Journal of Computational Electronics (2022) 21:10–20
https://doi.org/10.1007/s10825-021-01823-3

Parallel approach of Schrödinger-based quantum corrections for ultrascaled semiconductor devices

Gabriel Espiñeira[1] · Antonio J. García-Loureiro[1] · Natalia Seoane[1]

Received: 5 August 2021 / Accepted: 11 November 2021 / Published online: 27 December 2021
© The Author(s) 2021

Abstract

In the current technology node, purely classical numerical simulators lack the precision needed to obtain valid results. At the same time, the simulation of fully quantum models can be a cumbersome task in certain studies such as device variability analysis, since a single simulation can take up to weeks to compute and hundreds of device configurations need to be analyzed to obtain statistically significative results. A good compromise between fast and accurate results is to add corrections to the classical simulation that are able to reproduce the quantum nature of matter. In this context, we present a new approach of Schrödinger equation-based quantum corrections. We have implemented it using Message Passing Interface in our in-house built semiconductor simulation framework called VENDES, capable of running in distributed systems that allow for more accurate results in a reasonable time frame. Using a 12-nm-gate-length gate-all-around nanowire FET (GAA NW FET) as a benchmark device, the new implementation shows an almost perfect agreement in the output data with less than a 2% difference between the cases using 1 and 16 processes. Also, a reduction of up to 98% in the computational time has been found comparing the sequential and the 16 process simulation. For a reasonably dense mesh of 150k nodes, a variability study of 300 individual simulations can be now performed with VENDES in approximately 2.5 days instead of an estimated sequential execution of 137 days.

Keywords Drift-diffusion · Schrödinger quantum corrections · Gate-all-around nanowire FET · Finite element method · Message passing interface

1 Introduction

Silicon device architectures have been developed rapidly during the last few decades, the FinFET being the current standard for mass production. However, IRDS predictions [1] point toward a change in the leading architectures for the next technology nodes, with promising candidates such as the GAA NW FET and the GAA nanosheet. This shift in the industry standard calls for new architecture designs able to predict the performance and reliability of new devices. For the current transistor dimensions, the traditional process to develop and optimize devices based on trial and error during the manufacturing stage is unviable. Consequently, Technology Computer Aided Design (TCAD) has become an indispensable tool to both predict future device architectures and optimize the present ones [2–6].

Traditionally, when devices were in the micron scale, the resolution of classical models was the most popular choice to predict device behavior, for instance the drift-diffusion (DD) method. Currently, device dimensions have been scaled down to the nanometer regime what requires the use of more complicated and more time-consuming simulation methods. Some of these are the particle-based semiclassical Monte Carlo (MC) [7, 8], or the purely quantum non-equilibrium Green's function (NEGF) [9] [10, 11]. A common alternative, as it is a good compromise between simulation time and accurate results, is to include quantum corrections to classical models such as those based on the solution of the density gradient (DG) [12–14] or the Schrödinger (SCH) [15] [16–18] equations.

As billions of transistors coexist in a die, hundreds or thousands of devices need to be simulated to obtain realistic results in variability studies [4, 19–21], making this a very computationally demanding job regardless of the simulation

✉ Gabriel Espiñeira
g.espineira@usc.es

[1] CITIUS, Universidade de Santiago de Compostela, Santiago de Compostela, Spain

method. Therefore, the optimization and parallelization of simulation frameworks play a key role in current device design, enabling us to obtain valid physical results at a fraction of the computational cost.

In this paper, we propose and evaluate a resolution scheme of SCH equation-based corrections compatible with a highly parallel DD model explained in [22], used in the resolution of 3D finite element (FE) meshes. By implementing these corrections in a scalable manner, we will be able to obtain accurate results without perpetuating the simulation time. In the literature, there are several simulation toolboxes that also implement the resolution of the SCH equation to include quantum corrections in classical simulators, following a similar methodology. These simulators either solve the SCH equation in 2D [18] or in 1D [23]. The main advantage of our proposal is that the 3D unstructured mesh can be divided into separate domains and executed in parallel with MPI in distributed systems, decreasing simulation times considerably. We have used as a benchmark device a 12-nm gate-all-around nanowire FET, one of the main contenders to replace FinFETs as the preferred device architecture for mass production [24–27]. The structure of this paper starts in Sect. 2, where we describe the simulation software and the benchmark device used in this study. In Sect. 3, we present the 2D SCH scheme and the implemented parallelization. In Sect. 4, different simulation results and an efficiency discussion are presented. Finally, in Sect. 5 we draw the main conclusions.

2 Simulation framework and benchmark device

VENDES [22] is a semiconductor simulation framework which includes several tools for modeling complex ultrascaled semiconductor devices such as FinFETs [8], GAA NW FETs [25] or vertically stacked nanosheets [28].

An overall look of the VENDES framework workflow and its capabilities is shown in Fig 1. The first step in the simulation process consists of generating a 3D finite element mesh made of tetrahedra that represents the structure of the device. The modeling of the device has been done via an open-source software called Gmsh [29]. The geometry is constructed using parametric design according to the device dimensions (see Fig. 2 a). Then, using a Delaunay triangulation technique, the 3D FE mesh is generated and optimized as shown in Fig. 2b. Every node of the 3D FE mesh is given a set of physical properties needed for the simulation, such as material permittivities, doping profiles or carrier mobilities. Note that a FE scheme has been chosen, opposed to the finite difference method, since it is able to describe complex 3D irregular geometries. At this point, the user can choose to apply to the device several built-in variability models that

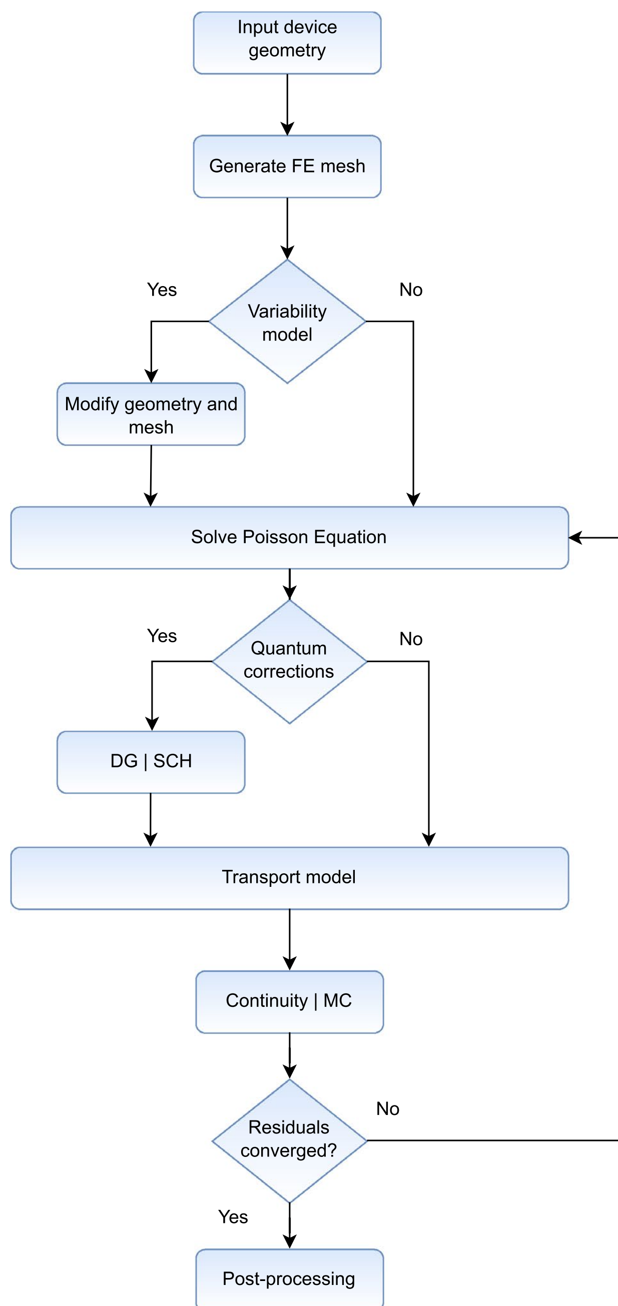

Fig. 1 VENDES flowchart. It can be seen the different steps needed to simulate, from the geometry input and FE mesh generation, to the calculations performed to solve the Poisson equation and the transport models

can either modify the properties of the nodes such as random discrete dopants and metal grain granularity or modify the geometry of the device like line edge roughness or gate edge roughness.

The device that has been used in this work is a state-of-the-art 12-nm Si gate-all-around (GAA) nanowire (NW) FET based on experimental data from [30]. The main dimensions and doping profile characteristics are given in Table 1.

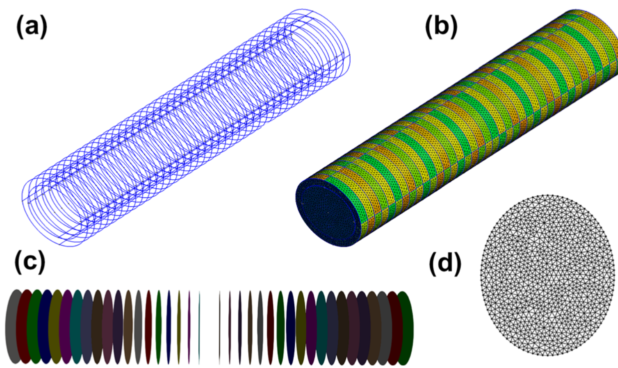

Fig. 2 The first step in the device modeling process is to design the geometry in Gmsh [29] (**a**) and generate the FE mesh (**b**). Using the spatial coordinates of the FE nodes, 2D slices are extracted using user-fixed criteria to obtain a 2D mesh (**c**). One of the FE 2D slices is presented in (**d**)

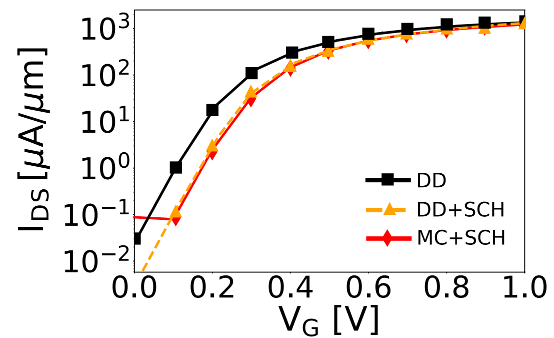

Fig. 3 I_D–V_G characteristics comparing 3D drift-diffusion (DD) simulations against: i) SCH quantum-corrected drift-diffusion (DD+SCH) and ii) Monte Carlo quantum-corrected simulations (MC+SCH)

Table 1 Device dimensions for the 12-nm-gate-length GAA NW FET

Gate length (L_G)[nm]	12.0
S/D length ($L_{S/D}$)[nm]	14.0
Channel width (W_{CH})[nm]	7.5
Channel height (H_{CH})[nm]	9.5
Equivalent oxide thickness (EOT)[nm]	1.0
Channel p-type doping (N_{CH})[cm^{-3}]	1×10^{15}
S/D n-type doping ($N_{S/D}$)[cm^{-3}]	5×10^{19}
S/D doping lateral straggle (σ_x)	3.45
S/D doping lateral peak (x_p)[nm]	11.3
Gate perimeter [nm]	26.8

The device channel has been uniformly doped, whereas the source/drain (S/D) regions have a Gaussian doping. These Gaussian doping profiles, reverse-engineered from experimental data [7], are characterized by the S/D doping lateral straggle (σ_x), which describes the slope of Gaussian profile, and the S/D doping lateral peak (x_p), which indicates the position where the Gaussian decay starts measured from the middle of the channel (see Table 1).

After the device geometry and properties have been defined, in the beginning of every simulation, an initial solution is calculated using only the Poisson equation at equilibrium at $V_G = 0.0$ V and $V_D = 0.0$ V. The output of this initial routine is a purely classical electrostatic potential. However, a solution that seeks a compromise between sound results and short simulation times has been to include quantum corrections [18, 31, 32]. Some of the most commonly used quantum corrections are the DG corrections that require calibration against a quantum mechanical simulation as explained in [12] or the SCH corrections, based on solving the FE 2D SCH equation that does not demand any fitting parameters.

The calibration of the I_D-V_G characteristics is shown in Fig. 3, comparing the classical DD simulations against SCH quantum-corrected DD and MC simulations. Note that the inclusion of quantum corrections is essential to properly characterize the device behavior. The DD simulation matches the MC+SCH in the ON region extraordinarily well, thanks to the fitting of the saturation velocity and the mobility. Finally, it is worth mentioning that at very low gate biases the MC+SCH is too noisy, and it is unable to obtain the correct off-current.

In VENDES, the simulation of the transport inside the channel of the device can be done with two different simulation schemes. The DD approach is a vastly implemented scheme used to calculate carrier transport in semiconductor device simulations [12, 33, 34]. This simulation method describes the relation between the electrostatic potential and the density of carriers in the device. Secondly, the ensemble MC technique solves the Boltzmann transport equation (BTE) using the charge distribution throughout the device to calculate both the electrostatic potential and electric field. A more detailed explanation of the implementation and solution of the DD model can be found in [12] and for the ensemble MC in [7].

3 Parallel Schrödinger-based quantum corrections

3.1 2D Schrödinger workflow

After the first Poisson iteration has been completed and an initial solution for the potential has been found, the SCH quantum corrections routine can be selected as shown in Fig. 4. Even though the device mesh is three-dimensional, the SCH algorithm is based on solving the 2D time-independent form of the equation to ensure a good compromise between accuracy and speed in the results since by removing one dimension from the SCH equation, the

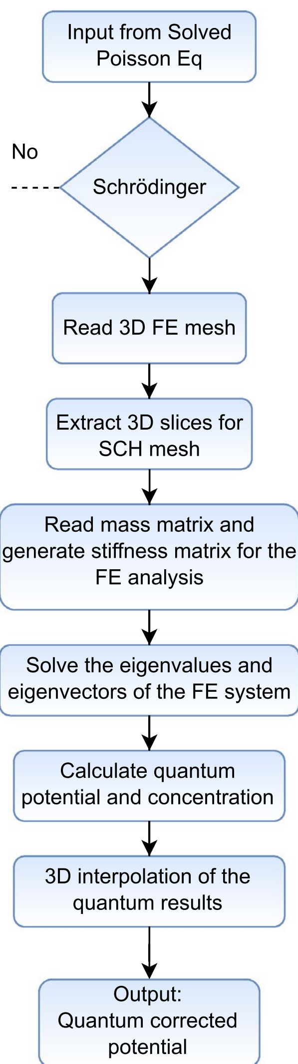

Fig. 4 SCH subroutine flowchart. In this scheme, the different processes of the SCH methodology are presented. First, the 3D information from the FE mesh is adapted to a 2D mesh for less complex computations. After the solution has been obtained, the results are used to interpolate the quantum corrections to the nodes in the FE mesh

complexity and resolution time drastically decrease. The 2D SCH equation is solved in YZ planes perpendicular to the transport direction. To create these planes, several X coordinates along the transport direction are defined. Since a 3D unstructured FE mesh is used in this study, the GAA NW FET has been designed in Gmsh by extruding the mesh specifically in the x-coordinates where the 2D SCH will be solved. Depending on the device and the concentration profiles, the number of 2D slices can vary in order to properly capture the quantum confinement. Using this method allows to solve the SCH equation in the 2D slice nodes and interpolate the results in the 3D nodes placed in between the 2D slices at a fraction of time compared to solving the 3D SCH equation.

Then, all 3D nodes with the same x-coordinate are grouped into a single 2D slice. In Fig. 2c, it can be seen the full 2D mesh of a GAA NW FET containing 41 cuts transversal to the transport direction. In Fig. 2d, the node structure of a single slice can be seen showing the 2D triangular mesh where the SCH equation is solved.

Prior to the resolution of the 2D time-independent SCH equation, it is necessary to read the mass matrix and to generate the stiffness matrix. This is done using the Galerkin method [35] [36]. However, since the SCH equation is solved continuously through the simulation, the mass lumping technique was implemented to diagonalize the mass matrix which makes the numerical resolution faster [35, 37, 38]. Once the FEM system has been generated, the eigenproblem is solved using EB13 [39], a library routine based on Arnoldi's iterative resolution method. The results of the standard generalized eigenproblem are both the eigenstates of the 2D SCH equation $\psi_i(y,z)$ and the eigenenergies $E_i(y,z)$. These two parameters are obtained for every node of the 2D slice and are used to calculate the quantum mechanical electron density $n_{sc}(y,z)$ following the Boltzmann approximation:

$$n_{sc}(y,z) = g\frac{\sqrt{2\pi m^* k_B T}}{\pi\hbar}\text{x} \\ \text{x}\sum_i |\psi_i(y,z)|^2 \exp\left[\frac{E_{F_n} - E_i}{k_B T}\right] \quad (1)$$

where g is the degeneracy factor, k_B is the Boltzmann constant, T is the temperature, m^* is the electron effective transport mass and E_{F_n} is the quasi-Fermi level as shown in [40].

Now that the quantum-corrected electron concentration has been calculated in the nodes of the 2D slices, it is used to correct the electron concentration in the nodes of the 3D mesh. This process is done using high-order Lagrange polynomials. When the value of the quantum concentration is needed for a 3D node that is not contained in a 2D slice, the position of the node is projected into the closest slices, four in the case of cubic interpolation. Then, using the shape functions from the finite element method, the quantum concentration is calculated at the triangular element of the 2D cut where the 3D node projection is. Once the quantum concentration is known at the node projections, the concentration profile at those coordinates can be obtained using the interpolation polynomials, and the quantum concentration at the original 3D point is calculated. A visual depiction of the 3D interpolation process is shown in Fig. 5.

The end result is a quantum-corrected potential assigned to every node of the full 3D mesh ($V_{node}(\mathbf{r})$) that is calculated using the following formula:

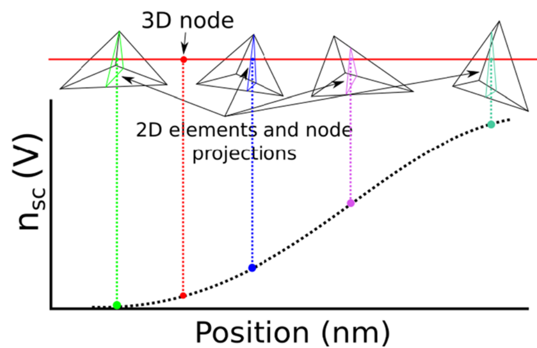

Fig. 5 Lagrange interpolation of the 3D quantum concentration. For a cubic interpolation, four slices are taken and the 2D quantum concentration in each one of them is calculated in the node projection using the FEM shape functions

$$V_{node}(\mathbf{r}) = V_{sc}(\mathbf{r}) + V_{cl}(\mathbf{r}) = \\ = \frac{k_B T}{q} \log(n_{sc}(\mathbf{r})/n_i(\mathbf{r})) \quad (2)$$

where $n_i(\mathbf{r})$ is the effective intrinsic carrier concentration of electrons and holes, $V_{sc}(\mathbf{r})$ is the 3D SCH quantum correction to the potential and $V_{cl}(\mathbf{r})$ is the classical potential obtained as the solution of the Poisson equation.

A full in-depth description of the SCH quantum corrections inclusion in the main DD scheme can be found in [13].

3.2 2D Schrödinger parallel implementation

Prior to this work, most of VENDES routines were implemented including MPI directives and could be executed in a distributed manner speeding up the simulation process, except for the SCH routines. In VENDES, the Poisson and electron continuity equations are first decoupled using Gummel methods and then linearized using Newton's algorithm. The resulting linear system is solved using a domain decomposition technique [12]. The problem domain is partitioned into several subdomains that can be solved in a parallel manner. In order to do this, the linear system is rearranged as shown for a two-domain example in Fig. 6. The whole domain is divided into internal nodes, whose solution is only stored locally in every process, internal boundary nodes, with a solution that is obtained locally and then shared with adjacent processes, and then external boundary nodes, whose solution is calculated locally in a different process and then shared from this process with its neighbors. The obtained solution in the internal and external nodes is interchanged among adjacent domains after every iteration to assure the consistency between the adjacent information stored in all the processes (see Fig. 6).

This node scheme is implemented not only as a way of distributing the problem domain, but also it has been shown that placing the external boundary nodes at the end of the linear matrices improves performance [41].

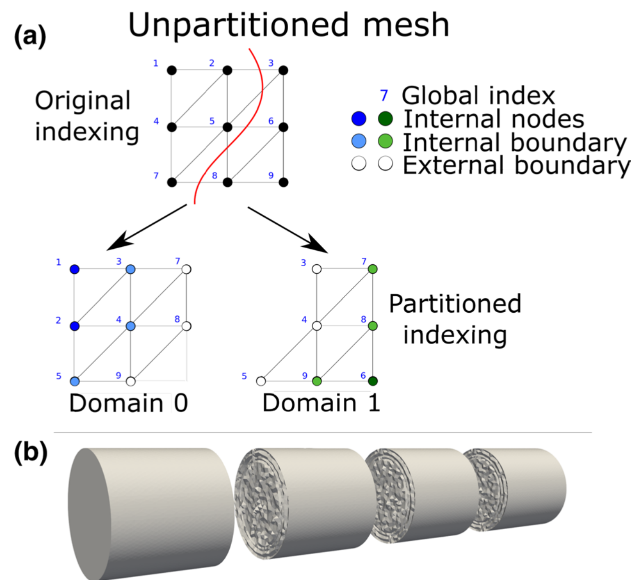

Fig. 6 **a** Example of a two-domain partitioned mesh. After the different domains have been determined, the node indexing is rearranged according to the local domain. Note that a halo of nodes is kept so that after every iteration of the solver, the results are communicated between domains for self-consistency. **b** Image showing a mesh divided into four domains. Note that the rough boundaries are made of the duplicated external boundary nodes between two adjacent processes

One of the first issues arising when trying to parallelize the SCH algorithm was that the SCH quantum concentration is obtained in 2D slices in several coordinates along the transport direction, some of which can be exactly located where the domain boundary is and be split between two adjacent processes. For a GAA NW FET with approximately 100k nodes, the SCH routine is solved 164 times in a single full IV curve simulation and the additional synchronization after every iteration of the routine would have too much impact on the resolution performance. To avoid having a 2D slice divided between two processors, a block partitioning scheme was implemented (see Fig. 7).

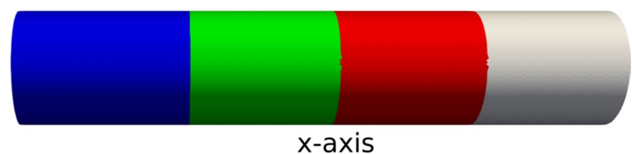

Fig. 7 Block partitioning scheme for the benchmark GAA NW FET. By using this decomposition method, the domain boundaries are not divided between adjacent processes and additional synchronization is avoided

After the 3D mesh is divided into the desired number of processors, the first step is to generate the 2D SCH mesh. As mentioned earlier, the user defines a set of coordinates along the x-axis where the 2D cuts will be performed. During this process, the boundary slices are duplicated so that both adjacent processes contain the slice with the external boundary nodes and do not have to share any information after every iteration in the SCH solution. After the duplication of the external boundary slice, all the 3D nodes of the mesh are in between slices that allow to interpolate the quantum-corrected concentration after solving the SCH equation. A visual representation of the parallel resolution is shown in Fig. 8.

Once all the slices are distributed, each process solves all the 2D slices it contains sequentially. Next, every node of the 2D cut is assigned a value of the quantum concentration and later interpolated to all the nodes of the local 3D mesh. With the coupled Poisson-SCH equation solver, using the quantum-corrected value of the concentration in the 3D mesh, the quantum-corrected potential is calculated in all the internal nodes of every domain. Then, the continuity equation is solved in the 3D mesh using Newton method until convergency is reached.

The linear solver algorithm is based on the additive Schwarz method that first solves the linear system locally. After every local iteration, the global linear system needs to be updated. This prompts that after every iteration all the processors exchange information, and the higher the number of processors, the more information that has to be synchronized. Therefore, the scalability of the code will be dependent on a compromise between the node communication network speed and the efficiency of the linear systems solver to avoid additional synchronization overheads.

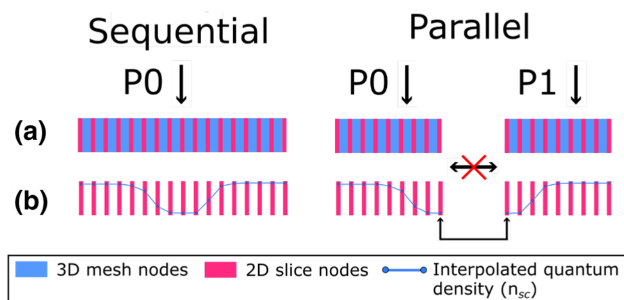

Fig. 8 Parallel Schrödinger (SCH) scheme. **a** In the parallel resolution of the SCH algorithm, the boundary slice between two domains is duplicated. This was done as the MPI communication between domains after every iteration was more time-consuming than the extra computing expense of an extra slice per domain. **b** After the quantum density (n_{SC}) is calculated in every node of the 2D slices, a set of high-order interpolation Lagrange polynomials are generated to evaluate n_{SC} at every node of the 3D mesh

4 Results and model efficiency

In this section, we analyze the performance of the proposed parallel SCH methodology. The first step to validate the effectiveness of this scheme is that it generates a I_D-V_G consistent with the sequential execution. This is shown in Fig. 9, where I_D-V_G curves simulated with a number of processors ranging from 1 to 32 are almost identical.

The small disagreement between the different output currents is due to discrepancies in the quantum density (n_{sc}). Once the quantum density is obtained, it is used to calculate the quantum potential (V_{sc}) which can be seen in Fig. 10. The n_{sc} variable is calculated in each one of the 2D slices, and then its value is obtained in all the 3D nodes of the FE mesh of the device using a high-order Lagrange interpolation method. Depending on the number of slices per process, the interpolation can be linear, quadratic or cubic. For instance, if a process contains only two or three slices, the interpolation polynomials must be first or second order, whereas if a process contains four or more slices, a cubic interpolation can be used. For the GAA NW FET, 40

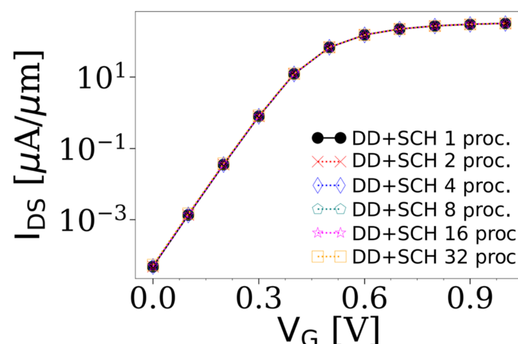

Fig. 9 Simulated I_D-V_G characteristics for the GAA NW FET in comparison with Schrödinger quantum-corrected drift-diffusion (DD-SCH) using 1, 2, 4, 8, 16 and 32 processes at $V_{D,sat} = 0.70$ V

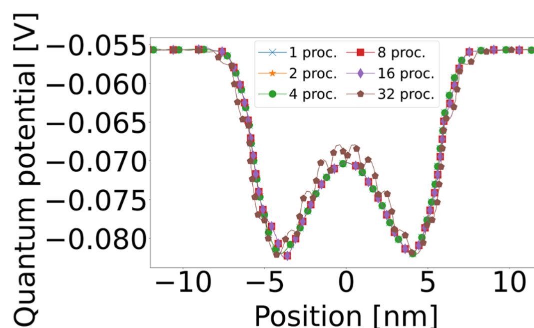

Fig. 10 Interpolated quantum potential profile (Vquan) in the transport direction with 1, 2, 4, 8, 16 and 32 processes. The middle of the gate corresponds to the zero value in the x-axis

x-coordinates, every 1 nm approximately, are defined where the 2D slices are extracted during the simulation. In the case of 32 running processes, usually each processor computes up to 2 slices and therefore the interpolation order can only be linear. Consequently, using more processes to decrease the simulation time induces a loss in accuracy because a lower order of interpolation has to be used.

Figure 11 shows the current percentage error with respect to the sequential execution that has been calculated at $V_{D,sat}$ = 0.70 V versus the number of processors. The disagreement between the sequential and parallel I_D-V_G values reaches a maximum error for the 32 processes case with up to 10% of difference between the extracted current at V_G = 0.0 V. At this gate bias, the potential barrier is at its maximum and the interpolation mechanism is not able to capture the steep slopes of the potential profile. When V_G increases, the potential barrier decreases, smoothing the potential profile and minimizing the interpolation error. Note that for high values of V_G, the percentage error of the current decreases and becomes less than 1% in the range from 0.6 to 1.0 V for all cases.

Interpolated quantum potential profile (V_{quan}) in the transport direction with 1, 2, 4, 8, 16 and 32 processes. The middle of the gate corresponds to the zero value in the x-axis

When using 16 or a lower number of processors, the current error percentage is always lower than 2%; therefore, for the rest of this work only cases from 1 to 16 processes have been simulated. Note that the use of more processes was not considered as the decrease in accuracy in the results due to the interpolation would not be negligible. Moreover, several simulations have been performed increasing the number of 2D slices while keeping the number of processors constant to obtain the optimum number of slices. With a low number of slices, the Schrödinger solution does not provide the correct quantum confinement of the device. When using 40 or more slices, we obtain the same drain current results.

4.1 Simulation times

After checking the integrity and validity of the new parallel SCH routines, the performance has been tested. The simulation results obtained for the S, M and L meshes (40k, 100k and 150k nodes, respectively) can be seen in Table 2. The time measurements correspond to the execution times needed to obtain a current point at V_D = 0.05 V and V_G = 0.0 V. These data have been obtained using a HP ProLiant BL685c G7 @ 3.40GHz with 64 cores and 256 GB of RAM.

The output simulation times show that the code is highly parallel with a time reduction using 16 processes of 97.1%, 97.9% and 98.1% w.r.t the sequential times for the S, M and L meshes. Similar results were found for the measurements of a full I_D-V_G, with times up to 98.2% faster using 16 processes as shown in Fig. 12.

The increase in performance has two main reasons. First, the parallelization of the SCH routines, which are run several hundreds of times during a full I_D-V_G curve, reduced the overall simulation times. On the other hand, the VENDES

Table 2 Execution time measurements of parallel VENDES using 1, 2, 4, 8 and 16 processes and the S, M and L meshes with 40k, 100k and 150k nodes, respectively, needed to obtain a current point at V_D = 0.05 V and V_G = 0.0 V

N proc	t_S	t_M	t_L
1	415	1944	3705
2	184	784	1456
4	78	294	519
8	29	99	180
16	12	42	72

These results have been obtained using a HP ProLiant BL685c G7 @ 3.40GHz of 64 cores and 256 GB of RAM

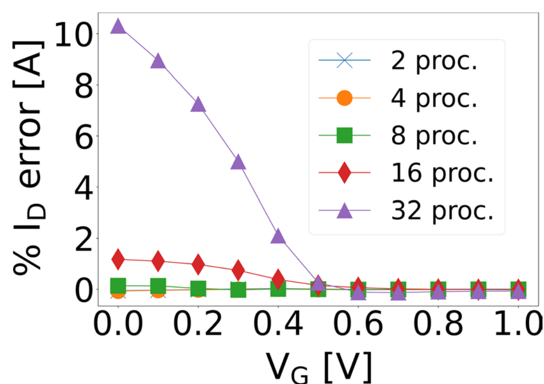

Fig. 11 Current percentage error with respect to the sequential case versus the gate voltage. Note that at low gate voltages for 32 processes a maximum of up to 10 % error is produced whereas, as the gate voltage is increased the percentage error vanishes

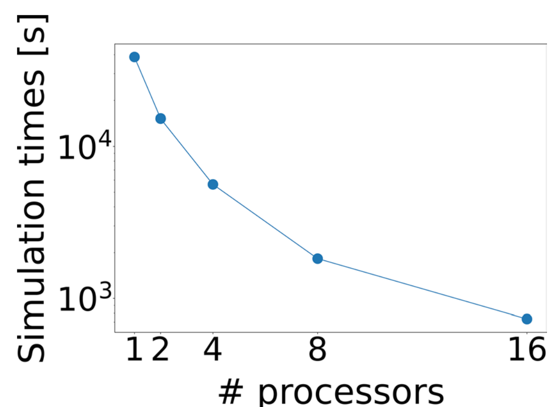

Fig. 12 Simulation times results of a full I_D-V_G curve at V_D = 0.7 V with a 150k node mesh executed with 1, 2, 4, 8 and 16 processes in a parallel system

framework was already parallelized with MPI. Therefore, with the proposed parallelization, the resolution of linear systems that are the most computationally expensive part of main scheme can now be executed in a distributed manner, decreasing simulation times drastically.

4.2 Efficiency of the SCH equation resolution

In order to calculate the efficiency results, the standard definition has been followed:

$$E(p) = \frac{t_1}{t_p \cdot p} \tag{3}$$

where t_1 is the sequential execution time, p is the number of processes and t_p is the execution time using p processes. The parallel efficiency results for one point of the I_D-V_G curve are shown in Fig. 13. These data have been simulated with and without activating the resolution of the SCH and the continuity equation routines so that it can be estimated the influence in the efficiency of the different routines that solve the Poisson, SCH and continuity equations. First, if only the Poisson equation is solved to reach a solution for the electrostatic potential in equilibrium (i.e., V_D, V_G = 0.0 V), the parallel efficiency increases with the number of processors up to approximately 2.8 for the 16 processes case. In this scenario, VENDES reaches super-linear efficiency values (higher than 1). As explained in [41], this behavior appears because when the number of processors increases the size of the local subdomains decreases, reducing the computational time; the linear solver takes more than the increment of the communication time. If instead both the Poisson and continuity equations are solved to calculate the current flow at V_D, V_G = 0.0 V, the efficiency decreases to a maximum of 2.6, which indicates the routines responsible of solving the continuity equation scale worse than the Poisson equation solver. Moreover, if routines that solve the Poisson and the SCH equation are utilized, the parallel efficiency drops to 1.9, meaning that the SCH implementation is not as efficient as the Poisson. Finally, the three functions that solve the Poisson, SCH and continuity equations are executed at the same time, yielding a parallel efficiency of 2.2. The addition of the resolution of the continuity equation increases the efficiency w.r.t the case of only using Poisson and SCH which indicates that the routines used to solve the SCH equation are the less efficient out of the three simulation methods.

We have also found that the results can vary depending on the number of subdomains the device is split into. For example, it has been seen that for four domains, the boundaries between these domains are placed closer to different material interfaces where the physical behavior changes, like gate-source or gate-drain boundaries. This domain-interface coincidence forces additional process synchronizations, increasing the time to achieve the convergency of the linear systems resolution.

Moreover, when more than 8 processes are used, the efficiency begins to saturate. After every iteration of the Poisson and SCH, a synchronization between the results of the boundary nodes in adjacent domains is performed. When using a large number of processes, the overhead of this communication can take a toll on the efficiency.

4.3 Simulation scalability

A similar analysis of the parallel efficiency of the code has been performed changing both the number of running processors and the number of mesh nodes to show how scalable VENDES is. The results are shown in Fig. 14. The efficiency increases with the number of processes, particularly with higher node density meshes. The larger the simulated mesh,

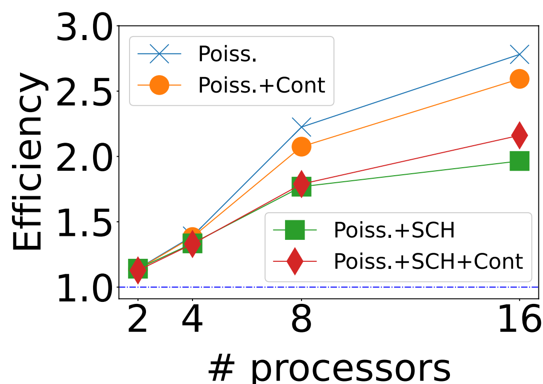

Fig. 13 Efficiency results using a 40k node mesh considering the different types of simulations VENDES has available: only solving the Poisson equation, solving both Poisson and continuity equations, solving Poisson plus SCH equations and solving Poisson, SCH and continuity equations

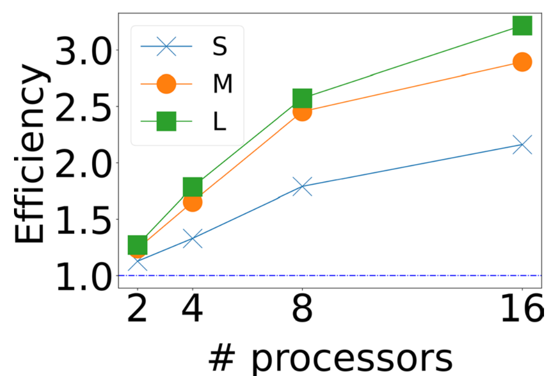

Fig. 14 Efficiency results using three different size meshes (40k, 100k and 150k). These data have been obtained by solving the Poisson, SCH and the continuity equations for a single point of the I_D-V_G curve

the more use it makes out of executing the code in a parallel manner. This is why the maximum efficiency for the 40k mesh is 2.2, whereas for the 100k and 150k the efficiency reaches 2.9 and 3.2, respectively. As explained before, beyond 8 processes, the synchronization between processors after every iteration of the solver produces a significant overhead and starts to decrease the scalability of the code.

Finally, to assess the impact this parallelization has on the performance, we are going to consider a typical variability study, where 300 simulations are performed. Considering that the sequential execution of a full IV curve is close to 11 hours, to obtain the results for 300 curves, it would take more than 137 days of computational time. Now, with the parallel software, since the results for a simulation can be obtained within approximately 12 minutes using 16 processes, the time it would take to simulate the 300 devices drops to 2.5 days using 16 processes for the L mesh.

5 Conclusions

A fully parallel implementation of 3D finite element (FE) Schrödinger (SCH) quantum corrections was developed for physical modeling of ultrascaled semiconductor devices such as GAA NW FETs. The incorporation of the SCH equation-based quantum corrections into the simulation framework increases the overall accuracy of the results, and its execution in distributed systems allows to keep reasonable time frame.

The new routines were tested in a state-of-the-art 12-nm Si GAA NW FET showing an almost perfect agreement w.r.t the sequential simulation in the I_D-V_G curve up to 16 processes. Also, in order to reduce synchronizations stalls, in the current nested multilevel domain decomposition scheme, the boundary nodes have been duplicated and an additional one or two 2D SCH slices have been computed per processor. This extra computation avoids a considerable overhead because after every iteration the SCH solution needs to be shared between adjacent processors to check for consistency. Finally, a comparison of simulation times and code efficiency was made. For three different 40k, 100k and 150k node meshes, S, M and L, respectively, the new parallel code was tested. The execution times dropped up to 98.2% for L meshes using 16 processors with respect to the sequential case. The efficiency is maximum when the simulator is running in a distributed system with 16 processes, reaching a super-linear result of 3.2.

Overall, the new implementation allows for rapid simulation of ultrascaled devices, a much desired characteristic in research studies such as variability analysis, where hundreds or thousands of non-ideal devices need to be simulated. For instance, a typical variability study with 300 device simulations, each one running for approximately for 11 hours, may take over 137 days of computational time. With the proposed parallel software, the total elapsed time drops to 2.5 days using 16 processes for a mesh of 150k nodes.

Acknowledgements This work is supported by the Spanish Government (PID2019-104834GB-100, RYC-2017-23312), by Xunta de Galicia and FEDER (GRC 2014/008, accreditation 2016-2019, ED431G/08, ED431F-2020/008).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. More Moore-Logic Core Device Technology Roadmap. https://irds.ieee.org/editions/2020/more-moore (2013)
2. Selmi, L., Caruso, E., Carapezzi, S., Visciarelli, M., Gnani, E., Zagni, N., Pavan, P., Palestri, P., Esseni, D., Gnudi, A., Reggiani, S., Puglisi, F.M., Verzellesi, G.: Modelling nanoscale n-MOSFETs with III-V compound semiconductor channels: From advanced models for band structures, electrostatics and transport to TCAD. In: 2017 IEEE International Electron Devices Meeting (IEDM), pp. 13-411344 (2017). https://doi.org/10.1109/IEDM.2017.8268384
3. Vasileska, D., Goodnick, S., Klimeck, G.: Computational Electronics: Semiclassical and Quantum Device Modeling and Simulation. CRC Press. pp. 1-764. (2017). https://doi.org/10.1201/b13776
4. Asenov, A., Cheng, B., Wang, X., Brown, A.R., Millar, C., Alexander, C., Amoroso, S.M., Kuang, J.B., Nassif, S.R.: Variability aware simulation based design-technology cooptimization (DTCO) Flow in 14 nm FinFET/SRAM Cooptimization. IEEE Trans. Electron Dev. **62**(6), 1682-1690 (2015). https://doi.org/10.1109/TED.2014.2363117
5. Asenov, A., Wang, Y., Cheng, B., Wang, X., Asenov, P., Al-Ameri, T., Georgiev,: V.P.: Nanowire transistor solutions for 5nm and beyond. In: 2016 17th International Symposium on Quality Electronic Design (ISQED), pp. 269-274. https://doi.org/10.1109/ISQED.2016.7479212 (2016)
6. Khodadadian, A., Taghizadeh, L., Heitzinger, C.: Three-dimensional optimal multi-level Monte-Carlo approximation of the stochastic drift-diffusion-Poisson system in nanoscale devices. J. Comput. Electron. **17**(1), 76-89 (2018). https://doi.org/10.1007/s10825-017-1118-0
7. Aldegunde, M., García-Loureiro, A.J., Kalna, K.: 3D finite element monte carlo simulations of multigate nanoscale transistors. IEEE Trans. Electron Dev. **60**(5), 1561-1567 (2013). https://doi.org/10.1109/TED.2013.2253465

8. Seoane, N., Indalecio, G., Comesana, E., Aldegunde, M., García-Loureiro, A.J., Kalna, K.: Random Dopant, Line-Edge Roughness, and Gate Workfunction Variability in a Nano InGaAs FinFET. IEEE Trans. Electron Dev. **61**(2), 466-472 (2014). https://doi.org/10.1109/TED.2013.2294213
9. Watling, J.R., Brown, A.R., Asenov, A., Svizhenko, A., Anantram, M.P.:Simulation of direct source-to-drain tunnelling using the density gradient formalism: Non-equilibrium greens function calibration. In: International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), pp. 267-270. https://doi.org/10.1109/SISPAD.2002.1034569 (2002)
10. Datta, S.: Nanoscale device modeling: the Green's function method. Superlattices Microstr. **28**(4), 253-278 (2000). https://doi.org/10.1006/spmi.2000.0920
11. Mo, F., Tagawa, Y., Saraya, T., Hiramoto, T., Kobayashi, M.: Scalability study on ferroelectric-hfo2 tunnel junction memory based on non-equilibrium green function method. In: 2019 19th Non-Volatile Memory Technology Symposium (NVMTS), pp. 1-5 (2019). https://doi.org/10.1109/NVMTS47818.2019.8986219
12. Garcia-Loureiro, A.J., Seoane, N., Aldegunde, M., Valin, R., Asenov, A., Martinez, A., Kalna, K.: Implementation of the density gradient quantum corrections for 3-D simulations of multigate nanoscaled transistors. IEEE Trans. Computer-Aided Design Integr. Circuits Syst. **30**(6), 841-851 (2011). https://doi.org/10.1109/TCAD.2011.2107990
13. Lindberg, J., Aldegunde, M., Nagy, D., Dettmer, W.G., Kalna, K., García-Loureiro, A.J., Perić, D.: Quantum corrections based on the 2-D Schrödinger Equation for 3-D finite element monte carlo simulations of Nanoscaled FinFETs. IEEE Trans. Electron Dev. **61**(2), 423-429 (2014). https://doi.org/10.1109/TED.2013.2296209
14. Ancona, M.G., Iafrate, G.J.: Quantum correction to the equation of state of an electron gas in a semiconductor. Phys. Rev. B **39**, 9536-9540 (1989)
15. Kajen, R.S., Chang, K.K.F., Bai, P., Li, E.: Gate leakage analysis of nano-mosfets using ensemble full band monte carlo with quantum correction. In: 2007 International Symposium on Integrated Circuits, pp. 135-138 (2007). https://doi.org/10.1109/ISICIR.2007.4441815
16. Winstead, B., Ravaioli, U.: A coupled Schrödinger/Monte Carlo technique for quantum-corrected device simulation. In: Device Research Conference. Conference Digest (Cat. No.01TH8561), pp. 169-170 (2001). https://doi.org/10.1109/DRC.2001.937918
17. Al-Ameri, T., Georgiev, V.P., Lema, F.A., Sadi, T., Towie, E., Riddet, C., Alexander, C., Asenov, A.: Performance of vertically stacked horizontal Si nanowires transistors: A 3D Monte Carlo/2D Poisson Schrödinger simulation study. In: 2016 IEEE Nanotechnology Materials and Devices Conference (NMDC), pp. 1-2 (2016). https://doi.org/10.1109/NMDC.2016.7777117
18. Dutta, T., Medina-Bailon, C., Carrillo-Nuñez, H., Badami, O., Georgiev, V., Asenov, A.: Schrödinger equation based quantum corrections in drift-diffusion: A multiscale approach. In: 2019 IEEE 14th Nanotechnology Materials and Devices Conference (NMDC), pp. 1-4 (2019). https://doi.org/10.1109/NMDC47361.2019.9084010
19. Paul, A., Bryant, A., Hook, T.B., Yeh, C.C., Kamineni, V., Johnson, J.B., Tripathi, N., Yamashita, T., Tsutsui, G., Basker, V., Standaert, T.E., Faltermeier, J., Haran, B.S., Kanakasabapathy, S., Bu, H., Cho, J., Iacoponi, J., Khare, M.: Comprehensive study of effective current variability and MOSFET parameter correlations in 14nm multi-fin SOI FINFETs. In: 2013 IEEE International Electron Devices Meeting (IEDM), pp. 13-511354 (2013). https://doi.org/10.1109/IEDM.2013.6724625
20. Nayak, K., Agarwal, S., Bajaj, M., Murali, K.V.R.M., Rao, V.R.: Random Dopant fluctuation induced variability in undoped channel si gate all around nanowire n-MOSFET. IEEE Trans. Electron Dev. **62**(2), 685-688 (2015). https://doi.org/10.1109/TED.2014.2383352
21. Wu, Y.S., Su, P.: Sensitivity of Gate-All-Around Nanowire MOSFETs to Process Variations - A Comparison With Multigate MOSFETs. IEEE Trans. Electron Devices **55**(11), 3042-3047 (2008). https://doi.org/10.1109/TED.2008.2008012
22. Seoane, N., Nagy, D., Indalecio, G., Espiñeira, G., Kalna, K., García-Loureiro, A.: A multi-method simulation toolbox to study performance and variability of nanowire fets. Materials (2019). https://doi.org/10.3390/ma12152391
23. Medina-Bailon, C., Padilla, J.L., Sadi, T., Sampedro, C., Godoy, A., Donetti, L., Georgiev, V.P., Gímiz, F., Asenov, A.: Multisubband ensemble monte carlo analysis of tunneling leakage mechanisms in ultrascaled fdsoi, dgsoi, and finfet devices. IEEE Trans. Electron Dev. **66**(3), 1145-1152 (2019). https://doi.org/10.1109/TED.2019.2890985
24. Yoon, J.-S., Rim, T., Kim, J., Meyyappan, M., Baek, C.-K., Jeong, Y.-H.: Vertical gate-all-around junctionless nanowire transistors with asymmetric diameters and underlap lengths. J. Appl. Phys. **105**(10), 102105 (2014). https://doi.org/10.1063/1.4895030
25. Nagy, D., Indalecio, G., García-Loureiro, A.J., Elmessary, M.A., Kalna, K., Seoane, N.: FinFET versus gate-all-around nanowire FET: performance, scaling, and variability. IEEE J. Electron Dev. Soc. **6**, 332-340 (2018). https://doi.org/10.1109/JEDS.2018.2804383
26. Feng, P., Song, S., Nallapati, G., Zhu, J., Bao, J., Moroz, V., Choi, M., Lin, X., Lu, Q., Colombeau, B., Breil, N., Chudzik, M., Chidambaram, C.: Comparative analysis of semiconductor device architectures for 5-nm node and beyond. IEEE Electron Dev. Lett. **38**(12), 1657-1660 (2017). https://doi.org/10.1109/LED.2017.2769058
27. Bufler, F.M., Ritzenthaler, R., Mertens, H., Eneman, G., Mocuta, A., Horiguchi, N.: Performance Comparison of *n*-Type Si Nanowires, Nanosheets, and FinFETs by MC Device Simulation. IEEE Electron Dev. Lett. **39**(11), 1628-1631 (2018). https://doi.org/10.1109/LED.2018.2868379
28. Kushwaha, P., Dasgupta, A., Kao, M.-Y., Agarwal, H., Salahuddin, S., Hu, C.: Design optimization techniques in nanosheet transistor for rf applications. IEEE Trans. Electron Dev. **67**(10), 4515-4520 (2020). https://doi.org/10.1109/TED.2020.3019022
29. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Int. J. Numer. Meth. Eng. **79**(11), 1309-1331 (2009). https://doi.org/10.1002/nme.2579
30. Bangsaruntip, S., Balakrishnan, K., Cheng, S.L., Chang, J., Brink, M., Lauer, I., Bruce, R.L., Engelmann, S.U., Pyzyna, A., Cohen, G.M., Gignac, L.M., Breslin, C.M., Newbury, J.S., Klaus, D.P., Majumdar, A., Sleight, J.W., Guillorn, M.A.: Density scaling with gate-all-around silicon nanowire MOSFETs for the 10 nm node and beyond. In: Proc. IEEE Electron Devices Meeting (IEDM), pp. 526-529 (2013). https://doi.org/10.1109/IEDM.2013.6724667
31. Rhee, S., Kim, D., Kim, K., Choi, S., Park, B., Park, Y.J.: Extension of the dg model to the second-order quantum correction for analysis of the single-charge effect in sub-10-nm mos devices. IEEE J. Electron Dev. Soc. **8**, 213-222 (2020). https://doi.org/10.1109/JEDS.2020.2971426
32. Medina-Bailon, C., Sampedro, C., Padilla, J.L., Godoy, A., Donetti, L., Gamiz, F., Asenov, A.: MS-EMC vs. NEGF: A comparative study accounting for transport quantum corrections. In: 2018 Joint International EUROSOI Workshop and International Conference on Ultimate Integration on Silicon (EUROSOI-ULIS), pp. 1-4 (2018). https://doi.org/10.1109/ULIS.2018.8354758

33. Bank, R., Rose, D., Fichtner, W.: Numerical methods for semiconductor device simulation. IEEE Trans. Electron Dev. **30**(9), 1031-1041 (1983). https://doi.org/10.1137/0904032
34. Lundstrom, M.: Drift-diffusion and computational electronics—still going strong after 40 years! In: 2015 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), pp. 1-3 (2015)
35. Burnett, D.S.: Finite element analysis: From concepts to applications. Addison-Wesley, MA, USA (1987). https://doi.org/10.1002/nme.1620260817
36. Ram-Mohan, L.R.: Finite element and boundary element applications in quantum mechanics. Oxford University Press, London, UK (2002)
37. Hughes, T.J.R.: The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Civil and Mechanical Engineering. Dover Publications (2000)
38. Duff, M., Rabitz, H., Askar, A., Cakmak, A., Ablowitz, M.: A comparison between finite element methods and spectral methods as applied to bound state problems. J. Chem. Phys. **72**(3), 1543-1559 (1980). https://doi.org/10.1063/1.439381
39. A Collection of Fortran Codes for Large Scale Scientific Computation.http://www.hsl.rl.ac.uk. Accessed: 2019-7-21 (2013)
40. Ramayya, E.B., Knezevic, I.: Self-consistent Poisson-Schrödinger-Monte Carlo solver: electron mobility in silicon nanowires. J. Comput. Electron. **9**(3), 206-210 (2010). https://doi.org/10.1007/s10825-010-0341-8
41. Seoane, N., Garcia Loureiro, A., Aldegunde, M.: Optimization of linear systems for 3d parallel simulation of semiconductor devices: application to statistical studies. Int. J. Numer. Model. EL. **22**, 235-258 (2009). https://doi.org/10.1002/jnm.695

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.