AD-NEGF: An End-to-End Differentiable Quantum Transport Simulator for Sensitivity Analysis and Inverse Problems

A Preprint

Yinzhanghao Zhou
College of Intelligence and Computing
Tianjin University, Tianjin, China
zhouyinzhanghao@gmail.com

Xiang Chen
Noah's Ark Lab
Huawei, Beijing, China
xiangchen.ai@outlook.com

Peng Zhang *
College of Intelligence and Computing
Tianjin University, Tianjin, China
pzhang@tju.edu.cn

Jun Wang *
University College London
London, United Kingdom
jun.wang@cs.ucl.ac.uk

Lei Wang
Institute of Physics
Chinese Academy of Sciences, Beijing, China
wanglei@iphy.ac.cn

Hong Guo
Department of Physics, McGill University
Montréal, Quebec, Canada
hong.guo@mcgill.ca

February 11, 2022

Abstract

Since proposed in the 70s, the Non-Equilibrium Green Function (NEGF) method has been recognized as a standard approach to quantum transport simulations. Although it achieves superiority in simulation accuracy, the tremendous computational cost makes it unbearable for high-throughput simulation tasks such as sensitivity analysis, inverse design, etc. In this work, we propose AD-NEGF, to our best knowledge the first end-to-end differentiable NEGF model for quantum transport simulations. We implement the entire numerical process in PyTorch, and design customized backward pass with implicit layer techniques, which provides gradient information at an affordable cost while guaranteeing the correctness of the forward simulation. The proposed model is validated with applications in calculating differential physical quantities, empirical parameter fitting, and doping optimization, which demonstrates its capacity to accelerate the material design process by conducting gradient-based parameter optimization.

Keywords Quantum Transport, Non-Equilibrium Green Function, Automatic Differentiation, Differentiable Programming, Deep Learning, Sensitivity Analysis, Inverse Design

1 Introduction

Electronic transport models are used to simulate electrical properties of devices, which are essential for circuit simulation, semiconductor device fabrication, and so on (Jacoboni [2010], Wimmer [2009], Pourfath [2014]). Traditionally, the transport process is formulated with the drift-diffusion model (Markowich et al. [2012]), where electrons and holes in the device are treated as flows. However, with the improvement of semiconductor manufacturing, the quantum effect is not negligible anymore (Anantram et al. [2008], Wang et al. [2008], Datta [1997]). Moreover, the classic

[*] Corresponding Authors: Peng Zhang and Jun Wang.

macroscopic model relies on empirical models and parameters, hence not capable of handling the emergence of new materials and new structures. These problems can be solved with first-principle quantum transport simulation, which formulates microscale electronic devices at the atomic level. An important technique for quantum transport simulation is the Non-Equilibrium Green Function (NEGF) method (Jacoboni [2010]), which solves the open system Schrödinger equation. More specifically speaking, the Green function of the system is calculated considering the effect of the contacting electrodes and the fixed bias voltage. It is then iterated with the Poisson equation which describes the electrostatic potential field self-consistently. The NEGF method not only acts as a standard computational tool of transport problems for researchers in academia, but is also widely adopted in the semiconductor industry, which vastly accelerates the device design process.

Despite its advantages in accuracy, the first-principle simulation is extremely time and computation consuming. One way to solve the dilemma is to build up learning-based surrogate models (Li et al. [2020], Bürkle et al. [2021], Pimachev and Neogi [2021]). By learning from data generated with first-principle simulations beforehand, the surrogate model is expected to maintain first-principle accuracy but achieves a much higher speed in usage. A fatal problem of such methods is that, there is no guarantee for the model accuracy. This is a fundamental limit for machine learning, and is especially non-negligible for high-dimensional and input-sensitive scenarios such as quantum transport simulations.

An alternative is to keep the forward numerical computation unchanged, but to make the whole process differentiable by utilizing automatic differentiation techniques. In this way, gradient information can be obtained at a similar cost to the forward pass, but faster and more accurate than the numerical differentiation. A direct application is to compute differential physical properties, which is ubiquitous in scientific computation. Specifically, in quantum transport, there are examples such as the differential conductance and resistance, as well as the Seebeck coefficient which describes the sensitive thermoelectric power, etc. Moreover, the availability of gradients makes it possible to conduct efficient gradient-based optimization, which can outperform black-box optimization methods. Recent advances have also shown the value to apply differentiable programming in scientific computation scenarios, such as fluid dynamics (Schenck and Fox [2018]), quantum chemistry (Kasim and Vinko [2021]), molecular dynamics (Schoenholz and Cubuk [2020]), etc.

In this work, we make the NEGF process differentiable with automatic differentiation techniques. The entire numerical process of the quantum transport simulation is implemented in PyTorch, including the computation of the self-energy, the Green function, the electrostatic potential, the transport properties, as well as a Slater-Koster Tight-Binding (SKTB) module to generate the block tri-diagonal Tight-Binding (TB) Hamiltonian from atomic coordinates and TB parameters (Klymenko et al. [2021]). The backward pass is improved by utilizing the implicit gradient techniques, the adjoint sensitivity method for PDE, and our proposed image charge gradient method. We demonstrate its capability to efficiently and accurately compute differential physical properties by comparing with numerical differentiation. Also, it is shown that by cooperating AD-NEGF with the gradient-based optimizer, it can perform high-dimensional optimization at a scale that is not affordable with conventional optimization approaches. Furthermore, in a more practical scenario of material doping optimization where we optimize the empirical SK parameters of injected atoms, our method shows significant advances in convergence speed and optimization results, compared with traditional black-box optimization methods.

Our contributions can be summarized as follows:

- We propose and implement AD-NEGF, as far as we know the first end-to-end differentiable quantum transport simulator, including the NEGF method, the Poisson equation module for self-consistent electrostatic potential computation, and the SKTB module to generate the tight-binding Hamiltonian from the coordinates and properties of the system atoms.
- The efficiency of the backward gradient computation is improved by applying the implicit gradient method, the adjoint method for PDEs, as well as our proposed gradient computation for the image charge method.
- We validate the advantages of AD-NEGF in calculating differential transport quantities, high-dimensional parameter fitting, and device optimization, where AD-NEGF outperforms numerical differentiation and black-box optimization methods.

2 Related Works

NEGF. Originated from Keldysh [1964], Kadanoff [2018], NEGF has been a well-received method in the quantum transport theory, which describes a system with a finite bias voltage and contact interactions in consideration. Recently, NEGF based computation methods gain increasing popularity for the simplicity of the formulation, and the easy implementation in programming (Ferry and Goodnick [1999], Taylor et al. [2001], Brandbyge et al. [2002], Fetter and Walecka [2012]), which makes NEGF one of the most widely applied methods in transport calculation. Several methods dedicated to improving its numerical stability and computational efficiency are proposed (Sancho et al. [1985],

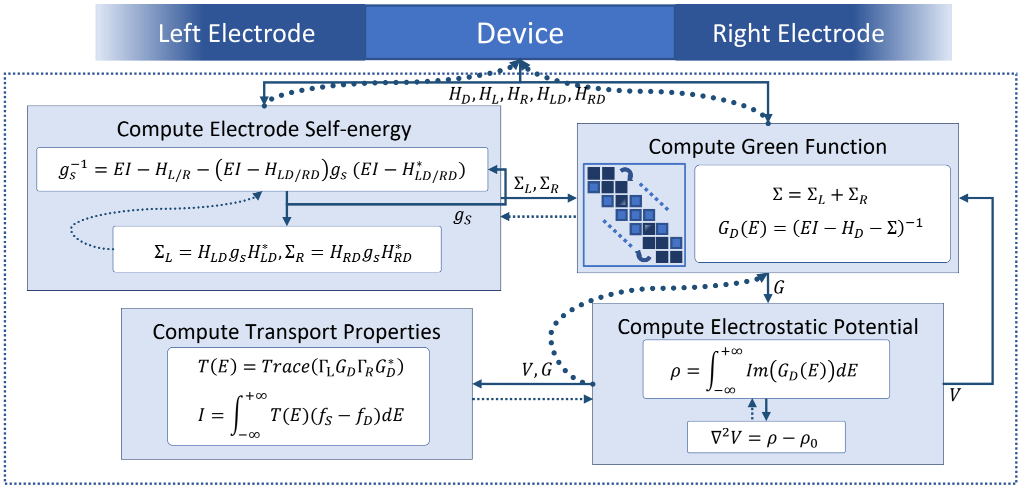

Left Electrode
Device
Right Electrode
$H_D, H_L, H_R, H_{LD}, H_{RD}$
Compute Electrode Self-energy
$g_s^{-1} = EI - H_{L/R} - (EI - H_{LD/RD})g_s\,(EI - H^*_{LD/RD})$
$\Sigma_L = H_{LD} g_s H^*_{LD}, \Sigma_R = H_{RD} g_s H^*_{RD}$
Σ_L, Σ_R
g_S
Compute Green Function
$\Sigma = \Sigma_L + \Sigma_R$
$G_D(E) = (EI - H_D - \Sigma)^{-1}$
G
Compute Electrostatic Potential
$\rho = \int_{-\infty}^{+\infty} Im(G_D(E))dE$
$\nabla^2 V = \rho - \rho_0$
V
V, G
Compute Transport Properties
$T(E) = Trace(\Gamma_L G_D \Gamma_R G_D^*)$
$I = \int_{-\infty}^{+\infty} T(E)(f_S - f_D)dE$

Figure 1: Workflow of the differentiable NEGF. The solid lines indicate the forward simulation pass. Loops in the forward pass are where self-consistent iterations apply. The dotted lines indicate the backward gradient computation pass.

Krstić et al. [2002], Rungger and Sanvito [2008]), some of which are widely implemented in modern quantum transport simulation software, including but not limited to Papior et al. [2017], Smidstrup et al. [2019], Steiger et al. [2011].

AI for Quantum Transport. There have been works to apply machine learning techniques in quantum transport, mostly by training a neural network with data generated from first-principle simulations, so that the neural network can serve as an efficient surrogate model to predict transport properties, such as conductance (Bürkle et al. [2021], Pimachev and Neogi [2021], Li et al. [2020]), transport coefficients (Lopez-Bezanilla and von Lilienfeld [2014]), etc. Most existing methods use relatively simple deep learning models such as multi-layer perceptrons (Župančić et al.) and convolutional networks (Han et al. [2021], Souma and Ogawa [2021, 2020]), while in some cases more advanced and specially designed models are utilized (Bürkle et al. [2021]).

Differentiable Programming. Deep learning has been applied to more and more diverse scenarios, which require the network structure to be more and more flexible. This generalization sometimes is referred to as differentiable programming. It requires the automatic differentiation framework to support more numerical operations, such as fixed-point iterations (Bai et al. [2019]), optimization (Amos and Kolter [2017]), ordinary differential equations (Chen et al. [2018]), etc. Differentiable programming has been widely applied to physical simulations (Hu et al. [2019], Innes et al. [2019]), such as rigid body dynamics (de Avila Belbute-Peres et al. [2018], Freeman et al. [2021]), computational fluid dynamics (Kochkov et al. [2021], Schenck and Fox [2018]), ray tracing (Li et al. [2018]), etc. More specifically in ab-initio simulations, there have been works for density functional theory (Li et al. [2021], Kasim and Vinko [2021]), Hartree–Fock (Tamayo-Mendoza et al. [2018]), coupled cluster methods (Pavošević and Hammes-Schiffer [2020]), and molecular dynamics (Schoenholz and Cubuk [2020]).

3 Methodology

3.1 Non-Equilibrium Green Function Method

In this section, we will first introduce the computation procedure of the NEGF method in brief, and then explain each module in details. Consider a transport system containing a device region and two semi-infinite contacts that attach to the left and right sides of the device, as shown in Figure 1. The contacts can also be referred to as leads or electrodes interchangeably. According to the theory of quantum mechanics, the whole system, including the device and the contacts, can be fully described by its Hamiltonian H. In this paper, we consider the Tight-Binding (TB) model (Slater and Koster [1954]), which makes H block tri-diagonal. We assume a set of basis has been selected, and hence the full process of NEGF can be expressed in the matrix form. The stationary Schrödinger equation of this open system is:

$$H\Psi = E\Psi, \tag{1}$$

where Ψ stands for the wave function of electrons, and E is a scalar value corresponding to the system energy. The characteristics of the system are contained in its Green function

$$G = [EI - H]^{-1}, \tag{2}$$

where I is the identity matrix. However, the Hamiltonian H is infinitely large hence intractable. This is resolved by computing the Green function only for the device part, while considering the effect of two semi-infinite contacts in a term Σ called self-energy. The device Green function G_D will be used to describe the non-equilibrium charge transfer process by solving a Poisson equation in a self-consistent iteration. The output self-consistent potential field V and device Green function G_D can be used to compute transport properties such as transmission, current, etc. Each part is further explained in the following.

Compute the Green Function. Although Equation (2) is mathematically correct, the Hamiltonian H describing an infinite system is also infinitely large, making it impractical to compute. Since it is only the properties of the center device that we are interested in, we can just describe the device Green function, and integrate all the other effects such as the electrodes, the lattice vibration, and the charge redistribution, in a term Σ, which is referred to as self-energy:

$$G_D = [EI - H_D - \Sigma]^{-1}. \tag{3}$$

Here G_D and H_D state for the Green function and the Hamiltonian matrix of the device region. Directly computing the matrix inversion is with complexity $O(N^3)$, which is unbearable as the matrix size is proportional to the vast amount of atoms. By utilizing the block tri-diagonal form of the Hamiltonian matrix, an efficient recursive algorithm (Anantram et al. [2008]) is implemented, which scales linearly with the system size.

Compute Electrode Self-Energy. One of the most critical parts to analyze the non-equilibrium state of a device is to describe the effect of electrodes. Since the system is made up of a device and two semi-infinite contacts on the side, Equation (2) can be expanded in the following form:

$$\begin{bmatrix} A_L & A_{LD} & 0 \\ A_{DL} & A_D & A_{DR} \\ 0 & A_{RD} & A_R \end{bmatrix} \begin{bmatrix} G_L & G_{LD} & G_{LR} \\ G_{DL} & G_D & G_{DR} \\ G_{RL} & G_{RD} & G_R \end{bmatrix} = I, \tag{4}$$

where $A = [EI - H]$, and the subscripts are used to distinguish the matrix elements corresponding to the left lead (L), the device (D), the right lead (R), and their interactions. Thanks to its block tri-diagonal form, the device Green function G_D satisfies

$$[A_D - A_{DL}A_L^{-1}A_{LD} - A_{DR}A_R^{-1}A_{RD}]G_D = I. \tag{5}$$

Since $A_D = [EI - H_D]$, compared with Equation (3), we have

$$\Sigma^L = A_{DL}A_L^{-1}A_{LD}, \tag{6}$$

$$\Sigma^R = A_{DR}A_R^{-1}A_{RD}, \tag{7}$$

$$\Sigma = \Sigma^L + \Sigma^R. \tag{8}$$

Here we assume only the neighbouring layers have interactions with each other, and denote the left lead layer connected to the device by l. Then the left self-energy can be simplified as $\Sigma^L = A_{Dl}A_l^{-1}A_{lD}$. The coupling matrix A_{lD} is given as input of NEGF. What remains unclear is A_l^{-1}, the bottom-right block of A_L^{-1}. This is known as the surface green function, denoted as g_s. By utilizing the ideal lead assumption that removing one layer of the lead will not change g_s, we obtain a self-consistent form:

$$g_s^{-1} = [A_l - A_{l,l-1}g_sA_{l-1,l}^{\dagger}], \tag{9}$$

where $A_{l,l-1}$ is its coupling with the neighbouring layer. This equation can be used to calculate g_s self-consistently. To speed up the process, we implement the Lopez-Sancho algorithm (Sancho et al. [1985]), as illustrated in Algorithm 1, which converges exponentially faster than the conventional self-consistent iteration. We also implement a modern method based on the generalized eigenvalue problem (Wang et al. [2008]) as an alternative.

Self-Consistent Iteration for Electrostatic Potential. In NEGF, charge transfer due to the applied bias voltage is modeled as an external potential, which is attained self-consistently by solving the Poisson equation for electrostatics with non-equilibrium charges. Denote the charge densities in the equilibrium and non-equilibrium states as ρ_0 and ρ, and the potential fields from the original neutral and redistributed charges as V_0 and V. The equilibrium and non-equilibrium

Algorithm 1 Lopez-Sancho algorithm for surface Green function

```
set $\epsilon_0^s = h_{0,0}, \epsilon_0 = h_{0,0}, \alpha_0 = h_{0,1} - ES_{0,1}, \beta_0 = h_{1,0} - ES_{1,0}$
repeat
    $\epsilon_i^s = \epsilon_{i-1}^s + \alpha_{i-1}(ES - \epsilon_{i-1})^{-1}\beta_{i-1},$
    $\epsilon_i = \epsilon_{i-1} + \beta_{i-1}(ES - \epsilon_{i-1})^{-1}\alpha_{i-1} + \alpha_{i-1}(ES - \epsilon_{i-1})^{-1}\beta_{i-1}$
    $\alpha_i = \alpha_{i-1}(ES - \epsilon_{i-1})^{-1}\alpha_{i-1}$
    $\beta_i = \beta_{i-1}(ES - \epsilon_{i-1})^{-1}\beta_{i-1}$
until converge
$g_{0,0} = (ES - \epsilon_m^s)^{-1}$
```

Hamiltonian can be expressed as $H_0 = T + V_0$, $H_{neq} = T + V$, where T is the kinetic energy. Poisson equations relate potentials to the corresponding charge densities:

$$\begin{cases} \nabla \cdot \epsilon(r)\nabla V(r) = -\rho(r), \\ \nabla \cdot \epsilon(r)\nabla V_0(r) = -\rho_0(r). \end{cases} \tag{10}$$

Therefore we have $\nabla \cdot \epsilon(r)\nabla[\Delta V(r)] = -[\rho(r) - \rho_0(r)]$, where $\Delta V = V - V_0$ is used to correct the Hamiltonian by $H_{neq} = H_0 + \Delta V$. The updated H_{neq} will again be used to update ΔV. Hence a self-consistent iteration is constructed:

$$\begin{cases} \nabla \cdot \epsilon(r)\nabla[\Delta V(r)] & = -[\rho(r; \Delta V) - \rho_0(r)], \\ \Delta V(r)|_{\{z_L, z_R\}} & = \{V_L, V_R\}. \end{cases} \tag{11}$$

Charge densities are necessary input for the above equation. Denote potentials in left and right electrodes as u_l and u_r (assume $u_l < u_r$), then the charge density $\rho(r) = -\frac{i}{2\pi}\int_{-\infty}^{+\infty} dEG(E)$, which can be decomposed into equilibrium and non-equilibrium terms:

$$\rho(r) = \rho_{eq}(r) + \rho_{neq}(r) \tag{12}$$

$$= \frac{1}{\pi} Im \left[\int_{-\infty}^{u_l} dEG_D(E)\right] + \frac{1}{2\pi}\int_{u_l}^{u_r} dEG_D(E). \tag{13}$$

The first integration up to infinity can be computed efficiently using contour integration with the residue theorem. It is achieved by expanding the Fermi-Dirac function, more details about which can be referred to in Ozaki [2007], Areshkin and Nikolić [2010]. On the other hand, the non-equilibrium charge density ρ_{neq} is computed directly by numerical integration. The density of neutral charges ρ_0 can be computed by setting $u_l = u_r = 0$.

In implementation, the Poisson equation can be solved using numerical PDE solvers with spherical charges. Meanwhile, a computationally more efficient image charge method using Fast Multipole Method (FMM) is preferred (Svizhenko and Anantram [2005], Zahn [1976]). After the procedure converges to a stable solution, transport properties can be computed accordingly.

Transport Electronic Properties. With the NEGF theory, electronic transport properties can be derived, such as transmission probability ($T(E)$), density of states (DOS), electronic current (I), equilibrium and non-equilibrium electronic densities (ρ_{eq} and ρ_{neq}), etc. Here we list some of the expressions.

$$T(E) = Trace[\Gamma_L(E)G_D(E)\Gamma_R(E)G_D^\dagger(E)], \tag{14}$$

$$DOS(E) = -\frac{1}{\pi}Trace[Im(G_D(E))], \tag{15}$$

$$I = \frac{2e}{\hbar}\int_{-\infty}^{+\infty}\frac{dE}{2\pi}T(E)[f(E - u_l) - f(E - u_r)], \tag{16}$$

$$\rho(r) = \frac{1}{\pi}Im\left[\int_{-\infty}^{u_l} dEG_D(E)\right] + \frac{1}{2\pi}\int_{u_l}^{u_r} dEG_D(E). \tag{17}$$

For Equation (16), the integral range of the current is decided by the subtraction of the Fermi-Dirac function, which is a little wider than (u_l, u_r).

3.2 Differentiating the NEGF Process

We choose to implement the differentiable NEGF model under PyTorch (Paszke et al. [2019]). We extend the autograd function with implicit gradient techniques for calculating gradients through self-consistent iterations and the adjoint sensitivity method for calculating gradients through Poisson equations (Pontryagin [1987]). We also derive the gradient formula for the image charge method (Svizhenko and Anantram [2005]), which is a more efficient solution for Poisson equations, with the Fast Multipole Method (FMM) adopted for acceleration. The derived formula can be regarded as a summation of point charges produced by the gradients, which can also be computed with FMM. Details of the customized backward propagation modules are explained as follows.

Implicit Gradient. The implicit gradient method is implemented when the direct automatic differentiation through function $y = f(x)$ is unavailable or expensive to compute, and instances often arise when one wants to calculate gradients through numerical solvers of equilibrium problems or complicated iterative algorithms. Based on the implicit function theorem (Krantz and Parks [2002]), if there exists such constrained function $h(y, x) = 0$ where y is taken as the converged output of function f, the gradient $\frac{dy}{dx}$ can be given as:

$$\frac{dy}{dx} = -\left[\frac{\partial h(y,x)}{\partial y}\right]^{-1} \frac{\partial h(y,x)}{\partial x}. \tag{18}$$

We use the implicit gradient techniques to derive the gradient of the surface Green function (Sancho et al. [1985]), where according to the ideal definition of the two semi-infinite leads, the converged surface Green function $g_s(\theta)$ must satisfy the self-consistent Equation (9). Hence $h(g_s, \theta) = [A_{ll} - A_{ll-1} g_s A^{\dagger}_{l-1l}] - gs^{-1} = 0$, where A_{ll} stands for $[ES_{ll} - H_{ll}]$, and θ denotes the input variables to compute g_s. Thus we could write down the gradient of g_s with respect to θ explicitly by:

$$\frac{dg_s}{d\theta} = -\left[\frac{\partial h(g_s,\theta)}{\partial g_s}\right]^{-1} \frac{\partial h(g_s,\theta)}{\partial \theta}. \tag{19}$$

Here we should notice that, since such a constraint h is generally independent with the algorithm to compute the surface Green function, this gradient form is also valid for other algorithms such as the method based on solving generalized eigenvalue problems (Wang et al. [2008]).

Another scenario that the implicit gradient method can be applied to is to compute gradients through the self-consistent Poisson equation loop, where the system electrostatic potential is updated until consistent with the bias voltage of contacts and other boundaries conditions.

Adjoint Method for PDE. In order to perform backpropagation through the Poisson equation solver, gradients can be evaluated with the adjoint sensitivity method (Pontryagin [1987]), which is often applied in constrained optimization problems. Recent application in machine learning includes the Neural ODE (Chen et al. [2018]) and PDEs. Briefly speaking, the adjoint method employs a solver similar to the original problem for calculating gradients.

Gradient of FMM image charge method. An alternative approach to solve the Poisson equation raised in Equation (11), is to apply the point charge approximation, where the charge density is considered as the linear combination of a series of point charges as $\Delta q(r) = \sum_i \Delta q_i \delta(r - r_i)$. Then by employing the linearity of the Poisson equation, the original form can be further decomposed into a Laplace equation with Dirichlet boundary conditions and a Poisson equation with zero Dirichlet boundary conditions:

$$\begin{cases} -\nabla^2(\Delta V_1(r)) = 0, \\ \Delta V_1(r)|_{\{z_L, z_R\}} = \{V_L, V_R\}. \end{cases} \tag{20}$$

$$\begin{cases} -\nabla^2(\Delta V_2(r)) = \frac{1}{\epsilon}\Delta\rho(r), \\ \Delta V_2(r)|_{\Sigma} = 0. \end{cases} \tag{21}$$

The first Laplace equation can be easily solved by a linear drop potential. The second equation can be solved by assuming the charge density as a combination of point charges of each atom site. The closed form solution can be obtained using the image charge method (Svizhenko and Anantram [2005], Harb [2019]), and the second potential can be written as:

$$\begin{aligned} V_2(r_i) = &\sum_{j \in N, j \neq i} \frac{q_j}{4\pi\epsilon} \frac{1}{\sqrt{t_{ij}^2 + (z_i - z_j)^2}} \\ &+ \sum_{j \in N} \frac{q_j}{4\pi\epsilon} \sum_{n=1}^{\infty} \left[\frac{1}{\sqrt{t_{ij}^2 + \Delta_1^2}} - \frac{1}{\sqrt{t_{ij}^2 + \Delta_2^2}} + \frac{1}{\sqrt{t_{ij}^2 + \Delta_3^2}} - \frac{1}{\sqrt{t_{ij}^2 + \Delta_4^2}} \right], \end{aligned} \tag{22}$$

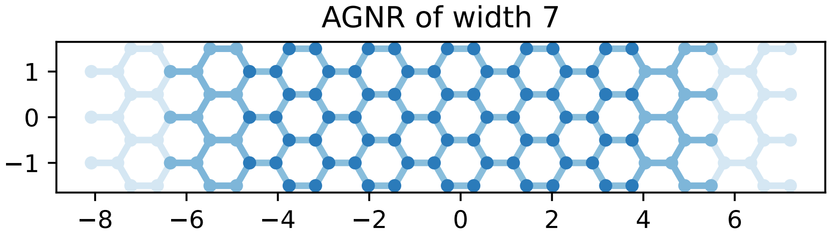


Figure 2: A transport system of AGNR, with width 7 and length 5.

where $t_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$, and Δ^2 stands for the distance in the transport direction between central charges and charges from two electrodes. Therefore, the first term here describes the interactions inside the device, while all the remaining terms simulate the effect of its coupling to charges outside. The summation of the second term is computed until achieving certain accuracy, which is empirically hundreds of site numbers. Hence a direct summation is also too expensive to compute. In this case, the Fast Multipole Method (Engheta et al. [1992]) is employed to reduce the computational complexity from $O(N^3)$ to $O(N^{4/3})$.

To perform backward propagation through the fast multipole layer, the gradient of the output potential to the charges is required. By taking the derivative of a target objective $L : C^d \rightarrow R$, the derivative of L with respect to charge q_j can be expanded as the image summation form of accumulated gradients from the last layer, which is:

$$\frac{\partial L(V)}{\partial q_j} = \sum_i \frac{\partial L}{\partial V_i} \frac{\partial V_i}{\partial q_j} \tag{23}$$

$$\begin{aligned} &= \sum_{i \in N, i \neq j} \frac{\partial L / \partial V_i}{4\pi\epsilon} \frac{1}{\sqrt{t_{ij}^2 + (z_j - z_i)^2}} \\ &+ \sum_{i \in N} \frac{\partial L / \partial V_i}{4\pi\epsilon} \sum_{n=1}^{\infty} \left[\frac{1}{\sqrt{t_{ij}^2 + \Delta_1^2}} - \frac{1}{\sqrt{t_{ij}^2 + \Delta_2^2}} + \frac{1}{\sqrt{t_{ij}^2 + \Delta_3^2}} - \frac{1}{\sqrt{t_{ij}^2 + \Delta_4^2}} \right]. \end{aligned} \tag{24}$$

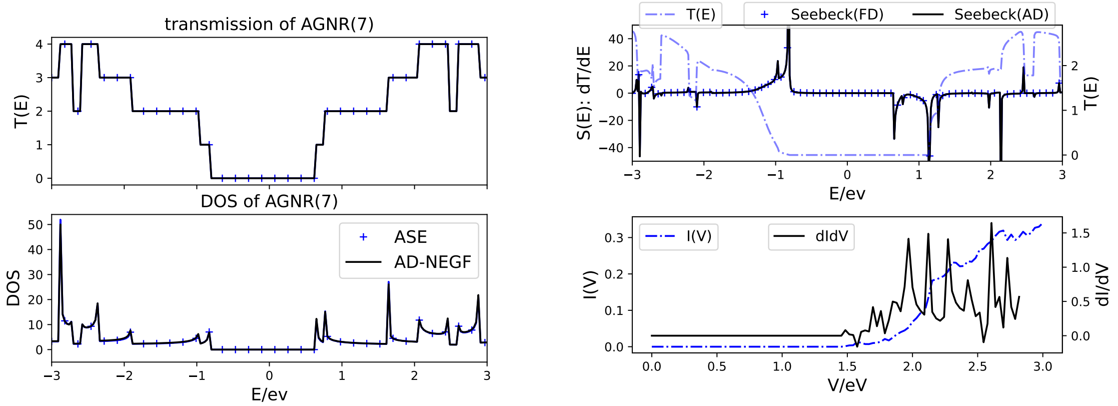
Similarly, computing gradients of this form can be accelerated by the Fast Multipole Method, which is also with complexity $O(N^{4/3})$ and much faster than solving adjoint Poisson equations.

4 Applications

In this section, the results on several applications are displayed. For all experiments, we take graphene as the transport system, including the Armchair Graphene NanoRibbon (AGNR) and the graphene nano-junction. The basic structure of graphene is displayed in Figure 2. The experiments are organized as follows. We first validate the result of the forward transport calculation of AD-NEGF by comparing with ASE (Larsen et al. [2017]), an atomistic simulation package including electron transport modules. The differential transport properties calculated by AD-NEGF are compared with numerical differentiation, including the Seebeck coefficient and the differential conductance, where it is shown that AD-NEGF can achieve better accuracy and numerical stability. In what follows, two examples of gradient based optimization are presented, one demonstrates the potential of conducting high-dimensional variable optimization with the AD-NEGF framework, and the other highlights solving more practical end-to-end inverse design by cooperating AD-NEGF with established material models.

4.1 Differential Transmission Quantity Computation

A direct and major application to perform differentiation on physical models is to evaluate differential physical quantities. Most of the times, the analytical form is difficult to obtain. For numerical differentiation, there is a trade-off between the round-off error and the truncation error when choosing the step-size ([Gautschi, 1997, Chap. 3]), and the computation will be very expensive when the input dimension is high. On the contrary, automatic differentiation can achieve machine precision while maintaining $O(1)$ complexity when the output dimension is low and the input dimension is high (Baydin et al. [2018]).



(a) Transmission and DOS calculated by AD-NEGF and confirmed with ASE.

(b) Seebeck coefficient and differential conductance calculated by AD-NEGF.

Figure 3: Transmission Quantity Computation with AD-NEGF.

In this experiment, we first validate the correctness of the forward computation of AD-NEGF. As shown in Figure 3(a), the transmission coefficient and the density of states (DOS) of an AGNR system with width 7 are computed by AD-NEGF, which perfectly match the results of ASE. Based on it, we compute two differential transmission quantities, the Seebeck coefficient and the differential conductance, which are shown in Figure 3(b). The Seebeck coefficient is the derivative of transmission $T(E)$ with respect to the chemical potential E (Reddy et al. [2007]): $S_{junction} = -\frac{\pi^2 k_B^2 T}{3e}\frac{\partial ln(T(E))}{\partial E}$, where T stands for the temperature and k_B is the Boltzmann constant. The differential conductance is the gradient of electronic current to voltage: $I_D = \frac{dI}{dV}$.

The singularity of the transmission function leads to peaks in the Seebeck coefficient curve, which is highly sensitive thus challenging for derivative calculation, as illustrated in Figure 4. To amplify the phenomenon for clearer demonstration, the output transmission coefficient $T(E)$ of the forward computation is transformed into half-precision floating-point format for both automatic and numerical differentiation, before it is used to compute the Seebeck coefficient. It can be seen that, with AD-NEGF, we can still generate high-quality results. However, for numerical differentiation, the trade-off between the truncation error and the round-off error is observed by selecting different step-sizes from 1e-2 to 1e-5. With a large step-size, peaks may be skipped or mistakenly generated due to truncation error. With a small step-size, lacking in machine precision causes noises on the curve. Specifically for step-size 1e-5, the calculated curve becomes totally meaningless. Moreover, even though this is not a high-dimensional input situation, evaluating the Seebeck coefficient with AD-NEGF can still be faster than numerical differentiation, since in AD-NEGF the backward pass is improved. According to our experiments, for a smaller system with 70 carbon atoms, to compute the Seebeck coefficient for 400 energy samples costs 71.1 seconds with AD-NEGF and 98.3 seconds with numerical differentiation. For a larger system with 240 carbon atoms, to compute the Seebeck coefficient for 400 energy samples costs 363.1 seconds with AD-NEGF and 512.6 seconds with numerical differentiation.

To summarize, by conducting the above experiments, the correctness and effectiveness of AD-NEGF are validated. With AD-NEGF, differential transport quantities can be calculated simply by calling one backward step. Moreover, the process of computing derivatives is itself differentiable, permitting the computation of higher order derivatives, which remains for further discovery.

4.2 Transmission Fitting

Inverse problems, which require to infer input parameters reversely from the output objectives, are in general difficult in first-principle simulations. Black-box optimization methods require sampling a large number of input combinations, the cost of which grows exponentially with the number of parameters. Based on the efficient and accurate gradient computation by AD-NEGF, performing gradient-based optimization holds the potential to outperform black-box optimization methods for high dimensional inverse problems.

We conduct a 10^4 dimensional optimization experiment to fit the transmission curve of one graphene nano-junction to another. The target system is a 7-4 nano-junction, with 7 graphene rings on the left and 4 on the right. The fitting system is a 5-2 nano-junction, and the fitting variables are the elements of its Hamiltonian, including the device, leads and the

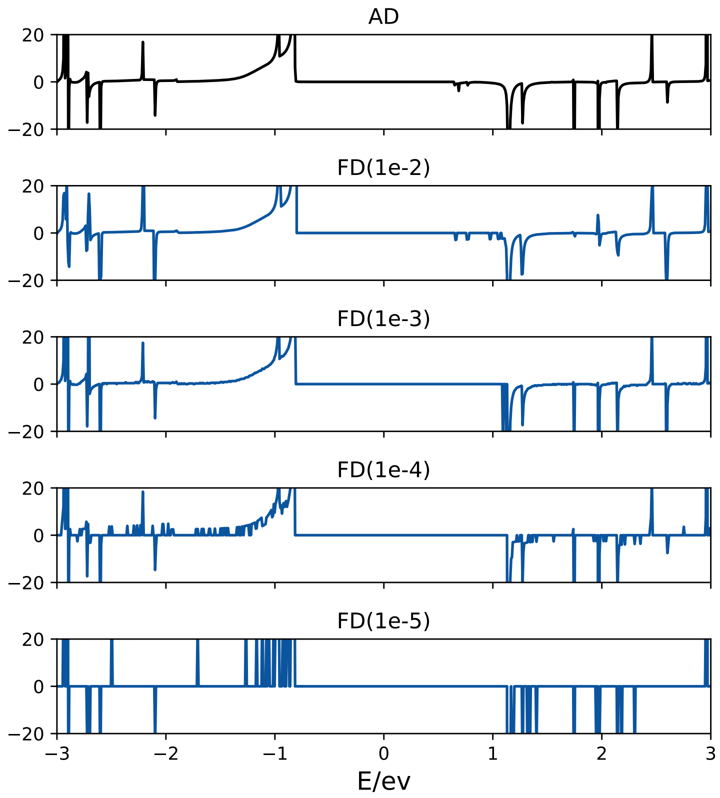


Figure 4: Comparison of Automatic Differentiation and Numerical Differentiation with different step-sizes.

corresponding couplings. The dimension of the optimizing variables is at the level of 10^4. The transmission curve, as shown in Figure 5, is sampled with 2000 energy points from $E \in (-5, 5)$. Since directly computing the gradients of all 2000 points would not be efficient for iterations, we apply the stochastic gradient descent algorithm to conduct mini-batch training, which has shown supremacy of efficiency and performance in high dimensional optimization problems. The fitting parameters are trained with the Adam optimizer (Kingma and Ba [2014]) built in PyTorch, as this task shares similarity with the process of training a neural network.

The results are displayed in Figure 5, where the loss is reduced to a considerably low level, which means the converged parameters of the 5-2 nano-junction fit nicely to the larger 7-4 nano-junction. The fitted curve is akin to a smoothed version of the curve of the 7-4 junction, which agrees with the intuition since a graphene junction of 5-2 is of less freedom than that of a 7-4 nano-junction. On the other hand, we have also tried traditional methods on this problem such as the Bayesian optimization, the genetic algorithm, and gradient-based optimization with numerical differentiation, but none of them can even work for this high-dimensional problem.

This experiment demonstrates that, AD-NEGF, by cooperating with gradient-based optimization methods, can handle inverse design tasks that are intractable for conventional parameter optimization methods because of the curse of dimensionality.

4.3 On-Site Doping Optimization

Modern material engineering is capable of manipulating at the atomic level. More specifically speaking, by performing processes such as deformation, doping, etc., microscopic physical quantities such as atomic spatial coordinates, bond lengths and doping positions can be changed, which further modify the macroscopic material properties. Doping process is one of the most common techniques in material development, which can dramatically change the properties of the original material, by injecting foreign atoms into specific positions. In this experiment, we further explore the possibility to solve practical inverse problems with AD-NEGF by performing an end-to-end doping optimization cooperated with established material models.

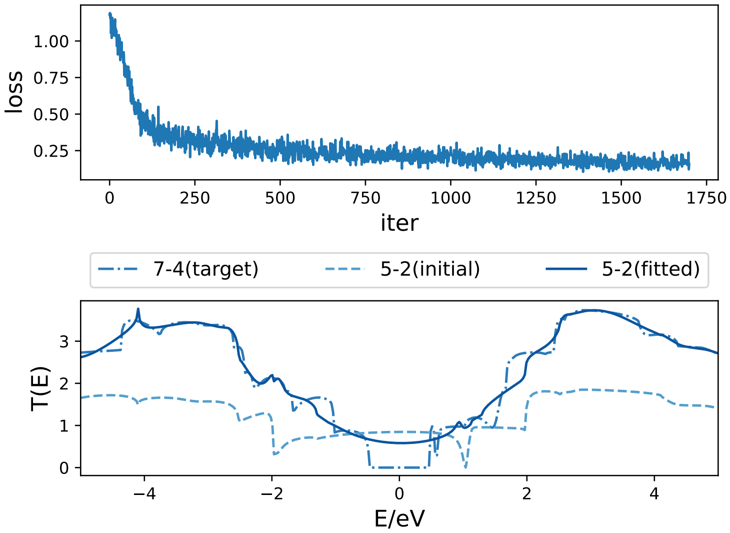
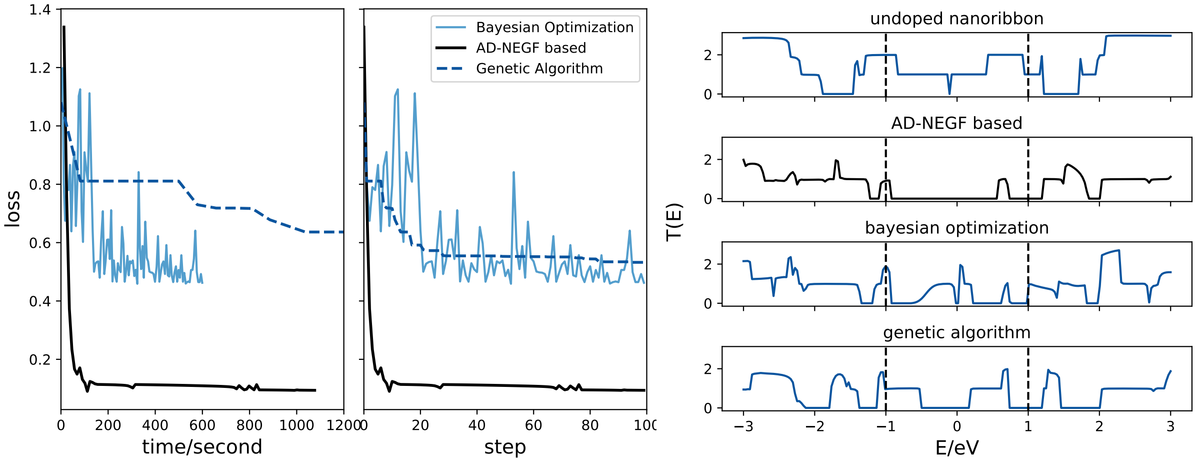


Figure 5: The fitting loss and the fitted transmission curve of a 5-2 graphene nano-junction.



(a) Loss curves with running time and numerical steps as the x-axis respectively.

(b) Original and optimized transmission curves.

Figure 6: Comparison between AD-NEGF and conventional black-box optimization methods in the doping optimization task.

In this experiment, we try to reduce the average transmission of AGNR (7) between energies -1eV and 1eV by injecting other atoms into the center of the AGNR system along the transmission direction. A reduction of transmission coefficient near zero Energy point would indicate an increase of the truncation voltage, which changes the semi-conductive properties of the device (Wu et al. [2013]). Doping can be modeled as an effective change in the site and the hopping terms in the tight-binding Hamiltonian, i.e., the diagonal and off-diagonal elements of the Hamiltonian matrix. This on-site approximation allows us to treat doping optimization as tuning local terms in the Hamiltonian influenced by the injected atoms. However, although the process above is applicable, the tuning terms in the TB Hamiltonian need to be distinguished carefully from those invariant ones. It will be more convenient to cooperate with an SKTB model, which constructs the TB Hamiltonian based on strict rules of local dependence of atom identities and their semi-empirical SK parameters. Beside convenience, it has more concrete physical interpretation than directly optimizing elements of the Hamiltonian, since it provides guidelines for practitioners to find the possible atom satisfying the SKTB parameters from the optimization result. In this way, doping optimization is modeled as an optimization of the SKTB parameters of the doped atoms, which include the orbital energy and parameters for two center integrals. The total number of the optimization variables is 13.

For comparison, we also perform black-box optimizations including the genetic algorithm and the Bayesian optimization. The results are displayed in Figure 6. In the loss diagram, the gradient based method converges significantly faster and better than the other approaches, especially in the beginning of the training. The loss curves of the genetic optimization and the Bayesian optimization are also dropping, but much slower and less effective, with either the running time or the iteration steps as the x-axis. Moreover, their performances are sensitive to preset higher-parameters. Corresponding to the loss curves, the results of optimized transmission curves demonstrate the advantages of AD-NEGF in a more straightforward way, where the gradient-based optimization gives a much cleaner band with low transmission in the target interval (-1eV, 1eV) compared to other methods. These results validate the effectiveness of the AD-NEGF method in conducting practical atomic level inverse design to optimize transport properties by cooperating with material models.

5 Conclusion

In this paper, we have proposed AD-NEGF, the first end-to-end differentiable quantum transport simulator to our best knowledge. It aims to improve the efficiency of first-principle transport simulations by providing gradient information based on differentiable programming. Compared with numerical differentiation, gradients can be computed more efficiently and accurately. Moreover, it accelerates parameter fitting and parameter optimization with gradient-based optimization. The results are validated on applications such as differential physical quantity computation, transmission parameter fitting, and device optimization.

Acknowledgements

This work is supported in part by the state key development program of China (grant No. 2017YFE0111900), Natural Science Foundation of China (grant No.61772363). L.W. is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDB30000000 and National Natural Science Foundation of China under Grant No. T2121001.

References

Carlo Jacoboni. *Theory of electron transport in semiconductors: a pathway from elementary physics to nonequilibrium Green functions*, volume 165. Springer Science & Business Media, 2010.

Michael Wimmer. *Quantum transport in nanostructures: From computational concepts to spintronics in graphene and magnetic tunnel junctions*. PhD thesis, 2009.

Mahdi Pourfath. *The Non-Equilibrium Green's Function Method for Nanoscale Device Simulation*, volume 3. Springer, 2014.

Peter A Markowich, Christian A Ringhofer, and Christian Schmeiser. *Semiconductor equations*. Springer Science & Business Media, 2012.

MP Anantram, Mark S Lundstrom, and Dmitri E Nikonov. Modeling of nanoscale devices. *Proceedings of the IEEE*, 96(9):1511–1550, 2008.

J-S Wang, Jian Wang, and JT Lü. Quantum thermal transport in nanostructures. *The European Physical Journal B*, 62 (4):381–404, 2008.

Supriyo Datta. *Electronic transport in mesoscopic systems*. Cambridge university press, 1997.

Kangyuan Li, Junqiang Lu, and Feng Zhai. Neural networks for modeling electron transport properties of mesoscopic systems. *Physical Review B*, 102(6):064205, 2020.

Marius Bürkle, Umesha Perera, Florian Gimbert, Hisao Nakamura, Masaaki Kawata, and Yoshihiro Asai. Deep-learning approach to first-principles transport simulations. *Physical Review Letters*, 126(17):177701, 2021.

Artem K Pimachev and Sanghamitra Neogi. First-principles prediction of electronic transport in fabricated semiconductor heterostructures via physics-aware machine learning. *npj Computational Materials*, 7(1):1–12, 2021.

Connor Schenck and Dieter Fox. Spnets: Differentiable fluid dynamics for deep neural networks. In *Conference on Robot Learning*, pages 317–335. PMLR, 2018.

Muhammad F Kasim and Sam M Vinko. Learning the exchange-correlation functional from nature with fully differentiable density functional theory. *arXiv preprint arXiv:2102.04229*, 2021.

Samuel S. Schoenholz and Ekin D. Cubuk. Jax m.d. a framework for differentiable physics. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

MV Klymenko, JA Vaitkus, JS Smith, and JH Cole. Nanonet: an extendable python framework for semi-empirical tight-binding models. *Computer Physics Communications*, 259:107676, 2021.

L. V. Keldysh. Diagram technique for nonequilibrium processes. *Zh. Eksp. Teor. Fiz.*, 47:1515–1527, 1964.

Leo P Kadanoff. *Quantum statistical mechanics*. CRC Press, 2018.

David Ferry and Stephen Marshall Goodnick. *Transport in nanostructures*. Number 6. Cambridge university press, 1999.

Jeremy Taylor, Hong Guo, and Jian Wang. Ab initio modeling of quantum transport properties of molecular electronic devices. *Physical Review B*, 63(24):245407, 2001.

Mads Brandbyge, José-Luis Mozos, Pablo Ordejón, Jeremy Taylor, and Kurt Stokbro. Density-functional method for nonequilibrium electron transport. *Physical Review B*, 65(16):165401, 2002.

Alexander L Fetter and John Dirk Walecka. *Quantum theory of many-particle systems*. Courier Corporation, 2012.

MP Lopez Sancho, JM Lopez Sancho, JM Lopez Sancho, and J Rubio. Highly convergent schemes for the calculation of bulk and surface green functions. *Journal of Physics F: Metal Physics*, 15(4):851, 1985.

PS Krstić, X-G Zhang, and WH Butler. Generalized conductance formula for the multiband tight-binding model. *Physical Review B*, 66(20):205319, 2002.

Ivan Rungger and Stefano Sanvito. Algorithm for the construction of self-energies for electronic transport calculations based on singularity elimination and singular value decomposition. *Physical Review B*, 78(3):035407, 2008.

Nick Papior, Nicolás Lorente, Thomas Frederiksen, Alberto García, and Mads Brandbyge. Improvements on non-equilibrium and transport green function techniques: The next-generation transiesta. *Computer Physics Communications*, 212:8–24, 2017.

Søren Smidstrup, Troels Markussen, Pieter Vancraeyveld, Jess Wellendorff, Julian Schneider, Tue Gunst, Brecht Verstichel, Daniele Stradi, Petr A Khomyakov, Ulrik G Vej-Hansen, et al. Quantumatk: an integrated platform of electronic and atomic-scale modelling tools. *Journal of Physics: Condensed Matter*, 32(1):015901, 2019.

Sebastian Steiger, Michael Povolotskyi, Hong-Hyun Park, Tillmann Kubis, and Gerhard Klimeck. Nemo5: A parallel multiscale nanoelectronics modeling tool. *IEEE Transactions on Nanotechnology*, 10(6):1464–1474, 2011.

Alejandro Lopez-Bezanilla and O Anatole von Lilienfeld. Modeling electronic quantum transport with machine learning. *Physical Review B*, 89(23):235411, 2014.

T Župančić, I Stresec, and M Poljak. Predicting the transport properties of silicene nanoribbons using a neural network. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 44–48. IEEE.

Seung-Cheol Han, Jonghyun Choi, and Sung-Min Hong. Acceleration of three-dimensional device simulation with the 3d convolutional neural network. In *2021 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pages 52–55. IEEE, 2021.

Satofumi Souma and Matsuto Ogawa. Neural network model for implementation of electron–phonon scattering in nanoscale device simulations based on negf method. In *2021 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pages 56–59. IEEE, 2021.

Satofumi Souma and Matsuto Ogawa. Acceleration of nonequilibrium green's function simulation for nanoscale fets by applying convolutional neural network model. *IEICE Electronics Express*, pages 17–20190739, 2020.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *arXiv preprint arXiv:1909.01377*, 2019.

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019.

Mike Innes, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B Shah, and Will Tebbutt. A differentiable programming system to bridge machine learning and scientific computing. *arXiv preprint arXiv:1907.07587*, 2019.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31:7178–7189, 2018.

C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax-a differentiable physics engine for large scale rigid body simulation. 2021.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

Li Li, Stephan Hoyer, Ryan Pederson, Ruoxi Sun, Ekin D Cubuk, Patrick Riley, Kieron Burke, et al. Kohn-sham equations as regularizer: Building prior knowledge into machine-learned physics. *Physical review letters*, 126(3): 036401, 2021.

Teresa Tamayo-Mendoza, Christoph Kreisbeck, Roland Lindh, and Alán Aspuru-Guzik. Automatic differentiation in quantum chemistry with applications to fully variational hartree–fock. *ACS central science*, 4(5):559–566, 2018.

Fabijan Pavošević and Sharon Hammes-Schiffer. Automatic differentiation for coupled cluster methods. *arXiv preprint arXiv:2011.11690*, 2020.

John C Slater and George F Koster. Simplified lcao method for the periodic potential problem. *Physical Review*, 94(6): 1498, 1954.

Taisuke Ozaki. Continued fraction representation of the fermi-dirac function for large-scale electronic structure calculations. *Physical Review B*, 75(3):035123, 2007.

Denis A Areshkin and Branislav K Nikolić. Electron density and transport in top-gated graphene nanoribbon devices: First-principles green function algorithms for systems containing a large number of atoms. *Physical Review B*, 81 (15):155450, 2010.

A Svizhenko and MP Anantram. Effect of scattering and contacts on current and electrostatics in carbon nanotubes. *Physical Review B*, 72(8):085430, 2005.

Markus Zahn. Point charge between two parallel grounded planes. *American Journal of Physics*, 44(11):1132–1134, 1976.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC press, 1987.

Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

Mohammed Aziz Harb. *Scattering Effects in Atomistic Quantum Transport Simulations*. McGill University (Canada), 2019.

Nader Engheta, William D Murphy, Vladimir Rokhlin, and Marius S Vassiliou. The fast multipole method (fmm) for electromagnetic scattering problems. *IEEE Transactions on Antennas and Propagation*, 40(6):634–641, 1992.

Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 1997.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18, 2018.

Pramod Reddy, Sung-Yeon Jang, Rachel A Segalman, and Arun Majumdar. Thermoelectricity in molecular junctions. *Science*, 315(5818):1568–1571, 2007.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Guo-xun Wu, Zhen-qing Wang, Yu-hang Jing, and Chao-ying Wang. I–v curves of graphene nanoribbons under uniaxial compressive and tensile strain. *Chemical Physics Letters*, 559:82–87, 2013.