npj | computational materials

Article

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

https://doi.org/10.1038/s41524-025-01604-7

# ELEQTRONeX: A GPU-accelerated exascale framework for non-equilibrium quantum transport in nanomaterials

Check for updates

Saurabh S. Sawant [1] ✉, François Léonard[2], Zhi Yao [1] & Andrew Nonaka[1]

Non-equilibrium electronic quantum transport is crucial for existing and envisioned electronic, optoelectronic, and spintronic devices. Encompassing atomistic to mesoscopic length scales in the same nonequilibrium device simulations has been challenging due to the computational cost of high-fidelity coupled multiphysics and multiscale requirements. In this work, we present ELEQTRONeX (**ELE**ctrostatic **Q**uantum **TR**ansport modeling **O**f **N**anomaterials at **eX**ascale), a massively parallel GPU-accelerated framework for self-consistently solving the nonequilibrium Green's function formalism and electrostatics in complex device geometries. By customizing algorithms for GPU multithreading, we achieve significant improvement in computational time, and excellent scaling on up to 512 GPUs and billions of spatial grid cells. We validate our code by computing band structures, current-voltage characteristics, conductance, and drain-induced barrier lowering for various 3D configurations of carbon nanotube field-effect transistors, and demonstrate its suitability for complex device/material geometries where periodic approaches are not feasible, such as arrays of misaligned carbon nanotubes requiring fully 3D simulations.

Non-equilibrium electronic transport determines the properties of many modern electronic devices. As device dimensions are reduced and new nanomaterials introduced, novel quantum phenomena are being explored for improved performance. In parallel, the inherent three-dimensional (3D) nature of nano- and meso-scale devices, including their constituent nanomaterials, requires a device modeling approach that can not only capture quantum phenomena but do so in a complex 3D geometry. In particular, such devices may involve multiple densely packed materials/channels, necessitating modeling the cross-talk caused by neighboring channels to ensure accurate performance evaluation. Thus, there is a need for an efficient approach capable of simulating multiple-channel devices on the order of hundreds of nanometer in length, while incorporating all of the atoms of the constituent materials, and capturing refined electrostatics over three orders of magnitude in size scale.

The Non-Equilibrium Green's Function (NEGF) formalism has emerged as a promising approach for device simulations. It has been applied to the study of devices based on carbon nanotubes[1,2], graphene[3], two-dimensional materials[4], nanowires[5], silicon nanosheets[6], and single molecules[7]. In addition it can describe time-dependent transport[8], as well as response under external stimuli[9,10]. However, despite its versatility, NEGF simulations incur significant computational costs, primarily due to the need for self-consistently solving NEGF equations with electrostatics in complex 3D geometries. While traditionally implemented through parallel deployment on CPUs, these simulations may still require hours to converge for a single device operating point[6]. Furthermore, this computational demand is further amplified when modeling multiple channels, as it necessitates simultaneous self-consistent computations of charge density across all channels. Although numerous parallel NEGF implementations exist for CPUs, their GPU counterparts are notably scarce, with none currently capable of modeling multiple channels.

Sophisticated NEGF implementations include Transiesta[11,12], QuantumATK[13], NEMO5[14,15], the latter one is extended to GPUs using libraries like MAGMA and cuSPARSE[16], while other solvers use message passing interface (MPI) for parallelizations. All of these implementations use independent computational units for parallel distribution such as energy points used for contour integration, discrete wavevectors, and voltage bias points. However, this strategy suffers from scalability and memory limitations since each rank must compute the entire matrices and require storage of the entire non-distributed matrices on the order of tens of gigabytes in the processor memory. These requirements can be alleviated by at most an order of magnitude using OpenMP threads per MPI rank[11,16]. However, for a more substantial improvement and extension to modeling of multiple channels, a low-level optimization approach is needed that allows for

[1]The Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, 94720 CA, USA. [2]Sandia National Laboratories, 7011 East Ave, Livermore, 94551 CA, USA. ✉e-mail: saurabhisreachable@gmail.com

multiple MPI ranks and GPUs to take part in computation of these matrices, say, at each energy point.

Initial attempts at low-level optimization for NEGF were made in ref. 17 with a GPU-based implementation of the recursive Green's function (RGF) algorithm. However, this strategy is limited to decomposing the computations into two MPI ranks, each with its own GPU. Others[18] are leading efforts to port the libNEGF solver to GPUs using OpenACC. Currently, the implementation supports parallelization across energy and wavevector points, while ongoing investigations focus on domain decomposition. These efforts have not yet addressed a parallelization strategy for self-consistently coupling with electrostatics, let alone extension to multiple channels.

We introduce ELEQTRONeX (**ELE**ctrostatic **Q**uantum **TR**ansport modeling **O**f **N**anomaterials at **eX**ascale)[19], an open-source self-consistent NEGF implementation built on the DOE Exascale Computing Project AMReX library[20–22], emphasizing low-level MPI/GPU parallelization strategies across key components: electrostatics, NEGF, and the self-consistency algorithm. The electrostatics module utilizes AMReX's GPU-accelerated multigrid capabilities and can handle intricate shapes of terminal leads represented as embedded boundaries. For NEGF parallelization, large matrices such as Green's and spectral functions are decomposed on distributed across MPI ranks, with GPU acceleration applied to kernel computations. We employ Broyden's modified second method for self-consistency and present an MPI/GPU parallelization approach suitable for modeling multiple channel materials, simultaneously. Overall we demonstrate excellent scaling up to 512 GPUs.

In this work, we demonstrate the capability of ELEQTRONeX by modeling nanodevices with arrays of up to 20 carbon nanotubes of 10 and 100 nm channel lengths with varied average spacing. Our focus is on investigating the effect of angular misalignment and non-uniform spacing. These non-idealities closely reflect experimental device configurations[23,24] and result in non-periodic configurations, requiring fully 3D simulations. By rigorously comparing several key device performance metrics, such as ON and OFF currents and subthreshold swing, against perfectly parallel configurations, we observe for the first time that carbon nanotube field effect transistors (CNTFETs) are robust against these non-idealities. Additionally, we highlight that the modular design of the code allows for straightforward extension to model a wide range of materials beyond nanotubes.

The remainder of this paper is organized as follows: the Methods section outlines our self-consistent NEGF approach and MPI/GPU optimization strategies. In the Results and Discussion, we validate ELEQTRONeX for gate-all-around and planar CNTFET configurations, demonstrate its parallel performance using up to 512 GPUs for 400 nm channel simulations, and examine current-voltage characteristics for nanotubes of hundred-nanometer lengths, while discussing challenges in modeling longer nanotubes. Finally, we study the effect of modeling multiple CNTs, both aligned and non-aligned, in fully 3D planar CNTFET configurations compared to corresponding periodic configurations.

## Methods

First we provide a brief overview of the general system modeled and the computational approach used.

For modeling quantum transport through nanodevices using the NEGF method, the system is decomposed into material channels connected to semi-infinite external leads such as source and drain contacts, as described in Fig. 1a. Here, Hamiltonian $\boldsymbol{H}$ describes the electronic structure of each material channel, self-energies $\boldsymbol{\Sigma}_l^R$ for each lead $l$ represent its coupling to external leads, and $V_{gs}$ and $V_{ds}$ represent the gate-source and drain-source voltages, respectively.

To model the complete system, the ELEQTRONeX framework comprises three major components: the electrostatic module, the quantum transport module, and their self-consistent coupling, as shown in Fig. 1b. The electrostatic module computes the electrostatic potential due to induced charges and device terminals such as source, drain, and gate, modeled as embedded boundaries with intricate shapes. Using this potential, the quantum transport module applies the NEGF method to compute the retarded Green's function $\boldsymbol{G}^R(\boldsymbol{r}, E)$ for each material channel, determining induced charges at $N$ sites (also referred to as system size), which correspond to the number of carbon rings in a nanotube, modeled here using the mode-space approximation (see Supplementary Note 5 and ref. 25). The NEGF-computed induced charges, in turn, affect the electrostatic potential, necessitating iterative solving for self-consistency, achieved using Broyden's modified second algorithm.

In the following sections, we provide a brief overview of the key features of the electrostatics module and its integration with the NEGF module. We also describe the NEGF module and our approach to achieving self-consistency, followed by the parallelization strategies for core computations in the NEGF approach. For detailed algorithmic information, we direct the reader to the supplementary information, while summarize here the algorithmic complexity and the amenability to parallelization, consistent with the parallel performance tests outlined later in the Results and Discussion section. We analyze our algorithmic components in terms of both serial time complexity and the parallel implementation time complexity, assuming that the number of computational resources increases in proportion to system size $N$.

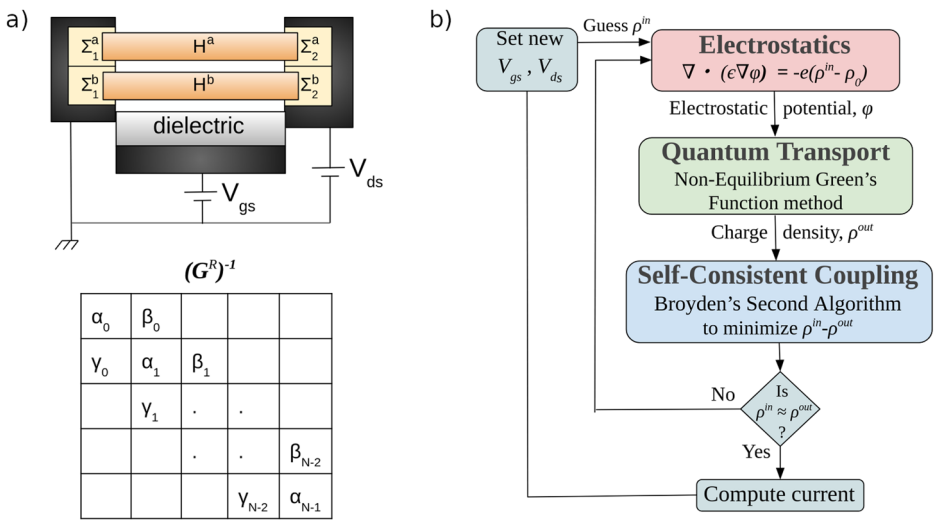


**Fig. 1 | Overview of simulating self-consistent quantum transport through field-effect transistors using the NEGF method. a** System setup for the NEGF method, featuring multiple material channels (denoted by superscripts **a** and **b**) and the structure of inverse Green's function matrix. **b** High-level workflow of the simulation process.

### Electrostastics, Potential Interpolation, and Charge Deposition

The electrostatic module solves Poisson's equation,

$$\nabla \cdot \left[\epsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r})\right] = -e\left[\rho(\boldsymbol{r}) - \rho_0(\boldsymbol{r})\right] \tag{1}$$

where $\epsilon(\boldsymbol{r})$ is the permittivity, $\phi(\boldsymbol{r})$ is the spatially varying electrostatic potential, $e$ is the electronic charge, $\rho(\boldsymbol{r})$ is the charge density, and $\rho_0(\boldsymbol{r})$ is the background charge density. The underlying spatial discretization is a finite-volume mesh with cuboid grid cells that store the electrostatic potential and permittivity.

For solving Eq. (1) robustly, the AMReX library provides core data structures, customized finite volume stencils, and geometric multigrid-based linear solvers[22]. Although we do not yet utilize the adaptive mesh refinement (AMR) feature of AMReX in our implementation, it offers significant advantages, such as support for 3D geometry representation using embedded boundaries (EBs) and Lagrangian particle data structures, robust infrastructure for CPU/GPU parallelization, portability across different hardware architectures, and coupling with external open-source libraries, as will be described shortly.

The multigrid solvers in AMReX use a series of V- or W-cycles, which are iterative procedures involving relaxation techniques, grid transfer operations, and coarse grid computations[20]. Specifically, at each iteration, the algorithm transfers residual information between a hierarchy of grid levels, utilizes relaxation methods to reduce errors, and performs computations at coarser grid levels[26]. This process continues until the geometry can no longer be accurately resolved on the coarsened grid representation. At the coarsest level of the grid hierarchy, the discretized equations are directly solved using a strategy termed as the 'bottom solver'. While the default choice for this task is BiCGSTAB (Bi-Conjugate Gradient STABilized)[27], users have the flexibility to select from a range of methods or leverage external libraries such as hypre (High-Performance Preconditioners) for enhanced robustness[28]. In our current implementation, we typically use two multigrid levels, although this can vary depending on the complexity of the geometry and the resolution required for accurate simulations. For all simulations described in this work involving carbon nanotubes, we employ a computational grid cell that is at least one-fourth the size of the carbon-carbon bond length of 0.142 nm.

To handle complex 3D device configurations, AMReX provides capabilities to specify heterogeneous permittivity distributions, diverse domain boundary conditions, and EBs (also known as cut-cell representations), which represent device terminal surfaces. In the AMReX library, the finite volume stencils are customized for the multigrid algorithm to accurately handle the complex shapes of EBs[22]. The Laplacian stencil near the boundary incorporates an extrapolation that includes the boundary value (which could be either Dirichlet or Neumann) and interpolation of nearby interior points. In our implementation, we have extended the EB capability to specify different boundary conditions on distinct EBs, such as the electrostatic potential on the source, drain, and gate, and provided support for some common configurations of these terminals. By specifying spatially varying voltages on the domain or EBs, the framework enables the computation of complete current-voltage characteristics of a device when coupled with NEGF.

For coupling with the NEGF module, AMReX provides support for Lagrangian particle data structure, which interacts with the underlying spatial mesh through interpolation and deposition kernels. We have customized this capability to represent atoms/sites of materials in real space as static Lagrangian particles, facilitated by electrostatic potential interpolation and charge density deposition schemes based on the cloud-in-cell algorithm[29]. In this approach, the potential at each atomic location is computed by tri-linear interpolation from eight neighboring cell-centered grid points, while charge deposition follows a similar tri-linear scheme to distribute charge density across these grid points. See Supplementary Note 1 for pseudo codes of these algorithms. Additionally, there is an option to average the potential over a subset of atomic locations before it is used in the NEGF calculation, and to distribute the average charge density obtained from the NEGF algorithm across those same atoms. This capability is particularly useful in carbon nanotubes modeled using the mode-space approximation (see Supplementary Note 5 and ref. 25), where we average the potential computed over atoms around a ring before using it in the NEGF method, and spread the charge density obtained from NEGF at each ring equally across all atoms around the ring.

Furthermore, AMReX is responsible for the structured mesh parallel decomposition of the computational domain, which includes handling regions with varying permittivity, EBs, and the Lagrangian particle representation, as well as managing grid/grid and particle/grid communication patterns. For floating-point computations, AMReX supports automatic GPU acceleration by launching kernels on GPU accelerators using C++ lambda functions. This functionality enables efficient multithreading and parallel execution of computationally intensive operations.

The serial time complexity of the linear solvers, as well as the charge deposition and potential interpolation algorithms, is $O(N)$. These algorithms are highly parallelizable, with scaling primarily constrained by inter-processor communication. In an ideal parallel implementation, the parallel time complexity approaches $O(1)$.

### Overview of the Self-Consistent NEGF Approach

In this work, we represent Hamiltonian $\boldsymbol{H}$ of size $(N \times N)$ using a tight-binding model, but the scheme is amenable to any representation. The Hamiltonian has block tri-diagonal form, with the diagonal entries of $\boldsymbol{H}$ corresponding to the electrostatic potential energy $-e\phi$ at each site within the material; in the present case, at each carbon ring.

Central to computing the DC transport properties of the device is the calculation of the retarded Green's function, expressed as

$$\begin{aligned} \boldsymbol{G}^R(\boldsymbol{r}, E) &= \left[(E + i\eta)\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{r}) - \sum_l^L \boldsymbol{\Sigma}_l^R\right]^{-1} \\ \boldsymbol{\Sigma}_l^R(E) &= \boldsymbol{\tau}_l^\dagger \boldsymbol{g}_l^R \boldsymbol{\tau}_l \end{aligned} \tag{2}$$

where $E$ denotes the electron energy, $\eta$ is an infinitesimal constant, $\boldsymbol{I}$ is the identity matrix, $L$ denotes the number of leads in the system, and $\boldsymbol{\tau}$ and $\boldsymbol{g}^R$ are matrices for the device-lead coupling and the surface Green's function, respectively. From Eq. (2) and Fig. 1a, we note that $\alpha_j = (E + i\eta) + H_{(j,j)} - \sum_l^L \Sigma_{l,(j,j)}$ for $j = 0$ to $N-1$, and $\beta_j = H_{(j,\,j+1)}$, $\gamma_j = H_{(j+1,\,j)}$ for $j = 0$ to $N-2$. For a system with 2 semi-infinite leads, only $\Sigma_{1,(0,\,0)}$ and $\Sigma_{2,(N-1,\,N-1)}$ are non-zero.

Using the retarded Green's function, we can compute the charge density matrix $\boldsymbol{\rho}$ as

$$\boldsymbol{\rho}(\boldsymbol{r}) = -\frac{g_s g_b}{2\pi i}\int_{-\infty}^{\infty} \boldsymbol{G}^<(\boldsymbol{r}, E + i\eta)dE \tag{3}$$

where $g_s = 2$ and $g_b$ are spin and band degeneracies, $\boldsymbol{G}^< = i\sum_l^L \boldsymbol{A}_l F_l$ represents the lesser Green's function, $\boldsymbol{A}_l = \boldsymbol{G}^R\boldsymbol{\Gamma}_l\boldsymbol{G}^{R\dagger}$ denotes the lead-specific spectral function matrix, and $\boldsymbol{\Gamma}_l = i(\boldsymbol{\Sigma}_l^R - \boldsymbol{\Sigma}_l^{R\dagger})$ stands for the broadening matrix. $F_l \equiv F_l(E - \mu_b, k_bT_l)$ represents the lead-specific Fermi function, depending on the electrochemical potential $\mu_l$ and temperature $T_b$ where $k_b$ is the Boltzmann constant. Using the above identities and following the approach described in refs. 11,30,31, We calculate $\boldsymbol{\rho}$ in two parts as

$$\boldsymbol{\rho}(\boldsymbol{r}) = \frac{1}{\pi}\Im\left[\int_{E_{lowest}}^{E_{min}} \boldsymbol{G}^R F_{min}\, dE\right] - \frac{1}{2\pi}\int_{E_{min}}^{E_{max}} \sum_l \boldsymbol{A}_l F_l\, dE \tag{4}$$

where $E_{lowest}$ is set below the bottom valence-band edge ($-10$ eV in this work), $F_{min} \equiv F_l(E - \mu_{min}, k_bT_{min})$, $E_{min} = \mu_{min} - f_1k_bT_{min}$, $E_{max} = \mu_{max} + f_2k_bT_{max}$, $(\mu_{min},\ T_{min})$ and $(\mu_{max},\ T_{max})$ are minimum and maximum electrochemical potentials and temperatures, respectively. The factors $f_1$ and $f_2$ are problem-dependent and typically set to 14 to capture the tail of the Fermi levels in the integration. We efficiently calculate the first part of the

integral using contour integration with the residue theorem, and use Gauss-Legendre mapping for accuracy with fewer integration points[11]. See Supplementary Note 3 for details. At equilibrium, i.e. when all leads have equal $\mu$ and $T$, the charge density can be calculated solely using the residue theorem by setting $E_l = E_u = \mu + fk_bT$. Similarly, the neutral charge density matrix $\boldsymbol{\rho}_0(\boldsymbol{r})$ is computed with $E_l = E_u = 0$. The diagonal entries of the density matrix, $\boldsymbol{\rho}_{mm}$ and $\boldsymbol{\rho}_{0,mm}$, represent the charge at each material site, which are spread on the computational grid using the cloud-in-cell algorithm before recomputing the potential using Eq. (1), as the previous section.

Equations (1), (2), and (4) are evaluated until the charge density reaches a self-consistent value. The self-consistent calculation involves finding the zero of the function $\boldsymbol{f} = \boldsymbol{\rho}^{in} - \boldsymbol{\rho}^{out}$, where $\boldsymbol{\rho}^{in}$ is the charge density used to compute the electrostatic potential, and $\boldsymbol{\rho}^{out}$ is the charge density obtained from the NEGF algorithm. To achieve faster convergence, we utilize Broyden's modified second algorithm[32–35], which avoids the need to store the inverse Jacobian matrix, significantly reducing memory usage. The serial time complexity of this algorithm is $O(m_{\text{avg}}N)$, where $m_{\text{avg}}$ represents the average number of iterations required for convergence for a system size of $N$. In Supplementary Note 4, we describe an MPI/GPU parallelization strategy which results in a parallel time complexity of $O(m_{avg})$, where $m_{\text{avg}}$ increases sub-linearly with the system size in our demonstration cases.

Subsequently, essential device properties such as current and transmission can be determined as

$$\begin{aligned} I_l &= \tfrac{g_s e}{h}\int_{E_{min}}^{E_{max}} \mathrm{Tr}\left[\mathrm{F_l}\boldsymbol{\Gamma}_\mathrm{l}\boldsymbol{A} + \mathrm{i}\boldsymbol{\Gamma}_\mathrm{l}\boldsymbol{G}^{<}\right]\mathrm{dE} \\ T_{pq}(E) &= \mathrm{Tr}\left[\boldsymbol{\Gamma}_\mathrm{p}\boldsymbol{G}^{\mathrm{R}}\boldsymbol{\Gamma}_\mathrm{q}\boldsymbol{G}^{\mathrm{R}\dagger}\right] \end{aligned} \tag{5}$$

where $I_l$ is the steady-state current through lead $l$, $h$ is Plank's constant, $\boldsymbol{A} = \sum_l^L \boldsymbol{A}_l$ is the total spectral function, and $T_{pq}$ is the transmission from lead $p$ to $q$. At zero source-drain bias, we may compute conductance as

$$G(E) = G_0 \int_{-\infty}^{\infty} T(E)\left(-\frac{\partial F}{\partial E}\right) dE \tag{6}$$

where $T$ is the transmission at equilibrium and $G_0 = g_s e^2/h$ is the quantum of conductance.

### Flexible Data Structure to Accommodate Different Materials

The composition of the matrices described above depends on the materials under consideration, the device structure, and the chosen representation for the Hamiltonian. Each block or element of these matrices can be a single number, a submatrix, or an array storing entries of a diagonal submatrix. For example, when modeling a CNT under the mode-space approximation[25] with a single subband, each block is represented as a single number. However, when considering multiple subbands, it is represented as an array, with each entry corresponding to a subband, as explained in Supplementary Note 5.

To ensure flexibility in data structure and facilitate future extensions to other materials, we employed various C++ object-oriented programming techniques, including templates, virtual functions, and operator overloading. The general procedure of the NEGF method is implemented as a templated base class, with specialized classes derived from it for specific materials and their unique data structures. The general code in the base class, such as the one used to compute Eq. (2), works regardless of the underlying data structure of the materials-specific matrices. This is achieved by overloading mathematical operators, such as matrix multiplication, for the specific data structures of a material. Furthermore, during the execution of specific steps in the NEGF algorithm, the code searches for material-specific specializations of that step. If none are found, it defaults to the general procedure. For instance, the surface Green's function $g^R$ can be computed using a decimation technique[36,37], but may be overridden, for example when an analytical expression is available (e.g. for CNTs[25]). This approach facilitates easy extension to new materials, as only the aspects specific to the new material need to be coded.

### Efficient Parallelization Strategies for Core NEGF Computations

In the NEGF approach, the two core computations are the matrix inversion to compute the retarded Green's function $\boldsymbol{G}^R$ and the calculation of the spectral function $\boldsymbol{A}$. These computations are performed multiple times for each integration point on all contour paths used to compute density matrix (see Eq. (4)). Typically this means $\mathcal{O}(100 - 10k)$ of these computations per iteration of the self-consistency algorithm, depending on the problem.

Existing parallelization approaches map each energy point to an MPI process, leading to an efficient parallel scenario with no communication overhead for these tasks. However, this approach has limitations. First, it restricts parallelism to the number of energy points, limiting scalability. Second, it imposes substantial memory demands, as each MPI process must store entire matrices for $\boldsymbol{G}^R$ and $\boldsymbol{A}$, alongside additional memory for other quantities[11]. This constraint can significantly restrict the utilization of available MPI processes, even on supercomputers. For instance, on the Perlmutter supercomputer[38], CPU-only nodes feature 64 cores with 512 GB DDR4 DRAM memory, allowing for the storage of only two complex floating-point matrices of size (16k × 16k). For complex 3D configurations, we require a scalable solver capable of managing matrices substantially larger in size-by one or two orders of magnitude-while also allocating sufficient memory for computational grids related to electrostatics on each node. Any future extension of the code to model time-dependent equations will further increase the requirements for the number of matrices stored[8].

To address these challenges and enable efficient computations of $\boldsymbol{G}^R$ and $\boldsymbol{A}$, we adopt a novel hybrid MPI/GPU parallelization strategy, employing a block tri-diagonal matrix inversion algorithm[39,40]. The algorithmic details of parallelization strategy are provided in Supplementary Note 2. In our implementation, each MPI process computes a predetermined number of columns of $\boldsymbol{G}^R$ and $\boldsymbol{A}$ matrices at each energy point, significantly reducing memory requirements per process. Consequently, each MPI process only needs to store Hamiltonian entries and compute charges for the assigned range of columns.

Furthermore, we implement fused computations for improved efficiency. The charge density integrand at a particular energy point involves both the Green's function and the spectral function, with the spectral function itself depending on the Green's function. Instead of computing these quantities separately, we calculate them together within the same GPU kernel, reducing the overhead of invoking multiple kernels and avoiding synchronization delays. This fusion also reduces the memory footprint, as temporary storage of intermediate results between kernels is no longer necessary.

We divide the calculation of the two terms in Eq. (4) based on their respective energy paths (see Supplementary Note 3) and invoke a fused GPU kernel to compute each integrand separately. For example, the integrand of the second term is computed using a GPU kernel that simultaneously calculates the lead-specific spectral function $\boldsymbol{A}_l$ and the Green's function. Once these are computed, the integrand only requires multiplying the diagonal elements of the lead-specific spectral functions with the Fermi function and constants for the Gauss-quadrature procedure, which is done within the same kernel.

Our hybrid MPI/GPU strategy also facilitates a high degree of parallelization for large systems, especially when the number of columns in the Hamiltonian matrix far exceeds the number of energy points. In extending this strategy to GPUs, each MPI process is bound to a GPU device, with GPU threads invoked to handle the assigned columns of $\boldsymbol{G}^R$ and $\boldsymbol{A}$, independently. The GPU-accelerated portion of the code ensures portability across different GPU architectures and vendors, maximizing computational efficiency while accommodating diverse hardware configurations.

The time complexity of the serial block-tridiagonal algorithm is $O(N^2)$, whereas our parallelization strategy achieves a parallel time complexity of $T(N) = O\left(\frac{N}{s_1}\right) + O\left(\frac{N^2}{s_2 P}\right)$ where $s_1 \in [1, 2)$ represents the speedup factor due to the optimization involving overlap of CPU comptuation and CPU-to-GPU asynchronous copy and $s_2$ represents potential speedup due to GPU acceleration. $P$ is the total number of parallel processing units, which, if kept proportional to $N$, allows the algorithm to scale as $O(N)$.

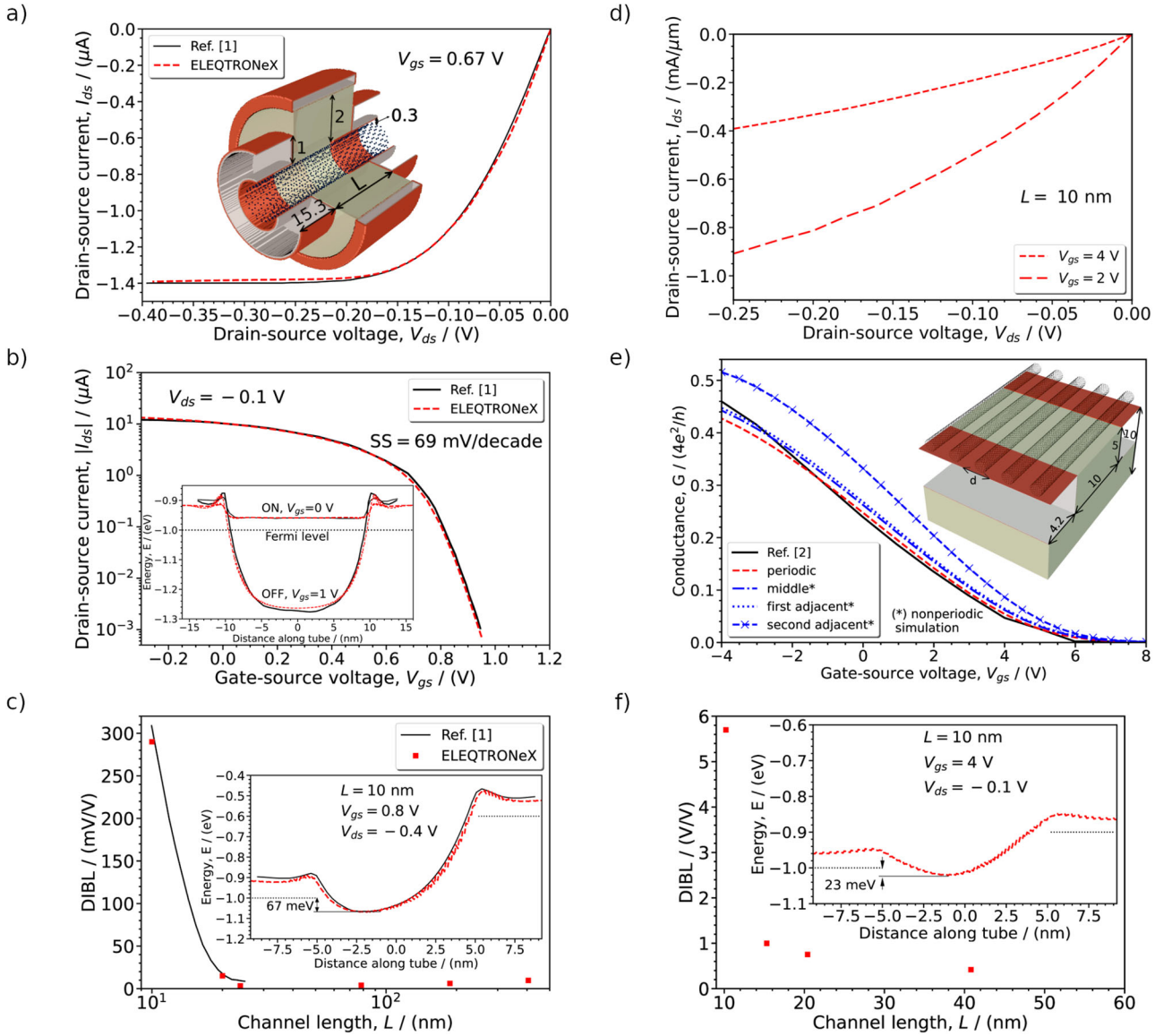


**Fig. 2 | Comparison of CNTFET characteristics with data from refs. 1 and [2], showing gate-all around configuration in the first column (panels a–c) and planar configuration in the second column (panels d–f).** **a**, **d** show $I_{ds}$-$V_{ds}$ characteristics for both configurations. **b**, **e** show $I - V_{gs}$ and $G - V_{gs}$ characteristics for gate-all-around and planar configurations, respectively. In panel **e**, we compare the periodic simulation (with periodicity distance $d$ = 3.2 nm) from ref. 2 with the nonperiodic simulation setup, showing conductance profiles for the `middle', `first adjacent', and `second adjacent' nanotubes are shown. **c**, **f** show drain-induced barrier lowering (DIBL) as a function of $L$ for both configurations. In all panels, insets show valence band edge and schematics with dimensions in nanometers.

## Results and Discussion

### Validation Studies

We validate ELEQTRONeX in obtaining DC transport properties by comparing our results with those reported in ref. 1 for the gate-all-around configuration (first column of Fig. 2) and those reported in ref. 2 for planar CNTFET configuration (second column of Fig. 2), respectively. Schematics of these configurations are shown in the insets of Fig. 2 indicating dimensions in nanometer. Further details specific to modeling CNTFETs can be found in the references. In brief, we consider a (17,0) zigzag CNT of diameter 1.3 nm and nearest-neighbor tight-binding coupling of 2.5 eV, giving a bandgap of 0.55 eV. The Hamiltonian is written in terms of nearest-neighbor coupling between parallel carbon rings. The gate oxide has a dielectric constant of 3.9 to simulate $SiO_2$.

The Fermi level of the source and drain contacts is set 1 eV below the CNT midgap before self-consistency, as would be expected from CNT/palladium (Pd) contacts. Most high-performance CNT devices utilize Pd since it leads to ohmic band alignment[41,42]. The role of the contact metal has been studied computationally in the past[1]; we verified that our current approach agrees with these previous calculations as described in Supplementary Note 6.

In both configurations, the source and drain contacts are modeled as EBs, as described in the Methods, while the gate is modeled as an EB in the gate-all-around configuration and as a domain boundary at the bottom of the gate oxide in the planar configuration. We set $V_{ds}$ and $V_{gs}$ by applying zero potential to the surface of the source metal, $V_{ds}$ to the surface of the drain metal, and $V_{gs}$ to the surface of the gate metal or the domain boundary, depending on how it is modeled. For both configurations the separation between the CNT and the metal is set to 0.3 nm.

For the gate-all-around CNTFET configuration, Fig. 2a, b, and c demonstrate good agreement for the drain-source current $I_{ds}$ versus drain-source voltage $V_{ds}$, $I_{ds}$ versus gate-source voltage $V_{gs}$, and drain-induced barrier lowering (DIBL) versus channel length $L$, respectively. (DIBL refers to the shift in the $I_{ds}$ vs $V_{gs}$ curve as the drain-source voltage is increased. In an ideal transistor DIBL would be zero; larger values of DIBL reflect loss of

**Table 1 | Parameters used and times obtained from the scaling studies**

| *Channel length of nanotube modeled / (nm) | 23.9 | 78.4 | 187.4 | 405.6 |
|---|---|---|---|---|
| Number of carbon rings (system size), $N$ | 512 | 1024 | 2048 | 4096 |
| Number of MPI ranks and GPUs used | 64 | 128 | 256 | 512 |
| Computational cells in the length-wise direction | 1536 | 3072 | 6144 | 12288 |
| Average number of Broyden's iterations for convergence, $m_{avg}$ | 21 | 22 | 28 | 46 |
| Time for electrostatics / (s) | 1.45 | 1.54 | 1.59 | 1.63 |
| Time for NEGF / (s) | 2.04 | 3.12 | 5.28 | 8.76 |
| Time for NEGF per integration point / (ms) | 0.17 | 0.26 | 0.44 | 0.78 |
| Time for interpolation of potential from mesh to atoms / (ms) | 3.0 | 4.0 | 3.4 | 5.8 |
| Time for charge deposition to mesh / (ms) | 0.63 | 0.67 | 0.79 | 1.20 |
| Time for self-consistency / (ms) | 0.50 | 0.59 | 0.74 | 1.00 |

*Contact length was 15.336 nm, computational cells in the transverse direction were 192 × 192.

**Table 2 | Comparison of time per iteration for running the $L$ = 23.9 nm case on a single CPU versus a GPU**

| Modules or routines | CPU | GPU | CPU/GPU ratio |
|---|---|---|---|
| Electrostatics / (s) | 612 | 16.2 | 37.8 |
| NEGF per integration point / (ms) | 0.60 | 0.134 | 4.5 |
| Interpolation of potential from mesh to atoms / (ms) | 9.01 | 0.55 | 16.38 |
| Charge deposition to mesh / (ms) | 48.3 | 1.46 | 33.08 |
| Self-consistency / (ms) | 0.26 | 0.15 | 1.73 |

control of the gate electrode compared to the source and drain electrodes. We compute DIBL quantitatively following the approach in ref. 1 from $\Delta I_{ds}/\Delta V_{ds}$ at 0.1 $\mu$A.)

The simulations agree despite methodological differences compared to ref. 1, including differences in solving Poisson's equation (finite-difference, successive overrelaxation method versus finite-volume, multigrid), charge deposition to mesh (Gaussian spreading versus cloud-in-cell), calculation of surface Green's function (iterative layer doubling technique[36] versus analytical[25]), and the type of Broyden's algorithm employed (first versus parallelized modified second method). Furthermore, in Fig. 2c, DIBL is small for large channel lengths and starts to increase rapidly as the channel approaches the 10 nm oxide thickness, consistent with the data reported by ref. 1. These additional runs (for $L > 20$ nm) are discussed later in the next two sections.

For the planar configuration, Fig. 2d, e, and f show $I - V_{ds}$, $G - V_{gs}$, and DIBL, respectively, although we only have data available for comparing the $G - V_{gs}$ characteristics from ref. 2. The planar periodic configuration is simulated with a single nanotube and imposing periodic boundary conditions in the lateral direction (domain width $d$ = 3.2 nm) to emulate an infinite array of nanotubes.

The $I - V_{ds}$ characteristics show that the device does not reach saturation as the magnitude of $V_{ds}$ increases, highlighting the impact of high DIBL observed for the $L$ = 10 nm case. In this case, the applied source-drain voltage reduces the energy barrier between the source Fermi level and the middle of the channel. As the channel length increases, DIBL plateaus, exhibiting behavior consistent with that observed in the gate-all-around CNTFET configuration.

Next, we validate the code's capability to model multiple channels in a full 3D geometry by simulating an array of nanotubes and comparing it with the planar periodic configuration, as shown in Fig. 2e. First we simulate a periodic configuration, and show that the conductance ($G$) compares well with the results of[2]. Subsequently, we simulate five nanotubes (domain width 5$d$), once again imposing periodic boundary conditions, to ensure consistent behavior of conductance for each nanotube. This validates the capability of our model to accurately simulate multiple nanotubes. Later in the final section, we will discuss the results from modeling only five CNTs to illustrate how finite-size effects can impact device performance.

## Parallel Performance

We conduct performance studies of ELEQTRONeX in parallel using up to 512 MPI ranks with one GPU per MPI rank, focusing on how the code runtime is impacted as computational resources increase in proportion to the number of rings $N$ in a carbon nanotube. ($N$ serves as an appropriate parameterization of system size since the number of grid cells and total atoms are proportional to $N$.)

We first simulate a gate-all-around CNTFET (see inset of Fig. 2a) with a carbon nanotube of 54.53 nm overall length ($L$=23.9 nm) using 64 GPUs, followed by simulations with 2, 4, and 8 times longer nanotubes, employing a proportional increase in the number of GPUs. We analyze changes in 'time per iteration' across three key components of the code-electrostatics, NEGF, and self-consistency. These times and other parameters are summarized in Table 1. For this study, we used approximately 12.5k integration points, chosen to ensure numerically converged solution for the largest channel length of 406 nm, although this required number decreases with channel lengths, as explained in detail in the following section.

These simulations were carried out using GPU resources at the National Energy Research Scientific Computing Center's (NERSC's) Perlmutter supercomputer[38], where each node consists of four NVIDIA A100 GPUs.

Table 1 illustrates that the time for electrostatics remains nearly constant indicating near-perfect parallelization of the electrostatic components. The small 12% increase in the largest case could be attributed to increase in the overall communication time. The time for the NEGF module increases proportionally to $N$/2, consistent with parallel time complexity discussed in Supplementary Note 2, where a factor of 1/2 is achieved due to overlapping a portion of the recursive array computations with simultaneous CPU-to-GPU asynchronous copying. In terms of absolute times, Table 1 reveals that the times per iteration for Broyden's algorithm, interpolating the electrostatic potential from mesh to atomic sites, and depositing charge from atomic locations to mesh, are orders of magnitude smaller compared to that for electrostatics and NEGF. Among these, the self-consistency module with the Broyden's algorithm exhibits time increase roughly proportional to the average number of iterations required for convergence, $m_{avg}$, as expected.

As discussed in the section on parallelization strategies for core NEGF computations, we employed a fused kernel approach in the NEGF module, dividing the calculation of the two integration terms in Eq. (4) based on their respective energy paths. Of these two terms, the first required only 90 integration points in total, whereas the second required over 12k integration points for the longest channel length presented in Table 1, dominating the computational load. As the total integration time is proportional to the number of integration points used, the majority of the reported NEGF time corresponds to the computation of the second term in Eq. (4).

Next, in Table 2, we compare the performance of a single CPU and a single GPU for simulating a channel length of 23.9 nm. In the electrostatics module, the GPU achieves a significant performance improvement, running 37.8 times faster than the CPU. This result is consistent with expectations, given the GPU-accelerated AMReX routines for multigrid linear solvers with EBs[22].

For the NEGF module, the speedup is more modest at 4.5 times for a system size of $N$ = 512. However, we expect that GPU performance will improve as the workload per GPU increases, specifically as the number of columns of spectral and Green's functions computed on each GPU grows. This behavior also explains why the NEGF simulation time for a single GPU is slightly shorter than that for the 64-GPU case, as presented earlier. When the GPU workload is already low, further reducing it causes the parallel

overhead to dominate the total computation time, diminishing further improvement from GPU acceleration. For bulk 3D materials, such as silicon, which have many atoms per unit volume, and as a result larger system sizes, we expect the GPU to perform better.

For the interpolation and deposition kernels, we observe a substantial speedup of over an order of magnitude, while the self-consistency module demonstrates a more limited speedup of 1.7 times. This relatively modest speedup in the self-consistency module is due to the small number of iterations required to achieve self-consistency, causing small size of intermediate matrices, and therefore, low GPU workload, for this particular case.

In the above calculations, we used only one doubly-degenerate transport mode. Next, we demonstrate the effect of increasing the number of modes on computational time using a simpler simulation involving a CNT embedded in an all-around circular lead, same as the CNT-lead portion of the gate-all-around configuration shown in Fig. 2a. We vary the potential on the lead metal from 0 to 1 V, while at each condition the Broyden iterations are carried out to achieve self-consistency. Since this is an equilibrium Green's function calculation, we only evaluate the first integral in Eq. (4), requiring few integration points without encountering convergence issues that are present in nonequilibrium cases, as described the following section. Therefore, for this study we simulate a very long CNT (55.84 $\mu$m), and vary the number of modes from 1 to 8 (all doubly-degenerate). For modes greater than one, each block of the block-tridiagonal matrix is represented in the form of an array with each array element corresponding to a mode. Table 3 shows that with increase in number of modes from 1 to 8, the time for NEGF increases by only 2.84 times. The sub-linear increase in time may be caused by two major reasons that benefit calculations on both CPUs and GPUs: better cache and memory bandwidth utilization, since the data for each mode is stored contiguously in an array, and better core-pipeline latency hiding due to increase in the load per thread.

**Table 3 | Parameters and times for the CNT-lead equilibrium calculations with multiple transport modes**

| | | | | |
|---|---|---|---|---|
| Number of carbon rings (system size), $N$ | 524,288 | | | |
| Computational cells | 72 × 1, 572, 864 × 72 | | | |
| Contour integration points | 90 | | | |
| Number of modes | 1 | 2 | 4 | 8 |
| Time for NEGF / (s) | 1.98 | 2.75 | 3.24 | 5.63 |
| Time for NEGF per integration point / (s) | 0.022 | 0.031 | 0.036 | 0.063 |
| Time for electrostatics / (s) | 2.2 - 2.4 | | | |

### Long-Channel Gate-All-Around CNTFET Configurations

Here, we discuss the results obtained from the nonequilibrium simulations run for the parallel performance cases with gate-all-around configurations, discussed in Table 1, as well as the $L = 10nm$ case used for code validation in Fig. 2c, as well as some considerations in choosing integration paths and points to obtain converged self-consistent solution for longer nanotubes.

Figure 3a, b depict the current-voltage characteristics, while Fig. 3c illustrates the band bending at $V_{gs}$ = 0.8V and $V_{ds} = -0.1$V. Initially, we compute the $I - V_{ds}$ characteristics at $V_{gs}$ = 0.8V when the transistor is nearly in the OFF state. Next, utilizing the charge density profile at $V_{ds} = -0.1$V, we begin simulations to compute the $I - V_{gs}$ characteristics, sweeping $V_{gs}$ from 0.8V to −0.3V while using the converged solution at the previous $V_{gs}$ to initialize the computation at the new $V_{gs}$.

In the subthreshold region, the $I - V_{ds}$ characteristics exhibit p-type FET behavior with current saturation at -0.1 V–except for the $L$ = 10.2 nm case. This lack of saturation is attributed to significant DIBL in short channels, as shown in Fig. 2c and first described in ref. 1. In short channels, the applied source-drain voltage reduces the energy barrier between the source Fermi level and the middle of the channel, preventing saturation. For longer channels, DIBL is significantly reduced, allowing current saturation to occur. However, DIBL still influences the device by increasing the drain-source saturation current as the channel length increases. Notably, the impact of DIBL on the drain-source current persists even when the channel length increases from 187.6 nm to 405.6 nm (greatly exceeding the gate oxide thickness of 10 nm). This channel-length dependence is also evident in the subthreshold region of the $I - V_{gs}$ characteristics. While the ON-state current remains comparable to that of the shortest channel length, the

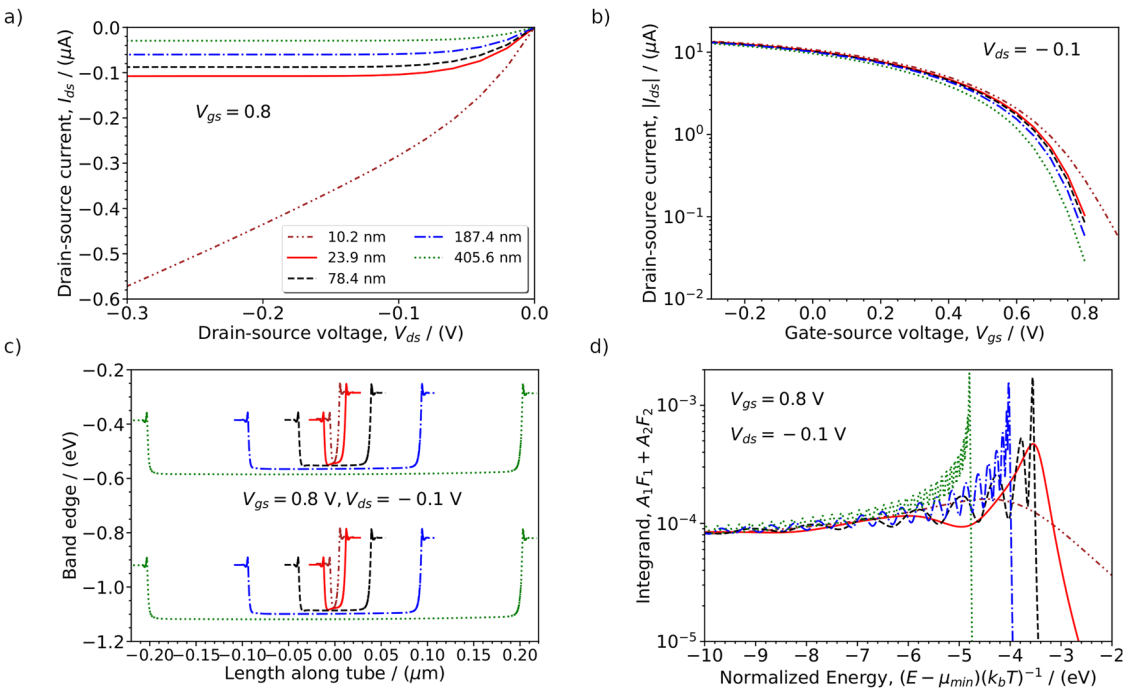


**Fig. 3 | Nonequilibrium simulation results obtained from cases run for parallel performance studies presented in Table 1. a**, **b** Current-voltage characteristics for four channel lengths analyzed for parallel performance studies. **c** Comparison of band bending across the four cases. **d** Integrand at the channel center for nonequilibrium charge density calculation. Legends for all figures are shown in **a**, representing channel lengths.

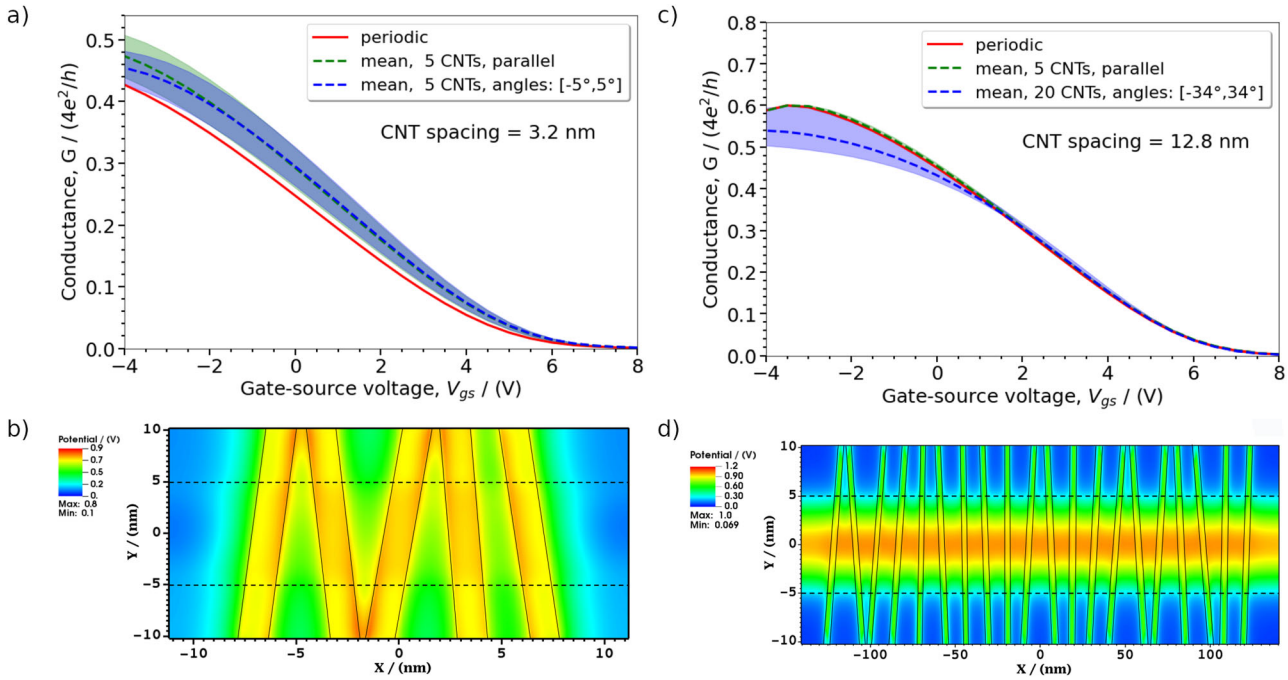


**Fig. 4 | Comparison of zero-bias conductance *G* versus gate-source voltage $V_{gs}$ for periodic, non-periodic and parallel, and non-parallel configurations with different CNT spacing *d*, and channel length *L*.** **a** $d = 3.2$ nm, $L = 10$ nm, **b** $d = 12.8$ nm, $L = 10$ nm. For nonperiodic simulations, we show the mean conductance with the colored spread around it marking the maximum and minimum conductance for that configuration. **c** and **d** show electrostatic potential in the non-parallel configurations at $V_{gs} = -2$ V on a Z-plane passing through CNTs. The black solid overlaid lines mark CNT edges, while the dashed horizontal black lines mark the boundary of the channel region.

subthreshold region shows performance degradation due to poor gate control over the channel electrostatics and tunneling across the hole barrier in the OFF regime, consistent with the trends described in ref. 1. A comparison of band structures further illustrates that longer channels exhibit more effective band bending, whereas the shortest channel shows a lower barrier for the same applied biases.

Turning to the question of integration points, we note that the band structure of an infinitely long carbon nanotube exhibits van hove singularities at the band edges. For a finite CNT, as the length gets longer, the singularity gets sharper, requiring more integration points for accurate calculation of the second part of the integral to compute charge density, shown in Eq. (4). To illustrate, the integrand $\boldsymbol{A}_1 F_1 + \boldsymbol{A}_2 F_2$ at the center of the channel is shown in Fig. 3d as a function of energy, *E*. We observe that for $L = 10.2$ nm channel, the integrand shows a slight peak without any oscillations, however, as the channel length increases, not only does the singularity get sharper, but it also exhibits oscillating structures on the lower energy side.

Therefore, to reduce the number of integration points, we use a semi-adaptive method and breakdown the integration path from $E_{min}$ to $E_{max}$ into three subparts, $E_{min}$ to $E_{l1}$, $E_{l1}$ to $E_{l2}$, and $E_{l2}$ to $E_{max}$. In each of these subregions we can specify the density of integration points, defined by the number of integration points per $kT$. Before starting the simulation, we do not know the exact location of the singularity, and therefore, we start the simulation with a guess for $E_{l1}$ and $E_{l2}$, compute the integrand at the center of the channel as a function of energy, obtain the peak of the singularity $E_s$, and refine limits as $E_{l1} = E_{l1} - akT$, $E_{l2} = E_s + bkT$, where constants $a$ and $b$ can be specified by the user, typically set to 3 and 1, respectively. The number of integration points change according to the density of integration points set by the user in each of these subregions. Note that during the self-consistency iterations the location and peak of singularity may change, and therefore, we update these limits at the first iteration for a given set of conditions and update it at a set period of 50 iterations. For the evaluation of integration in each subregion, we use Gauss-Legendre quadrature to further improve the accuracy. For instance, to simulate the largest channel length we used densities of integration points of (500, 1600, 50), respectively, which lead to approximately 12,500 integration points in total. For the performance analyses described above, we kept this total number of integration points constant for consistent scaling of the workload across various cases. However, in reality, to obtain a numerically accurate solution at 23.9, 78.4, and 187.4 channel lengths we only need approximately 500, 1200, and 6000 total integration points, respectively. In fact, the $L = 23.9$ nm can be simulated with just one integration path from $E_{min}$ to $E_{max}$. We suspect that a fully adaptive scheme might be necessary for simulating micron-scale CNTs self-consistently.

### Three-Dimensional Non-parallel Planar CNTFET Configurations

In the section on validation studies, Fig. 2d showed the conductance from a planar CNTFET configuration with periodic boundary conditions in the lateral direction, effectively emulating an infinite array of CNTs spaced apart by 3.2 nm. Although this setup is instructive for theoretical analysis, experimental devices typically contain a finite number of CNTs in the channel, necessitating 3D non-periodic simulations to reflect more realistic conditions. To showcase the capabilities of our approach, we simulated five nanotubes in the channel by applying Neumann boundary conditions instead of periodic ones. An additional spacing of *d* was introduced between the lateral boundaries and the outermost nanotubes.

The simulation results, compared in Fig. 2d, show that the conductance profiles of the middle three nanotubes (labeled as 'middle' and 'first adjacent') closely resemble those obtained with periodic boundary conditions. However, the conductance of the outermost nanotubes, labeled as 'second adjacent,' is significantly higher, because each outermost nanotube has only one neighboring nanotube influencing its potential. These results are shown again in Fig. 4a, where we present the conductance obtained from the non-periodic configuration as a shaded region around the mean conductance to provide a clearer visual representation of variability. The finite CNT case carries more current at a given gate voltage, 11.5% more current in the ON state (2.9 mS/$\mu$m vs 2.6 mS/$\mu$m), and a reduced substhrehsold swing (from 2986 mV/dec to 2011 mV/dec) due to the different gate capacitive coupling.

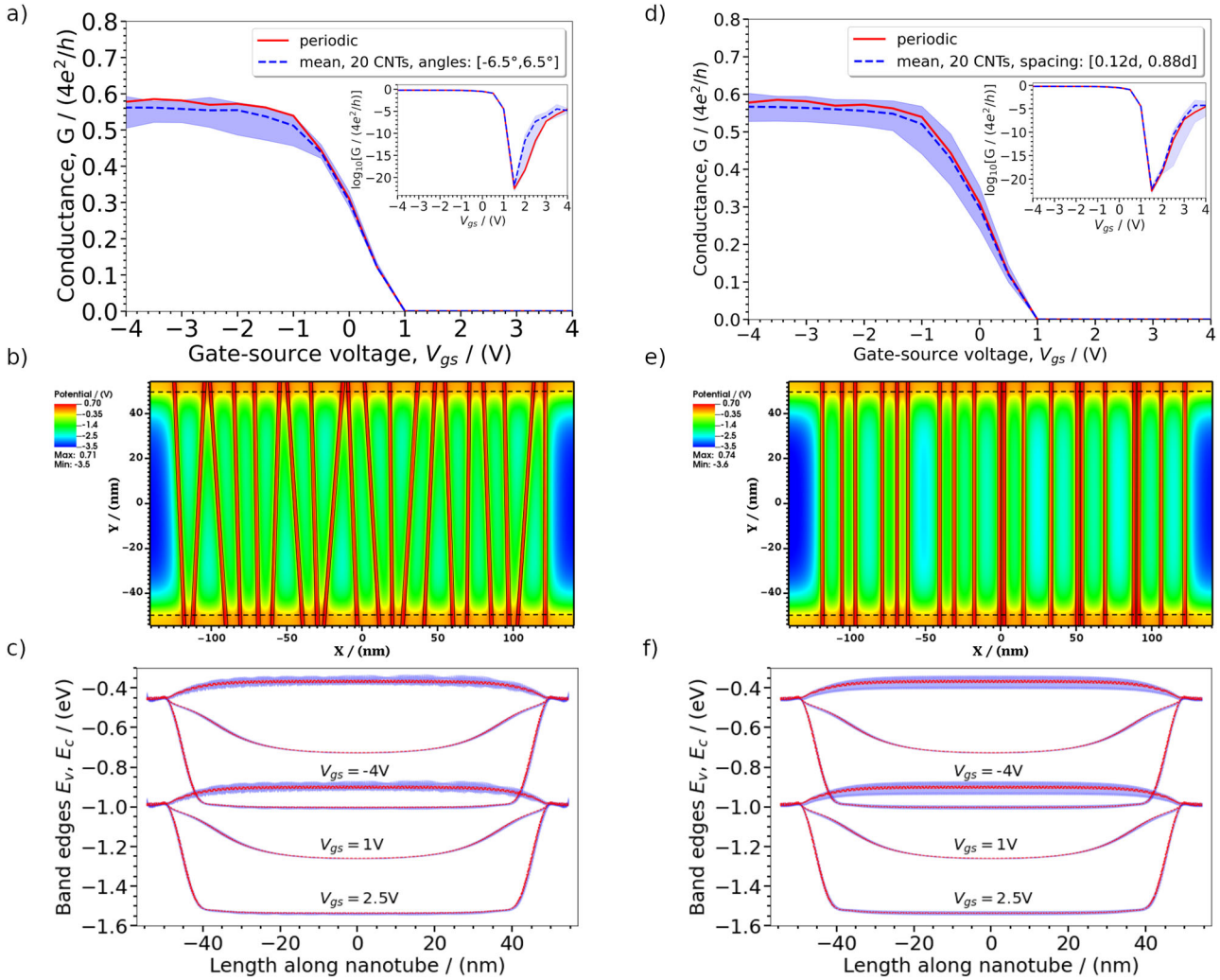


**Fig. 5 | Planar configurations of 20 CNTs with an average spacing of $d = 12.8$ nm and a channel length of $L = 100$ nm, illustrating the effects of variations in non-parallel orientation angles (panels a–c) and non-uniform spacing (panels d–f) on key characteristics compared to a periodic arrangement. a**, **d** show a comparison of $G$ versus $V_{gs}$, with an inset displaying the same plot on a $y$-log scale; **b**, **e** present the electrostatic potential at $V_{gs} = -4$ V on a Z-plane passing through CNTs, with black solid lines marking CNT edges and dashed horizontal lines marking the boundary of the channel region; and **c**, **f** compare band structures.

Next, in the same Fig. 4a, we also compare the impact of CNT misalignment, a common occurrence in CNT-array devices as noted in various studies[23,24]. In this configuration, 5 CNTs are rotated around their midpoint by angles sampled randomly in the range [− 5°, 5°], such that CNTs do not touch or overlap for this 10 nm channel case, as seen from the contours of electrostatic potential in Fig. 4b. Such configurations lack periodicity, requiring fully 3D simulations.

It is seen that the conductance for the non-parallel configuration only differs slightly from the parallel configuration when the FET is in the ON state (2.8 mS/$\mu$m vs 2.9 mS/$\mu$m). This small difference is attributed to the small range of possible angles due to the short channel and close proximity between CNTs. As a consequence, there is only a small portion of nanotubes that deviate from the mean spacing, resulting in minimal impact on the conductance. Also, the regions where the CNTs are closest are located under the metal contacts where charges are screened by the metal. Finally, since the CNT spacing is already smaller than the gate oxide thickness, the channel operates in the cross-talk regime, meaning CNT-CNT interactions are not fully screened by the gate. Thus, angular variations in CNT alignment do not significantly alter this behavior. In addition, the CNT misalignment has little impact on subthreshold swing (2010 mV/dec vs 2011 mV/dec) or threshold voltage compared to the parallel 5 CNT case, while the variation in the variation of the subthreshold swing from each CNT is only about 70 mV/decs.

Next, in Fig. 4c, we study the effect of modeling a finite number of parallel and non-parallel nanotubes when nanotubes are spaced farther apart ($d = 12.8$ nm) such that $d$ is larger than the gate-oxide thickness of 10 nm. In this case, contrary to the simulation with the smaller CNT spacing, the finite array of 5 parallel CNTs does not show much deviation from the corresponding infinite array because the CNT-CNT cross-talk is screened by the gate. As a result, the potential on each nanotube remains largely unaffected by its neighbors, leading to similar conductance for both outer and inner nanotubes in the parallel arrangement. (ON stage conductance 3.6 mS/$\mu$m in both cases, subthreshold swing 1778 mV/dec vs 1785 mV/dec.) However, for the case with misalignment with angles in the range [− 34°, 34°], we observe a decrease of the average ON state conductance by about 8.3% to 3.4 mS/$\mu$m This decrease is caused by cross-talk between the CNTs when the spacing between them decreases for part of the channel length, as seen from the contours of electrostatic potential in Fig. 4d. Indeed, some angled CNTs can be as close as 5 nm within the channel, bringing them into a regime with noticeable cross-talk, despite the average spacing exceeding the oxide thickness. This increased cross-talk effectively reduces the gate capacitance, making it harder to switch the channel[2], ultimately leading to a decrease in conductance.

To understand how the amount of variation in conductance changes with channel length, we increased the channel length to 100 nm and

**Table 4 | Impact of non-idealities on planar CNTFETs***

| L (nm) | d (nm) | Geometry | $G_{ON}$ ($mS/\mu m$) | SS (mV/dec) | $G_{OFF}$ ($\mu S/\mu m$) |
|---|---|---|---|---|---|
| 10 | 3.2 | parallel, infinite | 2.6 | 2986 | 6.7 |
| 10 | 3.2 | parallel, 5 CNTs | 2.9 ± 0.21 | 2011 ± 66 | 6.9 ± 0.4 |
| 10 | 3.2 | misaligned, 5 CNTs | 2.8 ± 0.17 | 2010 ± 70 | 6.8 ± 0.4 |
| 10 | 12.8 | parallel, infinite | 3.6 | 1785 | 1.8 |
| 10 | 12.8 | parallel, 5 CNTs | 3.6 ± 3.2e-3 | 1778 ± 11 | 1.8 ± 3.5e-3 |
| 10 | 12.8 | misaligned, 20 CNTs | 3.3 ± 0.2 | 1668 ± 56 | 1.4 ± 0.34 |
| 100 | 12.8 | parallel, infinite | 3.5 | 76.7 | 3.9e-4 |
| 100 | 12.8 | misaligned, 20 CNTs | 3.4 ± 0.11 | 76.7 ± 2.0 | 3.7e-4 ± 4.6e-5 |
| 100 | 12.8 | varying pitch, 20 CNTs | 3.4 ± 0.15 | 77.2 ± 1.0 | 4.2e-4 ± 4.4e-5 |

*Oxide thickness was kept constant to 10 nm. For cases with multiple nanotubes, one standard deviation variation is shown.

modeled 20 CNTs with an average spacing of $d = 12.8$ nm and angular rotation within the range $[-6.5^{\circ}, 6.5^{\circ}]$. Figure 5a, b, and c show conductance, electrostatic potential, and band structure for this simulation. From the conductance plot, we again note that the misalignment results in only about 3% lower conductance with respect to a periodic arrangement when the channel is ON (3.4 mS/$\mu$m vs 3.5 mS/$\mu$m), and has no impact on the subthreshold swing (76.7 mV/dec in both cases).

The insert shows the conductance on a y-log scale, revealing a minimum in the conductance and an increase in the current for higher gate-source voltages. This increase is caused by band-to-band tunneling as a result of the conduction band-edge approaching the Fermi level at -1 eV. We further note that the band-structure obtained from nonparallel simulation differs from a periodic simulation only when the channel is ON; when there is more charge in the channel, we see more cross-talk resulting in slightly lower conductance.

The above results suggest that CNT array devices have good resiliency to misalignment over a broad parameter range. This can be understood primarily based on screening arguments: CNT-CNT interactions due to misalignment occur primarily near the source and drain electrodes, but are effectively screened by the contact metal and the gate.

We also explored the effect of non-uniform spacing between nanotubes while keeping them parallel. The average spacing is kept constant at $d = 12.8$ nm, and the spacing between each nanotube is varied in the range [0.12d, 0.88d]. Figure 5d, e, and f show conductance, electrostatic potential, and band structure for this case. This simulation showed a broader variation in the conductance of individual CNTs compared to a periodic simulation, starting from the linearly increasing region of conductance ($V_{gs} = 1V$ to $-1V$). Yet, the total conductance of the device is close to that of the periodic system; the maximum variation seen is still only about 3% when the channel is ON. The band-structure shows a maximum variation with respect to periodic configuration, when the channel is ON. Variations in the subthreshold swing or the threshold voltage are also minimal in this case.

Another factor of importance for digital electronics is the CNTFET OFF current, which has been shown to be dominated by phonon-assisted band-to-band tunneling[43]. While we plan to add electron-phonon scattering in the future, here we provide a simple estimate based on the direct tunneling current. In CNTs, the phonon-assisted tunneling current is dominated by scattering with optical phonons of energy 0.2 eV[44]. Thus, we found the gate-source voltage at which the conduction band edge is 0.2 eV above the lead Fermi level and estimated the OFF state conductance from the conductance calculated at this value without electron-phonon scattering. The results can be found in Table 4. While the values are approximate, they show that the OFF state current is not strongly affected by the finite size of the array and misorientation.

In terms of computational cost, the simulations with 20 CNTs employed 10560, 768, and 768 cells in the X, Y, and Z directions, respectively, with a cell size of 0.026625 nm in each direction. The computation utilized 512 MPI ranks each associated with one GPU, averaging approximately 3.6 s per iteration for electrostatics and 0.65 seconds per iteration with 90 integration points for NEGF, while the time required for other steps in the iteration was an order of magnitude lower. The code required 42 iterations on average, i.e. approximately 3 minutes per gate-source voltage condition.

In summary, we presented ELEQTRONeX, a robust and portable implementation of the NEGF method coupled with electrostatics, leveraging modern supercomputing architectures with the AMReX library and MPI/GPU parallelization strategies. We demonstrated computational efficiency of ELEQTRONeX, particularly in modeling larger systems, such as long device channels and larger computational domains for electrostatics. We successfully modeled fully 3D non-periodic CNTFET configurations and investigate the accuracy of periodic calculations compared to realistic experimental setups, finding CNTFETs to be robust against misalignment and varying pitch for several key device performance metrics. Our future plans include adding electron-phonon interactions, trap charges, and other capabilities to address state-of-the-art problems, making ELEQTRONeX a versatile tool for simulating complex systems.

## Data availability

The input files used to generate the datasets in this work are available in the code repository at: https://github.com/AMReX-Microelectronics/ELEQTRONeX/tree/development/input/negf. All simulations in this work were performed using AMReX[20] (git hash ≤ ae3af4339) and ELEQTRONeX[19] (git hash ≤ 8483b65).

## Code availability

The code used in this work, ELEQTRONeX, is open-source and available in the GitHub repository maintained by the ELEQTRONeX team: https://github.com/AMReX-Microelectronics/ELEQTRONeX. Documentation for ELEQTRONeX can be found at https://amrex-microelectronics.github.io/. Use of this software is under a modified BSD license - the license agreement is included in the ELEQTRONeX home directory as license.txt.

Received: 25 July 2024; Accepted: 16 March 2025;
Published online: 24 April 2025

## References

1. Léonard, F. & Stewart, D. A. Properties of short channel ballistic carbon nanotube transistors with ohmic contacts. *Nanotechnology* **17**, 4699 (2006).
2. Léonard, F. Crosstalk between nanotube devices: contact and channel effects. *Nanotechnology* **17**, 2381 (2006).
3. Banadaki, Y. M. & Srivastava, A. Investigation of the width-dependent static characteristics of graphene nanoribbon field effect transistors using non-parabolic quantum-based model. *Solid-State Electron.* **111**, 80–90 (2015).
4. Ganapathi, K., Yoon, Y. & Salahuddin, S. Monolayer MoS2 transistors - ballistic performance limit analysis. In *69th Device Research Conference*, 79–80 (2011).
5. Nag Chowdhury, B. & Chattopadhyay, S. Dual-gate GaAs-nanowire FET for room temperature charge-qubit operation: A NEGF approach. *Adv. Quantum Technol.* **6**, 2200072 (2023).
6. Park, H.-H. et al. NEGF simulations of stacked silicon nanosheet FETs for performance optimization. In *2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, 1–3 (2019).

7. Cohen, G. & Galperin, M. Green's function methods for single molecule junctions. *J. Chem. Phys.* **152**, 090901 (2020).
8. Kienle, D. & Léonard, F. Terahertz response of carbon nanotube transistors. *Phys. Rev. Lett.* **103**, 026601 (2009).
9. Léonard, F., Spataru, C. D., Goldflam, M., Peters, D. W. & Beechem, T. E. Dynamic wavelength-tunable photodetector using subwavelength graphene field-effect transistors. *Sci. Rep.* **7**, 45873 (2017).
10. Stewart, D. A. & Léonard, F. Photocurrents in nanotube junctions. *Phys. Rev. Lett.* **93**, 107401 (2004).
11. Papior, N., Lorente, N., Frederiksen, T., García, A. & Brandbyge, M. Improvements on non-equilibrium and transport green function techniques: The next-generation TRANSIESTA. *Comput. Phys. Commun.* **212**, 8–24 (2017).
12. Papior, N., Gunst, T., Stradi, D. & Brandbyge, M. Manipulating the voltage drop in graphene nanojunctions using a gate potential. *Phys. Chem. Chem. Phys.* **18**, 1025–1031 (2016).
13. Smidstrup, S. et al. QuantumATK: an integrated platform of electronic and atomic-scale modelling tools. *J. Phys.: Condens. Matter* **32**, 015901 (2019).
14. Klimeck, G. & Luisier, M. Atomistic modeling of realistically extended semiconductor devices with NEMO and OMEN. *Comput. Sci. Eng.* **12**, 28–35 (2010).
15. Steiger, S., Povolotskyi, M., Park, H.-H., Kubis, T. & Klimeck, G. Nemo5: A parallel multiscale nanoelectronics modeling tool. *IEEE Trans. Nanotechnol.* **10**, 1464–1474 (2011).
16. Sahasrabudhe, H., Fonseca, J. & Klimeck, G. Accelerating nano-scale transistor innovation with NEMO5 on Blue Waters. Tech. Rep., University of Illinois at Urbana-Champaign https://bluewaters.ncsa.illinois.edu/liferay-content/document-library/NEIS-P2-C1-Reports/Klimeck_NEISP2_Final_Report.pdf (2014).
17. Jeong, Y. & Ryu, H. High performance simulations of quantum transport using manycore computing. In *The International Conference on High Performance Computing in Asia-Pacific Region Companion*, HPCAsia'21 Companion, 21–28 (Association for Computing Machinery, New York, NY, USA, 2021).
18. Pecchia, A. et al. libNEGF: A library for Non-Equilibrium Green's Function (NEGF) simulations. https://github.com/libnegf/libnegf (Accessed: April 2024).
19. ELEQTRONeX team. ELEQTRONeX. Lawrence Berkeley National Laboratory. GitHub link: https://github.com/AMReX-Microelectronics/ELEQTRONeX. Documentation available at https://amrex-microelectronics.github.io/ (2024).
20. AMReX team. AMReX Lawrence Berkeley National Laboratory. GitHub link: https://github.com/AMReX-Codes/amrex. Documentation release 24.03-dev available at https://amrex-codes.github.io/amrex/docs_html/ (2024).
21. Zhang, W. et al. AMReX: a framework for block-structured adaptive mesh refinement. *J. Open Source Softw.* **4**, 1370–1370 (2019).
22. Zhang, W., Myers, A., Gott, K., Almgren, A. & Bell, J. AMReX: Block-structured adaptive mesh refinement for multiphysics applications. *Int. J. High. Perform. Comput. Appl.* **35**, 508–526 (2021).
23. Jinkins, K. R. et al. Aligned 2D carbon nanotube liquid crystals for wafer-scale electronics. *Sci. Adv.* **7**, eabh0640 (2021).
24. Liu, L. et al. Aligned, high-density semiconducting carbon nanotube arrays for high-performance electronics. *Science* **368**, 850–856 (2020).
25. Guo, J., Datta, S., Lundstrom, M. & Anantam, M. P. Toward multiscale modeling of carbon nanotube transistors. *Int. J. Multiscale Comput. Engineer* **2**, 257–276 (2004).
26. Almgren, A. S., Bell, J. B., Colella, P., Howell, L. H. & Welcome, M. L. A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *J. Comput. Phys.* **142**, 1–46 (1998).
27. van der Vorst, H. A. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **13**, 631–644 (1992).
28. Falgout, R. D. & Yang, U. M. hypre: A library of high performance preconditioners. In Sloot, P. M. A., Hoekstra, A. G., Tan, C. J. K. & Dongarra, J. J. (eds.) *Computational Science — ICCS 2002*, 632–641 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2002).
29. Birdsall, C. K. & Fuss, D. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. *J. Comput. Phys.* **3**, 494–511 (1969).
30. Taylor, J., Guo, H. & Wang, J. *Ab* initio modeling of quantum transport properties of molecular electronic devices. *Phys. Rev. B* **63**, 245407 (2001).
31. Brandbyge, M., Mozos, J.-L., Ordejón, P., Taylor, J. & Stokbro, K. Density-functional method for nonequilibrium electron transport. *Phys. Rev. B* **65**, 165401 (2002).
32. Srivastava, G. P. Broyden's method for self-consistent field convergence acceleration. *J. Phys. A: Math. Gen.* **17**, L317 (1984).
33. Singh, D., Krakauer, H. & Wang, C. Accelerating the convergence of self-consistent linearized augmented-plane-wave calculations. *Phys. Rev. B* **34**, 8391 (1986).
34. Ihnatsenka, S., Zozoulenko, I. & Willander, M. Electron-electron interaction effects in transport through open dots: Pinning of resonant levels. *Phys. Rev. B* **75**, 235307 (2007).
35. Areshkin, D. A. & Nikolić, B. K. Electron density and transport in top-gated graphene nanoribbon devices: First-principles Green function algorithms for systems containing a large number of atoms. *Phys. Rev. B* **81**, 155450 (2010).
36. Sancho, M. P. L., Sancho, J. M. L., Sancho, J. M. L. & Rubio, J. Highly convergent schemes for the calculation of bulk and surface green functions. *J. Phys. F: Met. Phys.* **15**, 851 (1985).
37. Wang, J.-S., Zeng, N., Wang, J. & Gan, C. K. Nonequilibrium Green's function method for thermal transport in junctions. *Phys. Rev. E-Stat., Nonlinear, Soft Matter Phys.* **75**, 061128 (2007).
38. *System details*, https://docs.nersc.gov/systems/perlmutter/system_details/ (2021).
39. Godfrin, E. M. A method to compute the inverse of an n-block tridiagonal quasi-Hermitian matrix. *J. Phys.: Condens. Matter* **3**, 7843 (1991).
40. Hod, O., Peralta, J. E. & Scuseria, G. E. First-principles electronic transport calculations in finite elongated systems: A divide and conquer approach. *J. Chem. Phys.* **125**, 114704 (2006).
41. Javey, A., Guo, J., Wang, Q., Lundstrom, M. & Dai, H. Ballistic carbon nanotube field-effect transistors. *Nature* **424**, 654–657 (2003).
42. Brady, G. J. et al. Quasi-ballistic carbon nanotube array transistors with current density exceeding si and gaas. *Sci. Adv.* **2**, e1601240 (2016).
43. Lin, Q. et al. Band-to-band tunneling leakage current characterization and projection in carbon nanotube transistors. *ACS Nano* **17**, 21083–21092 (2023).
44. Koswatta, S. O., Lundstrom, M. S., Anantram, M. P. & Nikonov, D. E. Simulation of phonon-assisted band-to-band tunneling in carbon nanotube field-effect transistors. *Appl. Phys. Lett.* **87**, 253107 (2005).


## Acknowledgements

Work by S. S. Sawant, A. Nonaka, and Z. Yao was supported by the U.S. Department of Energy, Office of Science, under the Microelectronics Co-Design Research Program (Co-Design and Integration of Nano-sensors on CMOS), the Microelectronics Science Research Centers (Nanoscale hybrids: a new paradigm for energy-efficient optoelectronics) and Accelerate Innovations in Emerging Technologies Program (Phonon Control for Next-Generation Superconducting Systems and Sensors) under Contract DE-AC02-05-CH11231. Work by F. Léonard was supported by the same programs under Contract DE-NA-0003525. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This research used resources of the National Energy Research Scientific

Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award ASCR-ERCAP0026882.

## Author contributions

Conceptualization, methodology, investigation, validation: S.S. Sawant, Léonard. Data curation, formal analysis, visualization, writing–original draft: S.S. Sawant. Software: S.S. Sawant, A. Nonaka. Project administration, resources, supervision, funding acquisition, Writing–review & editing: F. Léonard, A. Nonaka, Z. Yao. These author contributions are defined according to the CRediT contributor roles taxonomy.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41524-025-01604-7.

**Correspondence** and requests for materials should be addressed to Saurabh S. Sawant.

**Reprints and permissions information** is available at http://www.nature.com/reprints

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

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, 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 you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. 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-nc-nd/4.0/.

© The Author(s) 2025