1991, pp. 34–47.
- [25] K. Derstine, E. Gelbard, Use of the preconditioned conjugate gradient method to accelerate $s_{\text{sub } n}$ iterations, *Trans. Am. Nucl. Soc.* 50 (1985).
- [26] D.E. Bruss, J.E. Morel, J.C. Ragusa, S_2SA preconditioning for the S_n equations with strictly nonnegative spatial discretization, *J. Comput. Phys.* 273 (2014) 706–719.
- [27] L.J. Lorence Jr, J. Morel, E.W. Larsen, An S_2 synthetic acceleration scheme for the one-dimensional S_n equations with linear discontinuous spatial differencing, *Nucl. Sci. Eng.* 101 (1989) 341–351.
- [28] R.E. Alcouffe, A stable diffusion synthetic acceleration method for neutron transport iterations, *Trans. Am. Nucl. Soc.* 23 (1976).
- [29] R.E. Alcouffe, Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations, *Nucl. Sci. Eng.* 64 (1977) 344–355.
- [30] C. Cercignani, I.M. Gamba, J.W. Jerome, C.-W. Shu, A domain decomposition method for silicon devices, *Transp. Theory Stat. Phys.* 29 (2000) 525–536.
- [31] Z. Hu, R. Li, T. Lu, Y. Wang, W. Yao, Simulation of an n^+-n-n^+ diode by using globally-hyperbolically-closed high-order moment models, *J. Sci. Comput.* 59 (2014) 761–774.
- [32] S. Selberherr, H. Stippel, E. Strasser, *Simulation of Semiconductor Devices and Processes*, vol. 5, Springer, Vienna, 2012, <https://books.google.com/books?id=QtuoCAAQBAJ>.
- [33] F. Golse, C.D. Levermore, Hydrodynamic limits of kinetic models, in: *Topics in Kinetic Theory*, Fields Inst. Commun., Amer. Math. Soc., 2005, pp. 1–75.
- [34] L. Saint-Raymond, *Hydrodynamic Limits of the Boltzmann Equation*, 1971, Springer Science & Business Media, 2009.
- [35] T. Boyd, J. Sanderson, *The Physics of Plasmas*, Cambridge University Press, 2003.
- [36] R. Hazeltine, F. Waelbroeck, *The Framework of Plasma Physics*, Frontiers in Physics, Avalon Publishing, 2004.
- [37] F. Poupaud, Diffusion approximation of the linear semiconductor Boltzmann equation: analysis of boundary layers, *Asymptot. Anal.* 4 (1991) 293–317.
- [38] P. Stichel, D. Strothmann, Asymptotic analysis of the high field semiconductor Boltzmann equation, *Physica A* 202 (1994) 553–576.
- [39] F. Poupaud, Runaway phenomena and fluid approximation under high fields in semiconductor kinetic theory, *J. Appl. Math. Mech./Z. Angew. Math. Mech.* 72 (1992) 359–372.
- [40] N.B. Abdallah, M.L. Tayeb, Diffusion approximation for the one dimensional Boltzmann-Poisson system, *Discrete Contin. Dyn. Syst., Ser. B* 4 (2004) 1129–1142.
- [41] N. Masmoudi, M. Tayeb, Diffusion limit of a semiconductor Boltzmann-Poisson system, *SIAM J. Math. Anal.* 38 (2007) 1788–1807.
- [42] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer Berlin Heidelberg, 1996.
- [43] R.B. Lowrie, J.E. Morel, Methods for hyperbolic systems with stiff relaxation, *Int. J. Numer. Methods Fluids* 40 (2002) 413–423.
- [44] E.W. Larsen, J. Morel, Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes II, *J. Comput. Phys.* 83 (1989) 212–236.
- [45] J.-L. Guermond, G. Kanschat, Asymptotic analysis of upwind discontinuous Galerkin approximation of the radiative transport equation in the diffusive limit, *SIAM J. Numer. Anal.* 48 (2010) 53–78.
- [46] M.L. Adams, Discontinuous finite element transport solutions in thick diffusive problems, *Nucl. Sci. Eng.* 137 (2001) 298–333.
- [47] S. Brenner, R. Scott, *The Mathematical Theory of Finite Element Methods*, Texts in Applied Mathematics, Springer, New York, 2007.
- [48] A. Ern, J. Guermond, *Theory and Practice of Finite Elements*, Applied Mathematical Sciences, Springer, New York, 2013.
- [49] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems, *SIAM J. Numer. Anal.* 47 (2009) 675–698.
- [50] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections, *Commun. Comput. Phys.* (2010).
- [51] Z. Chen, H. Huang, J. Yan, Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes, *J. Comput. Phys.* 308 (2016) 198–217.
- [52] A. Toth, C. Kelley, Convergence analysis for Anderson acceleration, *SIAM J. Numer. Anal.* 53 (2015) 805–819.
- [53] S. Jin, Asymptotic preserving (AP) schemes for multiscale kinetic and hyperbolic equations: a review, *Riv. Mat. Univ. Parma* 3 (2012) 177–216.
- [54] E.M. Gelbard, L.A. Hageman, The synthetic method as applied to the S_n equations, *Nucl. Sci. Eng.* 37 (1969) 288–298.
- [55] W.H. Reed, The effectiveness of acceleration techniques for iterative methods in transport theory, *Nucl. Sci. Eng.* 45 (1971) 245–254.

- [56] D. Valougeorgis, Boundary treatment of the diffusion synthetic acceleration method for fixed-source discrete-ordinates problems in x - y geometry, *Nucl. Sci. Eng.* 100 (1988) 142–148.
- [57] K.E. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1997.
- [58] M.M. Crockatt, A.J. Christlieb, C.K. Garrett, C.D. Hauck, An arbitrary-order, fully implicit, hybrid kinetic solver for linear radiative transport using integral deferred correction, *J. Comput. Phys.* 346 (2017) 212–241.
- [59] C. Hauck, R. McClarren, A collision-based hybrid method for time-dependent, linear, kinetic transport equations, *Multiscale Model. Simul.* 11 (2013) 1197–1227.