

Received December 6, 2019, accepted January 13, 2020, date of publication January 17, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967125

# Steady-State Simulation of Semiconductor Devices Using Discontinuous Galerkin Methods

LIANG CHEN<sup>1</sup>, (Member, IEEE), AND HAKAN BAGCI<sup>1</sup>, (Senior Member, IEEE)

Division of Computer, Electrical, and Mathematical Science and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

Corresponding author: Liang Chen (liang.chen@kaust.edu.sa)

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award 2016-CRG5-2953.

**ABSTRACT** Design of modern nanostructured semiconductor devices often calls for simulation tools capable of modeling arbitrarily-shaped multiscale geometries. In this work, to this end, a discontinuous Galerkin (DG) method-based framework is developed to simulate steady-state response of semiconductor devices. The proposed framework solves a system of Poisson equation (in electric potential) and stationary drift-diffusion equations (in charge densities), which are nonlinearly coupled via the drift current and the charge distribution. This system is “decoupled” and “linearized” using the Gummel method and the resulting equations are discretized using a local DG scheme. The proposed framework is used to simulate geometrically intricate semiconductor devices with realistic models of mobility and recombination rate. Its accuracy is demonstrated by comparing the results to those obtained by the finite volume and finite element methods implemented in a commercial software package.

**INDEX TERMS** Discontinuous Galerkin method, drift-diffusion equations, multiphysics modeling, Poisson equation, semiconductor device modeling.

## I. INTRODUCTION

Simulation tools capable of numerically characterizing semiconductor devices play a vital role in device/system design frameworks used by the electronics industry as well as various related research fields [1]–[8]. Indeed, in the last several decades, numerous commercial and open source technology computer aided design (TCAD) tools, which implement various transport models ranging from semi-classical to quantum mechanical models, have been developed for this purpose [9]. Despite the recent trend of device miniaturization that requires simulators to account for quantum transport effects, many devices with larger dimensions (at the scale of  $1\mu m$ ) and with more complex geometries are being designed and implemented for various applications. Examples of these nanostructured devices range from photodiodes and phototransistors to solar cells, light emitting diodes, and photoconductive antennas [10]. Electric field-charge carrier interactions on these devices can still be accurately accounted for using semi-classical models, however, their numerical

simulation in TCAD raises challenges due to the presence of multi-scale and intricate geometric features.

Among the semi-classical approaches developed for modeling charge carrier transport, the drift-diffusion (DD) model is among the most popular ones because of its simplicity while being capable of explaining many essential characteristics of semiconductor devices [1]–[3]. One well-known challenge in using the DD model is the exponential variation of carrier densities, which renders standard numerical schemes used for discretizing the model unstable unless an extremely fine mesh is used. This challenge traces back to the convection-dominated convection-diffusion equations, whose solutions show sharp boundary layers. Various stabilization techniques have been proposed and incorporated with different discretization schemes [11]–[21]. The Scharfetter-Gummel (SG) method [11] has been one of the workhorses in semiconductor device modeling; it uses exponential functions to approximate the carrier densities so that the fine mesh requirement can be alleviated. The SG method has been first proposed for finite difference discretization, and then generalized to finite volume method (FVM) [12]–[17] and finite element method (FEM) [18]–[21].

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei<sup>1</sup>.

As mentioned above, many modern devices involve geometrically intricate structures. Therefore, FVM and FEM, which allow for unstructured meshes, have drawn more attention in recent years. However, the SG generalizations making use of FVM and FEM pose requirements on the regularity of the mesh [14], [16], [20]–[22]. For example, FVM requires boundary conforming Delaunay triangulations for two dimensional (2D) simulations and admissible partitions for three dimensional (3D) ones [14], [16], [22]. These requirements cannot be easily satisfied in mesh generation for devices with complex geometries [21], [22]. In addition, FEM stabilization techniques, such as the streamline upwind Petrov-Galerkin (SUPG) method [23], [24] and the Galerkin least-square (GLS) method [25], [26], have been used in simulation of semiconductor devices. However, SUPG suffers from “artificial” numerical diffusion [27]–[29]; and GLS leads to unphysical smearing of the boundary layers and does not preserve current conservation [27], [30].

Although significant effort has been put into the numerical solution of the convection-dominated convection-diffusion problem in the last three decades, especially in the applied mathematics community, a fully-satisfactory numerical scheme for general industrial problems is yet to be formulated and implemented, for example see [27], [28], [31]–[33] for surveys.

The discontinuous Galerkin (DG) method has attracted significant attention in several fields of computational science [34]–[38]. DG can be thought of as a hybrid method that combines the advantages of FVM and FEM. It uses local high-order expansions to represent/approximate the unknowns to be solved for. Each one of these expansions is defined on a single mesh element and is “connected” to other expansions defined on the neighboring elements via numerical flux. This approach equips DG with several advantages: The order of the local expansions can be changed individually, the mesh can be non-conformal (in addition to being unstructured), and the numerical flux can be designed to control the stability and accuracy characteristics of the DG scheme. More specifically, for semiconductor device simulations, the instability caused by the boundary layers can be alleviated without introducing much numerical diffusion. We should note here that for a given order of expansion  $p$ , DG requires a larger number of unknowns than FEM. However, the difference decreases as  $p$  gets larger, and for many problems, DG benefits from  $h$ - and/or  $p$ -refinement schemes [36], [38] and easily compensates for the small increase in the computational cost.

These properties render DG an attractive option for multi-scale simulations [29], [34]–[39], and indeed, its time-domain version has been used to solve different types of time-dependent transport equations (e.g., the hydrodynamic and energy-transport model [40], [41], the DD model [42]–[44], and the Boltzmann transport equation [45]) and applied to the transient simulations of semiconductors. However, in device TCAD, the non-equilibrium steady-state responses, such as the dc I-V characteristics, are usually the most concerned

properties [1], [2], [46]. One can obtain the steady-state response by running the transient simulation for a long duration of time while waiting for the device to reach the steady state. However, this approach becomes computationally expensive especially for explicit time-domain solvers, whose time-step size is restricted to be small by stability requirements, since they have to be executed for large number of time steps [1], [42], [44], [46]. Therefore, numerical schemes, which operate on steady-state version of the time-dependent equations and directly produce the steady-state response, are often preferred for analysis of semiconductor devices [22], [30], [46]–[50].

The steady-state DD model calls for solution of a nonlinear system consisting of three coupled second-order partial differential equations (PDEs). The first of these equations is the Poisson equation in scalar potential and the other two are the convection-diffusion type DD equations in electron and hole densities. These three equations are nonlinearly coupled via the drift current and the charge distribution. The charge-density dependent recombination rate, together with the field-dependent mobility and diffusion coefficients, makes the nonlinearity even stronger. In this context, numerical methods developed for the steady-state DD system operate differently from those developed for the time-dependent DD system. Steady-state solvers account for the nonlinearity using a fixed-point iteration method [2], [22], while existing time-domain solvers use “explicit” updates between nonlinearly coupled system during time marching [42]–[44].

In this work, for the first time, a DG-based numerical framework is formulated and implemented to solve the coupled nonlinear system of the steady-state DD and Poisson equations. More specifically, we use the local DG (LDG) scheme [51] in cooperation with the Gummel method [47] to simulate the non-equilibrium steady-state response of semiconductor devices. To construct the (discretized) DG operator for the convection-diffusion type DD equations (linearized within the Gummel method), the LDG alternate numerical flux is used for the diffusion term [52] and the local Lax-Friedrichs flux is used for the convection term. Similarly, the discretized DG operator for the nonlinear Poisson equation (linearized within the Gummel method) is constructed using the alternate numerical flux. The resulting DG-based framework is used to simulate geometrically intricate semiconductor devices with realistic models of the mobility and the recombination rate [2]. Its accuracy is demonstrated by comparing the results to those obtained by the FVM and FEM solvers implemented within the commercial software package COMSOL [30]. We should note here that other DG schemes, such as discontinuous Petrov Galerkin [53], hybridizable DG [54], exponential fitted DG [55], and DG with Lagrange multipliers [56] could be adopted for the DG-based framework proposed in this work.

The rest of the paper is organized as follows. Section II starts with the mathematical model where the coupled nonlinear system of Poisson and DD equations is introduced, then it describes the Gummel method and provides the details

of the DG-based discretization. Section III demonstrates the accuracy and the applicability of the proposed framework via simulations of two realistic device examples. Finally, Section IV provides a summary and discusses possible future research directions.

## II. FORMULATION

### A. MATHEMATICAL MODEL

The DD model describes the (semi-classical) transport of electrons and holes in an electric field under the drift-diffusion approximation [1], [2]. It couples the Poisson equation that describes the behavior of the (static) electric potential and the two continuity equations that describe the behavior of electrons and holes. This (coupled) system of equations reads

$$-\nabla \cdot (\varepsilon(\mathbf{r})\nabla\varphi(\mathbf{r})) = q(C + n_h(\mathbf{r}) - n_e(\mathbf{r})) \quad (1)$$

$$\nabla \cdot \mathbf{J}_s(\mathbf{r}) = \pm qR(n_e, n_h), \quad s \in \{e, h\} \quad (2)$$

where  $\mathbf{r}$  represents the location vector,  $n_e(\mathbf{r})$  and  $n_h(\mathbf{r})$  are the electron and hole densities,  $\varphi(\mathbf{r})$  is the electric potential,  $\mathbf{J}_e(\mathbf{r})$  and  $\mathbf{J}_h(\mathbf{r})$  are the electron and hole current densities,  $\varepsilon(\mathbf{r})$  is the dielectric permittivity,  $q$  is the electron charge, and  $R(n_e, n_h)$  is the net recombination rate. In (15) and other equations in the rest of the text,  $s \in \{e, h\}$ , and the upper and lower signs should be selected for  $s = e$  and  $s = h$ , respectively. The current densities  $\mathbf{J}_s(\mathbf{r})$  are given by

$$\mathbf{J}_s(\mathbf{r}) = q\mu_s(\mathbf{E})\mathbf{E}(\mathbf{r})n_s(\mathbf{r}) \pm qV_T\mu_s(\mathbf{E})\nabla n_s(\mathbf{r}) \quad (3)$$

where  $\mu_e(\mathbf{E})$  and  $\mu_h(\mathbf{E})$  are the (field-dependent) electron and hole mobilities,  $V_T = k_B T/q$  is the thermal voltage,  $k_B$  is the Boltzmann constant,  $T$  is the absolute temperature, and

$$\mathbf{E}(\mathbf{r}) = -\nabla\varphi(\mathbf{r}) \quad (4)$$

is the (static) electric field intensity. Inserting (3) into (2) yields

$$\begin{aligned} \pm \nabla \cdot (\mu_s(\mathbf{E})\mathbf{E}(\mathbf{r})n_s(\mathbf{r})) + \nabla \cdot (V_T\mu_s(\mathbf{E})\nabla n_s(\mathbf{r})) \\ = R(n_e, n_h). \end{aligned} \quad (5)$$

Here,  $R(n_e, n_h)$  describes the recombination/generation of carriers. In this work, we only consider the two most common recombination processes, namely the trap assisted recombination described by the Shockley-Read-Hall (SRH) model [2] as

$$R_{\text{SRH}}(n_e, n_h) = \frac{n_e(\mathbf{r})n_h(\mathbf{r}) - n_i^2}{\tau_e(n_{h1} + n_h(\mathbf{r})) + \tau_h(n_{e1} + n_e(\mathbf{r}))}$$

and the three-particle band-to-band transition described by the Auger model [2] as

$$R_{\text{Auger}}(n_e, n_h) = (n_e(\mathbf{r})n_h(\mathbf{r}) - n_i^2)(C_e^A n_e(\mathbf{r}) + C_h^A n_h(\mathbf{r})).$$

Here,  $n_i$  is the intrinsic carrier concentration,  $\tau_e$  and  $\tau_h$  are the carrier lifetimes,  $n_{e1}$  and  $n_{h1}$  are SRH model parameters related to the trap energy level, and  $C_e^A$  and  $C_h^A$  are the Auger

coefficients. The net recombination rate  $R(n_e, n_h)$  is given by [2]

$$R(n_e, n_h) = R_{\text{SRH}}(n_e, n_h) + R_{\text{Auger}}(n_e, n_h). \quad (6)$$

We should note here that, without loss of generality in the formulation of the method we propose in this work,  $R(n_e, n_h)$  can include other physical models describing different recombination and generation mechanisms (see [1], chapter 4.2). A positive value means recombination is dominant while a negative value means generation is dominant.

The mobility models have a significant impact on the accuracy of semiconductor device simulations. Various field- and temperature-dependent models have been developed for different semiconductor materials and different device operating conditions [1], [2], [30], [49], [50]. Often, high-field mobility models, which account for the carrier velocity saturation effect, are more accurate [2], [30], [49], [50]. In this work, we use the Caughey-Thomas model [2], which expresses  $\mu_e(\mathbf{E})$  and  $\mu_h(\mathbf{E})$  as

$$\mu_s(\mathbf{E}) = \mu_s^0 \left[ 1 + \left( \frac{\mu_s^0 E_{\parallel}(\mathbf{r})}{V_s^{\text{sat}}} \right)^{\beta_s} \right]^{\beta_s^{-1}} \quad (7)$$

where  $E_{\parallel}(\mathbf{r})$  is amplitude of the electric field intensity parallel to the current flow,  $\mu_e^0$  and  $\mu_h^0$  are the low-field electron and hole mobilities, respectively, and  $V_s^{\text{sat}}$ ,  $\beta_e$  and  $\beta_h$  are fitting parameters obtained from experimental data.

### B. GUMMEL METHOD

The DD model described by (1)-(2) and (3)-(4) represents a nonlinear and coupled system of equations. The electric field moves the carriers through the drift term in the expressions of  $\mathbf{J}_e(\mathbf{r})$  and  $\mathbf{J}_h(\mathbf{r})$  [first term in (3)]. The carrier movements change  $n_e(\mathbf{r})$  and  $n_h(\mathbf{r})$ , which in turn affect  $\mathbf{E}(\mathbf{r})$  through the Poisson equation [see (1)]. Furthermore,  $R(n_e, n_h)$  [in (6)] and  $\mu_e(\mathbf{E})$  and  $\mu_h(\mathbf{E})$  [in (7)] are nonlinear functions of  $n_e(\mathbf{r})$  and  $n_h(\mathbf{r})$ , and  $\mathbf{E}(\mathbf{r})$ , respectively. This system can be solved using either a decoupled approach such as the Gummel method or a fully-coupled scheme such as the direct application of the Newton method [2], [22]. The Gummel method's memory requirement and computational cost per iteration are less than those of the Newton method. In addition, accuracy and stability of the solution obtained by the Gummel method are less sensitive to the initial guess [2], [22]. On the other hand, the Gummel method converges slower, i.e., takes a higher number of iterations to converge to the solution [2], [22]. Since the simulations of the nanostructured devices considered in this work are memory-bounded, we prefer to use the Gummel method.

The Gummel iterations operate as described next and shown in Fig. 1. To facilitate the algorithm, we first introduce the quasi-Fermi potentials [1], [2], [22]

$$\varphi_s(\mathbf{r}) = \varphi(\mathbf{r}) \mp V_T \ln(n_s(\mathbf{r})/n_i), \quad s \in \{e, h\}. \quad (8)$$

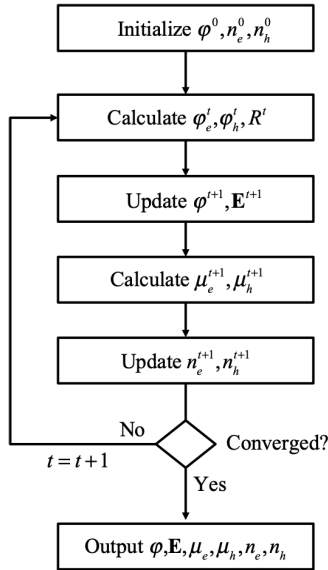


FIGURE 1. Gummel method.

Solving (8) for  $n_e(\mathbf{r})$  and  $n_h(\mathbf{r})$ , respectively, and inserting the resulting expressions into (1) yield

$$-\nabla \cdot (\varepsilon(\mathbf{r})\nabla\varphi(\mathbf{r})) = q(C + n_i e^{(\varphi_h(\mathbf{r})-\varphi(\mathbf{r}))/V_T} - n_i e^{(\varphi(\mathbf{r})-\varphi_e(\mathbf{r}))/V_T}). \quad (9)$$

Equation (9) is termed as the nonlinear Poisson (NLP) equation simply because it is nonlinear in  $\varphi(\mathbf{r})$ . Using  $\varphi_e(\mathbf{r})$  and  $\varphi_h(\mathbf{r})$ , one can easily write the Fréchet derivative of the NLP equation and solve the nonlinear problem with a fixed-point iteration technique such as the Newton method [1], [2], [22] (see below). The Gummel method decouples the NLP equation and the DD equations (2); the nonlinearity is “maintained” solely in the NLP equation and the DD equations are treated as linear systems [1], [2], [22] as shown by the description of the Gummel method below. To solve the NLP equation in (9), we write it as a root-finding problem

$$F(\varphi(\mathbf{r}), \varphi_e(\mathbf{r}), \varphi_h(\mathbf{r})) = \nabla \cdot (\varepsilon(\mathbf{r})\nabla\varphi(\mathbf{r})) + q(C + n_i e^{(\varphi_h(\mathbf{r})-\varphi(\mathbf{r}))/V_T} - n_i e^{(\varphi(\mathbf{r})-\varphi_e(\mathbf{r}))/V_T}) = 0. \quad (10)$$

The Fréchet derivative of  $F(\varphi(\mathbf{r}), \varphi_e(\mathbf{r}), \varphi_h(\mathbf{r}))$  with respect to  $\varphi(\mathbf{r})$  is

$$F'(\varphi(\mathbf{r}), \varphi_e(\mathbf{r}), \varphi_h(\mathbf{r}); \delta\varphi(\mathbf{r})) = \nabla \cdot (\varepsilon(\mathbf{r})\nabla\delta\varphi(\mathbf{r})) - \frac{qn_i}{V_T} (e^{(\varphi_h(\mathbf{r})-\varphi(\mathbf{r}))/V_T} + e^{(\varphi(\mathbf{r})-\varphi_e(\mathbf{r}))/V_T})\delta\varphi(\mathbf{r}). \quad (11)$$

The root-finding problem (10) is solved iteratively as

$$\varphi^{t+1}(\mathbf{r}) = \varphi^t(\mathbf{r}) + \delta\varphi^{t+1}(\mathbf{r}) \quad (12)$$

where subscript “ $t$ ” refers to the variables at iteration  $t$ . In (12),  $\delta\varphi^{t+1}(\mathbf{r})$  is obtained by solving

$$F'(\varphi^t, \varphi_e^t, \varphi_h^t; \delta\varphi^{t+1}) = -F(\varphi^t, \varphi_e^t, \varphi_h^t) \quad (13)$$

where  $\varphi^t(\mathbf{r})$  is the solution at iteration  $t$  (previous iteration),  $\varphi_e^t(\mathbf{r})$  and  $\varphi_h^t(\mathbf{r})$  are computed using using  $n_e^t(\mathbf{r})$  and  $n_h^t(\mathbf{r})$  in (8). At iteration  $t = 0$ , initial guesses  $\varphi^0(\mathbf{r})$ ,  $n_e^0(\mathbf{r})$  and  $n_h^0(\mathbf{r})$  are used to start the iterations. In this work, the initial guesses are solved from the equilibrium-state (without bias voltage) [2]. Note that, in practice, one can directly compute  $\varphi^{t+1}(\mathbf{r})$  without using the variable  $\delta\varphi^{t+1}(\mathbf{r})$ . This is done by adding  $F'(\varphi^t, \varphi_e^t, \varphi_h^t; \varphi^t)$  to both sides of (13), and using (4) and the fact that

$$F'(\varphi^t, \varphi_e^t, \varphi_h^t; \varphi^t + \delta\varphi^{t+1}) = F'(\varphi^t, \varphi_e^t, \varphi_h^t; \varphi^{t+1})$$

which result in the coupled system of equations in unknowns  $\varphi^{t+1}(\mathbf{r})$  and  $\mathbf{E}^{t+1}(\mathbf{r})$

$$\nabla \cdot (\varepsilon(\mathbf{r})\mathbf{E}^{t+1}(\mathbf{r})) + g(\mathbf{r})\varphi^{t+1}(\mathbf{r}) = f(\mathbf{r}) \quad (14a)$$

$$\mathbf{E}^{t+1}(\mathbf{r}) = -\nabla\varphi^{t+1}(\mathbf{r}). \quad (14b)$$

Here,

$$g(\mathbf{r}) = \frac{qn_i}{V_T} (e^{(\varphi_h^t(\mathbf{r})-\varphi^t(\mathbf{r}))/V_T} + e^{(\varphi^t(\mathbf{r})-\varphi_e^t(\mathbf{r}))/V_T})$$

and

$$f(\mathbf{r}) = \frac{qn_i}{V_T} (e^{(\varphi_h^t(\mathbf{r})-\varphi^t(\mathbf{r}))/V_T} + e^{(\varphi^t(\mathbf{r})-\varphi_e^t(\mathbf{r}))/V_T})\varphi^k(\mathbf{r}) + qn_i(C/n_i + e^{(\varphi_h^t(\mathbf{r})-\varphi^t(\mathbf{r}))/V_T} - e^{(\varphi^t(\mathbf{r})-\varphi_e^t(\mathbf{r}))/V_T})$$

are known coefficients obtained from the previous iteration.

Unknowns  $\varphi^{t+1}(\mathbf{r})$  and  $\mathbf{E}^{t+1}(\mathbf{r})$  are obtained by solving (14). Then,  $\mu_e(\mathbf{E}^{t+1})$  and  $\mu_h(\mathbf{E}^{t+1})$  are computed using  $\mathbf{E}^{t+1}(\mathbf{r})$  in (7). Finally,  $n_e^{t+1}(\mathbf{r})$  and  $n_h^{t+1}(\mathbf{r})$  can be obtained by solving

$$\pm\nabla \cdot (\mu_s(\mathbf{E}^{t+1})\mathbf{E}^{t+1}(\mathbf{r})n_s^{t+1}(\mathbf{r})) + \nabla \cdot (V_T\mu_s(\mathbf{E}^{t+1})\nabla n_s^{t+1}(\mathbf{r})) = R(n_e^t, n_h^t) \quad (15)$$

where  $R(n_e^t, n_h^t)$  on the right hand side is computed using  $n_e^t(\mathbf{r})$  and  $n_h^t(\mathbf{r})$  (from previous iteration) in (6). Note that a “lagging” technique may also be applied to  $R(n_e^t, n_h^t)$  to take advantage of the solutions at the current iteration. This technique expresses  $R(n_e, n_h)$  as a summation of functions of  $n_e^t(\mathbf{r})$  and  $n_h^t(\mathbf{r})$  and  $n_e^{t+1}(\mathbf{r})$  and  $n_h^{t+1}(\mathbf{r})$ , and moves the functions of  $n_e^{t+1}(\mathbf{r})$  and  $n_h^{t+1}(\mathbf{r})$  to the left hand side of (15). More details about this technique can be found in [57].

At this stage of the iteration,  $\varphi^{t+1}(\mathbf{r})$ ,  $n_e^{t+1}(\mathbf{r})$  and  $n_h^{t+1}(\mathbf{r})$  are known; one can use these to compute  $\varphi_e^{t+1}(\mathbf{r})$  and  $\varphi_h^{t+1}(\mathbf{r})$  and move to the next iteration. Convergence of the iterations can be checked by either the residuals of (10) and (15) or by the difference between the solutions of two successive iterations.

### C. DG DISCRETIZATION

As explained in the previous section, at every iteration of the Gummel method, one needs to solve three linear systems of equations, namely (14) and (15) ( $s = e, h$ ). This can only be done numerically for arbitrarily shaped devices. To this end, we use the LDG method [51], [52] to discretize and numerically solve these equations. We start with the description of

the discretization of (14). First, we re-write (14) in the form of the following boundary value problem

$$\nabla \cdot [\varepsilon(\mathbf{r})\mathbf{E}(\mathbf{r})] + g(\mathbf{r})\varphi(\mathbf{r}) = f(\mathbf{r}), \quad \mathbf{r} \in \Omega \quad (16)$$

$$\mathbf{E}(\mathbf{r}) = -\nabla\varphi(\mathbf{r}), \quad \mathbf{r} \in \Omega \quad (17)$$

$$\varphi(\mathbf{r}) = f_D(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega_D \quad (18)$$

$$\hat{\mathbf{n}}(\mathbf{r}) \cdot \varepsilon(\mathbf{r})\mathbf{E}(\mathbf{r}) = f_N(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega_N. \quad (19)$$

In (16)-(19),  $\varphi(\mathbf{r})$  and  $\mathbf{E}(\mathbf{r})$  are the unknowns to be solved for and  $\Omega$  is the solution domain. Note that in LDG,  $\mathbf{E}(\mathbf{r})$  is introduced as an auxiliary variable to reduce the order of the spatial derivative in (16). Here it is also a ‘‘natural’’ unknown to be solved for within the Gummel method. Dirichlet and Neumann boundary conditions are enforced on surfaces  $\partial\Omega_D$  and  $\partial\Omega_N$ , and  $f_D(\mathbf{r})$  and  $f_N(\mathbf{r})$  are the coefficients associated with these boundary conditions, respectively. In (19),  $\hat{\mathbf{n}}(\mathbf{r})$  denotes the outward normal vector  $\partial\Omega_N$ . For the problems considered in this work,  $\partial\Omega_D$  represents the metal contact surfaces with  $f_D(\mathbf{r}) = V_{\text{contact}}(\mathbf{r})$ , where  $V_{\text{contact}}(\mathbf{r})$  is the potential impressed on the contacts. The homogeneous Neumann boundary condition, i.e.,  $f_N(\mathbf{r}) = 0$ , is used to truncate the simulation domain [58].

To facilitate the numerical solution of the boundary value problem described by (16)-(19) (within the Gummel method),  $\Omega$  is discretized into  $k$  non-overlapping tetrahedrons. The (volumetric) support of each of these elements is represented by  $\Omega_k$ ,  $k = 1, \dots, K$ . Furthermore, let  $\partial\Omega_k$  denote the surface of  $\Omega_k$  and  $\hat{\mathbf{n}}(\mathbf{r})$  denote the outward unit vector normal to  $\partial\Omega_k$ . Testing equations (16) and (17) with the Lagrange polynomials  $\ell_i(\mathbf{r})$ ,  $i = 1, \dots, N_p$  on element  $k$  [36] and applying the divergence theorem to the resulting equation yield the following weak form

$$\int_{\Omega_k} g(\mathbf{r})\varphi_k(\mathbf{r})\ell_i(\mathbf{r})dV - \int_{\Omega_k} \varepsilon(\mathbf{r})\mathbf{E}_k(\mathbf{r}) \cdot \nabla\ell_i(\mathbf{r})dV + \oint_{\partial\Omega_k} \hat{\mathbf{n}}(\mathbf{r}) \cdot [\varepsilon(\mathbf{r})\mathbf{E}_k(\mathbf{r})]^* \ell_i(\mathbf{r})dS = \int_{\Omega_k} f(\mathbf{r})\ell_i(\mathbf{r})dV \quad (20)$$

$$\int_{\Omega_k} E_k^v(\mathbf{r})\ell_i(\mathbf{r})dV - \int_{\Omega_k} \varphi_k(\mathbf{r}) \frac{\partial}{\partial v} \ell_i(\mathbf{r})dV + \oint_{\partial\Omega_k} \hat{n}_v(\mathbf{r})\varphi_k(\mathbf{r})^* \ell_i(\mathbf{r})dS = 0. \quad (21)$$

Here,  $N_p = (p+1)(p+2)(p+3)/6$  is the number of interpolating nodes,  $p$  is the order of the Lagrange polynomials and subscript  $v \in \{x, y, z\}$  is used for identifying the components of the vectors in the Cartesian coordinate system. We note here  $\varphi_k(\mathbf{r})$  and  $\mathbf{E}_k(\mathbf{r})$  denote the local solutions on element  $k$  and the global solutions on  $\Omega$  are the sum of these local solutions.

$\varphi^*$  and  $(\varepsilon\mathbf{E})^*$  are numerical fluxes ‘‘connecting’’ element  $k$  to its neighboring elements. Here, the variables are defined on the interface between elements and the dependency on  $\mathbf{r}$  is dropped for simplicity of notation/presentation. In LDG, the alternate flux, which is defined as [51]

$$\varphi^* = \{\varphi\} + 0.5\hat{\beta} \cdot \hat{\mathbf{n}} \llbracket \varphi \rrbracket$$

$$(\varepsilon\mathbf{E})^* = \{\varepsilon\mathbf{E}\} - 0.5\hat{\beta}(\hat{\mathbf{n}} \cdot \llbracket \varepsilon\mathbf{E} \rrbracket)$$

is used in the interior of  $\Omega$ . Here, averaging operators are defined as  $\{a\} = 0.5(a^+ + a^-)$  and  $\{\mathbf{a}\} = 0.5(\mathbf{a}^+ + \mathbf{a}^-)$  and ‘‘jumps’’ are defined as  $\llbracket a \rrbracket = a^- - a^+$  and  $\llbracket \mathbf{a} \rrbracket = \mathbf{a}^- - \mathbf{a}^+$ , where superscripts ‘‘-’’ and ‘‘+’’ refer to variables defined in element  $k$  and in its neighboring element, respectively. The vector  $\hat{\beta}$  determines the upwinding direction of  $\varphi$  and  $(\varepsilon\mathbf{E})$ . In LDG, it is essential to choose opposite directions for  $\varphi$  and  $(\varepsilon\mathbf{E})$ , while the precise direction of each variable is not important [36], [38], [51]. In this work, we choose  $\hat{\beta} = \hat{\mathbf{n}}$  on each element surface. On boundaries of  $\Omega$ , the numerical fluxes are chosen as  $\varphi^* = f_D$  and  $(\varepsilon\mathbf{E})^* = (\varepsilon\mathbf{E})^-$  on  $\partial\Omega_D$  and  $\varphi^* = \varphi^-$  and  $(\varepsilon\mathbf{E})^* = f_N$  on  $\partial\Omega_N$ , respectively [52].

We expand  $\varphi_k(\mathbf{r})$  and  $E_k^v(\mathbf{r})$  with the same set of Lagrange polynomials  $\ell_i(\mathbf{r})$

$$\varphi_k(\mathbf{r}) \simeq \sum_{i=1}^{N_p} \varphi(\mathbf{r}_i)\ell_i(\mathbf{r}) = \sum_{i=1}^{N_p} \varphi_k^i \ell_i(\mathbf{r}) \quad (22)$$

$$E_k^v(\mathbf{r}) \simeq \sum_{i=1}^{N_p} E_v(\mathbf{r}_i)\ell_i(\mathbf{r}) = \sum_{i=1}^{N_p} E_k^{v,i} \ell_i(\mathbf{r}) \quad (23)$$

where  $\mathbf{r}_i$ ,  $i = 1, \dots, N_p$ , denote the location of interpolating nodes, and  $\varphi_k^i$  and  $E_k^{v,i}$ ,  $v \in \{x, y, z\}$ ,  $k = 1, \dots, K$ , are the unknown coefficients to be solved for.

Substituting (22) and (23) into (20) and (21) for  $k = 1, \dots, K$ , we obtain a global matrix system

$$\begin{bmatrix} \bar{M}^g & \bar{D}\bar{\varepsilon} \\ \bar{G} & \bar{M} \end{bmatrix} \begin{bmatrix} \bar{\Phi} \\ \bar{E} \end{bmatrix} = \begin{bmatrix} \bar{B}^\varphi \\ \bar{B}^E \end{bmatrix}. \quad (24)$$

Here, the global unknown vectors  $\bar{\Phi} = [\bar{\Phi}_1, \dots, \bar{\Phi}_K]^T$  and  $\bar{E} = [\bar{E}_1^x, \bar{E}_1^y, \bar{E}_1^z, \dots, \bar{E}_K^x, \bar{E}_K^y, \bar{E}_K^z]^T$  are assembled from elemental vectors  $\bar{\Phi}_k = [\varphi_k^1, \dots, \varphi_k^{N_p}]$  and  $\bar{E}_k^v = [E_k^{v,1}, \dots, E_k^{v,N_p}]$ ,  $v \in \{x, y, z\}$ . The dimension of (24) can be further reduced by substituting  $\bar{E} = \bar{M}^{-1}(\bar{B}^E - \bar{G}\bar{\Phi})$  (from the second row) into the first row, which results in

$$(\bar{M}^g - \bar{D}\bar{\varepsilon}\bar{M}^{-1}\bar{G})\bar{\Phi} = \bar{B}^\varphi - \bar{D}\bar{\varepsilon}\bar{M}^{-1}\bar{B}^E \quad (25)$$

where  $(\bar{M}^g - \bar{D}\bar{\varepsilon}\bar{M}^{-1}\bar{G})$  is the Schur complement of the matrix in (24). In (24) and (25),  $\bar{M}^g$  and  $\bar{M}$  are mass matrices.  $\bar{M}^g$  is a  $K \times K$  block diagonal matrix, where each  $N_p \times N_p$  block is defined as

$$\bar{M}_{kk}^g(i, j) = \int_{\Omega_k} g(\mathbf{r})\ell_i(\mathbf{r})\ell_j(\mathbf{r})dV.$$

$\bar{M}$  is also a  $K \times K$  block diagonal matrix, where each block is a  $3 \times 3$  block diagonal matrix with  $N_p \times N_p$  identical blocks defined as

$$\bar{M}_{kk}^{(m)}(i, j) = \int_{\Omega_k} \ell_i(\mathbf{r})\ell_j(\mathbf{r})dV, \quad m = 1, 2, 3.$$

$\bar{\varepsilon}$  is a diagonal matrix with entries  $(\bar{\varepsilon}_1, \dots, \bar{\varepsilon}_K)$ , where  $\bar{\varepsilon}_k = (\bar{\varepsilon}_k^x, \bar{\varepsilon}_k^y, \bar{\varepsilon}_k^z)$ ,  $\bar{\varepsilon}_k^v(i) = \varepsilon_k(\mathbf{r}_i)$ ,  $k = 1, \dots, K$ ,  $v \in \{x, y, z\}$ . We note that  $\varepsilon(\mathbf{r})$  is assumed isotropic and constant in each element.

Matrices  $\bar{G}$  and  $\bar{D}$  represent the gradient and divergence operators, respectively. For LDG, one can show that

$\bar{D} = -\bar{G}^T$  [52]. The gradient matrix  $\bar{G}$  is a  $K \times K$  block sparse matrix, where each block is of size  $3N_p \times N_p$  and has contribution from the volume integral term and the surface integral term in (21). The volume integral term only contributes to diagonal blocks as  $\bar{G}_{kk}^{vol} = [\bar{S}_k^x \ \bar{S}_k^y \ \bar{S}_k^z]^T$ , where

$$\bar{S}_k^v(i, j) = - \int_{\Omega_k} \frac{d\ell_i(\mathbf{r})}{dv} \ell_j(\mathbf{r}) dV, \quad v \in \{x, y, z\}.$$

The surface integral term contributes to both the diagonal blocks  $\bar{G}_{kk}$  and off-diagonal blocks  $\bar{G}_{kk'}$ , where  $k'$  corresponds to the index of the elements connected to element  $k$ . Let  $\partial\Omega_{kk'}$  be the interface connecting elements  $k$  and  $k'$ , and let  $\theta_k(j)$  select the interface nodes from element  $k$ ,

$$\theta_k(j) = \begin{cases} 1, & \mathbf{r}_j \in \Omega_k, \mathbf{r}_j \in \partial\Omega_{kk'} \\ 0, & \text{otherwise.} \end{cases}$$

Then, the contributions from the surface integral term to the diagonal block and the off-diagonal blocks are  $\bar{G}_{kk}^{surf} = [\bar{L}_k^x \ \bar{L}_k^y \ \bar{L}_k^z]^T$  and  $\bar{G}_{kk'}^{surf} = [\bar{L}_{k'}^x \ \bar{L}_{k'}^y \ \bar{L}_{k'}^z]^T$ , where

$$\bar{L}_k^v(i, j) = \frac{1 + \text{sign}(\hat{\boldsymbol{\beta}} \cdot \hat{\mathbf{n}})}{2} \theta_k(j) \oint_{\partial\Omega_{kk'}} \hat{n}_v(\mathbf{r}) \ell_i(\mathbf{r}) \ell_j(\mathbf{r}) dS$$

and

$$\bar{L}_{k'}^v(i, j) = \frac{1 - \text{sign}(\hat{\boldsymbol{\beta}} \cdot \hat{\mathbf{n}})}{2} \theta_{k'}(j) \oint_{\partial\Omega_{kk'}} \hat{n}_v(\mathbf{r}) \ell_i(\mathbf{r}) \ell_j(\mathbf{r}) dS$$

respectively,  $v \in \{x, y, z\}$ . The right hand side terms in (24) and (25) are contributed from the force term and boundary conditions and are expressed as

$$\bar{B}_k^f(i) = \int_{\Omega_k} f(\mathbf{r}) \ell_i(\mathbf{r}) dV + \oint_{\partial\Omega_k \cap \partial\Omega_N} f_N(\mathbf{r}) \ell_i(\mathbf{r}) dS$$

$$\bar{B}_k^{\mathbf{E},v}(i) = \oint_{\partial\Omega_k \cap \partial\Omega_D} \hat{n}_v(\mathbf{r}) f_D(\mathbf{r}) \ell_i(\mathbf{r}) dS, \quad v \in \{x, y, z\}.$$

The DD equations in (15) (within the Gummel method) are also discretized using the LDG scheme as described next. Note that, here, we only discuss the discretization of the electron DD equation ( $s = e$ ) and that of the hole DD equation ( $s = h$ ) only differs by the sign in front of the drift term and the values of physical parameters. To simplify the notation/presentation, we drop the subscript denoting the species (electron and hole). The electron DD equation in (15) is expressed as the following boundary value problem

$$\nabla \cdot [d(\mathbf{r})\mathbf{q}(\mathbf{r})] + \nabla \cdot [\mathbf{v}(\mathbf{r})n(\mathbf{r})] = R(\mathbf{r}), \quad \mathbf{r} \in \Omega \quad (26)$$

$$\mathbf{q}(\mathbf{r}) = \nabla n(\mathbf{r}), \quad \mathbf{r} \in \Omega \quad (27)$$

$$n(\mathbf{r}) = f_D(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega_D \quad (28)$$

$$\hat{\mathbf{n}}(\mathbf{r}) \cdot [d(\mathbf{r})\mathbf{q}(\mathbf{r}) + \mathbf{v}(\mathbf{r})n(\mathbf{r})] = f_R(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega_R. \quad (29)$$

Here  $n(\mathbf{r})$  and  $\mathbf{q}(\mathbf{r})$  are the unknowns to be solved for and  $\Omega$  is the solution domain. The auxiliary variable  $\mathbf{q}(\mathbf{r})$  is introduced to reduce the order of the spatial derivative.  $d(\mathbf{r}) = V_T \mu(\mathbf{E})$  and  $\mathbf{v}(\mathbf{r}) = \mu(\mathbf{E})\mathbf{E}(\mathbf{r})$  become known coefficients during the solution of (15) within the Gummel method. Dirichlet and Robin boundary conditions are enforced on surfaces  $\partial\Omega_D$  and

$\partial\Omega_R$ , and  $f_D(\mathbf{r})$  and  $f_R(\mathbf{r})$  are the coefficients associated with these boundary conditions, respectively.  $\hat{\mathbf{n}}(\mathbf{r})$  denotes the outward normal vector of the surface. For the problems considered in this work,  $\partial\Omega_D$  represents electrode/semiconductor interfaces and, based on local charge neutrality [58],  $f_D(\mathbf{r}) = (C + \sqrt{C^2 + 4n_i^2})/2$  and  $f_D(\mathbf{r}) = n_i^2/n_e^s$  for electron and hole DD equations, respectively. The homogeneous Robin boundary condition, i.e.,  $f_R(\mathbf{r}) = 0$ , is used on semiconductor/insulator interfaces, indicating no carrier spills out those interfaces [58].

Following the same procedure used in the discretization of (14), we discretize the domain into non-overlapping tetrahedrons and test equations (26) and (27) using Lagrange polynomials on element  $k$ . Applying the divergence theorem yield the following weak form:

$$\begin{aligned} & - \int_{\Omega_k} d(\mathbf{r})\mathbf{q}_k(\mathbf{r}) \cdot \nabla \ell_i(\mathbf{r}) dV + \oint_{\partial\Omega_k} \hat{\mathbf{n}}(\mathbf{r}) \cdot [d(\mathbf{r})\mathbf{q}_k(\mathbf{r})]^* \ell_i(\mathbf{r}) dS \\ & - \int_{\Omega_k} \mathbf{v}(\mathbf{r})n_k(\mathbf{r}) \cdot \nabla \ell_i(\mathbf{r}) dV + \oint_{\partial\Omega_k} \hat{\mathbf{n}}(\mathbf{r}) \cdot [\mathbf{v}(\mathbf{r})n_k(\mathbf{r})]^* \ell_i(\mathbf{r}) dS \\ & = \int_{\Omega_k} R(\mathbf{r}) \ell_i(\mathbf{r}) dV \end{aligned} \quad (30)$$

$$\begin{aligned} & \int_{\Omega_k} q_k^v(\mathbf{r}) \ell_i(\mathbf{r}) dV + \int_{\Omega_k} n_k(\mathbf{r}) \frac{\partial}{\partial v} \ell_i(\mathbf{r}) dV \\ & - \oint_{\partial\Omega_k} \hat{n}_v(\mathbf{r}) n_k^*(\mathbf{r}) \ell_i(\mathbf{r}) dS = 0 \end{aligned} \quad (31)$$

where  $n^*$ ,  $(d\mathbf{q})^*$ , and  $(\mathbf{v}n)^*$  are numerical fluxes ‘‘connecting’’ element  $k$  to its neighboring elements. Here, for the simplicity of notation, we have dropped the explicit dependency on  $\mathbf{r}$  on element surfaces. For the diffusion term, the LDG alternate flux is used for the primary variable  $n^*$  and the auxiliary variable  $(d\mathbf{q})^*$  [51]

$$\begin{aligned} n^* &= \{n\} + 0.5\hat{\boldsymbol{\beta}} \cdot \hat{\mathbf{n}} [\![n]\!] \\ (d\mathbf{q})^* &= \{d\mathbf{q}\} - 0.5\hat{\boldsymbol{\beta}} \cdot \hat{\mathbf{n}} [\![d\mathbf{q}]\!]. \end{aligned}$$

Here, averages and ‘‘jumps’’, and the vector coefficient  $\hat{\boldsymbol{\beta}}$  are same as those defined before. For the drift term, the local Lax-Friedrichs flux is used to mimic the path of information propagation [36]

$$(\mathbf{v}n)^* = \{\mathbf{v}n\} + \alpha \hat{\mathbf{n}}(n^- - n^+), \quad \alpha = \frac{\max(|\hat{\mathbf{n}} \cdot \mathbf{v}^-|, |\hat{\mathbf{n}} \cdot \mathbf{v}^+|)}{2}.$$

On boundaries, the numerical fluxes are chosen as  $n^* = f_D$ ,  $(d\mathbf{q})^* = (d\mathbf{q})^-$  and  $(\mathbf{v}n)^* = \mathbf{v}^- f_D$  on  $\partial\Omega_D$  and  $n^* = n^-$  and  $(d\mathbf{q})^* + (\mathbf{v}n)^* = f_R$  on  $\partial\Omega_R$ , respectively. We note  $(d\mathbf{q})^*$  and  $(\mathbf{v}n)^*$  are not assigned independently on  $\partial\Omega_R$ .

Expanding  $n_k(\mathbf{r})$  and  $q_k^v(\mathbf{r})$  with Lagrange polynomials  $\ell_i(\mathbf{r})$

$$n_k(\mathbf{r}) \simeq \sum_{i=1}^{N_p} n(\mathbf{r}_i) \ell_i(\mathbf{r}) = \sum_{i=1}^{N_p} n_k^i \ell_i(\mathbf{r}) \quad (32)$$

$$q_k^v(\mathbf{r}) \simeq \sum_{i=1}^{N_p} q_v(\mathbf{r}_i) \ell_i(\mathbf{r}) = \sum_{i=1}^{N_p} q_k^{v,i} \ell_i(\mathbf{r}) \quad (33)$$

where  $\mathbf{r}_i$ ,  $i = 1, \dots, N_p$ , denote the location of interpolating nodes,  $n_k^i$  and  $q_k^{v,i}$ ,  $v \in \{x, y, z\}$ ,  $k = 1, \dots, K$  are the unknown coefficients to be solved for. Substituting (32) and (33) into (30) and (31), we obtain a global matrix system

$$\begin{bmatrix} \bar{C} & \bar{D}\bar{d} \\ -\bar{G} & \bar{M} \end{bmatrix} \begin{bmatrix} \bar{N} \\ \bar{Q} \end{bmatrix} = \begin{bmatrix} \bar{B}^n \\ \bar{B}^q \end{bmatrix}. \quad (34)$$

Here, the global unknown vectors  $\bar{N} = [\bar{N}_1, \dots, \bar{N}_K]^T$  and  $\bar{Q} = [\bar{Q}_1^x, \bar{Q}_1^y, \bar{Q}_1^z, \dots, \bar{Q}_K^x, \bar{Q}_K^y, \bar{Q}_K^z]^T$  are assembled from elemental vectors  $\bar{N}_k = [n_k^1, \dots, n_k^{N_p}]$  and  $\bar{Q}_k^v = [q_k^{v,1}, \dots, q_k^{v,N_p}]$ ,  $v \in \{x, y, z\}$ . Substituting  $\bar{Q} = \bar{M}^{-1}(\bar{B}^q + \bar{G}\bar{N})$  into (34) yields

$$(\bar{C} + \bar{D}\bar{d}\bar{M}^{-1}\bar{G})\bar{N} = \bar{B}^n - \bar{D}\bar{d}\bar{M}^{-1}\bar{B}^q \quad (35)$$

where  $(\bar{C} + \bar{D}\bar{d}\bar{M}^{-1}\bar{G})$  is the Schur complement of the matrix in (34).

In (34) and (35), the mass matrix  $\bar{M}$ , the gradient matrix  $\bar{G}$  and the divergence matrix  $\bar{D}$  are same as those defined before.  $\bar{d}$  is a diagonal matrix with entries  $(\bar{d}_1, \dots, \bar{d}_K)$ , where  $\bar{d}_k = (\bar{d}_k^x, \bar{d}_k^y, \bar{d}_k^z)$ ,  $\bar{d}_k^v(i) = d_k(\mathbf{r}_i)$ ,  $k = 1, \dots, K$ ,  $v \in \{x, y, z\}$ .

The block sparse matrix  $\bar{C}$  has contribution from the third term (the volume integral) and the fourth term (the surface integral) in (30). Each block is of size  $N_p \times N_p$ . The volume integral term only contributes to diagonal blocks as  $\bar{C}_{kk}^{\text{vol}} = \sum_v \bar{C}_k^v$ , where

$$\bar{C}_k^v(i, j) = - \int_{\Omega_k} v_v(\mathbf{r}) \frac{d\ell_i(\mathbf{r})}{dv} \ell_j(\mathbf{r}) dV, \quad v \in \{x, y, z\}.$$

The surface integral term contributes to both the diagonal and off-diagonal blocks as

$$\bar{C}_{kk}^{\text{surf}}(i, j) = \theta_k(j) \oint_{\partial\Omega_{kk'}} \left( \frac{1}{2} \sum_v \hat{n}_v(\mathbf{r}) v_v(\mathbf{r}) + \alpha(\mathbf{r}) \right) \ell_i(\mathbf{r}) \ell_j(\mathbf{r}) dS$$

and

$$\bar{C}_{kk'}^{\text{surf}}(i, j) = \theta_{k'}(j) \oint_{\partial\Omega_{kk'}} \left( \frac{1}{2} \sum_v \hat{n}_v(\mathbf{r}) v_v(\mathbf{r}) - \alpha(\mathbf{r}) \right) \ell_i(\mathbf{r}) \ell_j(\mathbf{r}) dS$$

respectively, where  $v \in \{x, y, z\}$ , and  $k'$ ,  $\partial\Omega_{kk'}$ , and  $\theta_k(j)$  are defined the same as before.

The right hand side terms in (34) are contributed from the force term and boundary conditions and are expressed as

$$\begin{aligned} \bar{B}_k^n(i) &= \int_{\Omega_k} \mathbf{R}(\mathbf{r}) \ell_i(\mathbf{r}) dV + \oint_{\partial\Omega_k \cap \partial\Omega_R} f_R(\mathbf{r}) \ell_i(\mathbf{r}) dS \\ &+ \oint_{\partial\Omega_k \cap \partial\Omega_D} \hat{\mathbf{n}}(\mathbf{r}) \cdot \mathbf{v}(\mathbf{r}) f_D(\mathbf{r}) \ell_i(\mathbf{r}) dS \\ \bar{B}_k^q(i) &= \oint_{\partial\Omega_k \cap \partial\Omega_D} \hat{n}_v(\mathbf{r}) f_D(\mathbf{r}) \ell_i(\mathbf{r}) dS, \quad v \in \{x, y, z\}. \end{aligned}$$

#### D. COMMENTS

Several comments regarding the implementation and computational cost of the steady-state DG scheme and its comparison to the existing time-domain DG schemes are in order.

1) The mass matrix  $\bar{M}$  is a block diagonal matrix with  $3K$  blocks that are denoted by  $\bar{M}_{kk}$  and of dimensions  $N_p \times N_p$ .

Therefore, during the construction of the matrix systems (25) and (35),  $\bar{M}^{-1}$  is directly assembled from  $\bar{M}_{kk}^{-1}$ . We note that the computation of  $\bar{M}_{kk}^{-1}$  requires only one matrix inversion since  $\bar{M}_{kk}$  is computed using a reference element [36]. The sparse block matrices  $\bar{G}$  and  $\bar{C}$  have approximately  $15K$  and  $5K$  (non-zero) blocks of dimensions  $N_p \times N_p$ , respectively. Therefore, the memory required to store these matrices and the computational cost of sparse matrix multiplications required to construct the matrix systems (25) and (35) scale with  $O(KN_p^2)$ . Nevertheless, the most computationally expensive step of the steady-state DG scheme is solution of these matrix systems.

2) The sparse linear systems (25) and (35) are solved in MATLAB. For small systems, one can use a direct solver. For large systems, when the number of unknowns is larger than 1 000 000 (when using double precision on a computer with 128GB RAM), it is preferable to use sparse iterative solvers to reduce the memory requirement. During our numerical experiments, we have found that the generalized minimum residual (GMRES) (the MATLAB built-in function ‘‘gmres’’) outperforms other iterative solvers in execution time. Incomplete lower-upper (ILU) factorization is used to obtain a preconditioner for the iterative solver. The drop tolerance of the ILU is critical to keep the balance between the memory requirement and the convergence speed of the preconditioned iterative solution. A smaller drop tolerance usually results in a better preconditioner, however, it also increases the amount of fill-in, which increases the memory requirement.

We note here that one can reuse the preconditioner throughout the Gummel iterations. Because the matrix coefficients change gradually between successive iterations, we can store the preconditioner obtained in the first iteration ( $t = 0$ ) and reuse it as the preconditioner in the following few iterations. In practice, the preconditioner only needs to be updated when the convergence of the sparse iterative solver becomes slower than it is in the previous Gummel iteration. For the devices considered in this work, the number of Gummel iterations is typically less than 50 and we find the preconditioners of the initial matrices work well throughout these iterations.

3) Semiconductor devices might have multiple steady states under certain biasing conditions [48]. A good approach to overcome this issue, is to gradually increase the bias voltage from the equilibrium-state and compute the steady-state solution at every bias voltage value. The steady-state solution at one voltage value is used as the initial guess while computing the solution at the next voltage value. Using this approach, we have simulated the same physical problem with different voltage steps. For the simulations with smaller voltage steps, the number of Gummel iterations at each voltage value has become smaller. This is expected since the initial guess becomes closer to the actual solution. With this approach, one can expect that the numerical solution is the steady-state closest to the equilibrium-state. If one needs to solve for other states (assuming multiple of them exist), other techniques might be required [49].

4) One can use a time-domain DG scheme for steady-state analysis of semiconductor devices [40]–[44]. However, this approach is rarely used for steady-state simulations of practical devices (see Section 3.1.7 of [46]) simply because the number of time steps, which depends on the time-step size required to maintain the numerical stability and the time duration required for the physical system to reach the steady state, is often very large.

Existing time-domain DG schemes use “explicit” updates between the (explicit and/or implicit) solutions of matrix systems arising from the DG-based discretization at every time step during the execution of the time-evolution (which is done using a time integration scheme) [40]–[44]. This approach does not call for a nonlinear solver, but the time-step size is subject to stability conditions required by the spatial discretization of the DD equations and the explicit updates of the nonlinear coupling between solutions of the matrix equations (Section 6.4 of [1]).

We should also note here that, in theory, a fully implicit time-domain DG method that uses a nonlinear scheme (such as Newton method) to solve the globally and (nonlinearly) coupled system of equations could be used. Even though the time-step size restriction would be relaxed for such a method, its application to steady-state analysis would be computationally very expensive, simply because it would call for solution of a global nonlinear system at every time step [1], [22]. To the best of our knowledge, such a time-domain DG method has yet to be reported (even for transient analysis of semiconductor devices).

In our tests we have done for a one dimensional (1D) system with physical parameters same as those in the metal-oxide semiconductor field-effect transistor (MOSFET) example in Section III-A, and using the equilibrium-state solution as the initial solution at the start of time marching, we have observed that the physical time required to reach the steady state (with a tolerance of  $10^{-6}$ ) is around 0.5ps. For an explicit solver, the above physical parameters require the mesh size to be around  $0.01\mu\text{m}$ , which in return requires the time-step size to be around  $5^{-6}$ ps. This results in a total number of time steps around  $10^5$ . Similar numbers have been reported in [42].

### III. NUMERICAL EXAMPLES

In this section, we demonstrate the accuracy and the applicability of the proposed DG-based framework by numerical experiments as detailed in the next two sections. We have simulated two practical devices and compared the results to those obtained by the COMSOL semiconductor module [30].

#### A. METAL-OXIDE FIELD EFFECT TRANSISTOR

First, we simulate a MOSFET that is illustrated in Fig. 2. The background is uniformly-doped p-type silicon and source and drain regions are uniformly-doped n-type silicon. The doping concentrations in p- and n-type regions are  $10^{17}\text{cm}^{-3}$  and  $10^{18}\text{cm}^{-3}$ , respectively. The source and drain are ideal Ohmic contacts attached to n-type regions. The gate contact is

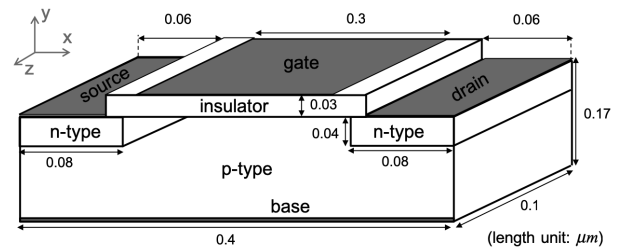


FIGURE 2. Schematic diagram of the MOSFET device.

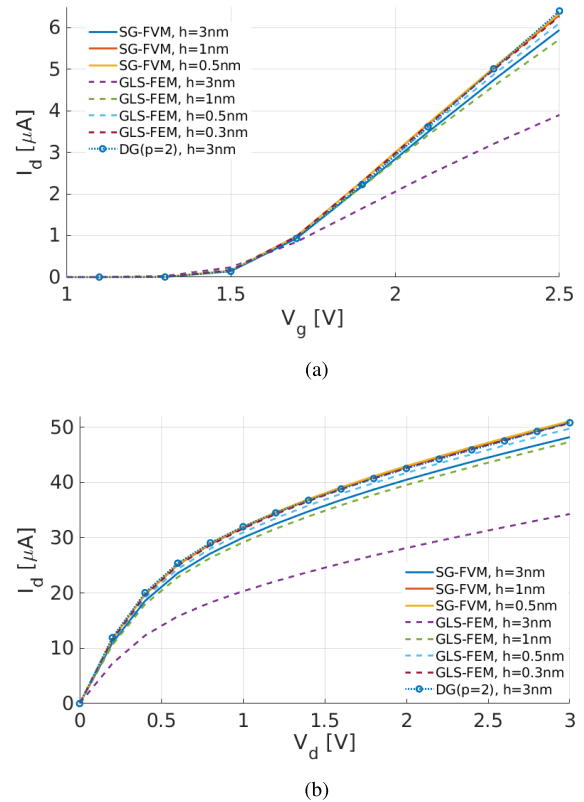


FIGURE 3. (a) The drain current  $I_d$  versus gate voltage  $V_g$  for drain voltage  $V_d = 0.1\text{ V}$ . (b) The drain current  $I_d$  versus drain voltage  $V_d$  for gate voltage  $V_g = 3\text{ V}$ .

separated from the semiconductor regions by a silicon-oxide insulator layer. The dimensions of the device and the different material regions are shown in Fig. 3. Material parameters at 300 K are taken from [59].

Special care needs to be taken to enforce the boundary conditions [58]. The DD equations are only solved in the semiconductor regions. Dirichlet boundary conditions are imposed on semiconductor-contact interfaces, where the electron and hole densities are determined from the local charge neutrality as  $n_e = (C + \sqrt{C^2 + 4n_i^2})/2$  and  $n_h = n_i^2/n_e$ , respectively. Homogeneous Robin boundary condition, which enforces zero net current flow, is imposed on semiconductor-insulator interfaces. Poisson equation is solved in all regions. Dirichlet boundary conditions that enforce impressed external potentials are imposed on metal contacts (the contact barrier is ignored for simplicity).



Homogeneous Neumann boundary condition is used on other exterior boundaries.

The semiconductor regions are discretized using a total of 122 350 elements and the order of basis functions in (32) and (33) is 2. This makes the dimension of the system in (35) 1 223 500. The regions where the Poisson equation is solved are discretized using a total of 170 902 elements and the order of basis functions in (23) and (23) is 2, making the dimension of the system in (25) 1 709 020. The systems (25) and (35) are solved iteratively with a residual tolerance of  $10^{-11}$ . The drop tolerance of the ILU preconditioner is  $10^{-5}$ . The convergence tolerance of the Gummel method is  $10^{-7}$ .

Fig. 3 compares  $I(V)$  curves computed using the proposed DG solver to those computed by the COMSOL semiconductor module. Note that this module includes two solvers: SG-FVM and GLS-FEM. For all three solvers, we refine the mesh until the corresponding  $I(V)$  curve converges with a relative deviation of  $10^{-2}$  (from the reference solution). This relative deviation is defined as  $\sum_V |I(V) - I_{\text{ref}}(V)| / \sum_V |I_{\text{ref}}(V)|$ , where the reference  $I_{\text{ref}}(V)$  curve is obtained from the solution computed by the SG-FVM solver on a mesh with element size  $h = 0.5$  nm. Fig. 3 shows that all  $I(V)$  curves obtained by the three methods converge to  $I_{\text{ref}}(V)$  curve as the mesh they use is made finer. Fig. 3 (a) plots the drain current  $I_d$  versus gate voltage  $V_g$  under a constant drain voltage  $V_d = 0.1$  V. It shows that  $I_d$  increases dramatically as  $V_g$  becomes larger than a turn-on voltage  $V_{\text{th}}$  of approximately 1.5V. This indicates that a tunneling channel is formed between the source and the drain as expected. Fig. 3 (b) plots  $I_d$  versus  $V_d$  for  $V_g = 3$  V. It shows that  $I_d$  increases continuously with  $V_d$  and gradually saturates with a smaller slope of the  $I(V)$  curve.

Comparing the  $I(V)$  curves obtained by the three solvers using meshes with different element sizes, one can clearly see that the GLS-FEM solver requires considerably finer meshes than the SG-FVM and the DG solvers. The relative deviations corresponding to different solvers and element sizes are listed in Table I. To reach a relative deviation of  $10^{-2}$ , the SG-FVM solver uses a mesh with  $h = 1$  nm, the DG solver uses a mesh with  $h = 3$  nm, while the GLS-FEM solver requires  $h$  to be as small as 0.3 nm. The computation time of the DG solver is about 8 hours for each curve shown in Fig. 3 on a workstation with Intel(R) Xeon(R) E5-2680 CPU. We should note here that, since our solver is developed within MATLAB, and the sparse linear solvers are not parallelized (as of 2018 version), it is not straightforward (and also not very meaningful) to compare this computation time to those of the commercial software.

Figs. 4 (a) and (b) and Figs. 4 (c) and (d), respectively, compare the electron density and electric field intensity distributions computed by the DG and GLS-FEM solvers on the plane  $z = 0$  for  $V_g = 3$  V and  $V_d = 0.5$  V. Figs. 4 (a) and (b) illustrate the “field-effect” introduced by the gate voltage, i.e., a sharp conducting channel forms near the top interface facing the gate ( $y = 0.2\mu\text{m}$ ). This sharp boundary layer of carriers is the reason why a very

TABLE 1. Relative deviation in  $I(V)$  curves.

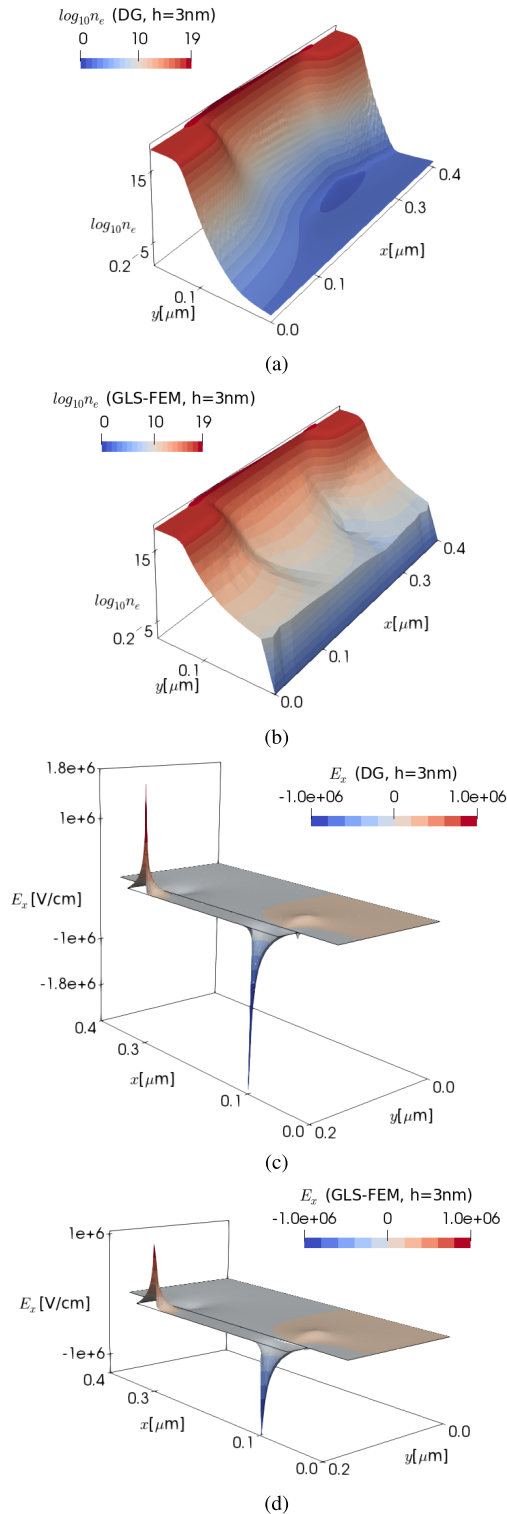
	$I_d(V_g)$	$I_d(V_d)$
SG-FVM, $h = 3$ nm	$5.83 \times 10^{-2}$	$5.91 \times 10^{-2}$
SG-FVM, $h = 1$ nm	$3.90 \times 10^{-3}$	$4.97 \times 10^{-3}$
GLS-FEM, $h = 3$ nm	$3.35 \times 10^{-1}$	$3.49 \times 10^{-1}$
GLS-FEM, $h = 1$ nm	$7.83 \times 10^{-2}$	$8.19 \times 10^{-2}$
GLS-FEM, $h = 0.5$ nm	$2.94 \times 10^{-2}$	$2.98 \times 10^{-2}$
GLS-FEM, $h = 0.3$ nm	$9.11 \times 10^{-3}$	$9.13 \times 10^{-3}$
DG, $h = 3$ nm	$3.05 \times 10^{-3}$	$5.28 \times 10^{-3}$

fine mesh is required to obtain accurate results from this simulation. In Fig. 4 (b), the carrier density decays more slowly [compared to the result in Fig. 4 (a)] away from the gate interface and suddenly drops to the Dirichlet boundary condition values at the bottom interface ( $y = 0$ ). This demonstrates the unphysical smearing of the boundary (carrier) layers observed in GLS-FEM solutions. Figs. 4 (c) and (d) show the x-component of the electric field intensity computed by the DG and the GLS-FEM solvers, respectively. One can clearly see that the solution computed by the GLS-FEM solver is smoother (compared to the DG solution) at the sharp corners of the gate. The unphysical effects, as demonstrated in Figs. 4 (b) and (d), result from the GLS testing, which lacks of control on local conservation law [30].

The SG-FVM solver requires the mesh to be admissible, which is often difficult to satisfy for 3D devices [21], [22], [27]. Implementation of SG-FVM in COMSOL uses a prism mesh generated by sweeping triangles defined on surfaces (for 3D devices) [30]. However, this leads to a considerable increase in the number of elements compared to the number of tetrahedral elements used by the DG and the SG-FEM solvers. In this example, the number of elements used by the SG-FVM is 545 342 ( $h = 1$  nm), which results in 1 499 646 unknowns. The DG solver refines the tetrahedral mesh near the boundaries where the solution changes fast, which is not possible to do using the prism mesh generated by sweeping triangles (Fig. 5). This mesh flexibility compensates for the larger number of unknowns required by the DG solver, which results from defining local expansions only connected by numerical flux.

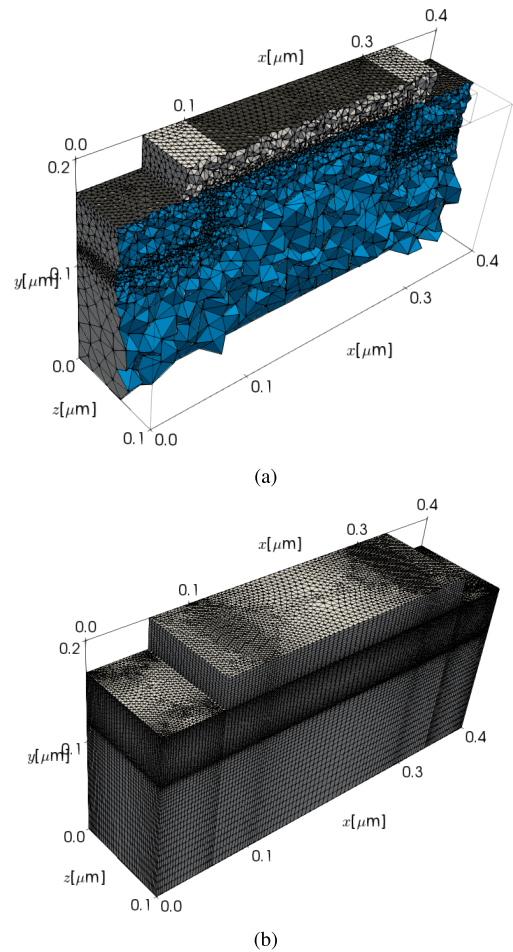
## B. PLASMONIC-ENHANCED PHOTOCONDUCTIVE ANTENNA

For the second example, we consider a plasmonic-enhanced photoconductive antenna (PCA). The operation of a PCA relies on photoelectric effect: it “absorbs” optical wave energy and generate terahertz (THz) short-pulse currents. Plasmonic nanostructures dramatically enhances the optical-THz frequency conversion efficiency of the PCAs. The steady-state response of the PCAs, especially the static electric field and the mobility distribution in the device region, strongly influences their performance. Here, we use the proposed DG solver to simulate the device region of a

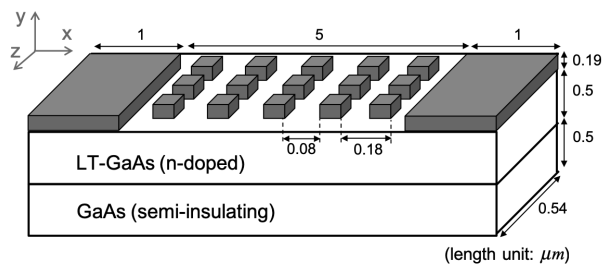


**FIGURE 4.** Electron density distribution computed on the plane  $z = 0$  by (a) the DG solver (b) the GLS-FEM solver for gate voltage  $V_g = 3$  V and drain voltage  $V_d = 0.5$  V.  $x$ -component of the electric field intensity computed on the plane  $z = 0$  by (c) the DG solver (d) the GLS-FEM solver for gate voltage  $V_g = 3$  V and drain voltage  $V_d = 0.5$  V.

PCA shown in Fig. 6, and compare the results to those obtained by the SG-FVM solver in COMSOL semiconductor module.



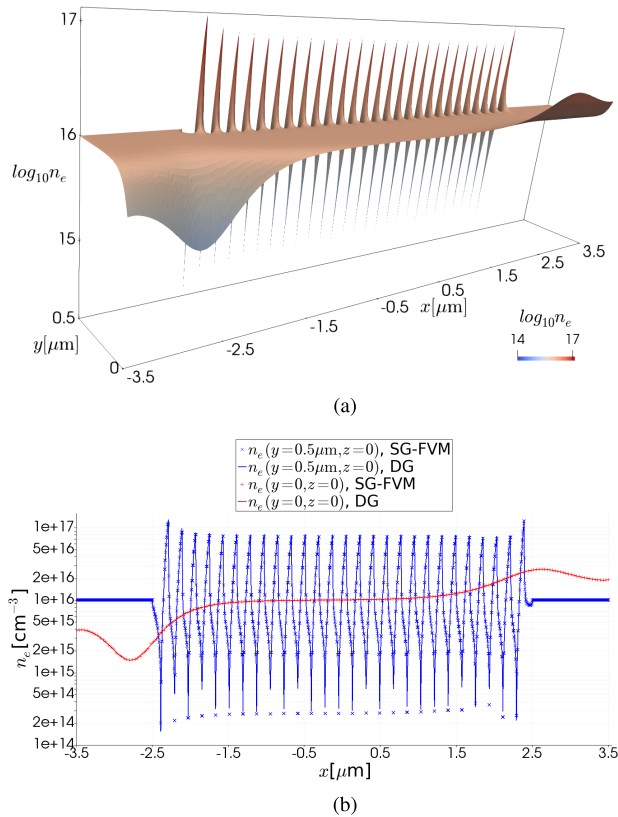
**FIGURE 5.** (a): Tetrahedral mesh used by DG and GLS-FEM. (b): Prism mesh used by SG-FVM in COMSOL [30].



**FIGURE 6.** Schematic diagram of the plasmonic PCA.

Fig. 6 illustrates the device structure that is optimized to enhance the plasmonic fields near the operating optical frequency [60]. The semiconductor layer is LT-GaAs that is uniformly doped with a concentration of  $10^{16} \text{ cm}^{-3}$ . The substrate layer is semi-insulating GaAs. We should note here that it is crucial to employ the appropriate field-dependent mobility models to accurately simulate this device [61]. The Caughey-Thomas model is used here. Other material parameters same as those used in [61]. The bias voltage is set to 10 V.

The DD equations are solved in the semiconductor layer with Dirichlet boundary conditions on the



**FIGURE 7. (a) Electron density distribution computed by the DG solver on the plane  $z = 0$  in the device region of the plasmonic PCA. (b) Electron density computed by the DG and the SG-FVM solvers along lines  $(x, y = 0.5 \mu\text{m}, z = 0)$  and  $(x, y = 0, z = 0)$  versus  $x$ .**

semiconductor-contact interfaces and homogeneous Robin boundary condition on the semiconductor-insulator interfaces. Poisson equation is solved in the whole domain, which includes an extended air background. Dirichlet boundary conditions that enforce impressed external potentials are imposed on metal contacts. Floating potential condition is enforced on metals of the nanograting [62]. Homogeneous Neumann boundary condition is used on exterior boundaries.

The semiconductor region is discretized using a total of 173 711 elements and the order of basis functions in (32) and (33) is 2. This makes the dimension of the system in (35) 1 737 110. The regions where the Poisson equation is solved are discretized using a total of 228 921 elements and the order of basis functions in (23) and (23) is 2, making the dimension of the system in (25) 2 289 210. The systems (25) and (35) are solved iteratively with a residual tolerance of  $10^{-11}$ . The drop tolerance of the ILU preconditioner is  $10^{-5}$ . The convergence tolerance of the Gummel method is  $10^{-7}$ .

Fig. 7 (a) shows the electron density distribution computed by the proposed DG solver. Fig. 7 (b) plots the electron density computed by the DG and the SG-FVM solvers along lines  $(x, y = 0.5 \mu\text{m}, z = 0)$  and  $(x, y = 0, z = 0)$  versus  $x$ . The results agree well. The relative difference, which is defined as  $\|n_e^{\text{DG}} - n_e^{\text{FVM}}\|_2 / \|n_e^{\text{FVM}}\|_2$ , between the solutions obtained

by the DG and the SG-FVM solvers is 0.78%. Here,  $\|\cdot\|_2$  denotes L2 norm and  $n_e^{\text{DG}}$  and  $n_e^{\text{FVM}}$  are the electron densities obtained by the two solvers. Note that  $n_e^{\text{DG}}$  is interpolated to the nodes where  $n_e^{\text{FVM}}$  is computed. The computation time of the DG solver is about 15 hours on a workstation with Intel(R) Xeon(R) E5-2680 CPU.

#### IV. CONCLUSION

In this paper, we report on a DG-based numerical framework for simulating steady-state response of geometrically intricate semiconductor devices with realistic models of mobility and recombination rate. The Gummel method is used to “decouple” and “linearize” the system of the Poisson equation (in electric potential) and the DD equations (in electron and hole charge densities). The resulting linear equations are discretized using the LDG scheme. The accuracy of this framework is validated by comparing simulation results to those obtained by the state-of-the-art FEM and FVM solvers implemented within the COMSOL semiconductor module. Just like FEM, the proposed DG solver is high-order accurate but it does not require the stabilization techniques (such as GLS and SUPG), which are used by FEM. The main drawback of the proposed method is that it requires a larger number of unknowns than FEM for the same geometry mesh. But the difference in the number of unknowns gets smaller with the increasing order of basis function expansion. Additionally, DG can account for non-conformal meshes and benefit from local h/p- refinement strategies. Indeed, we are currently working on a more “flexible” version of the current DG scheme, which can account for multi-scale geometric features more efficiently by making use of these advantages.

#### ACKNOWLEDGMENT

The authors would like to thank the King Abdullah University of Science and Technology (KAUST) Supercomputing Laboratory (KSL) for providing the required computational resources.

#### REFERENCES

- [1] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. New York, NY, USA: Springer-Verlag, 1984.
- [2] D. Vasileska, S. M. Goodnick, and G. Klimeck, *Computational Electronics: Semiclassical and Quantum Device Modeling and Simulation*. Boca Raton, FL, USA: CRC Press, 2010.
- [3] Y. Fu, Z. Li, W. T. Ng, and J. K. Sin, *Integrated Power Devices and TCAD Simulation*. Boca Raton, FL, USA: CRC Press, 2014.
- [4] M. Z. Szymanski, D. Tu, and R. Forchheimer, “2-D drift-diffusion simulation of organic electrochemical transistors,” *IEEE Trans. Electron Devices*, vol. 64, no. 12, pp. 5114–5120, Dec. 2017.
- [5] S. Li, W. Chen, Y. Luo, J. Hu, P. Gao, J. Ye, K. Kang, H. Chen, E. Li, and W.-Y. Yin, “Fully Coupled multiphysics simulation of crosstalk effect in bipolar resistive random access memory,” *IEEE Trans. Electron Devices*, vol. 64, no. 9, pp. 3647–3653, Sep. 2017.
- [6] D.-W. Wang, W. Chen, W.-S. Zhao, G.-D. Zhu, Z.-G. Zhao, J. E. Schutt-Aine, and W.-Y. Yin, “An improved algorithm for drift diffusion transport and its application on large scale parallel simulation of resistive random access memory arrays,” *IEEE Access*, vol. 7, pp. 31273–31285, 2019.
- [7] D. Nagy, G. Indalecio, A. J. Garcia-Loureiro, G. Espineira, M. A. Elmessary, K. Kalna, and N. Seoane, “Drift-diffusion versus Monte Carlo simulated ON-current variability in nanowire FETs,” *IEEE Access*, vol. 7, pp. 12790–12797, 2019.

- [8] D. Rossi, F. Santoni, M. Auf Der Maur, and A. Di Carlo, "A multiparticle drift-diffusion model and its application to organic and inorganic electronic device simulation," *IEEE Trans. Electron Devices*, vol. 66, no. 6, pp. 2715–2722, Jun. 2019.
- [9] TCAD Central. Accessed: Nov. 30, 2019. [Online]. Available: <https://tcad.com/Software.html>
- [10] J. Piprek, *Handbook of Optoelectronic Device Modeling and Simulation*. Boca Raton, FL, USA: CRC Press, 2018.
- [11] D. Scharfetter and H. Gummel, "Large-signal analysis of a silicon Read diode oscillator," *IEEE Trans. Electron Devices*, vol. 16, no. 1, pp. 64–77, Jan. 1969.
- [12] R. Bank, D. Rose, and W. Fichtner, "Numerical methods for semiconductor device simulation," *IEEE Trans. Electron Devices*, vol. 30, no. 9, pp. 1031–1041, Sep. 1983.
- [13] R. Sacco and F. Saleri, "Mixed finite volume methods for semiconductor device simulation," *Numer. Methods Partial Differ. Eq.*, vol. 13, no. 3, pp. 215–236, May 1997.
- [14] R. E. Bank, W. Coughran, Jr., and L. C. Cowsar, "The finite volume Scharfetter-Gummel method for steady convection diffusion equations," *Comput. Vis. Sci.*, vol. 1, no. 3, pp. 123–136, Nov. 1998.
- [15] C. Chainais-Hillairet, J.-G. Liu, and Y.-J. Peng, "Finite volume scheme for multi-dimensional drift-diffusion equations and convergence analysis," *Math. Model. Numer. Anal.*, vol. 37, no. 2, pp. 319–338, Mar. 2003.
- [16] M. Bessemoulin-Chatard, "A finite volume scheme for convection-diffusion equations with nonlinear diffusion derived from the Scharfetter-Gummel scheme," *Numer. Math.*, vol. 121, no. 4, pp. 637–670, Aug. 2012.
- [17] M. Bessemoulin-Chatard and F. Filbet, "A finite vol. scheme, for nonlinear degenerate parabolic equations," *SIAM J. Sci. Comput.*, vol. 34, no. 5, pp. B559–B583, 2012.
- [18] F. Brezzi, L. D. Marini, and P. Pietra, "Two-dimensional exponential fitting and applications to drift-diffusion models," *SIAM J. Numer. Anal.*, vol. 26, no. 6, pp. 1342–1355, Dec. 1989.
- [19] J. Xu and L. Zikatanov, "A monotone finite element scheme for convection-diffusion equations," *Math. Comp.*, vol. 68, no. 228, pp. 1429–1447, May 1999.
- [20] R. D. Lazarov and L. T. Zikatanov, "An exponential fitting scheme for general convection-diffusion equations on tetrahedral meshes," Nov. 2012, *arXiv:1211.0869*. [Online]. Available: <https://arxiv.org/abs/1211.0869>
- [21] A. Mauri, A. Bortolossi, G. Novielli, and R. Sacco, "3D finite element modeling and simulation of industrial semiconductor devices including impact ionization," *J. Math. Ind.*, vol. 5, no. 1, p. 1, 2015.
- [22] P. Farrell, N. Rotundo, D. H. Doan, M. Kantner, J. Fuhrmann, and T. Koprucki, "Drift-diffusion models," in *Handbook of Optoelectronic Device Modeling and Simulation*. Boca Raton, FL, USA: CRC Press, 2017, pp. 733–772.
- [23] T. Hughes and A. Brooks, "A multi-dimensionsoal upwind scheme with no crosswind diffusion," *Amer. Soc. Mech. Eng., Appl. Mech. Division*, vol. 34, pp. 19–35, Jan. 1979.
- [24] A. N. Brooks and T. J. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Comput. Methods Appl. Mech. Eng.*, vol. 32, nos. 1–3, pp. 199–259, Sep. 1982.
- [25] T. J. Hughes, L. P. Franca, and G. M. Hulbert, "A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations," *Comput. Methods Appl. Mech. Eng.*, vol. 73, no. 2, pp. 173–189, May 1989.
- [26] L. P. Franca and R. Stenberg, "Error analysis of Galerkin least squares methods for the elasticity equations," *SIAM J. Numer. Anal.*, vol. 28, no. 6, pp. 1680–1697, Dec. 1991.
- [27] G. F. Carey, A. L. Pardhanani, and S. W. Bova, "Advanced numerical methods and software approaches for semiconductor device simulation," *VLSI Des.*, vol. 10, no. 4, pp. 391–414, Jan. 2000.
- [28] M. Stynes, "Steady-state convection-diffusion problems," *Acta Numerica*, vol. 14, pp. 445–508, May 2005.
- [29] A. Ngo, P. Bastian, and O. Ippisch, "Numerical solution of steady-state groundwater flow and solute transport problems: Discontinuous Galerkin based methods compared to the Streamline Diffusion approach," *Comput. Methods Appl. Mech. Eng.*, vol. 294, pp. 331–358, Sep. 2015.
- [30] *Semiconductor Module User's Guide, Version 5.3a*, COMSOL Multiphysics, Stockholm, Sweden, 2017.
- [31] F. Brezzi, L. Marini, S. Micheletti, P. Pietra, R. Sacco, and S. Wang, "Discretization of semiconductor device problems (I)," *Handbook Numer. Anal. Numer. Methods Electromagn.*, vol. 13, pp. 317–441, Jan. 2005.
- [32] H.-G. Roos, M. Stynes, and L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems*, vol. 24. Springer, 2008.
- [33] H.-G. Roos, "Robust numerical methods for singularly perturbed differential equations: A survey covering 2008–2012," *ISRN Appl. Math.*, vol. 2012, pp. 1–30, Dec. 2012.
- [34] B. Cockburn, "Discontinuous Galerkin methods for convection-dominated problems," in *High-Order Methods for Computational Physics*. Springer, 1999, pp. 69–224.
- [35] B. Riviere, *Discontinuous Galerkin Methods for Solving Elliptic Parabolic Equations: Theory implementation*. Philadelphia, PA, USA: SIAM, 2008.
- [36] J. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. New York, NY, USA: Springer, 2008.
- [37] D. A. A. Pietro and A. Ern, "Mathematical aspects of discontinuous Galerkin methods," in *Mathématiques Applications*, vol. 69. Springer, 2012.
- [38] C.-W. Shu, *Discontinuous Galerkin Methods for Time-Dependent Convection Dominated Problems: Basics, Recent Developments Comparison with Other Methods*. Cham, Switzerland: Springer, 2016, pp. 371–399.
- [39] M. Augustin, A. Caiazzo, A. Fiebach, J. Fuhrmann, V. John, A. Linke, and R. Umla, "An assessment of discretizations for convection-dominated convection-diffusion equations," *Comput. Methods Appl. Mech. Eng.*, vol. 200, nos. 47–48, pp. 3395–3409, Nov. 2011.
- [40] Y. Liu and C.-W. Shu, "Local discontinuous Galerkin methods for moment models in device simulations: Formulation and one dimensional results," *J. Comput. Electron.*, vol. 3, nos. 3–4, pp. 263–267, Oct. 2004.
- [41] Y. Liu and C.-W. Shu, "Local discontinuous Galerkin methods for moment models in device simulations: Performance assessment and two-dimensional results," *Appl. Numer. Math.*, vol. 57, nos. 5–7, pp. 629–645, May 2007.
- [42] Y. Liu and C.-W. Shu, "Analysis of the local discontinuous Galerkin method for the drift-diffusion model of semiconductor devices," *Sci. China Math.*, vol. 59, no. 1, pp. 115–140, Jan. 2016.
- [43] G. Kumar, M. Singh, A. Bulusu, and G. Trivedi, "A framework to simulate semiconductor devices using parallel computer architecture," *J. Phys., Conf. Ser.*, vol. 759, Oct. 2016, Art. no. 012098.
- [44] M. Harmon, I. M. Gamba, and K. Ren, "Numerical algorithms based on Galerkin methods for the modeling of reactive interfaces in photoelectrochemical (PEC) solar cells," *J. Comput. Phys.*, vol. 327, pp. 140–167, Dec. 2016.
- [45] Y. Cheng, I. M. Gamba, A. Majorana, and C.-W. Shu, "A brief survey of the discontinuous Galerkin method for the Boltzmann-Poisson equations," *SeMA J.*, vol. 54, no. 1, pp. 47–64, Apr. 2011.
- [46] D. Vasilevka and S. M. Goodnick, "Computational electronics," *Synth. Lectures Comput. Electromagn.*, vol. 1, no. 1, pp. 1–216, 2005.
- [47] H. Gummel, "A self-consistent iterative scheme for one-dimensional steady state transistor calculations," *IEEE Trans. Electron Devices*, vol. 11, no. 10, pp. 455–465, Oct. 1964.
- [48] P. A. Markowich, *The Stationary Semiconductor Device Equations*. New York, NY, USA: Springer-Verlag, 1986.
- [49] *Atlas User's Manual: Device Simulation Software*, Silvaco, Int., Santa Clara, CA, USA, 2016.
- [50] *Minimos-Nt User Manual*, Inst. for Microelectron. (TU Vienna) Global TCAD Solutions GmbH, Santa Clara, CA, USA, 2017.
- [51] B. Cockburn and C.-W. Shu, "The local discontinuous Galerkin method for time-dependent convection-diffusion systems," *SIAM J. Numer. Anal.*, vol. 35, no. 6, pp. 2440–2463, Dec. 1998.
- [52] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau, "An A Priori error analysis of the local discontinuous Galerkin method for elliptic problems," *SIAM J. Numer. Anal.*, vol. 38, no. 5, pp. 1676–1706, Jan. 2000.
- [53] P. Causin and R. Sacco, "A discontinuous petrov-Galerkin method with lagrangian multipliers for second order elliptic problems," *SIAM J. Numer. Anal.*, vol. 43, no. 1, pp. 280–302, Jan. 2005.
- [54] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, and R. Sacco, "A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems," *SIAM J. Sci. Comput.*, vol. 31, no. 5, pp. 3827–3846, Jan. 2009.
- [55] A. L. Lombardi and P. Pietra, "Exponentially fitted discontinuous Galerkin schemes for singularly perturbed problems," *Numer. Methods Partial Differ.*, vol. 28, no. 6, pp. 1747–1777, Nov. 2012.

- [56] R. Borker, C. Farhat, and R. Tezaur, "A discontinuous Galerkin method with Lagrange multipliers for the solution of Helmholtz problems in the mid-frequency regime," *Comput. Methods Appl. Mech. Eng.*, vol. 327, pp. 93–117, Mar. 2017.
- [57] J. W. Jerome, *Analysis of Charge Transport: A Mathematical Study of Semiconductor Devices*. Springer, 2012.
- [58] D. Schroeder, *Modelling of Interface Carrier Transport for Device Simulation*. New York, NY, USA: Springer-Verlag, 1994.
- [59] M. E. Levinshtein and S. L. Rumyantsev, "Silicon (Si)," in *Handbook Series On Semiconductor Parameters: Si, Ge, C (Diamond), GaAs, GaP, GaSb, InAs, InP, InSb*, vol. 1. Singapore: World Scientific, 1996, pp. 1–32.
- [60] M. Bashirpour, S. Ghorbani, M. Kolahdouz, M. Neshat, M. Masnadi-Shirazi, and H. Aghababa, "Significant performance improvement of a terahertz photoconductive antenna using a hybrid structure," *RSC Adv.*, vol. 7, no. 83, pp. 53010–53017, Nov. 2017.
- [61] E. Moreno, M. F. Pantoja, S. G. Garcia, A. R. Bretones, and R. G. Martin, "Time-domain numerical modeling of THz photoconductive antennas," *IEEE Trans. THz Sci. Technol.*, vol. 4, no. 4, pp. 490–500, Jul. 2014.
- [62] A. Konrad and M. Graovac, "The finite element modeling of conductors and floating potentials," *IEEE Trans. Magn.*, vol. 32, no. 5, pp. 4329–4331, Sep. 1996.



**LIANG CHEN** (Member, IEEE) received the B.S. degree in applied physics from Southeast University, Nanjing, China, in 2007, and the M.S. and Ph.D. degrees in microelectronic and solid-state electronics from the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, in 2009 and 2012, respectively. From 2012 to 2013, he was a High-Performance Computing Engineer with the HPC Solution Center, Dawning Information Industry Company, Ltd. From 2013 to 2017, he was a Principal Developer of the electromagnetics simulation software EastWave and also served as the Director of the Technical Solution Department, DongJun Technology Company, Ltd. Since 2017, he has been with the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, as a Postdoctoral Research Fellow. His research interests include computational physics (electromagnetics, semiconductor devices, quantum transport, and multiphysics simulation), computer-aided engineering software, high-performance computing, and various topics in photonics and condensed matter physics.



**HAKAN BAGCI** (Senior Member, IEEE) received the B.S. degree in electrical and electronics engineering from the Bilkent University, Ankara, Turkey, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, USA, in 2003 and 2007, respectively. From June 1999 to July 2001, he worked as an Undergraduate Researcher with the Computational Electromagnetics Group, Bilkent University. From August 2001 to December 2006, he was a Research Assistant with the Center for Computational Electromagnetics and Electromagnetics Laboratory, UIUC. From January 2007 to August 2009, he was a Research Fellow with the Radiation Laboratory, University of Michigan, Ann Arbor, MI, USA. Since August 2009, he has been with the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, where he is currently an Associate Professor of electrical engineering. He has authored or coauthored more than 90 journal articles and more than 200 papers in conference proceedings. His research interests include various aspects of theoretical and applied computational electromagnetics with emphasis on well-conditioned frequency and time domain integral equation formulations and their discretization, hybrid time domain integral and differential equation solvers, accurate, stable, and efficient marching schemes for time domain solvers, stochastic characterization of electromagnetic field and wave interactions on complex geometries, and solution of two and three dimensional electromagnetic inverse scattering problem using signal processing techniques.

Dr. Bagci was a recipient of the 2008 International Union of Radio Scientists (URSI) Young Scientist Award and the 2004–2005 Interdisciplinary Graduate Fellowship from the Computational Science and Engineering Department, UIUC. His article titled Fast and Rigorous Analysis of EMC/EMI Phenomena on Electrically Large and Complex Structures Loaded With Coaxial Cables was one of the three finalists (with honorable mention) for the 2008 Richard B. Schulz Best Transactions Paper Award given by the IEEE Electromagnetic Compatibility Society. He has authored (as a student) or coauthored (as a student and an advisor) 17 finalist/honorable mention articles in the student article competitions at the 2005, 2008, 2010, 2014, 2015, 2016, 2017, and the 2018 IEEE Antennas and Propagation Society International Symposia and 2013, 2014, 2016, 2017, and the 2018 Applied Computational Electromagnetics Society Conferences. He is also an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, the IEEE JOURNAL ON MULTISCALE AND MULTIPHYSICS COMPUTATIONAL TECHNIQUES, and the *IEEE Antennas and Propagation Magazine*.

• • •