JPhys Materials

PAPER • OPEN ACCESS

Phoebe: a high-performance framework for solving phonon and electron Boltzmann transport equations

To cite this article: Andrea Cepellotti *et al* 2022 *J. Phys. Mater.* **5** 035003

View the article online for updates and enhancements.

You may also like

- THE IRREGULAR SATELLITES: THE MOST COLLISIONALLY EVOLVED POPULATIONS IN THE SOLAR SYSTEM
William F. Bottke, David Nesvorný, David Vokrouhlický et al.
- The Orbits of the Main Saturnian Satellites, the Saturnian System Gravity Field, and the Orientation of Saturn's Pole
Robert. A. Jacobson
- Cladistical Analysis of the Jovian and Saturnian Satellite Systems
Timothy. R. Holt, Adrian. J. Brown, David Nesvorný et al.

This content was downloaded from IP address 128.103.24.178 on 15/05/2023 at 05:44

Journal of Physics: Materials

PAPER

Phoebe: a high-performance framework for solving phonon and electron Boltzmann transport equations

Andrea Cepellotti[1], **Jennifer Coulter**[1], **Anders Johansson**[1], **Natalya S Fedorova**[1,2] and **Boris Kozinsky**[1,3,*]

[1] Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, United States of America
[2] Materials Research and Technology Department, Luxembourg Institute of Science and Technology, 5 Avenue des Hauts-Fourneaux, L-4362 Esch/Alzette, Luxembourg
[3] Robert Bosch LLC, Research and Technology Center, Cambridge, MA, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: bkoz@g.harvard.edu

Keywords: thermal transport, electron transport, electron–phonon interaction, phonon–phonon interaction, thermoelectrics

OPEN ACCESS

RECEIVED
21 June 2022

REVISED
1 August 2022

ACCEPTED FOR PUBLICATION
4 August 2022

PUBLISHED
19 August 2022

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Abstract

Understanding the electrical and thermal transport properties of materials is critical to the design of electronics, sensors, and energy conversion devices. Computational modeling can accurately predict material properties but, in order to be reliable, requires accurate descriptions of electron and phonon states and their interactions. While first-principles methods are capable of describing the energy spectrum of each carrier, using them to compute transport properties is still a formidable task, both computationally demanding and memory intensive, requiring integration of fine microscopic scattering details for estimation of macroscopic transport properties. To address this challenge, we present Phoebe—a newly developed software package that includes the effects of electron–phonon, phonon–phonon, boundary, and isotope scattering in computations of electrical and thermal transport properties of materials with a variety of available methods and approximations. This open source C++ code combines MPI-OpenMP hybrid parallelization with GPU acceleration and distributed memory structures to manage computational cost, allowing Phoebe to effectively take advantage of contemporary computing infrastructures. We demonstrate that Phoebe accurately and efficiently predicts a wide range of transport properties, opening avenues for accelerated computational analysis of complex crystals.

1. Introduction

As technology use and its energy footprint continue to grow, so does the importance of improving the properties of the materials which underlie such technology. Materials transport properties, and in particular the capacity of a crystal to conduct heat and electricity, are critical to the function of virtually any electronic device. These properties are determined by the spectrum and velocities of electrons and lattice vibrations in a crystal lattice, and the interactions of these carriers among themselves and with crystal defects.

The task of engineering better materials can be greatly accelerated through computational studies, as exemplified by successful computational discovery of new thermoelectric compounds [1]. In particular, the prediction of materials transport properties has reached high accuracy and robustness in recent years, thanks to the combination of first-principles density functional theory (DFT) simulations and numerical Boltzmann transport equation (BTE) solvers, the former capable of accurately parameterizing the microscopic characteristics of crystal quasiparticles and the latter providing macroscopic transport properties from microscopic data. Both components are still the subject of study and are continuously evolving, as there are several challenges, both technical and conceptual, in both the calculation of the relevant microscopic scattering interactions, as well as in increasing the sophistication at which the BTE can be solved. There are several active efforts on software implementations of these simulation techniques, and the

© 2022 The Author(s). Published by IOP Publishing Ltd

challenge is to achieve the best possible accuracy for the least computational cost. Various codes for solving the BTE are currently available, including EPW [2], BoltzWann [3], PERTURBO [4], and BoltzTrap [5] for electronic properties, as well as phonopy/phono3py [6, 7], ShengBTE [8], ALAMODE [9], kALDo [10] and GPU_PBTE [11] for the calculation of phonon transport. The physics of heat and electricity conduction are intrinsically linked one to the other, and, as we will discuss in further detail, the electron and phonon BTE also share several common features. Despite the similarities, no single currently developed software package is capable of describing both kinds of transport processes.

Our vision behind the development of this new software was to take advantage of the parallels between physical descriptions of thermal and electrical conduction to provide a single framework to fully characterize materials transport properties. In addition, we aimed not only to provide a tool for the prediction of materials transport properties; instead, we developed a package containing the abstraction and generic data structures needed to calculate conduction. In doing so, we created a flexible computational framework designed to easily accommodate extensions and improvements, as ongoing work in this field continues to reveal new and exciting developments. Lastly, we capitalized on the rapid evolution of computational hardware, in particular with respect to the rise of GPU accelerators and the increasingly heterogeneous architecture of modern supercomputing facilities. As a result, we aspired to develop a new software which could run efficiently on any available computing environment.

This work presents the new open-source software package Phoebe—a framework of PHOnon and Electron Boltzmann equation solvers—for the prediction of electron and phonon transport properties from first principles. Phoebe implements the construction and solution of the BTE for crystals using a scattering matrix formalism that takes into account the electron–phonon, phonon–phonon, boundary, and phonon–isotope scattering contributions to transport properties. Phoebe implements several unique features, including Wigner transport formalism, the electron–phonon averaged approximation, and hydrodynamic transport properties. Additionally, the simultaneous availability of both phonon and electron effects enables the calculation of properties such as the thermal conductivity of metals.

This code is designed to take advantage of contemporary supercomputing facilities to tackle the intensive computational workload of *ab-initio* transport calculations using multiple levels of parallelization. The package is written in object-oriented `C++` and utilizes a hybrid MPI-OpenMP parallelization scheme, together with GPU acceleration, distributed memory data-structures, and parallel HDF5 input/output operations. We discuss the capability of Phoebe to scale on supercomputing hardware and to reduce the computational cost of the calculation with respect to other codes, streamlining the calculation of transport properties. With its flexible, object-oriented structure, Phoebe is designed for easy implementation of future features, such as additional quasiparticle interactions, and the possibility of interfacing with a wide range of *ab-initio* software for the calculation of interaction matrix elements.

We showcase Phoebe's capabilities with the calculations of the transport properties of GaN, a III–V semiconductor used for high-power electronics. After validating the results against experimental measurements, we argue that Phoebe is able to accurately and efficiently predict a wide range of material transport properties such as the electrical and thermal conductivity and thermoelectric performance in large and complex crystal structures.

2. Functionalities

Phoebe provides the tools necessary for a complete first-principles study of electrical, thermal, and thermoelectric transport properties of crystalline materials in a single package. The code takes the output of *ab-initio* calculations of carrier spectra and coupling matrix elements and processes them in order to predict transport properties. Phoebe provides the capability to compute a wide range of properties, such as:

- Electron and phonon band-structure, density of states, and velocity operator;
- Linewidths and lifetimes of electrons, limited by electron–phonon interaction and boundary scattering;
- Linewidths and lifetimes of phonons, limited by three-phonon, isotope and boundary scattering;
- Lattice thermal conductivity;
- Electrical conductivity, mobility, electronic thermal conductivity and Seebeck coefficient;
- Phonon and electron viscosity;

with more properties planned for future releases. Moreover, we provide different approaches for modeling these properties at various levels of accuracy and/or computational cost. For example, the electron–phonon coupling can be evaluated either with the Wannier-interpolation technique, or the electron–phonon averaged approximation; transport properties can be evaluated using the BTE or the Wigner distribution function; and the BTE can be solved with various solvers, from relaxation time

models to exact solutions. Phoebe is made available freely on GitHub [12] under the MIT open-source license, and is written using object-oriented C++, with the intention of allowing future extensions and implementation of new features with minimal effort. We rely on a number of open-source libraries, including Eigen [13] for small linear algebra problems and ScaLAPACK [14] for massively parallel linear algebra problems, as well as Spglib [15] for analyzing symmetries, PugiXML [16] for reading XML files, and HighFive [17] to handle I/O based using HDF5 [18].

Additionally, Phoebe is designed to efficiently scale on contemporary HPC infrastructure. Parallelization is achieved with a hybrid MPI and OpenMP architecture. The most computationally expensive parts of the code are further accelerated with Kokkos [19], which allows for the optional use of GPU accelerators. Finally, we take advantage of several GitHub functionalities, such as GitHub pages to create a static website for user accessing the code and GitHub Actions to perform a continuous integration. The documentation built with Doxygen [20] and Sphinx [21] is hosted online on Read The Docs [22].

3. Microscopic properties

In this section, we report a brief summary of the main approaches used and implemented in Phoebe, that are needed to obtain a description of microscopic electron and phonon properties, referring to other works for detailed explanations.

3.1. Phonon properties

In crystals, the motion of atoms is typically an oscillation around their equilibrium positions, and is thus described with the atomic displacements $u(s\alpha\boldsymbol{R})$, where $\boldsymbol{R}$ is the Bravais lattice vector labeling a unit cell, s is an index over the unit cell basis of atoms, and α is the Cartesian direction of the atomic displacement. To lowest orders in perturbation theory, the lattice potential energy near the minimum is described with a Taylor expansion [23], where the first two important terms are the second derivative $V^{(2)}$ of the crystal energy with respect to atomic displacements

$$V^{(2)}(s\alpha\boldsymbol{R}, s'\beta\boldsymbol{R}') = \frac{\partial^2 E}{\partial u(s\alpha\boldsymbol{R})\partial u(s'\beta\boldsymbol{R}')}\,, \tag{1}$$

and the third derivative $V^{(3)}$

$$V^{(3)}(s\alpha\boldsymbol{R}, s'\beta\boldsymbol{R}', s''\gamma\boldsymbol{R}'') = \frac{\partial^3 E}{\partial u(s\alpha\boldsymbol{R})\partial u(s'\beta\boldsymbol{R}')\partial u(s''\gamma\boldsymbol{R}'')}\,, \tag{2}$$

while higher order derivatives are here neglected. These quantities can be computed using existing *ab-initio* codes. In particular, Phoebe interfaces with phonopy/phono3py [6] and Quantum ESPRESSO (QE) [24] for the calculation of the second derivative and with phono3py [7] and ShengBTE [25] for the calculation of the third derivative of the energy.

These two derivatives allow the characterization of phonon properties and of their scattering mechanisms by means of a Fourier transform. For example, the Fourier transform of the second derivative provides the dynamical matrix D defined as [26]

$$D(s\alpha|s'\alpha')(\boldsymbol{q}) = \sum_{\boldsymbol{R}} V^{(2)}(s\alpha\boldsymbol{0}|s'\alpha'\boldsymbol{R})e^{i\boldsymbol{q}\cdot\boldsymbol{R}}\,, \tag{3}$$

where $\boldsymbol{q}$ is an arbitrary wavevector, and one Bravais lattice vector component of the second derivative $V^{(2)}$ has been set to a reference unit cell $\boldsymbol{0}$ without loss of generality thanks to the translational periodicity of the crystal. The phonon harmonic properties are then found by diagonalizing the dynamical matrix

$$D(s\alpha|s'\alpha')(\boldsymbol{q})z_{s'\alpha'\nu}(\boldsymbol{q}) = (\hbar\omega_\nu(\boldsymbol{q}))^2 z_{s\alpha\nu}(\boldsymbol{q})\,, \tag{4}$$

where ν is an eigenvalue index (i.e. the phonon branch), $\hbar\omega_{\boldsymbol{q}\nu}$ the phonon energy and $z^{s\alpha}_{\boldsymbol{q}\nu}$ the phonon eigenvector. Polar corrections to the dynamical matrix are taken into account in Phoebe as explained in [26].

The Fourier transform of the third derivative $V^{(3)}$ instead describes the coupling strength of a three-phonon scattering event, and is computed as [27, 28]

$$\begin{aligned} &V^{(3)}(\boldsymbol{q}\nu, \boldsymbol{q}'\nu', \boldsymbol{q}''\nu'') \\ &= \frac{1}{N_0^3}\left(\frac{2}{\hbar}\right)^{3/2} \sum_{\boldsymbol{R}\boldsymbol{R}'} \sum_{\alpha\beta\gamma} \sum_{ss's''} \sqrt{\frac{m_s m_{s'} m_{s''}}{\omega_{\boldsymbol{q}\nu}\omega_{\boldsymbol{q}'\nu'}\omega_{\boldsymbol{q}''\nu''}}} z^{s\alpha *}_{\boldsymbol{q}\nu} z^{s'\beta *}_{\boldsymbol{q}'\nu'} z^{s''\gamma *}_{\boldsymbol{q}''\nu''} e^{-i(\boldsymbol{q}\cdot\boldsymbol{R}+\boldsymbol{q}'\cdot\boldsymbol{R}')} V^{(3)}(s\alpha\boldsymbol{R}, s'\beta\boldsymbol{R}', s''\gamma\boldsymbol{0})\,, \end{aligned} \tag{5}$$

where again one Bravais lattice vector component can be set to a reference unit cell **0**.

The equations (3) and (5) allow us to compute phonon properties (such as phonon energies or scattering coupling strength) at arbitrary wavevectors, starting from the derivatives computed in real space that can be obtained from *ab-initio* codes. This interpolation procedure is a major computational bottleneck due to the large number of tensor indices and the fact that phonon properties need to be computed for all wavevectors used to sample the Brillouin zone (BZ). As a result, it is critical to efficiently implement this interpolation procedure, in particular for the three-phonon coupling strength, as will be discussed in section 5.5.

3.2. Electron properties

In order to compute the electronic transport properties, it is necessary to characterize electrons and the electron–phonon interaction. Phoebe relies on an external *ab-initio* code to compute the electronic energies $\epsilon_{\boldsymbol{k}b}$ and wavefunctions $|\psi_{\boldsymbol{k}b}\rangle$, where $\boldsymbol{k}$, the electron wavevector, and b, the band index, are Bloch quantum numbers (spin is omitted for simplicity), which correspond to eigenvalues and eigenstates of the electronic Hamiltonian

$$H^B(\boldsymbol{k})|\psi_{\boldsymbol{k}b}\rangle = \epsilon_{\boldsymbol{k}b}|\psi_{\boldsymbol{k}b}\rangle \ . \tag{6}$$

Once the electronic wavefunction is known, it is possible to compute other properties such as the interaction of electrons with the lattice. The electron–phonon coupling g is defined as [29]

$$g_{b'b\nu}(\boldsymbol{k},\boldsymbol{q}) = \left(\frac{1}{2m\omega_{\boldsymbol{q}\nu}}\right)^{1/2} \langle\psi_{\boldsymbol{k}+\boldsymbol{q},b'}|\partial_{\boldsymbol{q}\nu}V|\psi_{\boldsymbol{k}b}\rangle = \left(\frac{1}{2m\omega_{\boldsymbol{q}\nu}}\right)^{1/2} g^0_{b'b\nu}(\boldsymbol{k},\boldsymbol{q}) \ , \tag{7}$$

and provides the coupling strength of the interaction between two electrons and a phonon, and $\partial_{\boldsymbol{q}\nu}V$ is the potential originating in response to the perturbation of the lattice with the ionic displacements of a phonon mode $\boldsymbol{q}\nu$. We define g^0 here as a quantity smoother to interpolate than g, as will be discussed below.

In this initial version, Phoebe interfaces with QE [24] in order to obtain values of the electronic energies and the electron–phonon coupling. In principle, *ab-initio* codes can directly compute electronic properties at any wavevector $\boldsymbol{k}$. The calculation of transport properties using momentum-space integrals typically requires very fine sampling of $\boldsymbol{k}$ throughout the BZ, making a direct first-principles calculation of every state unfeasible in most cases. To circumvent this issue, transport calculations rely on an interpolation scheme, in which QE computes electronic properties on a coarse grid of wavevectors, $\{\boldsymbol{k}_{\text{co}}\}$, and Phoebe interpolates these properties to an arbitrary fine mesh of wavevectors $\{\boldsymbol{k}_{fi}\}$.

We implemented the Wannier interpolation technique, interfacing Phoebe with Wannier90 [30] for the construction of maximally localized Wannier functions (MLWF) [31, 32]. The Wannier function $|\boldsymbol{R}m\rangle$ computed by Wannier90 is defined as

$$|\boldsymbol{R}m\rangle = \sum_{b\boldsymbol{k}_{\text{co}}} e^{-i\boldsymbol{k}_{\text{co}}\cdot\boldsymbol{R}} U_{\boldsymbol{k}_{\text{co}},mb}|\psi_{\boldsymbol{k}_{\text{co}}b}\rangle \ , \tag{8}$$

where m is a Wannier function index, and U is a rotation matrix. To briefly explain the MLWF procedure, we first recall that the wavefunction $|\psi_{\boldsymbol{k}b}\rangle$ is not unique, as it has a gauge degree of freedom: the wavefunction can be multiplied by an arbitrary factor $e^{i\phi(\boldsymbol{k},b)}$. In practice, the phase $\phi(\boldsymbol{k},b)$ is a random result of the numerical diagonalization of a Hamiltonian matrix, and depends unpredictably on the linear algebra algorithm implementation. As a result of this phase arbitrariness, the Wannier function are not uniquely defined. The MLWF procedure aims at reducing this arbitrariness, by finding for a rotation matrix U such that the spread of the Wannier function $|\boldsymbol{R}m\rangle$ is minimized.

The interpolation of electronic energies proceeds as follows [32]: we first compute the Hamiltonian matrix in the basis of Wannier functions $\langle\boldsymbol{0}n|H|\boldsymbol{R}m\rangle$ as

$$\langle\boldsymbol{0}n|H|\boldsymbol{R}m\rangle = \frac{1}{N_k}\sum_{\boldsymbol{k}_{\text{co}},bb'} e^{-i\boldsymbol{k}_{\text{co}}\cdot\boldsymbol{R}} U_{\boldsymbol{k}_{\text{co}},nb}[\delta_{bb'}\epsilon_{\boldsymbol{k}_{\text{co}}b}]U^{\dagger}_{\boldsymbol{k}_{\text{co}},mb'} \ , \tag{9}$$

where N_k is the total number of wavevectors in the coarse grid $\{\boldsymbol{k}_{\text{co}}\}$. From the Hamiltonian in the Wannier representation, we interpolate electronic energies at any arbitrary wavevector $\boldsymbol{k}$ by computing a Fourier back-transform

$$H^W_{\boldsymbol{k},nm} = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}}\langle\boldsymbol{0}n|H|\boldsymbol{R}m\rangle \ , \tag{10}$$

followed by a diagonalization of the resulting matrix

$$H^B_{\boldsymbol{k},bb'} = [U^\dagger_{\boldsymbol{k}} H^W_{\boldsymbol{k}} U_{\boldsymbol{k}}]_{bb'} = \delta_{bb'}\epsilon_{\boldsymbol{k}b}\,, \tag{11}$$

which provides an interpolation of the electronic energies $\epsilon_{\boldsymbol{k}b}$ and rotation matrix $U_{\boldsymbol{k}}$ at any desired wavevector.

The electron–phonon coupling can be interpolated in a conceptually similar way to the electronic energies. Following the approach of [33], we first note that the matrix g^0 is smoother than g (as defined in equation (7)) in the reciprocal space, since it removes a dependence on the phonon dispersion relation, and g^0 is thus more suitable for an interpolation algorithm. The matrix $g^0(\boldsymbol{k}_{\rm co}, \boldsymbol{q}_{\rm co})$ (with band indices omitted for simplicity) is computed by QE on a coarse grid of wavevectors and is then transformed by Phoebe to the Wannier representation as

$$g^0(\boldsymbol{R}_{\rm e}, \boldsymbol{R}_{\rm p}) = \frac{1}{N_q N_k} \sum_{\boldsymbol{k}_{\rm co}\boldsymbol{q}_{\rm co}} e^{-i\boldsymbol{k}_{\rm co}\cdot\boldsymbol{R}_{\rm e} - i\boldsymbol{q}_{\rm co}\cdot\boldsymbol{R}_{\rm p}} U^\dagger_{\boldsymbol{k}_{\rm co}+\boldsymbol{q}} g(\boldsymbol{k},\boldsymbol{q}) U_{\boldsymbol{k}_{\rm co}} z^{-1}_{\boldsymbol{q}_{\rm co}}\,, \tag{12}$$

where N_q is the number of phonon wavevectors in the coarse sampling, z is the phonon eigenvector introduced in equation (4), and $\boldsymbol{R}_{\rm e}$ ($\boldsymbol{R}_{\rm p}$) is the set of Bravais lattice vectors associated to the electron (phonon) Fourier transform of the set $\{\boldsymbol{k}_{\rm co}\}$ ($\{\boldsymbol{q}_{\rm co}\}$). Once this transformation is performed, the electron–phonon matrix is finally interpolated to arbitrary wavevectors $\boldsymbol{k}$ and $\boldsymbol{q}$ as

$$g^0(\boldsymbol{k},\boldsymbol{q}) = \frac{1}{N_{\rm e}} \sum_{\boldsymbol{R}_{\rm e}\boldsymbol{R}_{\rm p}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}_{\rm e} + i\boldsymbol{q}\cdot\boldsymbol{R}_{\rm p}} U_{\boldsymbol{k}+\boldsymbol{q}} g^0(\boldsymbol{R}_{\rm e}, \boldsymbol{R}_{\rm p}) U^\dagger_{\boldsymbol{k}} z_{\boldsymbol{q}}\,, \tag{13}$$

where the U and z matrices at the interpolating wavevectors can be found by constructing the dynamical matrix at the desired $\boldsymbol{q}$ wavevector and by interpolating the electronic energy at wavevector $\boldsymbol{k}$. As a result of this interpolation procedure, one can accurately sample the the electron–phonon coupling over the BZ and compute transport properties.

Similar to the interpolation of the three-phonon coupling, the interpolation of the electron–phonon coupling is a computationally expensive effort due to the large number of indices and the sheer number of interpolating wavevectors often needed to converge transport integrals. The implementation of the coupling interpolation for both electron–phonon and three-phonon coupling will be discussed together in detail in section 5.5, as equations (5) and (13) are similar from a mathematical point of view.

Finally, polar corrections to the electron–phonon coupling are implemented in Phoebe following the procedure outlined in [34]. The coupling is decomposed into short- and long-range contributions,

$$g_{bb'\nu}(\boldsymbol{k},\boldsymbol{q}) = g^S_{bb'\nu}(\boldsymbol{k},\boldsymbol{q}) + g^L_{bb'\nu}(\boldsymbol{k},\boldsymbol{q}), \tag{14}$$

where the long-range contribution is given by

$$g^L_{b'b\nu}(\boldsymbol{k},\boldsymbol{q}) = i\frac{4\pi}{V}\frac{e^2}{4\pi\epsilon_0} \sum_s \left(\frac{\hbar}{2N_q m_s \omega_{\boldsymbol{q}\nu}}\right)^{1/2} \sum_{\boldsymbol{G}\neq\boldsymbol{q}} \frac{(\boldsymbol{q}+\boldsymbol{G})\cdot\boldsymbol{Z}^*_s\cdot\boldsymbol{z}^s_{\boldsymbol{q}\nu}}{(\boldsymbol{q}+\boldsymbol{G})\cdot\boldsymbol{\epsilon}^\infty\cdot(\boldsymbol{q}+\boldsymbol{G})} \left\langle \psi_{\boldsymbol{k}+\boldsymbol{q},b'} \middle| e^{i(\boldsymbol{q}+\boldsymbol{G})\cdot\boldsymbol{r}} \middle| \psi_{\boldsymbol{k}+\boldsymbol{q},b} \right\rangle, \tag{15}$$

and the short-range contribution is calculated on the coarse grid by subtracting the long-range term from the total electron–phonon coupling calculated by the *ab-initio* code. Here, $\boldsymbol{Z}^*_s$ is the Born effective charge tensor of atom s, and ε_0 and $\boldsymbol{\epsilon}^\infty$ are the low- and high-frequency permittivities.

3.3. Velocity

For charge carriers, we compute the matrix elements of the velocity operator v as the derivative with respect to the wavevector of the non-interacting Hamiltonian [35]

$$v_{\alpha\boldsymbol{k},bb'} = \sum_{mn} U^\dagger_{\boldsymbol{k},mb} \frac{\partial H^W_{\boldsymbol{k},mn}}{\partial k_\alpha} U_{\boldsymbol{k},nb'} \approx \frac{1}{2\delta_\alpha} \sum_{mn} U^\dagger_{\boldsymbol{k},mb} \left(H^W_{\boldsymbol{k}+\boldsymbol{\delta}_\alpha,mn} - H^W_{\boldsymbol{k}-\boldsymbol{\delta}_\alpha,mn}\right) U_{\boldsymbol{k},nb'}\,, \tag{16}$$

where H^W is the Hamiltonian matrix in the Bloch representation computed at $\boldsymbol{k}$ with the Wannier interpolation technique described above, and δ_α is a small displacement vector along Cartesian direction α. If the eigenvalue $\epsilon_{\boldsymbol{k}b}$ is non-degenerate, the diagonal matrix element $v_{\alpha\boldsymbol{k},bb}$ represents the electronic group velocity. However if $\epsilon_{\boldsymbol{k}b}$ is a degenerate eigenvalue, we cannot readily identify the velocity matrix element $v_{\alpha\boldsymbol{k},bb}$ with the group velocity: in this case, we select the degenerate subspace of the matrix $v_{\alpha\boldsymbol{k}}$ and diagonalize it. After this subspace diagonalization, the new diagonal matrix elements represent the group velocity of the degenerate state.

The definition of the phonon velocity is similar to that of electrons, allowing us to use the same code for both velocity calculations. Simply, the matrix H is replaced with the square root of the dynamical matrix $\sqrt{D_{\boldsymbol{q}}}$ and the rotation matrix U is replaced with the phonon eigenvector z [27].

4. Transport methods

Moving on to the transport properties, we show in this section how the microscopic properties described above can be used to evaluate macroscopic transport coefficients.

4.1. Phonon Boltzmann transport equation

Given the microscopic quantities discussed in the previous sections, we discuss now how to obtain the lattice thermal conductivity. We describe lattice thermal transport in the framework of the phonon BTE [23, 36]. This semiclassical model describes the motion of a phonon (wavepacket) through the crystal. A phonon state $(\boldsymbol{q},\nu)$ at thermal equilibrium is populated with $\bar{n}_{\boldsymbol{q}\nu}$ phonon quanta, where $\bar{n}_{\boldsymbol{q}\nu} = 1/(e^{\hbar\omega_{\boldsymbol{q}\nu}/k_{\mathrm{B}}T} - 1)$ is the Bose–Einstein distribution. The application of a thermal gradient ∇T to the crystal forces the system out-of-equilibrium, so that each phonon state is then populated with an out-of-equilibrium population $n_{\boldsymbol{q}\nu}$. The phonon BTE provides a way to find such out-of-equilibrium population and consists of a balance between two terms:

$$\frac{\partial \bar{n}_{\boldsymbol{q}\nu}}{\partial T} v_{\alpha\boldsymbol{q}\nu} \nabla_\alpha T = -\frac{1}{V N_q} \sum_{\boldsymbol{q}'\nu'} \Omega_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} \delta n_{\alpha\boldsymbol{q}'\nu'} \,. \tag{17}$$

The left-hand side is the diffusion operator, and describes the phonon drift in presence of a thermal gradient ∇T, while the right-hand side describes how scattering changes the population number of a phonon state. Here, V is the crystal unit cell volume, $\Omega_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'}$ is the scattering matrix and $\delta n_{\boldsymbol{q}\nu} = n_{\boldsymbol{q}\nu} - \bar{n}_{\boldsymbol{q}\nu}$ is the deviation from equilibrium population. For computational convenience, we also work with a rescaled scattering matrix A defined as $A_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} = \Omega_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} \bar{n}_{\boldsymbol{q}'\nu'}(\bar{n}_{\boldsymbol{q}'\nu'} + 1)$. The phonon scattering matrix is built taking into account the three-phonon interaction, phonon–isotope, phonon–boundary and phonon–electron scattering [27] as

$$\begin{aligned} A_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} = &\left[\sum_{\boldsymbol{q}''\boldsymbol{q}'''\nu''\nu'''} \left(P^{\boldsymbol{q}''\nu''}_{\boldsymbol{q}\nu,\boldsymbol{q}'''\nu'''} + \frac{P^{\nu}_{\boldsymbol{q}'''\nu''',\boldsymbol{q}''\nu''}}{2} \right) + \sum_{\boldsymbol{q}''\nu''} P^{\mathrm{isot}}_{\boldsymbol{q}\nu,\boldsymbol{q}''\nu''} + P^{\mathrm{be}}_{\boldsymbol{q}\nu} + P^{\mathrm{el-ph}}_{\boldsymbol{q}\nu} \right] \delta_{\nu,\nu'} \delta_{\boldsymbol{q},\boldsymbol{q}'} \\ &- \sum_{\boldsymbol{q}''\nu''} \left(P^{\boldsymbol{q}'\nu'}_{\boldsymbol{q}\nu,\boldsymbol{q}''\nu''} - P^{\boldsymbol{q}''\nu''}_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} + P^{\boldsymbol{q}\nu}_{\boldsymbol{q}'\nu',\boldsymbol{q}''\nu''} \right) + P^{\mathrm{isot}}_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} \,, \end{aligned} \tag{18}$$

where we introduced the three-phonon scattering rates for coalescence events as

$$P^{\boldsymbol{q}''\nu''}_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} = \frac{2\pi}{N_q \hbar^2} \sum_{\boldsymbol{G}} \left| V^{(3)}(\boldsymbol{q}j, \boldsymbol{q}'\nu', -\boldsymbol{q}''\nu'') \right|^2 \bar{n}_{\boldsymbol{q}j} \bar{n}_{\boldsymbol{q}'\nu'} (\bar{n}_{\boldsymbol{q}''\nu''} + 1) \delta_{\boldsymbol{q}+\boldsymbol{q}'-\boldsymbol{q}'',\boldsymbol{G}} \delta(\hbar\omega_{\boldsymbol{q}\nu} + \hbar\omega_{\boldsymbol{q}'\nu'} - \hbar\omega_{\boldsymbol{q}''\nu''}) \,, \tag{19}$$

while the scattering rate for phonon decay is

$$\begin{aligned} P^{\boldsymbol{q}'\nu',\boldsymbol{q}''\nu''}_{\boldsymbol{q}\nu} = &\frac{2\pi}{N_q \hbar^2} \sum_{\boldsymbol{G}} \left| V^{(3)}(\boldsymbol{q}\nu, -\boldsymbol{q}'\nu', -\boldsymbol{q}''\nu'') \right|^2 \bar{n}_{\boldsymbol{q}\nu} (\bar{n}_{\boldsymbol{q}'\nu'} + 1)(\bar{n}_{\boldsymbol{q}''\nu''} + 1) \delta_{\boldsymbol{q}-\boldsymbol{q}'-\boldsymbol{q}'',} \\ &\times \boldsymbol{G} \delta(\hbar\omega_{\boldsymbol{q}\nu} - \hbar\omega_{\boldsymbol{q}'\nu'} - \hbar\omega_{\boldsymbol{q}''\nu''}) \,, \end{aligned} \tag{20}$$

where $\boldsymbol{G}$ is a reciprocal lattice vector, N_q is the number of phonon wavevectors for the BZ integration, and $V^{(3)}(\boldsymbol{q}\nu, \boldsymbol{q}'\nu', \boldsymbol{q}''\nu'')$ is the three-phonon coupling strength defined in equation (5).

The phonon–electron interaction contributes to the scattering matrix with a term on the diagonal part [37]

$$P^{\mathrm{el-ph}}_{\boldsymbol{q}\nu} = -\frac{2\pi}{N_{\boldsymbol{k}}\hbar} \bar{n}_{\boldsymbol{q}\nu}(\bar{n}_{\boldsymbol{q}\nu} + 1) \sum_{\boldsymbol{k}bb'} |g_{bb'\nu}(\boldsymbol{k},\boldsymbol{q})|^2 \left(f_{\boldsymbol{k}b} - f_{\boldsymbol{k}+\boldsymbol{q}b'} \right) \delta\left(\varepsilon_{\boldsymbol{k}b} - \varepsilon_{\boldsymbol{k}+\boldsymbol{q}b'} - \omega_{\boldsymbol{q}\nu} \right) \,, \tag{21}$$

with $f_{\boldsymbol{k}b} - f_{\boldsymbol{k}+\boldsymbol{q}b'} \approx \frac{\partial f_{\boldsymbol{k}b}}{\partial \varepsilon_{\boldsymbol{k}b}} \hbar\omega_{\boldsymbol{q}\nu}$ for numerical stability. Additionally, the scattering rate for phonon–isotope scattering is [38]

$$P^{\mathrm{isot}}_{\boldsymbol{q}\nu,\boldsymbol{q}'\nu'} = \frac{\pi}{2N_q} \omega_{\boldsymbol{q}\nu}\omega_{\boldsymbol{q}'\nu'} \left[\bar{n}_{\boldsymbol{q}\nu}\bar{n}_{\boldsymbol{q}'\nu'} + \frac{\bar{n}_{\boldsymbol{q}\nu} + \bar{n}_{\boldsymbol{q}'\nu'}}{2} \right] \sum_s g^{\mathrm{isot}} \left| \sum_\alpha z^{s\alpha *}_{\boldsymbol{q}\nu} \cdot z^{s\alpha}_{\boldsymbol{q}'\nu'} \right|^2 \delta(\omega_{\boldsymbol{q}\nu} - \omega_{\boldsymbol{q}'\nu'}) \,, \tag{22}$$

where the coupling strength g_s^{isot} depends on the variance of masses as

$$g_s^{\text{isot}} = \frac{1}{\langle m_s \rangle^2} \sum_i f_{is}(m_{is} - \langle m_s \rangle)^2 \,, \tag{23}$$

where m_{is} is the atomic mass of atom of species s and isotope i, f_{is} is the abundance of the isotope (which by default is set to natural abundances [39]) and $\langle m_s \rangle$ is the average atomic mass. Finally, the boundary scattering rate is described with the term

$$P^{\text{be}}_{\boldsymbol{q}\nu} = \frac{v_{\boldsymbol{q}\nu}}{L} \bar{n}_{\boldsymbol{q}\nu}(\bar{n}_{\boldsymbol{q}\nu} + 1), \tag{24}$$

where L is the user-specified macroscopic crystal size.

Using the first-principles phonon properties discussed in section 3.1 it is possible to construct the phonon BTE and in particular the scattering matrix. In section 4.3, we will discuss how to solve the BTE to obtain the out-of-equilibrium population $n_{\boldsymbol{q}\nu}$. Assuming that $n_{\boldsymbol{q}\nu}$ is found, the thermal conductivity tensor $\kappa_{\alpha\beta}$ is computed as

$$\kappa_{\alpha\beta} = \frac{1}{N_q V} \sum_{\boldsymbol{q}\nu} \hbar\omega_{\boldsymbol{q}} v_{\alpha\boldsymbol{q}\nu} \left(\frac{\delta n_{\boldsymbol{q}\nu}}{\nabla T} \right)_\beta \,, \tag{25}$$

where $v_{\alpha\boldsymbol{q}\nu}$ is the phonon group velocity.

4.2. Electron Boltzmann transport equation

The semiclassical description of electron transport shares several similarities with the phonon transport theory. In the phonon case, we solved for out-of-equilibrium phonon populations. Now, for the case of electron transport, we deal with the problem of finding the out-of-equilibrium occupation number of a charge carrier. At equilibrium, an electronic state $(\boldsymbol{k}, b)$ is occupied according to the Fermi–Dirac distribution $f_{\boldsymbol{k}b} = 1/(e^{(\epsilon_{\boldsymbol{k}b}-\mu)/k_B T} + 1)$, where μ is the chemical potential and $\epsilon_{\boldsymbol{k}b}$ is the electronic energy. When we apply a thermal gradient to a material, we can expect to induce the diffusion of carriers just as in the phonon case, resulting in an out-of-equilibrium population $f_{\boldsymbol{k}b}$. However, each carrier also possesses a charge e, and therefore can diffuse also when an external electric field $\boldsymbol{E}$ is applied to the material. As we are only concerned with the linear response to the perturbation induced by the thermal gradient or the electric field, the electronic BTE consists of two separate problems for the two different perturbations $\boldsymbol{E}$ and ∇T. The BTE describing the response to an electric field in direction α is

$$\frac{\partial \bar{f}_{\boldsymbol{k}b}}{\partial \epsilon} v_{\alpha\boldsymbol{k}b} e E_\alpha = -\frac{1}{N_k V} \sum_{\boldsymbol{k}'b'} \Omega_{\boldsymbol{k}b,\boldsymbol{k}'b'} \delta f_{\alpha\boldsymbol{k}'b'} \,, \tag{26}$$

where $\delta f_{\boldsymbol{k}b} = f_{\boldsymbol{k}b} - \bar{f}_{\boldsymbol{k}b}$ is the deviation from the equilibrium population, $v_{\alpha\boldsymbol{k}b}$ is the electronic group velocity in direction α, and Ω is the electronic scattering matrix. The BTE describing the response to a thermal gradient perturbation closely resembles the phonon case and is

$$\frac{\partial \bar{f}_{\boldsymbol{k}b}}{\partial T} v_{\alpha\boldsymbol{k}b} \nabla_\alpha T = -\frac{1}{N_k V} \sum_{\boldsymbol{k}'b'} \Omega_{\boldsymbol{k}b,\boldsymbol{k}'b'} \delta f_{\alpha\boldsymbol{k}'b'} \,. \tag{27}$$

When taking the electron–phonon scattering into account, the electron scattering matrix Ω can be computed as [40]

$$\begin{aligned} \Omega_{\boldsymbol{k}b,\boldsymbol{k}'b'} = {} & \frac{1}{\tau_{\boldsymbol{k}b}} \delta_{\boldsymbol{k}b,\boldsymbol{k}'b'} + (1 - \delta_{\boldsymbol{k}b,\boldsymbol{k}'b'}) \frac{2\pi}{N_k V} \sum_{\boldsymbol{q}\nu} |g_{bb'\nu}(\boldsymbol{k}, \boldsymbol{k}')|^2 \\ & \times \Big[(1 - \bar{f}_{\boldsymbol{k}'b'} + \bar{n}_{\boldsymbol{q}\nu}) \delta(\epsilon_{\boldsymbol{k}b} - \hbar\omega_{\boldsymbol{q}\nu} - \epsilon_{\boldsymbol{k}'b'}) + (\bar{f}_{\boldsymbol{k}'b'} + \bar{n}_{\boldsymbol{q}\nu}) \delta(\epsilon_{\boldsymbol{k}b} + \hbar\omega_{\boldsymbol{q}\nu} - \epsilon_{\boldsymbol{k}'b'}) \Big] \delta(\boldsymbol{k} - \boldsymbol{k}' + \boldsymbol{q}) \,, \end{aligned} \tag{28}$$

where the lifetime $\tau_{\boldsymbol{k}b}$ is

$$\begin{aligned} \frac{1}{\tau_{\boldsymbol{k}b}} = {} & \frac{2\pi}{N_k V} \sum_{\boldsymbol{k}'b'\boldsymbol{q}\nu} |g_{bb'\nu}(\boldsymbol{k}, \boldsymbol{k}')|^2 \Big[(1 - f_{\boldsymbol{k}'b'} + \bar{n}_{\boldsymbol{q}\nu}) \delta(\epsilon_{\boldsymbol{k}b} - \hbar\omega_{\boldsymbol{q}\nu} - \epsilon_{\boldsymbol{k}'b'}) \\ & + (f_{\boldsymbol{k}'b'} + \bar{n}_{\boldsymbol{q}\nu}) \delta(\epsilon_{\boldsymbol{k}b} + \hbar\omega_{\boldsymbol{q}\nu} - \epsilon_{\boldsymbol{k}'b'}) \Big] \delta(\boldsymbol{k} - \boldsymbol{k}' + \boldsymbol{q}) \end{aligned} \tag{29}$$

where $g_{bb'\nu}(\boldsymbol{k},\boldsymbol{k}')$ is the coupling strength of the electron–phonon interaction. Electron scattering against surfaces or boundaries can be taken into account by adding to A a term of the form $\frac{v_{\alpha k b}}{L}\delta_{\boldsymbol{k},\boldsymbol{k}'}\delta_{b,b'}$, where L is a characteristic length.

We note that the scattering matrix Ω can be brought to a symmetric form with the scaling

$$\tilde{\Omega}_{\boldsymbol{k}b,\boldsymbol{k}'b'} = \Omega_{\boldsymbol{k}b,\boldsymbol{k}'b'}\sqrt{\frac{\bar{f}_{\boldsymbol{k}'b'}(1-\bar{f}_{\boldsymbol{k}'b'})}{\bar{f}_{\boldsymbol{k}b}(1-\bar{f}_{\boldsymbol{k}b})}} \tag{30}$$

it can readily be checked that the matrix $\tilde{\Omega}$ is symmetric (i.e. $\tilde{\Omega}_{\boldsymbol{k}b,\boldsymbol{k}'b'} = \tilde{\Omega}_{\boldsymbol{k}'b',\boldsymbol{k}b}$), since the matrix elements are

$$\begin{aligned}\tilde{\Omega}_{\boldsymbol{k}b,\boldsymbol{k}'b'} = \frac{1}{\tau_{\boldsymbol{k}b}}\delta_{\boldsymbol{k}b\boldsymbol{k}'b'} + (1-\delta_{\boldsymbol{k}b,\boldsymbol{k}'b'})\frac{2\pi}{N_k V}\sum_{\boldsymbol{k}'b'\boldsymbol{q}\nu}|g_{bb'\nu}(\boldsymbol{k},\boldsymbol{k}')|^2\left[\delta(\epsilon_{\boldsymbol{k}b}-\hbar\omega_{\boldsymbol{q}\nu}-\epsilon_{\boldsymbol{k}'b'}) + \delta(\epsilon_{\boldsymbol{k}b}+\hbar\omega_{\boldsymbol{q}\nu}-\epsilon_{\boldsymbol{k}'b'})\right] \\ \times\,\delta(\boldsymbol{k}-\boldsymbol{k}'+\boldsymbol{q})\frac{1}{2\sinh(\frac{\hbar\omega_{\boldsymbol{q}\nu}}{2k_B T})}.\end{aligned} \tag{31}$$

This symmetric expression for the scattering matrix is more suitable for implementing linear algebra solvers to the BTE, and has been developed in analogy to the rescaling of the phonon scattering matrix described in [41].

The solution to the electronic BTE is discussed in section 4.3. Solving the BTE provides a way of calculating the linear response of the electronic out-of-equilibrium population in response to an electric field $\delta f^{E\alpha}$ or a thermal gradient $\delta f^{T\alpha}$ in direction α, defined as $\delta f_{\alpha\boldsymbol{k}b} = \delta f^{E\alpha}E^{\alpha}$ and $\delta f_{\alpha\boldsymbol{k}b} = \delta f^{T\alpha}\nabla_\alpha T$ respectively. These perturbations induce both a charge flux $\boldsymbol{J}$ and an energy flux $\boldsymbol{Q}$ defined as

$$\boldsymbol{J} = \frac{e}{N_k V}\sum_{\boldsymbol{k}b} v_{\alpha\boldsymbol{k}b}\delta f_{\boldsymbol{k}b}\,, \tag{32}$$

$$\boldsymbol{Q} = \frac{1}{N_k V}\sum_{\boldsymbol{k}b} v_{\alpha\boldsymbol{k}b}(\epsilon_{\boldsymbol{k}b}-\mu)\delta f_{\boldsymbol{k}b}\,. \tag{33}$$

These fluxes can be compared with the Onsager relations

$$\boldsymbol{J} = L_{\mathrm{EE}}\boldsymbol{E} + L_{\mathrm{ET}}\nabla T\,, \tag{34}$$

$$\boldsymbol{Q} = L_{\mathrm{TE}}\boldsymbol{E} + L_{\mathrm{TT}}\nabla T\,, \tag{35}$$

from which we can define and compute the electronic transport coefficients, such as the electrical conductivity σ

$$\sigma = L_{\mathrm{EE}}\,, \tag{36}$$

the drift mobility $\mu = \frac{\sigma}{ed}$, where d is the carrier concentration, the Seebeck coefficient S

$$S = -L_{\mathrm{EE}}^{-1}L_{\mathrm{ET}}\,, \tag{37}$$

and electronic thermal conductivity tensor κ_{el}

$$\kappa_{\mathrm{el}} = L_{\mathrm{TT}} - L_{\mathrm{TE}}L_{\mathrm{EE}}^{-1}L_{\mathrm{ET}}\,. \tag{38}$$

With these definitions, we are finally ready to predict the thermal and electronic transport properties of crystals.

4.3. BTE solvers

The similarity between electron and phonon BTEs allows us to readily implement solvers which can be applied generically to either the electron or phonon cases. We first look for a solution that is linear in the external perturbation of the system, i.e. an out-of-equilibrium population in the form $f_\lambda \to f_\lambda E$ or $f_\lambda \to f_\lambda \nabla T$, with λ as a combined index labeling the relevant band and momentum state (phonon or electron). Under this linear approximation, we write the BTE (equations (17), (26) and (27)) in the more concise form of a linear algebra problem

$$\sum_{\lambda'} \Omega_{\lambda\lambda'} f_{\lambda'} = b_\lambda \,, \tag{39}$$

where b is the diffusion term, Ω is the scattering matrix, all chosen appropriately according to the transport problem under consideration. Note that for the phonon transport problem, Phoebe uses at times a symmetrized scattering matrix $\tilde{\Omega}$ in place of Ω (see [41]) or the scattering matrix A (see [27]).

On the surface, the BTE appears as a trivial linear algebra problem; however, there are several complexities which arise when solving the BTE. First, the scattering matrix as in equations (28) and (18) cannot be directly inverted to solve the problem [42] due to the presence of zero eigenvalues (arising from energy conservation and/or charge conservation).

The second challenge is the sheer size of the problem: in order to obtain converged results, the summation over the BZ (λ includes a wavevector index) must be performed for an extremely fine mesh of wavevectors. It is often necessary to use more than 10^5 wavevectors. Therefore, Phoebe contains a number of different solvers which operate at various levels of complexity and computational cost, each with distinct advantages and disadvantages, and lets the user decide which one is best suited for a particular material. Taking advantage of the similarity between charge and heat transport, the solvers have been implemented for both the electron and phonon BTE.

4.3.1. Constant relaxation time approximation

The constant relaxation time approximation (CRTA) is a rudimentary, approximate solution to the BTE. In this case, the scattering matrix is replaced with a single scalar

$$\Omega_{\lambda\lambda'} \approx \frac{1}{\tau} \,, \tag{40}$$

where τ is a user-specified constant relaxation time. Most of the complexity of the BTE is removed and the BTE is readily solved with the out-of-equilibrium population $f_{\lambda'} \approx \tau b_\lambda$. The CRTA may provide reasonable results for the Seebeck coefficient, as this often has a weak dependence on the scattering rate, and may be used for fast estimates of trends in other transport properties. However, the choice of τ is largely arbitrary and in many crystals a more detailed description of scattering processes must be included to quantitatively predict transport properties or even capture qualitative transport anomalies [43, 44].

4.3.2. Relaxation time approximation

The relaxation time approximation (RTA) is another approximate solution to the BTE, in which only the diagonal elements of the scattering matrix are retained so that

$$\Omega_{\lambda\lambda'} \approx \Omega_{\lambda\lambda'}\delta_{\lambda\lambda'} = \delta_{\lambda\lambda'}\frac{1}{\tau_\lambda} \,, \tag{41}$$

where τ_λ is a carrier's lifetime, and the off-diagonal matrix elements are neglected. With this approximation, the out-of-equilibrium population is readily found as $f_{\lambda'} \approx b_\lambda \tau_\lambda$. The RTA is a fast and memory-efficient BTE solver, and can be used when more accurate solvers become prohibitively expensive. For electronic transport, it often shows good agreement with more accurate solvers. On the other hand, there are reported cases for which the RTA approximation has failed, particularly for materials where the conservation of crystal momentum is of the utmost relevance, such as in hydrodynamic phonon transport regimes [45–47]. In these cases, one must use a solver which utilizes the complete scattering matrix, such as the solvers listed below.

4.3.3. Iterative solver

The iterative BTE solver [48–50] constructs the out-of-equilibrium populations with a geometric series. First, we decompose the scattering matrix into its diagonal terms Ω^{out} and off-diagonal components Ω^{in}. A solution is found by iteratively correcting the RTA solution, $f^{\text{RTA}} = \frac{1}{\Omega^{\text{out}}} b = \tau b$, as

$$f^i = \sum_{j=0}^{i} \left(-\frac{1}{\Omega^{\text{out}}}\Omega^{\text{in}} \right)^j \frac{1}{\Omega^{\text{out}}} b \,, \tag{42}$$

where j is an iteration index that can be stopped when transport coefficients are converged. Note also that because Ω^{out} is a diagonal matrix, its inverse is trivially constructed.

Additionally, we implemented crystal symmetries in this solver to further accelerate simulations (see section 4.8 below). However, it is worth pointing out that the iterative solver can only converge as a geometric series, i.e. only if the absolute value of the determinant $|\det(\frac{1}{\Omega^{\text{out}}}\Omega^{\text{in}})|$ is smaller than 1. If the convergence criterion is not satisfied, the algorithm may diverge. The iterative method can be used both by opting to store the scattering matrix in memory, which allows a fast solution once the scattering matrix is built but at the expense of a large usage of memory, or not storing Ω in memory, so that we lower memory requirements but require to compute the action of the scattering matrix $\Omega \cdot f$ several times.

4.3.4. Variational solver

The variational BTE solver [27, 40] operates by minimizing the following quadratic functional of the carrier population f

$$\mathcal{F}[f] = \frac{1}{2} f \cdot A \cdot f - b \cdot f . \tag{43}$$

Since the scattering matrix is semi-positive definite [27, 40, 51] (a property related to the ever increasing value of entropy [23]), the functional $\mathcal{F}$ has a single minimum which coincides with the solution of the BTE. We implemented a conjugate gradient algorithm [27] that minimizes the functional $\mathcal{F}$ and finds the exact solution to the BTE. In contrast to the iterative solver, the variational solver is guaranteed to converge to the solution provided that the scattering matrix is semi-positive definite. Like the iterative solver, the user can choose to store the matrix in memory to quickly build the solution, or not, so that the solution is found more slowly but with smaller memory requirements.

4.3.5. Relaxons solver

The last solver is based on a direct diagonalization of the BTE [41, 52]. The scattering matrix is diagonalized with a ScaLAPACK solver, and we compute the complete set of eigenvalues and eigenvectors

$$\sum_{\lambda'} \Omega_{\lambda\lambda'} \theta^{\alpha}_{\lambda'} = \frac{1}{\tau_\alpha} \theta^{\alpha}_{\lambda} , \tag{44}$$

where α is an eigenvalue index, τ_α is the inverse eigenvalue and $\theta^{\alpha}_{\lambda}$ the eigenvector. Thanks to the completeness of the basis set of eigenvectors, we expand the out-of-equilibrium population as

$$f_\lambda = \sum_\alpha f_\alpha \theta^{\alpha}_{\lambda} , \tag{45}$$

where we introduced auxiliary coefficients f_α. Next, we write the BTE in the eigenvector (relaxons) basis as

$$\sum_\lambda b_\lambda \theta^{\alpha}_{\lambda} = \frac{1}{\tau_\alpha} f_\alpha , \tag{46}$$

and note that in this form we can analytically solve the BTE by substituting $f_\alpha = \tau_\alpha \sum_\lambda b_\lambda \theta^{\alpha}_{\lambda}$ in equation (45).

This last solver has the benefit of always providing the exact solution, as it is not based on any iterative scheme and therefore has none of the related numerical instabilities. Furthermore, this analytical solution to the BTE allows for further physical insights, such as the calculation of viscosity [53] (see section 4.5). On the other hand, this solver has the longest time-to-solution for calculations of transport properties.

4.4. Electron–phonon averaged approximation

The electron–phonon averaged (EPA) approximation offers another approximate solution to the BTE using an average over the phonon energies and coarse-mesh electron–phonon matrix elements [54]. Although the Wannier function technique discussed in section 3.2 provides an accurate interpolation of the electronic properties, the number of fine-mesh momentum vectors needed to converge transport calculations can be very large. Furthermore, the construction of the MLWF is a challenging task, not yet fully automated (progress is being made in this direction [55]), and therefore is not always suitable for materials studies. Avoiding momentum space integration and Wannier interpolation altogether, EPA provides a way to estimate electron transport properties in a much faster and automatable way, suitable for rapid materials screening applications.

Consider a set of electronic energies $\epsilon(\boldsymbol{k}_{\text{co}})$ (at fixed band index) computed by a first-principles code on a coarse grid of wavevectors in the BZ. To interpolate the energies, we implemented a plane-wave based interpolation method [56], which fits an electronic band with the periodic function

$$\tilde{\epsilon}(\boldsymbol{k}) = \sum_{m=0}^{M-1} c_m e^{i\boldsymbol{k}\cdot\boldsymbol{R}_m} , \tag{47}$$

where $\boldsymbol{R}_m$ is a set of $m = 0, 1, \ldots, M-1$ Bravais lattice vectors chosen as all $\boldsymbol{R}_m$ such that $|\boldsymbol{R}_m| < R_{\text{cut}}$, where R_{cut} is a user-specified parameter, and c_m is an expansion coefficient. The expansion coefficients are found by optimizing a target error function against a set of training points. Further details can be found in [56]. This technique does not involve Wannier functions and is thus simpler to implement and automate. In practice, compared to the Wannier interpolation, the sampling of wavevectors $\{\boldsymbol{k}_{\text{co}}\}$ typically needs to be denser in order to converge transport results. With this interpolation procedure, it is possible to evaluate the electronic density of states $\rho(\epsilon) = \frac{1}{VN_k}\sum_{kb}\delta(\epsilon - \tilde{\epsilon}_b(\boldsymbol{k}))$, where $\tilde{\epsilon}_b(\boldsymbol{k})$ are the interpolated energies of each band b.

The next component of the EPA approach is a scheme for estimating the electron–phonon coupling. The idea underlying the EPA method is to replace momentum-space integrations of the BZ with integrations over the quasiparticle energies, simplifying some of the microscopic details of electronic transport to gain computational performance. The electron–phonon coupling is approximated as

$$|g^2_{bb'\nu}(\boldsymbol{k},\boldsymbol{q})|^2 \to g^2_\nu(\epsilon_1,\epsilon_2) , \tag{48}$$

which replaces a six-dimensional wavevector dependence on $\boldsymbol{k}$ and $\boldsymbol{q}$ with a two-dimensional dependence on electronic energies. Additionally, as g often has a weak energy dependence, it is possible to sample g^2 with just a few energy values. There are several ways to construct the function g^2 [54], and we implemented the moving-least-squares method [57], which constructs g^2 by minimizing the integral

$$\sum_{bb'}\int \mathrm{d}\boldsymbol{k}\int \mathrm{d}\boldsymbol{q}\left(|g_{bb'\nu}(\boldsymbol{k},\boldsymbol{q})|^2 - g^2_\nu(\epsilon_1,\epsilon_2)\right)^2 \exp\left(-\frac{(\epsilon_{\boldsymbol{k}b}-\epsilon_1)^2+(\epsilon_{\boldsymbol{k}b'}-\epsilon_2)^2}{2\sigma^2_{\text{gauss}}}\right) , \tag{49}$$

where σ_{gauss} is a user-specified smoothing scale parameter. The energy dependence on ε_1 and ε_2 is implemented numerically with a user-specified uniform sampling of an energy interval.

Once the coupling is constructed, the energy-dependent electronic lifetime is calculated as

$$\begin{aligned}\tau^{-1}(\epsilon,\mu,T) = \frac{2\pi V}{g_s\hbar}\sum_\nu \big\{ & g^2_\nu(\epsilon,\epsilon+\bar{\omega}_\nu))^2\left[n(\bar{\omega}_\nu,T)+f(\epsilon+\bar{\omega}_\nu,\mu,T)\right]\rho(\epsilon+\bar{\omega}_\nu) \\ & + g^2_\nu(\epsilon,\epsilon-\bar{\omega}_\nu))^2\left[n(\bar{\omega}_\nu,T)+1-f(\epsilon-\bar{\omega}_\mu,\mu,T)\right]\rho(\epsilon-\bar{\omega}_\nu)\big\} ,\end{aligned} \tag{50}$$

where $\bar{\omega}_\nu = \frac{1}{N_q}\sum_{\boldsymbol{q}}\omega_{\boldsymbol{q}\nu}$ is the average phonon frequency of the phonon branch ν, f and n are energy-dependent Fermi–Dirac and Bose–Einstein distributions respectively, and g_s is the spin-multiplicity (equal to 2 for non spin-polarized calculations).

Using this energy-dependent lifetime, the BTE is readily solved at the RTA level, and it is possible to evaluate the electronic transport coefficients using an energy-dependent approximation for equations (36)–(38), as discussed in greater details in [54]. Typically, the computational bottleneck for this method in Phoebe is the construction of the electronic density of states, which simply has a cost linear in the number of wavevectors used to sample the BZ.

4.5. Viscosity

In some materials with large conductivities, transport does not follow the conventional macroscopic diffusion equations such as Fourier's law for heat or Ohm's law for charge currents. Recent studies highlighted the necessity to complement conventional transport coefficients such as conductivity with hydrodynamic transport coefficients such as viscosity. Following [53], we implemented the phonon viscosity as defined from the response coefficient η,

$$\Pi^{ij} = \eta^{ijkl}\frac{\partial u^k}{\partial r^l} , \tag{51}$$

where i, j, k, l are Cartesian indices, u is the phonon drift velocity in direction k, $\boldsymbol{r}$ is the position, and Π is the crystal momentum flux $\Pi^{ij} = \frac{1}{VN_q}\sum_{\boldsymbol{q}\nu}\hbar q^i v^j \delta n^D_{\boldsymbol{q}\nu}$, where $\delta n^D_{\boldsymbol{q}\nu}$ is the out-of-equilibrium phonon population in response to a perturbation of the drift velocity $\boldsymbol{u}$. The out-of-equilibrium population can be found solving

a modified version of the BTE, as reported in full details in [53]. Finally, the phonon viscosity μ is found by symmetrizing η as

$$\mu^{ijkl} = \frac{\eta^{ijkl} + \eta^{ilkj}}{2} . \tag{52}$$

The phonon viscosity can be used to parametrize a generalization of the Fourier's law for describing heat transport. We also implemented the analogous expression for electronic viscosity, which is readily obtained by replacing phonon properties such as group velocity, equilibrium distribution and lifetimes with their electronic counterparts.

4.6. Transport with the Wigner distribution

In systems where carriers' energies overlap with each other, such as in strongly anharmonic systems [58] or narrow-gap semiconductors [59], it is important to include contributions to transport from the coupling of multiple bands, an effect which is not described by the BTE. To accurately describe these contributions, we implemented the Wigner transport corrections for phonon and electron transport properties. The starting point of this modeling is the Wigner distribution function of phonons $N_{\nu\nu'}(\boldsymbol{r},\boldsymbol{q},t)$ or electrons $F_{bb'}(\boldsymbol{r},\boldsymbol{k},t)$. The Wigner distribution cannot be interpreted anymore as a probability distribution, as is the case for electron and phonon population appearing in the BTE. Instead, N and F are quasiprobability distributions that describe the state of a quantum system.

It is possible to construct an approximate equation of motion for the Wigner distribution function. As in the spirit of the BTE, this expression aims to describe the out-of-equilibrium state of a quantum system. In particular, the equation of motion for the phonon Wigner distribution function N is

$$\frac{\partial N(\boldsymbol{r},\boldsymbol{q},t)}{\partial t} + i[\tilde{\Omega}(\boldsymbol{q}), N(\boldsymbol{r},\boldsymbol{q},t)] + \frac{1}{2}\{\boldsymbol{V}(\boldsymbol{q}), \cdot\nabla_{\boldsymbol{r}} N(\boldsymbol{r},\boldsymbol{q},t)\} = \left.\frac{\partial N(\boldsymbol{r},\boldsymbol{q},t)}{\partial t}\right|_{\text{coll}} , \tag{53}$$

where $\left.\frac{\partial N(\boldsymbol{r},\boldsymbol{k},t)}{\partial t}\right|_{\text{coll}}$ is the phonon scattering operator, $\tilde{\Omega}(\boldsymbol{q})$ is a matrix of energies with elements $\tilde{\Omega}_{\nu\nu'}(\boldsymbol{q}) = \delta_{\nu\nu'}\hbar\omega_{\boldsymbol{q}\nu}$, $[,]$ is a commutator and $\{,\}$ an anticommutator. The electronic version of the Wigner transport equation is

$$\frac{\partial F(\boldsymbol{r},\boldsymbol{k},t)}{\partial t} + i[\mathcal{E}(\boldsymbol{k}) + \boldsymbol{d}(\boldsymbol{k})\cdot\boldsymbol{E}, F(\boldsymbol{r},\boldsymbol{k},t)] + \frac{1}{2}\{\boldsymbol{V}(\boldsymbol{k}), \cdot\nabla_{\boldsymbol{r}} F(\boldsymbol{r},\boldsymbol{k},t)\} - e\boldsymbol{E}\cdot\nabla_{\boldsymbol{k}} F(\boldsymbol{r},\boldsymbol{k},t) = \left.\frac{\partial F(\boldsymbol{r},\boldsymbol{k},t)}{\partial t}\right|_{\text{coll}} , \tag{54}$$

where $\left.\frac{\partial F(\boldsymbol{r},\boldsymbol{k},t)}{\partial t}\right|_{\text{coll}}$ is the electron scattering operator, $\mathcal{E}(\boldsymbol{k})$ is a matrix of energies with elements $\mathcal{E}_{bb'}(\boldsymbol{k}) = \delta_{bb'}\epsilon_{\boldsymbol{k}b}$, and $\boldsymbol{d}_{bb'}(\boldsymbol{k})$ is the electronic dipole [59]. The scattering operator for electrons is

$$\left.\frac{\partial F(\boldsymbol{r},\boldsymbol{k},t)}{\partial t}\right|_{\text{coll}} = -\sum_{\boldsymbol{k}'b'} \Omega_{\boldsymbol{k}b,\boldsymbol{k}'b'} F_{b'b'}(\boldsymbol{r},\boldsymbol{k},t) - (1-\delta_{bb'})\frac{\tau_{\boldsymbol{k}b}^{-1} + \tau_{\boldsymbol{k}b'}^{-1}}{2} F_{bb'}(\boldsymbol{r},\boldsymbol{k},t) , \tag{55}$$

where we distinguished the scattering matrix acting on the diagonal components of the Wigner distribution matrix from the off-diagonal terms of the scattering operator. The scattering operator for the phonon Wigner distribution is obtained in full analogy by replacing in the previous equation electron lifetimes and scattering matrix with the phonon counterparts. The details of the derivation of these equations are given in [58, 59].

There are both similarities and notable differences between these equations and the electron or phonon BTE. First of all, the BTE is obtained from the Wigner transport equation as the limit in which the off-diagonal components of the Wigner distributions are neglected (i.e. $N_{\nu\nu'} \approx N_{\nu\nu'}\delta_{\nu\nu'}$ or $F_{bb'} \approx F_{bb'}\delta_{bb'}$). As a result, the Wigner formalism expands on the well-established predictive capability of the BTE. Critically, the equation of motion for the Wigner distribution adds a commutator, which allows coupling between different off-diagonal elements of the Wigner distribution function.

We note that the diagonal (single-band) elements of these equations coincide with the electron and phonon BTE discussed previously, and that results for transport coefficients can be interpreted as corrections to the BTE predictions. Furthermore, these Wigner corrections to the BTE transport coefficients only depend on the carrier lifetimes. Because these quantities are already computed when constructing the scattering operator for the BTE, the computational cost of the Wigner corrections is negligible compared to the cost of constructing and solving the BTE.

Just as described for the BTE, the Wigner transport equations for electrons and phonons can be solved considering linear perturbations of the Wigner distribution from equilibrium. Once the equations are solved, the transport coefficient are readily found. For example, the expectation value of charge flux is

$$J(\boldsymbol{r},t) = \frac{g_s}{2VN_k}\sum_{\boldsymbol{k}b}\left[\{\boldsymbol{V}(\boldsymbol{k}),F(\boldsymbol{r},\boldsymbol{k},t)\}\right]_{bb}, \tag{56}$$

where g_s takes into account the spin degeneracy (here the equation is shown for a system without spin polarization). As shown in [59], the electrical conductivity can be written as $\sigma = \sigma^{\mathrm{BTE}} + \Delta\sigma$, where σ^{BTE} is the conductivity predicted by the BTE, and $\Delta\sigma$ is a correction introduced by the Wigner distribution function:

$$\Delta\sigma_{\alpha\beta} = \frac{2g_s e^2}{VN_k}\sum_{\boldsymbol{k}bb'}\frac{\bar{f}_{b'}(\boldsymbol{k}) - \bar{f}_b(\boldsymbol{k})}{\epsilon_{b'}(\boldsymbol{k}) - \epsilon_b(\boldsymbol{k})}v_{\alpha\boldsymbol{k},bb'}v^*_{\beta\boldsymbol{k}b'b}\frac{\tau^{-1}_{\boldsymbol{k}b} + \tau^{-1}_{\boldsymbol{k}b'}}{4(\epsilon_{b'}(\boldsymbol{k}) - \epsilon_b(\boldsymbol{k}))^2 + (\tau^{-1}_{\boldsymbol{k}b} + \tau^{-1}_{\boldsymbol{k}b'})^2}. \tag{57}$$

The correction to conductivity introduced by the Wigner distribution can be significant, for example when the difference between the energies of quasiparticles in valence and conduction bands are small, i.e. for narrow-gap semiconductors, or when the dipole coupling between valence and conduction states is strong. Corrections to the other Onsager coefficients can similarly be derived.

A similar result holds for the phonon transport, where the energy current can be computed as

$$\boldsymbol{Q}(\boldsymbol{r},t) = \frac{1}{2VN_q}\sum_{\boldsymbol{q}\nu}\left[\{\boldsymbol{V}(\boldsymbol{q}),N(\boldsymbol{r},\boldsymbol{q},t)\}\Omega(\boldsymbol{q})\right]_{\nu\nu}. \tag{58}$$

Similarly, the lattice thermal conductivity is $k = k^{\mathrm{BTE}} + \Delta k$, with the first term being the lattice thermal conductivity predicted by the phonon BTE, and the Wigner distribution resulting in the correction

$$\Delta k_{\alpha\beta} = \frac{\hbar^2}{k_{\mathrm{B}}T^2VN_q}\sum_{\boldsymbol{q}\nu\nu'}\frac{\omega_{\boldsymbol{q}\nu}+\omega_{\boldsymbol{q}\nu'}}{2}v_{\alpha\boldsymbol{q},\nu\nu'}v^*_{\beta\boldsymbol{q}\nu'\nu}(\omega_{\boldsymbol{q}\nu}\bar{n}_{\boldsymbol{q}\nu}(\bar{n}_{\boldsymbol{q}\nu}+1) + \omega_{\boldsymbol{q}\nu'}\bar{n}_{\boldsymbol{q}\nu'}(\bar{n}_{\boldsymbol{q}\nu'}+1))\frac{\tau^{-1}_{\boldsymbol{q}\nu}+\tau^{-1}_{\boldsymbol{q}\nu'}}{4(\omega_{\boldsymbol{q}\nu'}-\omega_{\boldsymbol{q}\nu})^2 + (\tau^{-1}_{\boldsymbol{q}\nu}+\tau^{-1}_{\boldsymbol{q}\nu'})^2}. \tag{59}$$

It has been shown [58] that this correction is especially relevant for crystals with localized vibrational modes, when the small phonon group velocities suppresses the BTE lattice thermal conductivity and small lifetimes and small energy differences between phonon modes enhance the correction described by the Wigner distribution.

4.7. Dirac delta and smearing

Calculation of materials properties such as scattering rates often requires a numerical approximation of the Dirac–delta function, for example, the density of states, ρ

$$\rho(\epsilon) = \frac{1}{2\pi^3 V}\sum_b\int\delta(\epsilon-\epsilon_{\boldsymbol{k}b})\mathrm{d}\boldsymbol{k}. \tag{60}$$

We offer three different schemes for the numerical approximation of the Dirac–delta for the user to choose from:

(a) *Gaussian smearing*: the Dirac–delta is replaced with a Gaussian function

$$\delta(\epsilon_{\boldsymbol{k}b}-\epsilon_{\boldsymbol{k}'b'}) \approx \frac{1}{\sqrt{\pi}\sigma}exp\left(-\frac{(\epsilon_{\boldsymbol{k}b}-\epsilon_{\boldsymbol{k}'b'})^2}{\sigma^2}\right), \tag{61}$$

where σ is a user-specified smearing parameter. This method is simple and fast, but requires the user to converge results with respect to σ in conjunction with the number of wavevectors used for the BZ integration.

(b) *Adaptive Gaussian smearing*: the Gaussian smearing parameter σ introduced above can be approximated [60, 61] as

$$\sigma = \frac{A}{\sqrt{12}}\sqrt{\sum_\beta\left(\sum_\alpha\left(v^\alpha_{\boldsymbol{k}b}-v^\alpha_{\boldsymbol{k}'b'}\right)\frac{C_{\alpha\beta}}{N_\beta}\right)}, \tag{62}$$

where C is the matrix describing the crystal lattice vectors, N_β is the size of the Monkhorst–Pack mesh in crystal direction β, and the parameter A is a prefactor that can be fixed to 1 [61]. As this method self-adjusts σ with respect to the chosen BZ sampling and the local derivative of the carrier's energy, it tends to yield easier convergence of results compared the fixed-width Gaussian smearing scheme.

(c) *Tetrahedron method*: discussed in [62], this method provides a truly parameter-free linear approximation of the integrand. For simplicity, the tetrahedron method is currently implemented only for the calculation of the DoS. The method is accurate and parameter-free, although considerably slower than the above two alternatives.

4.8. Symmetries

A crystal is characterized by a set of symmetry operations $S = \{R, t\}$, with each operation S consisting of a rotation R and a fractional translation t such that the application of a translation and a rotation to the atomic positions leaves the overall crystal unchanged. In reciprocal space, instead of working with a set of wavevectors $\{\boldsymbol{k}\}$ that uniformly samples the BZ, symmetries allow the selection of a subset of irreducible wavevectors $\{\boldsymbol{k}^*\}$, such that the complete set of wavevectors can be obtained by applying rotations to the irreducible wavevectors set $\boldsymbol{k} = R\boldsymbol{k}^*$. As a result, integration over the BZ can be sped up by restricting the domain to the irreducible wedge of the BZ.

We recall here that microscopic carrier properties also obey symmetry relations. For example, scalars such as the carriers energy or lifetimes are left invariant by symmetry operations between an irreducible point $\boldsymbol{k}^*$ and its symmetry equivalent point $\boldsymbol{k} = R\boldsymbol{k}^*$ [52]

$$\epsilon_{\boldsymbol{k}b} = \epsilon_{\boldsymbol{k}^*b} \,, \tag{63}$$

and

$$\tau_{\boldsymbol{k}b} = \tau_{\boldsymbol{k}^*b} \,. \tag{64}$$

Instead, the group velocity transforms like a vector as

$$v_{\alpha\boldsymbol{k}b} = \sum_\beta R_{\alpha\beta} v_{\beta\boldsymbol{k}^*b} \,, \tag{65}$$

and similarly, the out-of-equilibrium population [52] transform as

$$\delta f_{\alpha\boldsymbol{k}b} = \sum_\beta R_{\alpha\beta} \delta f_{\beta\boldsymbol{k}^*b} \,. \tag{66}$$

As a result, the calculation of several properties can be sped up by restricting states to the irreducible wedge of the BZ, for example when computing the density of states, or solving the BTE at the RTA level.

The relaxation time approximation and iterative solvers (see sections 4.3.2 and 4.3.3) support the use of symmetries. The BTE for an irreducible point $\boldsymbol{k}^*$ (omitting band indices for simplicity) is

$$b^\alpha_{\boldsymbol{k}^*} = \sum_{\boldsymbol{k}'} A_{\boldsymbol{k}^*\boldsymbol{k}'} f^\alpha_{\boldsymbol{k}'} = \sum_{\beta\boldsymbol{k}'} A_{\boldsymbol{k}^*\boldsymbol{k}'} R^{\alpha\beta} f^\beta_{\boldsymbol{k}'} = \sum_{\beta\boldsymbol{k}'^*} \tilde{A}_{\alpha\beta;\boldsymbol{k}^*\boldsymbol{k}'^*} f^\beta_{\boldsymbol{k}'^*} \,, \tag{67}$$

where we used the symmetry properties of the out-of-equilibrium population, and $\tilde{A}$ is a tensor describing the scattering properties limited to the irreducible BZ wedge. As a result of symmetries, the BTE's linear algebra problem can be solved with a much smaller number of momentum states, and thus with a better computational efficiency.

5. Software

Phoebe is written in `C++` following the object-oriented paradigm. Since Phoebe computes a range of extremely different physical properties, each of these properties are organized in Apps. A main function simply initializes global properties (such as starting the MPI or Kokkos environments), reads the user input and launches the appropriate App through a class factory. Currently, Phoebe offers the following Apps:

- `ElectronWannierBandsApp`, `ElectronFourierBandsApp`, `PhononBandsApp`: calculation of electron or phonon energies along a BZ path;
- `ElectronWannierDosApp`, `ElectronFourierDosApp`, `PhononDosApp`: calculation of electron or phonon DOS;

- `ElPhQeToPhoebeApp`: processing of the electron–phonon coupling from QE to a Phoebe format. An EPA processing computes the electron–phonon averaged coupling, while the Wannier processing transforms the coupling to its real-space Wannier representation;
- `ElectronLifetimesApp` and `PhononLifetimesApp`: calculation of carriers' lifetimes along a BZ path;
- `ElectronWannierTransportApp`: calculation of electronic transport coefficients with Wannier interpolation method;
- `TransportEpaApp`: calculation of electronic transport coefficients with the EPA approximation;
- `PhononTransportApp`: evaluation of phonon transport properties.

The physical properties computed by these Apps are either printed to standard output or written to JSON files for user's convenience. We also provide a few `Python` scripts to generate some of the most common plots of physical properties.

Each App uses a number of classes built to reduce code duplication and to provide convenient data structures. In fact, a central goal of the project is to guarantee a simple deployment of methods and algorithms for future releases. The most important among the several classes available are

- `Crystal`: description of the crystal unit cell and crystal symmetries;
- `Points`: handling of wavevectors in the BZ;
- `HarmonicHamiltonian`: base class for the calculation of non-interacting carriers properties; subclassed into `PhononH0`, `ElectronH0Wannier` and `ElectronH0Fourier` for evaluating phonon properties, Wannier- and Fourier-interpolated electron properties respectively;
- `BandStructure`: storage of carriers' energies, velocities and eigenvectors;
- `Context`: description of user input variables;
- `MPIcontroller`: wrapper for calls to MPI functions;
- `Matrix`: storage of MPI-distributed matrices and a wrapper to ScaLAPACK subroutines;
- `StatisticsSweep`: calculation and description of values for temperature, chemical potentials and carrier concentrations;
- `Interaction`: base class for the evaluation of the scattering coupling strength. Its subclasses describe the three-phonon interaction, electron–phonon coupling with Wannier interpolation and the EPA coupling;
- `ScatteringMatrix`: base class for the calculation, storage and manipulation of the electron or phonon scattering matrix;
- `VectorBTE`: data structure for storage and manipulation of the out-of-equilibrium population;
- `Observable`: base class of a number of classes for evaluating transport coefficients.

In the following we will discuss some of these classes in greater detail.

5.1. Points class

The `Points` class in Phoebe manages the wavevectors used to sample the BZ. We defined two constructors, one for instantiating a class for uniform sampling of the BZ and another for paths of wavevectors in the BZ. When uniformly sampling the BZ with a Monkhorst–Pack grid, the complete list of points (wavevectors) is not explicitly stored for memory management convenience. This implicit list contains points sorted by their coordinates (incrementing the coordinates z, y and x in this order). When describing BZ paths, instead, points are not sorted, and their coordinates are explicitly stored in a list. Points are internally manipulated in crystal coordinates (i.e. in scaled lattice vectors units), and we provide options to output points in Cartesian coordinates or to fold them in the first BZ.

In the `Points` class we implemented the logic for reducing a set of wavevectors to the set of symmetry-irreducible wavevectors in the BZ. The class method `setIrreduciblePoints()`, takes the set of crystal symmetries (from the `Crystal` class) and finds the list of irreducible BZ points, i.e. finds the irreducible set of points $\boldsymbol{k}^*$ such that all remaining points $\boldsymbol{k}$ can be found applying a rotation R to $\boldsymbol{k}^*$.

It's worth noting that symmetry operations are not unique, so that a symmetry operation may rotate a wavevector correctly as $\boldsymbol{k} = R\boldsymbol{k}^*$ but other quantities such as the group velocity may not be rotated correctly $\boldsymbol{v}(\boldsymbol{k}) \neq R\boldsymbol{v}(\boldsymbol{k}^*)$. Since the carrier populations transform like group velocities, the method `setIrreduciblePoints()` can optionally find a set of symmetries that rotate both points and velocities.

The `Points` class also has a `setActiveLayer()` method that allows discarding some points that do not provide a sizeable contribution to a physical observable, a functionality that will be explained in section 5.2 in conjunction with the `ActiveBandStructure` class. After a call to this method, the functionalities in this class operate as if only the restricted set of points are present in the class.

Finally, several other parts of the code require the lookup of a wavevector in the list of points and, due to the large number of points used to sample the BZ (often in the excess of 10^4), this search operation can take a

substantial part of the calculation if not implemented efficiently. If `Points` describes a uniform BZ sampling with a complete Monkhorst–Pack grid, the point search in the list can be done in $o(0)$ operations, taking advantage of the implicit ordering of points. If the list of points is sorted (e.g. after a call to `setActiveLayer()`) the lookup operation can be done with a binary search algorithm, which scales logarithmically with the number of points. Only if points are not sorted (e.g. when they describe a BZ path), the point search is made with a slow element-by-element comparison.

5.2. Band structure classes

We developed a data structure dedicated to the storage of the band structure, that is, the energies ($\epsilon_{\boldsymbol{k}b}$ and $\hbar\omega_{\boldsymbol{q}\nu}$), eigenvectors ($U_{\boldsymbol{k}mb}$ and $z^{s\alpha}_{\boldsymbol{q}\nu}$) and velocities ($v_{\alpha\boldsymbol{k}bb'}$ and $v_{\alpha\boldsymbol{q}\nu\nu'}$) of carriers over the BZ.

To handle this information, we defined two classes: `FullBandStructure` and `ActiveBandStructure`. Both classes take a `Points` instance as input, since `Points` defines the set of wavevectors over which the carriers' properties are computed. While both classes store the band structure, there is an important difference. The `FullBandStructure` class stores energies, eigenvectors and velocities for all wavevectors defined in the `Points` instance. For example, given N_k points and N_b electronic bands, `FullBandStructure` stores the complete list of $N_k \times N_b$ values of e.g. energies $\epsilon_{\boldsymbol{k}b}$ (and similar for eigenvectors or velocities). Since wavevector meshes can be extremely large, carriers properties can optionally be stored in a MPI-distributed fashion, using the `Matrix` class described in section 5.3.

`ActiveBandStructure` contains similar functionalities of `FullBandStructure` in that it stores information of carriers properties (in fact they share the same virtual parent class). However, the constructor of this class discards some Bloch states ($\boldsymbol{k}b$) that do not contribute significantly to the calculation of transport properties. Because transport properties at finite temperature only depend on thermally excited electron and phonon states, there are many states which do not contribute and are readily discarded to save computational resources. For example, only long-wavelength acoustic phonon states contribute to transport at low temperatures T as their excitation energy goes to 0 with T. Similarly, only electrons with energy close to the chemical potential μ contribute to charge currents and thus we can restrict calculations to electronic states such that $|\epsilon_{\boldsymbol{k}b} - \mu|$ is smaller than a few $k_B T$. As a result, states that are not thermally excited can be discarded, and in Phoebe we term this band and momentum-space state filtering procedure as a "window".

We implemented two kinds of windows. The first window type imposes an energy cutoff and discards all states with energy larger than a user-specified cutoff ϵ_{cutoff}: for phonons we retain states such that $|\hbar\omega_{\boldsymbol{q}\nu}| < \epsilon_{\text{cutoff}}$ and for electrons $|\epsilon_{\boldsymbol{q}\nu} - \mu| < \epsilon_{\text{cutoff}}$. However, this kind of window is impractical, as the cutoff is temperature dependent and, in practice, a user would need to perform a convergence test on ϵ_{cutoff}.

We implemented a second window type, noting that transport coefficients contain terms of the form $\bar{f}_{\boldsymbol{k}b}(1-\bar{f}_{\boldsymbol{k}b})$ for electrons or $\bar{n}_{\boldsymbol{k}b}(1+\bar{n}_{\boldsymbol{k}b})$ for phonons (which arise from the energy or temperature derivative of equilibrium distributions). For both carriers, these terms decay exponentially to zero if the carrier's energy is much larger than $k_B T$, thus suppressing contributions of high-energy carriers to transport properties. We therefore define a "population window" by retaining only states such that, for electrons

$$\bar{f}_{\boldsymbol{k}b}(1-\bar{f}_{\boldsymbol{k}b}) > \delta\,, \tag{68}$$

or for phonons

$$\bar{n}_{\boldsymbol{k}b}(1+\bar{n}_{\boldsymbol{k}b}) > \delta\,, \tag{69}$$

where δ is a small number. Since the temperature dependence is inside the definition of Bose–Einstein and Fermi–Dirac distributions, we found that δ can be hard-coded to a fixed value and that transport properties are not affected. In this way, we achieve a reduction in the number of Bloch states used in the various calculations, and avoid the introduction of an additional parameter to be converged, for greater user convenience.

As a result of this filtering procedures, the number of electron bands N_b or phonon branches N_ν is no longer a constant but depends on the wavevector index. As a result, quantities like the electronic energy $\epsilon_{\boldsymbol{k}b}$ cannot be stored as a matrix of size $N_k \times N_b$. Instead, energies are stored as a flattened vector, and we provide class methods to access energies through combined band and wavevector state indices. A similar strategy is adopted for storing eigenvectors or velocities.

5.3. Matrix

As described in sections 4.1 and 4.2, the BTE is a linear algebra problem and due to the number of Bloch states involved, requires distributed matrices and parallel linear algebra operations. In order to perform these massively parallel dense linear algebra operations, we utilize the ScaLAPACK library [14]. In order to both avoid explicit calls to this library in the main body of code and to allow future flexibility should we desire to

swap ScaLAPACK routines with those of a different library, we developed the `Matrix` class as a wrapper to this library.

Because ScaLAPACK is mostly built in Fortran and C languages, it follows an imperative procedural programming paradigm, i.e. a set of routines that must be called in the right order, resulting in a departure from the object-oriented programming style adopted in the rest of Phoebe. The `Matrix` class has therefore been designed to map this procedural-style library into an object-oriented style. Therefore, the class constructor takes care of ScaLAPACK initializations, such as starting the BLACS environment and allocating the memory locations, while deallocations take place in the class destructor. Various class methods have been implemented to provide wrappers around the needed linear algebra operations, such as matrix–matrix multiplication and matrix diagonalization, with the benefit of hiding within the class method all the necessary auxiliary calls (such as temporary memory allocations or index mappings) in order to call use the ScaLAPACK library. As a result, the rest of Phoebe's code can act on distributed memory with a simplified interface that does not obfuscate the main body of the transport code.

`C++` class templates further allow us to generalize the class to store different data-types (in particular, real and complex floating point values). We also allow the code to be compiled without the ScaLAPACK or MPI libraries. In this case, the `Matrix` class falls back to a "serial matrix" which relies on the `Eigen` library, losing all MPI-parallel characteristics (therefore becoming mostly suitable for debugging purposes).

5.4. Scattering matrix

The `ScatteringMatrix` class, as the name suggest, implements operations related to BTE's scattering operator. A key challenge in developing this class has been to try generalize this code to achieve different purposes. In order to implement the calculation of scattering rates, we defined `ScatteringMatrix` as a virtual base class that defines a virtual method `builder()`; the method is implemented in the subclasses `PhononScatteringMatrix` and `ElectronScatteringMatrix` and evaluates equation (28) or (18). In this way, auxiliary methods such as the calculation of a matrix-vector product or the scattering matrix diagonalization can be defined directly in the base class and are shared by the two different carriers.

The scattering matrix has a few modes of operation. If the BTE is to be solved only at the RTA level, only the calculation of lifetimes is performed by the class instance. For exact BTE solutions, for which the full scattering matrix must be dealt with, we distinguish two cases: first, when the complete scattering matrix is stored in memory or alternatively, when only the product of the scattering matrix on a population vector is computed (without the need of saving the matrix in memory). In the former case, the calculation has a larger memory requirement, however, this allows to quickly solve the BTE. In the latter case, the memory footprint of the calculation is reduced, but as a drawback each iteration of a BTE solver will require more computational time. We thus let the user choose the desired memory impact, which will depend on both the material studied and on the available hardware. To summarize, the `builder()` method, depending on the input, is tasked with computing (a) the carriers' lifetimes, (b) the scattering matrix or (c) its action on a vector. Note that due to the size of the scattering matrix $A_{\nu\nu'}$, it is stored as a class member using an MPI-distributed `Matrix` instance.

Great care has been taken to optimize the efficiency of the scattering rate calculations implemented in `builder`, as these are some of the most computationally intensive functions in the code. Significant improvements to the calculation speed have been achieved by precomputing all independent particle properties whenever possible. Additionally, we used MPI parallelization to accelerate loops over $\boldsymbol{k}$ and $\boldsymbol{k}'$ wavevectors, while other loops over band indices are accelerated with OpenMP parallelization. The calculation of the interaction coupling strength and its GPU acceleration is described in the next section.

5.5. Kokkos (interaction)

In Phoebe, great care has been taken to accelerate the calculation of interaction matrix elements and scattering rates. The calculation of the scattering coupling strength for either three phonon and electron–phonon scattering in reciprocal space, $f_{\alpha\beta\gamma}(\boldsymbol{k},\boldsymbol{k}')$, starting from a real-space representation, $g_{abc}(\boldsymbol{R},\boldsymbol{R}')$, is obtained with a tensor operation in the form

$$f_{\alpha\beta\gamma}(\boldsymbol{k},\boldsymbol{k}') = \sum_{\boldsymbol{R}\boldsymbol{R}'}\sum_{abc} e^{i\boldsymbol{k}\cdot\boldsymbol{R}+i\boldsymbol{k}'\cdot\boldsymbol{R}'} U_{\alpha a}(\boldsymbol{k}')g_{abc}(\boldsymbol{R},\boldsymbol{R}')U_{\beta b}(\boldsymbol{k})U_{\gamma c}(\boldsymbol{k}'-\boldsymbol{k}) \,, \tag{70}$$

where U are rectangular matrices indicating Wannier rotation matrices or phonon eigenvectors, a, b and c are indices labeling Wannier functions or atomic displacements, α, β and γ are Bloch band indices, $\boldsymbol{k}$ is a wavevector, and $\boldsymbol{R}$ is a Bravais lattice vector.

In order to maximise performance, we first note that this expression typically appears inside a double loop on $\boldsymbol{k}$ and $\boldsymbol{k}'$. Therefore, if $\boldsymbol{k}$ is the outer loop, it is convenient to precompute the quantity

$$\tilde{f}_{\alpha bc}(\boldsymbol{k},\boldsymbol{R}') = \sum_{\boldsymbol{R}b} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} g_{abc}(\boldsymbol{R},\boldsymbol{R}') U_{\beta b}(\boldsymbol{k}) \,, \tag{71}$$

at fixed value of $\boldsymbol{k}$, and then complete the transformation inside the loop over $\boldsymbol{k}'$. It is necessary to adopt the most convenient tensor contractions in order to minimize the size of loops needed to perform the interpolation.

Additionally, because GPUs are extremely effective at performing tensor operations, we implemented the calculation of the coupling with the Kokkos library [63]. This library provides a hardware-agnostic programming model for parallelizing operations, so that tensor operations are implemented once following Kokkos' syntax and then, at compilation time, the code is optimized for the desired architecture. The code can thus run both on standard CPU-only architecture as well as on GPUs when available without code duplication. We therefore achieve strong parallel performance across a varied range of underlying architectures.

In our preliminary tests, we observed that the interpolation at fixed values of $\boldsymbol{k}$ and $\boldsymbol{k}'$ is extremely fast on a GPU, causing communication latency between GPU and CPU to be the dominant bottleneck. To counteract this effect, we increase the workload on the GPU by performing the interpolation for a fixed value of $\boldsymbol{k}$ and a set of values for $\{\boldsymbol{k}'\}$ at once. The size of the set $\{\boldsymbol{k}'\}$ is optimized at runtime so that most of the on-board GPU's memory is used. Finally, as the size of the electron–phonon coupling matrix in the Wannier representation $g_{bb'\nu}(\boldsymbol{R}_\mathrm{e},\boldsymbol{R}_\mathrm{p})$ may be too large to fit in a single GPU's memory, we adopt an additional layer of MPI parallelization. The electron–phonon coupling can be distributed over the index $\boldsymbol{R}_\mathrm{e}$ across a 'pool' of MPI processes, and we allow the user to specify the size of the pool at runtime through the command line.

The electron–phonon matrix interpolation of polar materials takes an additional correction, specified in equation (15), due to long-range interaction. In order to optimize the evaluation of this long-range correction, we note that most terms, except the overlap matrix $\left\langle \psi_{\boldsymbol{k}+\boldsymbol{q},b'} \left| e^{i(\boldsymbol{q}+\boldsymbol{G})\cdot\boldsymbol{r}} \right| \psi_{\boldsymbol{k}+\boldsymbol{q},b} \right\rangle$, depend exclusively on the phonon wavevector $\boldsymbol{q}$. We therefore precompute these factors for all possible values of $\boldsymbol{q}$ once, at the start of the coupling interpolation. Next, we note that the overlap term can be evaluated as $\left\langle \psi_{\boldsymbol{k}+\boldsymbol{q},b'} \left| e^{i(\boldsymbol{q}+\boldsymbol{G})\cdot\boldsymbol{r}} \right| \psi_{\boldsymbol{k}+\boldsymbol{q},b} \right\rangle = \left[U(\boldsymbol{k}+\boldsymbol{q})U^\dagger(\boldsymbol{k})\right]_{b'b}$, and, since the U matrices are readily available for the calculation of the short-range component of the electron–phonon matrix, it is readily possible to add this contribution without much overhead to the calculation.

5.6. Quantum ESPRESSO interface for electron–phonon coupling

Phoebe interfaces with QE `ph.x` and Wannier90 in order to obtain the relevant *ab-initio* input for the electron–phonon coupling calculations. However, this interface presented an important technical challenge. As mentioned in section 3.2, gauge fixing is a critical component of the Wannier interpolation. In order for the Wannier interpolation to work, it is critical that the gauge of the wavefunctions used by these two separate codes stays the same throughout the calculations. Unfortunately, there is no built-in QE function to fix the gauge of a wavefunction and therefore the wavefunction may take a random phase every time it is computed. As a result, we developed a gauge-fixing procedure that ensures the wavefunction has a consistent gauge across `pw.x` (the QE code that finds the ground state wavefunctions), `ph.x` (the QE code that computes the electron–phonon coupling) and `wannier90` (which computes the maximally-localized Wannier functions).

Therefore, in addition to the main Phoebe code, we provide a patched version of QE that must be used to compute electronic properties, which guarantees the gauge of the wavefunctions is appropriately taken into account. In order to explain this gauge fixing procedure, we first recall that QE is a plane-wave *ab-initio* code, so that the wavefunction is computed as

$$\psi_{\boldsymbol{k}b}(\boldsymbol{r}) = \sum_{\boldsymbol{G}} c_{\boldsymbol{k}b}(\boldsymbol{G}) e^{i\boldsymbol{G}\cdot\boldsymbol{r}+i\boldsymbol{k}\cdot\boldsymbol{r}} \,, \tag{72}$$

where $\boldsymbol{G}$ runs over a set of reciprocal lattice vectors (typically defined up to a user-defined cutoff value on $|\boldsymbol{k}+\boldsymbol{G}|^2$), c are the plane wave coefficients, and $\boldsymbol{r}$ is the position. The subroutine `c_bands()` of QE is responsible for building and diagonalizing the Hamiltonian matrix, and therefore also computes the eigenvalues $\epsilon_{\boldsymbol{k}b}$ and the plane wave coefficients $c_{\boldsymbol{k}b}(\boldsymbol{G})$. The phase arbitrariness manifests as a random phase on the complex plane-wave coefficients $c_{\boldsymbol{k}b}(\boldsymbol{G})$. In our patch, we modified `c_bands()` so that if `c_bands()` is called by `pw.x` and QE has found the ground state wavefunction (i.e. a scf calculation), we save $c_{\boldsymbol{k}b}(\boldsymbol{G})$ to file. These coefficients will serve as a reference wavefunction to fix the gauge of all subsequent calculations. If `c_bands()` is called by another code, for example, when generating auxiliary data for Wannier90 or `ph.x`, we read the reference plane-wave coefficients and use them to fix the gauge of the newly computed

wavefunction. In this way, we can ensure that—at the phase level—the same wavefunction is used throughout these multiple calculations.

An additional complexity in this patch is caused by the use of symmetries in the *ab-initio* code. The reference plane-wave coefficients built in a `pw.x` calculation are typically computed only for wavevectors $\{\tilde{\boldsymbol{k}}\}$ in the irreducible BZ wedge. Therefore, we save to file only the coefficients $c_{\tilde{\boldsymbol{k}}b}(\boldsymbol{G})$ computed at irreducible points. The wavefunction at a symmetry-equivalent wavevector $\boldsymbol{k}$ can be reconstructed from the irreducible wavevector $\tilde{\boldsymbol{k}} = R\boldsymbol{k}$ as

$$\psi_{R\boldsymbol{k}b}(\boldsymbol{r}) = \psi_{\boldsymbol{k}b}(R^{-1}(\boldsymbol{r}-\boldsymbol{t})) \,, \tag{73}$$

which in terms of plane-wave coefficients implies

$$c_{R\boldsymbol{k}b}(\boldsymbol{G}) = e^{-i(R\boldsymbol{k}+\boldsymbol{G})\cdot t} c_{\boldsymbol{k}b}(R^{-1}\boldsymbol{G}) \,, \tag{74}$$

where $S = \{R, t\}$ is a crystal symmetry as introduced in section 4.8. An additional symmetry that is taken into account is the time reversal operation

$$\psi_{-\boldsymbol{k}b}(\boldsymbol{r}) = \psi^*_{\boldsymbol{k}b}(\boldsymbol{r}) \,, \tag{75}$$

which in terms of plane-wave coefficients is

$$c_{-\boldsymbol{k}b}(\boldsymbol{G}) = c^*_{\boldsymbol{k}b}(-\boldsymbol{G}) \,. \tag{76}$$

Another constraint comes from the translational symmetry of the crystal, i.e.

$$\psi_{\boldsymbol{k}b}(\boldsymbol{r}) = \psi_{\boldsymbol{k}+\boldsymbol{G}',b}(\boldsymbol{r}) \,, \tag{77}$$

or equivalently

$$c_{\boldsymbol{k}+\boldsymbol{G},b}(\boldsymbol{G}) = c_{\boldsymbol{k}b}(\boldsymbol{G}+\boldsymbol{G}') \,. \tag{78}$$

Capability for using symmetries in the presence of magnetism are currently being developed. Using these relations and knowing the wavefunction at $\tilde{\boldsymbol{k}}$, it is possible to reconstruct the wavefunction at any symmetry-equivalent wavevector $\boldsymbol{k}$.

However, in QE, the wavefunction cannot be built from symmetries just by mapping $c_{\tilde{\boldsymbol{k}},b}(\boldsymbol{G})$ into $c_{\boldsymbol{k},b}(\boldsymbol{G})$. In fact, the set of $\boldsymbol{G}$ vectors is chosen such that $|\boldsymbol{G}+\boldsymbol{k}|^2 < \delta$, where δ is a user-specified cutoff. As a result, we cannot map all $c_{\tilde{\boldsymbol{k}},b}(\boldsymbol{G})$ into $c_{\boldsymbol{k},b}(\boldsymbol{G})$ as some plane-wave coefficients might be missing at the rotated wavevector, causing errors in the normalization of the wavefunction. To address this, we first rotate the reference wavefunction $|\psi^{\mathrm{ref}}_{\tilde{\boldsymbol{k}}b}\rangle$ into the desired wavevector $\boldsymbol{k}$ finding $|\psi^{\mathrm{ref}}_{\boldsymbol{k}b}\rangle = |S\psi^{\mathrm{ref}}_{\tilde{\boldsymbol{k}}b}\rangle$. Next, we compute the overlap matrix O as

$$O_{\boldsymbol{k}bb'} = \langle \psi^{\mathrm{ref}}_{\boldsymbol{k},b} | \psi^{\mathrm{arb-gauge}}_{\boldsymbol{k},b'} \rangle_{bb'} \,, \tag{79}$$

where $|\psi^{\mathrm{arb-gauge}}_{\boldsymbol{k},b}\rangle$ is the wavefunction with an arbitrary gauge that has been computed by QE at wavevector $\boldsymbol{k}$. With a complete basis set O would be a unitary matrix, however, due to the finite set of $\boldsymbol{G}$-vectors, the unitary property is violated by a small amount Δ defined as

$$\Delta = 1 - OO^\dagger \,. \tag{80}$$

We can re-enforce the unitary property of O by first noting that O is expected to be semi-positive definite (due to the lack of completeness of the finite set of $\boldsymbol{G}$ vectors). We then write

$$1 = OO^\dagger + \Delta \tag{81}$$

$$= OO^\dagger + (OO^\dagger + \Delta)\Delta(OO^\dagger + \Delta) \tag{82}$$

$$\approx OO^\dagger + OO^\dagger \Delta OO^\dagger \tag{83}$$

$$= OLL^\dagger O^\dagger \tag{84}$$

where L is the Cholesky decomposition of the matrix $LL^\dagger = 1 + O^\dagger \Delta O$ and introduced an approximation that has an error of order Δ^3 (expected to be small for well-converged calculations). By construction, the matrix OL is a unitary matrix which can now be used to rotate the wavefunction computed at a symmetry-equivalent point to the desired gauge, while preserving the orthonormality of the wavefunctions. We therefore fix the gauge of the wavefunction at $\boldsymbol{k}$ by applying the rotation

$$\psi_{\boldsymbol{k}}^{\text{fixed}-\text{gauge}} = OL\psi_{\boldsymbol{k}}^{\text{arb}-\text{gauge}} . \tag{85}$$

This procedure makes the wavefunctions obey the symmetries of the crystals and greatly simplifies the electron–phonon coupling calculation. As explained in [33], if the wavefunction obeys crystal symmetries, the electron–phonon coupling satisfies the symmetry relation

$$g_{bb'\nu}(S\boldsymbol{k}, \boldsymbol{q}) = g_{bb'\nu}(\boldsymbol{k}, S^{-1}\boldsymbol{q}) , \tag{86}$$

where S is a symmetry operation of a crystal and

$$g_{bb'\nu}(\boldsymbol{k}, \boldsymbol{q}) = g_{bb'\nu}(\boldsymbol{k} + \boldsymbol{G}, \boldsymbol{q} + \boldsymbol{G}) , \tag{87}$$

due to the periodicity of the crystal. As a result, we can compute the electron–phonon coupling on a set of irreducible wavevectors, then construct the remaining matrix elements by means of the symmetry relations, reducing the burden of the *ab-initio* calculation.

We stress that this QE patch avoids the need for re-implementing calculations of the electron–phonon matrix elements $\langle \psi_{\boldsymbol{k}+\boldsymbol{q},b'} | \partial_{\boldsymbol{q}\nu} V | \psi_{\boldsymbol{k}b} \rangle$, which are already computed in the code `ph.x`. As a result, Phoebe can readily take advantage of all features supported by `ph.x`, such as ultra-soft or PAW pseudopotentials [64], U corrections [24] and others, without the need for re-implementing them. The only modification necessary in the code `ph.x` has been the introduction of a subroutine that unfolds the symmetry of the coupling g and writes it to file output. The Wannier90 code does not need to be modified, as it simply reads the wavefunctions pre-computed by QE. As a last detail, we stress that the gauge problem must also be taken care when dealing with the phonon eigenvector z, and the eigenvector z present in equation (12) must be exactly the same phonon eigenvector (phase factor included) used by `ph.x` to compute the electron–phonon coupling. Additionally, z must obey the symmetries of the crystal, as discussed in [33].

5.7. Workflow overview

Having introduced all the major features and computational structures of Phoebe, we demonstrate how the code can be used to calculate transport properties of materials. In figure 1 we schematically show a user's workflow for computing electronic and thermal transport properties with Phoebe. The electronic transport workflow, as shown in the top panel, starts with the user going through the *ab-initio* calculation. The first step requires the calculation of the ground state wavefunction and electronic energies. Due to the wavefunction gauge requirements explained in section 5.6, the ground state calculation must be performed using our patched version of QE. Next, the user proceeds to compute the phonon properties with the patched version of `ph.x`. The output of this calculation, specifically files containing the force constants and the electron–phonon coupling, are later passed to Phoebe. At the same time, the user must also compute the Wannier functions with Wannier90, which outputs the real-space representation of the electronic Hamiltonian and the U rotation matrices that must be used to rotate the wavefunction to the Wannier gauge. After these *ab-initio* calculations have concluded, the user must run the `qe2PhoebeApp`, which uses the data from the *ab-initio* calculations to compute the electron–phonon coupling in the Wannier representation $g(\boldsymbol{R}_e, \boldsymbol{R}_p)$ starting from $g(\boldsymbol{k}, \boldsymbol{q})$ computed by `ph.x`, as described in equation (12). This operation needs to be run only once per material, and generates an output (HDF5) file with the electron–phonon coupling in real space, ready to be used for the interpolation. Finally, the user has all the input data necessary to compute the electron transport properties by launching the `electronWannierTransportApp`. Results are output as JSON files, easily parsed and analyzed by the user.

The workflow for computing the phonon transport properties is shown in the lower panel of figure 1. First, the user must select an *ab-initio* code (from any that work with phono3py or ShengBTE) to calculate the equilibrium ground state of the crystal. After a reference equilibrium crystal structure is found, the user runs an external library, e.g. phonopy and phono3py or ShengBTE, in conjunction with the *ab-initio* code to generate all atomic displacements needed to compute the second and third derivatives of the total energy (see section 3.1). Once both $V^{(2)}$ and $V^{(3)}$ are computed, the `PhononTransportApp` is ready to be launched. This app will interpolate the phonon–phonon coupling, compute phonon lifetimes, and then solve the BTE to evaluate the phonon transport properties such as the thermal conductivity. As in the electronic transport case, the results are written in easily parsable JSON files that can be conveniently analyzed.

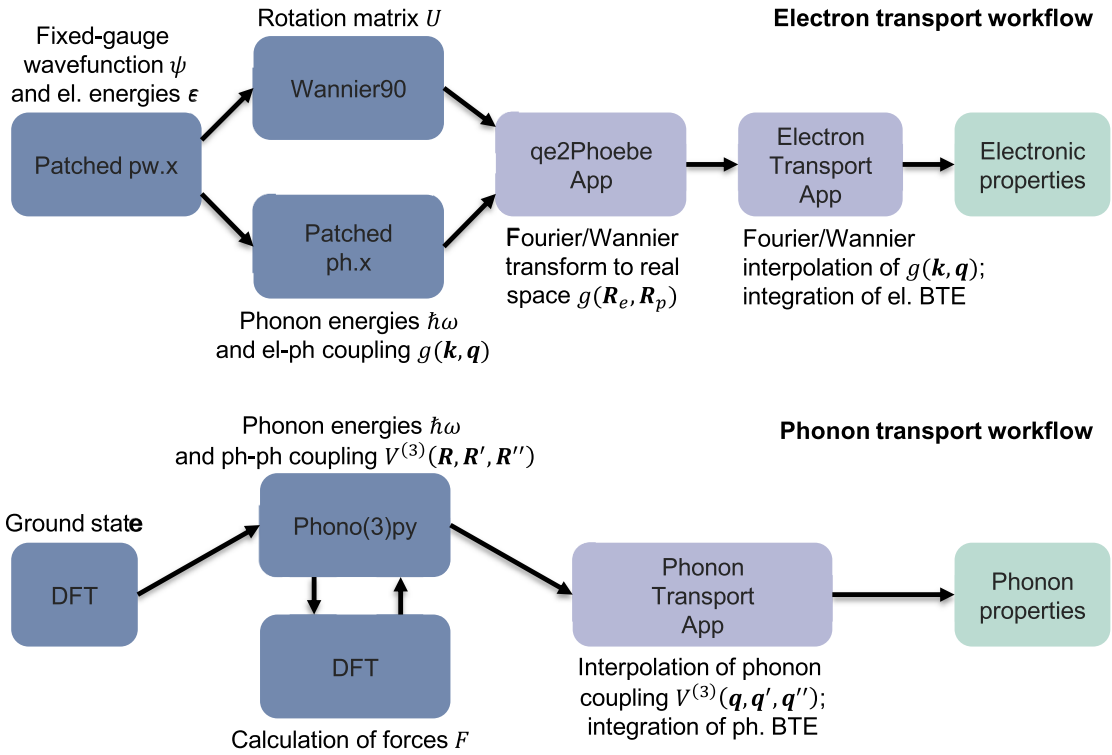

Figure 1. Schematic depiction of the user workflows to compute electronic transport properties with Phoebe (top panel) or phonon transport properties (bottom panel). Generally, the user must first perform *ab-initio* simulations (blue boxes), using our patched version of Quantum ESPRESSO for studying electron–phonon interactions or any DFT code that can provide anharmonic phonon interactions. Next, shown in purple boxes, the user must launch the Phoebe executable. The electronic transport properties calculation must be preceded by a one-time post-processing of the electron–phonon coupling taken in output from Quantum ESPRESSO, while the phonon transport calculation can be directly started from the output of *ab-initio* calculations. Finally, results are provided in output either in the standard output file, or as a parsable JSON format.

6. Results

Next, we showcase the calculation of transport properties by applying these two computational workflows to gallium nitride (GaN), a semiconductor often used for high-power applications.

6.1. GaN

GaN is a well-studied semiconductor that crystallizes in a wurtzite structure. We calculate the electronic structure of this material via QE using optimized norm-conserving Vanderbilt pseudopotentials [65, 66] and the local-density approximation exchange-correlation functional [67]. For the phonon workflow, we construct the force constants in a $4 \times 4 \times 3$ supercell, and the third derivative using a $3 \times 3 \times 2$ supercell. The supercell DFT calculation is performed using a $2 \times 2 \times 2$ mesh for the $\boldsymbol{k}$-point integration and a wavefunction cutoff of 110 Ry. Next, the phonon BTE is built using the adaptive smearing method for approximating the Dirac-delta and converged with a $15 \times 15 \times 15$ mesh of $\boldsymbol{q}$-points. For the electron transport workflow, *ab-initio* properties are computed using a 110 Ry plane-wave energy cutoff, a $12 \times 12 \times 7$ coarse mesh for both $\boldsymbol{k}$- and $\boldsymbol{q}$-points and the same pseudopotential choice. The maximally-localized Wannier function basis for this material was established by starting with an initial guess of 14 random Wannier centers. The electronic BTE is build using an adaptive smearing scheme (section 4.7), and states are discarded with the population window described in section 5.2. The BTE is integrated using a variable mesh of k-points, starting from the densest mesh of $170 \times 170 \times 106$ at the lowest temperature of 100 K and gradually decreased down to $90 \times 90 \times 56$ at the highest temperature of 450 K, as the convergence properties depend on temperature.

In figure 2 we plot some of the transport properties of GaN as a function of temperature. In the top left panel of figure 2, we plot the lattice thermal conductivity as a function of temperature, and compare results of the BTE at various levels of approximation with the experimental values [68]. We find a good agreement between our simulations and experiments, validating the results of this work. The RTA solution tends to underestimate the thermal conductivity predicted by exact BTE, as expected due to the variational principle underlying the formalism [23, 27]. To solve the phonon BTE exactly, we use the variational and the relaxons

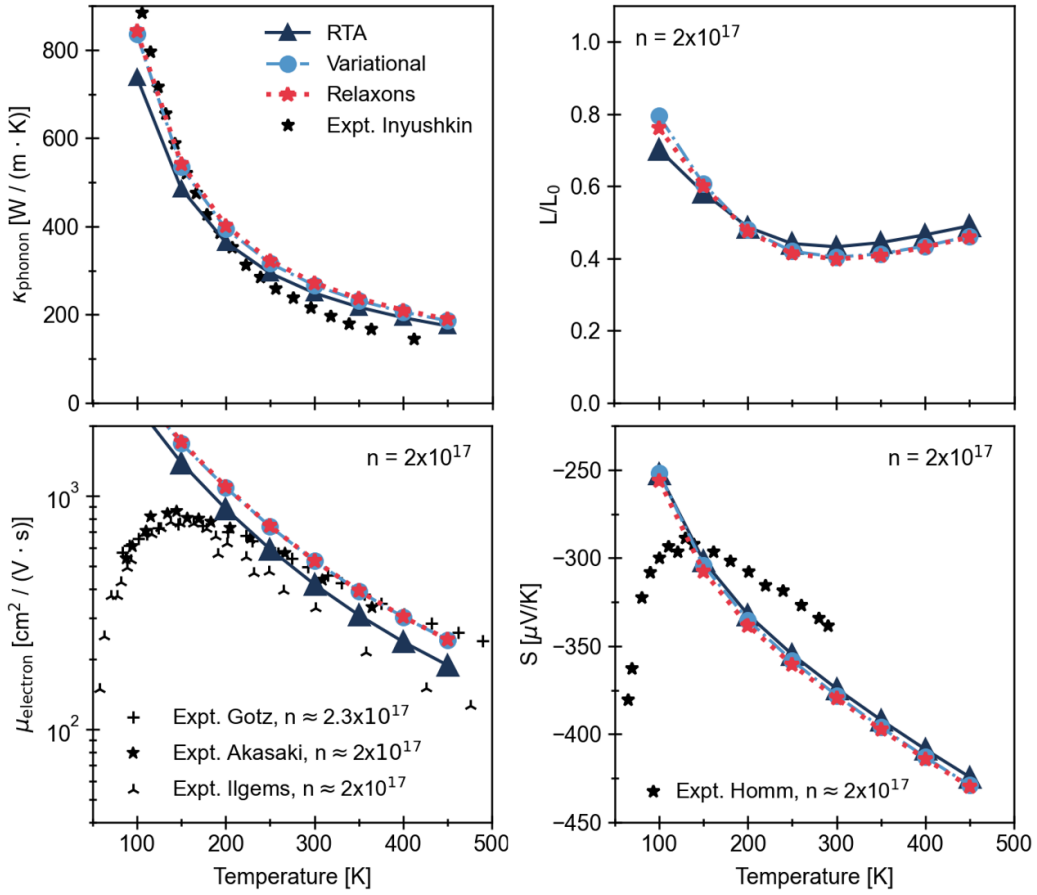

Figure 2. The calculated lattice thermal conductivity (top left), electron mobility (bottom left), Lorenz number (top right), and Seebeck coefficient (bottom right) (with μ, L, and S calculated for carrier density $n = 2 \times 10^{17}$ cm^{-3}) of GaN along the a-direction crystal axis determined using the RTA, variational and relaxons solutions to the BTE as implemented in Phoebe. Experimental data for κ_{phonon} [68], $\mu_{electron}$ [69–71], and S [72] are shown for comparison.

solvers, mainly to note how the two solvers are able to provide, as expected, approximately the same solution to the BTE.

The electron drift mobility at a n-carrier concentration of about 2×10^{17} cm^{-3} is plotted versus temperature in the bottom left panel of figure 2 and compared against some of the available experiments [69–71]. Here we distinguish two regimes. At higher temperatures, where a sufficient number of phonons is thermally excited, we expect electron–phonon scattering to be the dominant factor limiting charge transport in GaN. Indeed, for temperatures above approximately 150 K, we obtain a good agreement with experimental values, and correctly reproduce the decreasing trend of mobility with respect to temperature. At lower temperatures we observe a departure in computed mobility from experimental values, which tend to decrease as temperature goes to zero, while our results are showing the opposite trend. This is likely due to the fact that at lower temperature fewer phonons are thermally excited, and electrons have a higher probability of scattering against defects or impurities in the crystal, which have not been taken into account for this simulation. We also note that the RTA already provides a good approximation to the mobility (and also for the Seebeck coefficient and the resistivity), and the variational and relaxon solvers provide essentially the same estimates. Overall, we observe a good agreement with experiments for simulations in the phonon-limited regime of charge transport.

Additionally, in the top right panel of figure 2, we include a calculation of the Lorenz ratio, defined as $L/L_0 = \frac{k_e}{\sigma T L_0}$, where k_e is the electronic contribution to the thermal conductivity, and L_0 is the standard Lorenz constant for an ideal metal [GIVE VALUE]. In GaN we confirm that L/L_0 shows sensible behavior and is close to unity, i.e. the limit in which the Wiedemann–Franz law is obeyed. A deviation from $L/L_0 = 1$ is to be expected, since the Wiedemann–Franz law is typically more closely followed by metals.

In figure 2, bottom right panel, we plot the Seebeck coefficient as a function of temperature. Here, we obtain a good agreement between our results and the experimental values for large temperatures [72], i.e. larger than 150 K. Since the Seebeck coefficient is strongly related to the density of states of the material [73], the results seem to indicate that the DFT simulation provided a good approximation to the electronic band

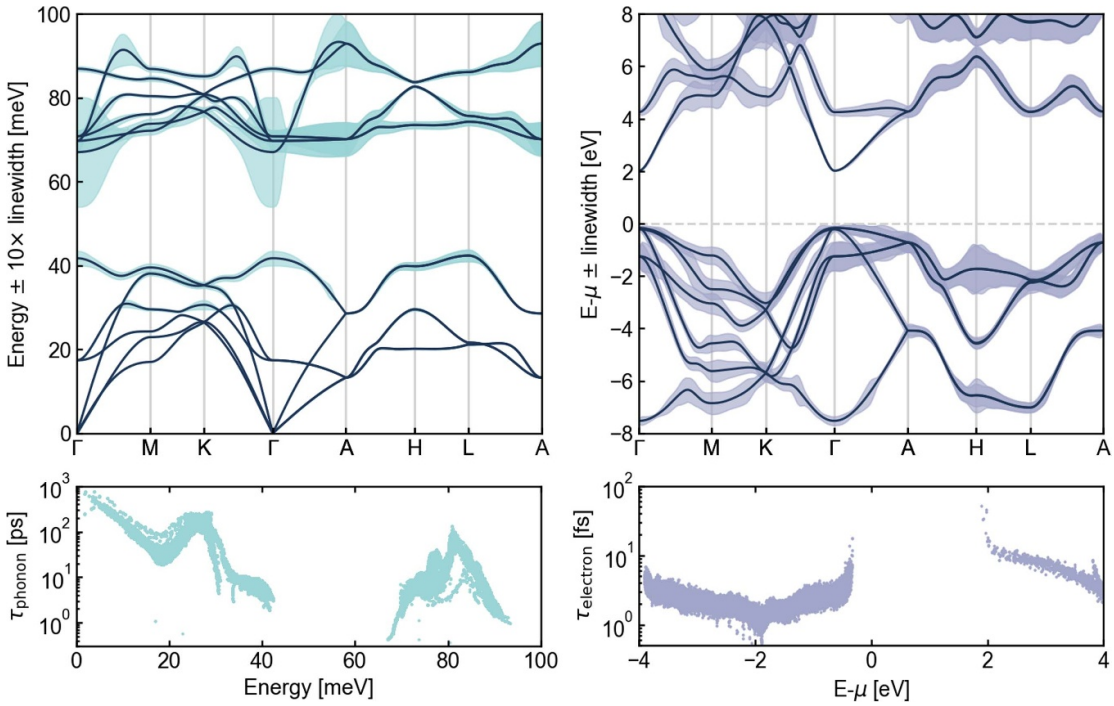

Figure 3. Plots of phonon (left panels) and electron (right panels) linewidths and lifetimes of GaN. In the top panels, linewidths are plotted as the thickness of lines against the band structure of the carrier on a high-symmetry path of the Brillouin zone. In the bottom panels, we plot lifetimes as a function of carrier energy. The reference zero energy corresponds to the chemical potential.

structure. Also in this case, we observe a departure between our results and experiments at the lowest temperatures. Electronic scattering by defects is not expected to change the Seebeck coefficient significantly [74]. Instead, this deviation could be attributed to phonon drag, which for example has been investigated in GaAs [74], and often manifests with a sudden change in the Seebeck coefficient at low temperatures.

In the top panels of figure 3 we plot the phonon (left panels) and electronic (right panels) band structure. On top of the band structure, we superimpose the carrier linewidths computed at a temperature of 300 K. These pictures are produced with `Python` scripts that have been included in the Phoebe release for user's convenience. The top panel provides an easy way to identify some particular bands or areas of the BZ which tend to scatter the most, as the linewidths are a measure of the total scattering rate. In the upper-left panel of 3, we can observe how optical phonons tend to scatter more, especially at the Γ or *A* high-symmetry points. In the upper-right panel of figure 3 instead we can observe how the holes close to the chemical potential have larger linewidths than electrons close to the band gap, but otherwise we do not observe any pocket of large scattering rates. The bottom panels describe the energy-dependence of the lifetimes, which sometimes is useful to derive simplified models of transport. For phonons, in the lower-left panel of figure 3, we can clearly see that lifetimes are largest for the three acoustic phonon modes whose energies tend to zero at the Γ point. The calculated phonon lifetimes tend to decrease as the energy increases. However, one can notice how the energy dependence cannot be described with a simple empirical power-law expression: an accurate microscopic description, as done here, is often necessary to describe phonon lifetimes. In the lower-right panel 3, we instead show the energy dependence of electron lifetimes due to electron–phonon scattering. We observe a spike in the electron lifetimes for carriers closest to the band edge due to the vanishing density of states; often these carriers are responsible for determining the transport properties of a material. Away from the gap, the relaxation times simply show a weaker energy dependence.

6.2. Performance and scaling

Phoebe uses MPI, OpenMP and Kokkos for scaling and acceleration at different levels. MPI enables large-scale simulations on multiple nodes, while OpenMP allows shared-memory parallelization within one node. As explained in section 5.5, the interpolation of the particle interaction is implemented with Kokkos, so that if no GPU is available, Kokkos's OpenMP backend will add to the intra-node parallelization, otherwise its CUDA, HIP, SYCL and OpenMPTarget backends compile the code for GPUs from different vendors.

In this section, we show some examples of the computational performance of Phoebe. As a benchmark, we take the GaN calculation of electronic transport properties discussed in the previous section, and integrate the BTE at a temperature of 300 K on a grid of $90 \times 90 \times 56$ wavevectors. For this test, we reduced

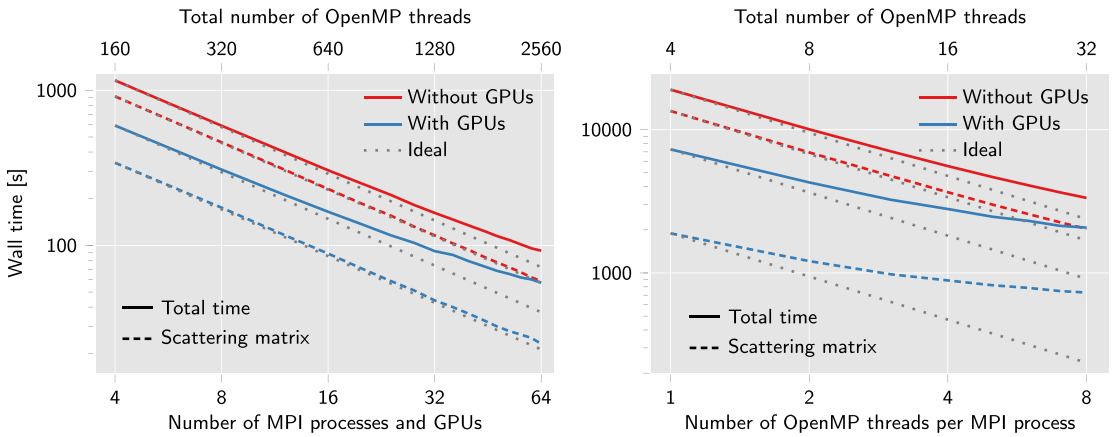

Figure 4. Strong scaling with MPI and OpenMP. The solid lines show the total execution time of Phoebe for the calculation of electronic transport properties of GaN, while the dashed lines show the time used to calculate the scattering matrix. On the left panel, the number of MPI processes is varied while the number of OpenMP threads is kept constant, and vice-versa on the right panel. In blue lines, we show results when the GPU is used (one V100 GPU per MPI process), which is contrasted with results in red lines performed on the same computational nodes but without GPUs. GPUs speedup simulations by approximately a factor of three.

the coarse *ab-initio* grid of wavevectors to $7 \times 7 \times 4$. The simulation is run on the TACC Longhorn supercomputer, where each node has four NVIDIA V100 GPUs and two IBM Power 9 AC922 20-core CPUs.

In figure 4 we plot the scaling of the `ElectronWannierTransportApp` with respect to the two levels of parallelization (MPI and OpenMP), while also demonstrating how the GPU accelerates the calculation (with one GPU per MPI process). The solid lines show the total wall time of the calculation as a function of the number of MPI processes (left panel) or OpenMP threads per MPI thread (right panel), while dashed lines show the wall time for the calculation of the scattering matrix (typically the most computationally intensive algorithm present in the code). Dotted lines represent the ideal scaling of a calculation—a straight line in this logarithmic scale—and the color code labels simulations performed with (blue) or without (red) GPU acceleration.

In the left panel of figure 4, we test the MPI scaling by increasing the number of nodes used, with four MPI ranks, four GPUs and 160 OpenMP threads per node. When running without GPUs, the overall scaling of the simulation is very close to ideal. In this case, the calculation of the scattering matrix dominates the total cost of the simulation and since it is efficiently distributed across MPI processes over the wavevectors, we achieve a good scaling performance. The GPUs, shown in blue lines, greatly accelerate the calculation of the scattering matrix, which is now a much smaller contribution to the overall computation time. Remarkably, we achieve a GPU acceleration of a factor 2 or 3 to the overall simulation wall time. Dashed lines in figure 4 show in detail the time taken by the construction of the scattering matrix. This part, dominant for larger systems and grids, is very close to ideal scaling.

The right panel of figure 4 showcases the scaling performance of Phoebe with respect to the number of OpenMP threads per MPI process. Here, we used a single node with four GPUs and four MPI ranks, while the number of OpenMP threads per MPI rank is gradually increased from 1 to 8. Because not all steps of the calculation can be parallelized with OpenMP (in particular, the input files reading at the start of the calculation), the total simulation time deviates from ideal scaling as more threads are used. Nevertheless, we achieve good OpenMP scaling performance when GPUs are not used, especially for the construction of the scattering matrix. When GPUs are used, we observe a significant deviation from the linear scaling, which can be traced to the lack of OpenMP scaling of the scattering matrix construction. This is not surprising: since the majority of the scattering matrix calculation is offloaded to the GPU, it becomes largely independent of the number of OpenMP threads being used on the CPU, which are only responsible for minor precomputations and data transfer.

Figure 5 shows the performance breakdown of the most significant steps in a single electronic transport calculation. This simulation is computed using a $80 \times 80 \times 50$ k-mesh, and has been run on one node of Harvard University's Cannon supercomputer with four MPI processes and eight OpenMP threads, where each node has four V100 GPUs and two Intel Xeon Gold 6142 16-core CPUs. The simulation is profiled using the KokkosP Profiling tools. A vast majority of the initial simulation time is spent reading input files, in particular the large HDF5 files containing the electron–phonon coupling in real space precomputed from equation (12). While file-reading cannot be accelerated with OpenMP or GPUs, we use parallel HDF5 to

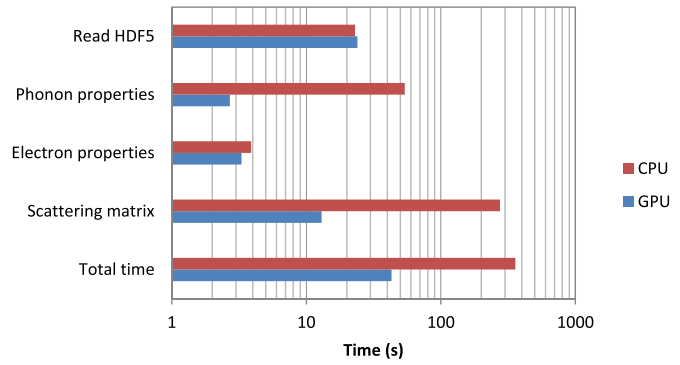

Figure 5. Performance breakdown during a simulation of GaN electronic transport properties, with and without GPUs. The simulation is run on a single node with four MPI ranks, eight OpenMP threads and four GPUs. The first step is to read the HDF5 file, which only depends on file system speed. Phoebe then computes the energies, eigenvectors and velocities for the phonon and electron Hamiltonians at different wavevectors, and finally the scattering matrix. Overall, we find the GPU greatly accelerates the total computation time.

maximize the I/O speed. In the next step of the calculation, Phoebe precomputes electron and phonon properties as described in sections 3.1 and 3.2, i.e. energies, eigenvectors, and velocities at all points $\boldsymbol{k}$, $\boldsymbol{q}$ and $\boldsymbol{k}+\boldsymbol{q}$. This operation is trivially parallelized with both MPI processes and either OpenMP threads or a GPU, thus driving the overall scaling towards linearity. The GPU is particularly efficient for diagonalizing the phonon and electron Hamiltonians at different wavevectors by using the `cusolverDnZheevjBatched` function from the cuSOLVER library, which diagonalizes a batch of small matrices. At this stage, we also precompute all factors in the long-range polar correction (equation (15)), except for the wavefunction overlap which is added during the calculation of the scattering rates. In this way, part of the calculation of the polar correction has a linear scaling with $\boldsymbol{q}$, rather than being embedded in a loop over $\boldsymbol{k}$ during the construction of the scattering matrix. In the last part of the simulation, Phoebe computes the scattering matrix, which is the largest contribution to the total wall time. When the GPU is being used, we can see how the calculation of the scattering matrix is significantly accelerated, to the point where the GPU calculation is about ten times faster than the calculation on CPU, and the GPU simulation is dominated by the cost of I/O. The use of non-blocking MPI reductions (`MPI_Ireduce`) minimizes the time spent on the CPU and further enhances the scaling. The calculation of transport properties after the scattering matrix has been built is done at a negligible cost compared to the operations shown in figure 5.

Overall, we demonstrate that Phoebe successfully leverages multi-level parallelism and GPU acceleration even for a moderately expensive calculation. With just a single GPU node, the calculation is fast enough that it becomes dominated by unavoidable overhead tasks like file reading and data transfer.

7. Conclusion

In this article we present Phoebe, a high-performance software for characterising electron and phonon transport properties from first-principles. It provides various tools to predict transport properties at various levels of theory and accuracy, such as the BTE or models based on the Wigner distribution. Phoebe interfaces with Phono3py and ShengBTE to obtain the first-principles coupling strength of the phonon–phonon interaction, and with QE for the electron–phonon interaction. The accurate integration of the BTE and related transport properties is achieved with Fourier/Wannier based interpolation techniques or with energy-averaging approximations. Lastly, the code is optimized for speed and scaling, using a mix of parallelization levels based on MPI, OpenMP and GPU (through Kokkos). The code has been made publicly available on Github (https://mir-group.github.io/phoebe/) under an open-source basis (MIT license).

The ongoing goal for this project is to provide the community with a scalable software to efficiently perform simulation of materials transport properties, while providing an extensible platform for implementing new methods as they are being developed.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/mir-group/phoebe.

Acknowledgments

We acknowledge funding support from the Star-Friedman Fund for Promising Scientific Research, the Harvard Quantum Initiative, the Harvard Climate Change Solutions Fund, the STC Center for Integrated Quantum Materials, NSF Grant No. DMR-1231319. Computational resources were provided by the Harvard FAS Research Computing and TACC, allocation DMR20013. J C acknowledges support from the Department of Energy Computational Science Graduate Fellowship (DOE CSGF) under Award Number DE-FG02-97ER25308. A J acknowledges support from the Aker Scholarship. N S F acknowledges support from Swiss National Science Foundation (Project Number P2EZP2_178532).

ORCID iDs

Andrea Cepellotti https://orcid.org/0000-0002-8129-1895
Jennifer Coulter https://orcid.org/0000-0002-0620-6558
Natalya S Fedorova https://orcid.org/0000-0001-6189-2266
Boris Kozinsky https://orcid.org/0000-0002-0638-539X

References

[1] Kozinsky B and Singh D J 2021 *Annu. Rev. Mater. Res.* **51** 565
[2] Poncé S, Margine E, Verdi C and Giustino F 2016 *Comput. Phys. Commun.* **209** 116
[3] Pizzi G, Volja D, Kozinsky B, Fornari M and Marzari N 2014 *Comput. Phys. Commun.* **185** 422
[4] Zhou J-J, Park J, Lu I-T, Maliyov I, Tong X and Bernardi M 2021 *Comput. Phys. Commun.* **264** 107970
[5] Madsen G K H, Carrete J and Verstraete M J 2018 *Comput. Phys. Commun.* **231** 140
[6] Togo A and Tanaka I 2015 *Scr. Mater.* **108** 1
[7] Togo A, Chaput L and Tanaka I 2015 *Phys. Rev.* B **91** 094306
[8] Li W, Carrete J, Katcho N A and Mingo N 2014 *Comput. Phys. Commun.* **185** 1747
[9] Tadano T, Gohda Y and Tsuneyuki S 2014 *J. Phys.: Cond. Matter* **26** 225402
[10] Barbalinardo G, Chen Z, Lundgren N W and Donadio D 2020 *J. Appl. Phys.* **128** 135104
[11] Zhang B, Fan Z, Zhao C Y and Gu X 2021 *J. Phys.: Condens. Matter* **33** 495901
[12] Cepellotti A, Coulter J, Johansson A, Fedorova N and Kozinsky B 2021 Phoebe (available at: https://mir-group.github.io/phoebe/)
[13] Guennebaud G *et al* 2010 Eigen v3 (available at: http://eigen.tuxfamily.org)
[14] ScaLAPACK team 2020 ScaLAPACK (available at: https://github.com/Reference-ScaLAPACK/scalapack)
[15] Togo A and Tanaka I 2018 `Spglib`: a software library for crystal symmetry search (arXiv:1808.01590 [cond-mat.mtrl-sci])
[16] Kapoulkine A 2020 Pugixml (available at: https://github.com/zeux/pugixml)
[17] Devresse A 2021 Highfive (available at: https://github.com/BlueBrain/HighFive)
[18] Group T H D F 2000–2010 Hierarchical data format version 5 (available at: www.hdfgroup.org/HDF5)
[19] Carter E H, Trott C R and Sunderland D 2014 *J. Parallel Distrib. Comput.* **74** 3202–16
[20] Dimitri van H 1997–2021 Doxygen, generate documentation from source code (available at: www.doxygen.nl/)
[21] B (The Sphinx Team) 2007–2021 Python documentation generator (available at: www.sphinx-doc.org)
[22] Read The Docs, Inc & contributors 2010–2021 Read the docs (available at: https://readthedocs.org/)
[23] Ziman J 2001 *Electrons and Phonons: The Theory of Transport Phenomena in Solids (Oxford Classic Texts in the Physical Sciences)* (New York: Oxford University Press)
[24] Giannozzi P *et al* 2017 *J. Phys.: Condens. Matter* **29** 465901
[25] Li W, Carrete J, Katcho N A and Mingo N 2014 *Comp. Phys. Commun.* **185** 1747
[26] Baroni S, de Gironcoli S, Dal Corso A and Giannozzi P 2001 *Rev. Mod. Phys.* **73** 515
[27] Fugallo G, Lazzeri M, Paulatto L and Mauri F 2013 *Phys. Rev.* B **88** 045430
[28] Paulatto L, Mauri F and Lazzeri M 2013 *Phys. Rev.* B **87** 214303
[29] Giustino F 2017 *Rev. Mod. Phys.* **89** 015003
[30] Pizzi G *et al* 2020 *J. Phys.: Condens. Matter* **32** 165902
[31] Marzari N, Mostofi A A, Yates J R, Souza I and Vanderbilt D 2012 *Rev. Mod. Phys.* **84** 1419
[32] Souza I, Marzari N and Vanderbilt D 2001 *Phys. Rev.* B **65** 035109
[33] Giustino F, Cohen M L and Louie S G 2007 *Phys. Rev.* B **76** 165108
[34] Verdi C and Giustino F 2015 *Phys. Rev. Lett.* **115** 176401
[35] Blount E 1962 *Solid State Phys.* (New York: Academic) pp 305–73
[36] Peierls R 1929 *Ann. Phys., Lpz.* **395** 1055
[37] Liao B, Qiu B, Zhou J, Huberman S, Esfarjani K and Chen G 2015 *Phys. Rev. Lett.* **114** 115901
[38] Garg J, Bonini N, Kozinsky B and Marzari N 2011 *Phys. Rev. Lett.* **106** 045901
[39] Meija J *et al* 2016 *Pure Appl. Chem.* **88** 293
[40] Fiorentini M and Bonini N 2016 *Phys. Rev.* B **94** 085204
[41] Cepellotti A and Marzari N 2016 *Phys. Rev.* X **6** 041013
[42] Guyer R A and Krumhansl J A 1966 *Phys. Rev.* **148** 766
[43] Fedorova N S, Cepellotti A and Kozinsky B 2022 *Adv. Funct. Mater.* 2111354
[44] Kozinsky B and Singh D J 2021 *Annu. Rev. Mater. Res.* **51** 565
[45] Cepellotti A, Fugallo G, Paulatto L, Lazzeri M, Mauri F and Marzari N 2015 *Nat. Commun.* **6** 6400
[46] Fugallo G, Cepellotti A, Paulatto L, Lazzeri M, Marzari N and Mauri F 2014 *Nano Lett.* **14** 6109
[47] Lindsay L, Broido D A and Mingo N 2010 *Phys. Rev.* B **82** 115427
[48] Omini M and Sparavigna A 1997 Heat transport in dielectric solids with diamond structure. *Nuovo Cimento* D **19** 1537
[49] Omini M and Sparavigna A 1996 *Phys. Rev.* B **53** 9064

[50] Broido D A, Ward A and Mingo N 2005 *Phys. Rev.* B **72** 014308
[51] Hardy R J 1970 *Phys. Rev.* B **2** 1193
[52] Chaput L 2013 *Phys. Rev. Lett.* **110** 265506
[53] Simoncelli M, Marzari N and Cepellotti A 2020 *Phys. Rev.* X **10** 011019
[54] Samsonidze G and Kozinsky B 2018 *Adv. Energy Mater.* **8** 1800246
[55] Vitale V, Pizzi G, Marrazzo A, Yates J R, Marzari N and Mostofi A A 2020 *npj Comput. Mater.* **6** 66
[56] Pickett W E, Krakauer H and Allen P B 1988 *Phys. Rev.* B **38** 2721
[57] Bang S, Kim J, Wee D, Samsonidze G and Kozinsky B 2018 *Mater. Today Phys.* **6** 22
[58] Simoncelli M, Marzari N and Mauri F 2019 *Nat. Phys.* **15** 809
[59] Cepellotti A and Kozinsky B 2021 *Mater. Today Phys.* **19** 100412
[60] Yates J R, Wang X, Vanderbilt D and Souza I 2007 *Phys. Rev.* B **75** 195121
[61] Li W, Mingo N, Lindsay L, Broido D A, Stewart D A and Katcho N A 2012 *Phys. Rev.* B **85** 195436
[62] Blöchl P E, Jepsen O and Andersen O K 1994 *Phys. Rev.* B **49** 16223
[63] Carter Edwards H, Trott C R and Sunderland D 2014 *J. Parallel Distrib. Comput.* **74** 3202–16
[64] Giannozzi P *et al* 2009 *J. Phys.: Condens. Matter* **21** 395502
[65] Hamann D 2013 *Phys. Rev.* B **88** 085117
[66] van Setten M J, Giantomassi M, Bousquet E, Verstraete M J, Hamann D R, Gonze X and Rignanese G-M 2018 *Comput. Phys. Commun.* **226** 39
[67] Perdew J P and Wang Y 1992 *Phys. Rev.* B **45** 13244
[68] Inyushkin A V, Taldenkov A, Chernodubov D A, Voronenkov V V and Shreter Y G 2020 *JETP Lett.* **112** 106
[69] Götz W, Johnson N, Chen C, Liu H, Kuo C and Imler W 1996 *Appl. Phys. Lett.* **68** 3144
[70] Akasaki I, Amano H, Koide Y, Hiramatsu K and Sawaki N 1989 *J. Cryst. Growth* **98** 209
[71] Ilegems M and Montgomery H 1973 *J. Phys. Chem. Solids* **34** 885
[72] Homm G, Klar P, Teubert J and Heimbrodt W 2008 *Appl. Phys. Lett.* **93** 042107
[73] May A F, Singh D J and Snyder G J 2009 *Phys. Rev.* B **79** 153101
[74] Protik N H and Broido D A 2020 *Phys. Rev.* B **101** 075202