



RESEARCH ARTICLE | NOVEMBER 03 2023

Stress and heat flux via automatic differentiation

Special Collection: [Software for Atomistic Machine Learning](#)

Marcel F. Langer   ; J. Thorben Frank  ; Florian Knoop 



J. Chem. Phys. 159, 174105 (2023)

<https://doi.org/10.1063/5.0155760>

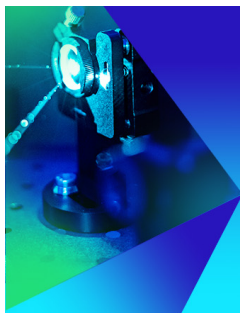


View
Online



Export
Citation

CrossMark



The Journal of Chemical Physics
Special Topic: Time-resolved
Vibrational Spectroscopy

Submit Today

Stress and heat flux via automatic differentiation

Cite as: J. Chem. Phys. 159, 174105 (2023); doi: 10.1063/5.0155760

Submitted: 23 April 2023 • Accepted: 25 September 2023 •

Published Online: 3 November 2023



View Online



Export Citation



CrossMark

Marcel F. Langer,^{1,2,3,a)}  J. Thorben Frank,^{1,2}  and Florian Knoop⁴ 

AFFILIATIONS

¹Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany

²BIFOLD–Berlin Institute for the Foundations of Learning and Data, Berlin, Germany

³The NOMAD Laboratory at the Fritz Haber Institute of the Max Planck Society and Humboldt University, Berlin, Germany

⁴Theoretical Physics Division, Department of Physics, Chemistry and Biology (IFM), Linköping University, SE-581 83 Linköping, Sweden

Note: This paper is part of the JCP Special Topic on Software for Atomistic Machine Learning.

a) Author to whom correspondence should be addressed: mail@marcel.science

ABSTRACT

Machine-learning potentials provide computationally efficient and accurate approximations of the Born–Oppenheimer potential energy surface. This potential determines many materials properties and simulation techniques usually require its gradients, in particular forces and stress for molecular dynamics, and heat flux for thermal transport properties. Recently developed potentials feature high body order and can include equivariant semi-local interactions through message-passing mechanisms. Due to their complex functional forms, they rely on automatic differentiation (AD), overcoming the need for manual implementations or finite-difference schemes to evaluate gradients. This study discusses how to use AD to efficiently obtain forces, stress, and heat flux for such potentials, and provides a model-independent implementation. The method is tested on the Lennard–Jones potential, and then applied to predict cohesive properties and thermal conductivity of tin selenide using an equivariant message-passing neural network potential.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0155760>

I. INTRODUCTION

Molecular dynamics (MD) simulations enable computational prediction of thermodynamic quantities for a wide range of quantum systems, and constitute a cornerstone of modern computational science.¹ In MD, systems are simulated by propagating Newton's equations of motion – potentially modified to model statistical ensembles – numerically in time, based on the forces acting on each atom due to their movement on the Born–Oppenheimer potential-energy surface (PES). Therefore, the quality of the underlying PES is important for the predictive ability of this method. First-principles electronic structure methods such as density-functional theory (DFT) can be used to perform high-accuracy *ab initio* molecular dynamics (aiMD) simulations,² provided the exchange-correlation approximation is reliable.³ Such approaches are, however, restricted by high computational cost, severely limiting accessible size and time scales. Computationally efficient approximations to the underlying PES are therefore required for the atomistic simulation of larger systems: Forcefields (FFs) are built on analytical functional forms that are often based on physical bonding principles, parametrized

to reproduce quantities of interest for a given material.⁴ They are computationally cheap, but parametrizations for novel materials are not always available, and their flexibility is limited by their fixed functional form. Machine-learning interatomic potentials (MLIPs),^{5–11} where a potential is inferred based on a small set of reference calculations, aim to retain near first-principles accuracy while remaining linear scaling with the number of atoms. While MLIPs are limited, in principle, to modeling the physical mechanisms present in the training data, they have nevertheless emerged as an important tool for MD,^{12–16} often combined with active learning schemes.^{17–21} Modern MLIPs can include semi-local interactions through message-passing (MP) mechanisms,²² internal equivariant representations,²³ and body-order expansions,²⁴ which enable the efficient construction of flexible many-body interactions. In such complex architectures, the manual implementation of derivatives is often unfeasible. Finite-difference approaches require tuning of additional parameters, as well as repeated energy evaluations. Automatic differentiation (AD)^{25,26} presents an intriguing alternative: If the computation of the potential energy U is implemented in a suitable framework, derivatives such as the forces F or the stress

σ can be computed with the same asymptotic computational cost as computing the energy. This is accomplished by decomposing the underlying “forward” computation into elementary operations with analytically tractable derivatives, computing local gradients for a given input, and then combining the resulting values using the chain rule.

This work provides a systematic discussion of and an introduction to the use of AD to compute forces, stress, and heat flux for MLIPs. While the calculation of forces with AD is common practice,^{27–30} stress and heat flux are not yet commonly available for many MLIPs. Both quantities have been the focus of much previous work due to the difficulty of defining and implementing them for many-body potentials and periodic boundary conditions.^{31–42} Introducing an abstract definition of MLIPs as functions of a graph representation of atomistic systems, unified formulations of stress and heat flux are given, which can be implemented generally for any such graph-based machine-learning potential (GLP). An example implementation using `jax`⁴³ is provided in the `glp` package.⁴⁴ To validate the approach, different formulations of stress and heat flux are compared for the Lennard-Jones potential,⁴⁵ where analytical derivatives are readily available for comparison, as well as a state-of-the-art message-passing neural network (MPNN), `SO3KRATES`.⁴⁶ Having established the correctness of the proposed implementation, the ability of `SO3KRATES` to reproduce first-principles cohesive properties and thermal conductivity of tin selenide (SnSe) is studied.

II. AUTOMATIC DIFFERENTIATION

Automatic differentiation (AD) is a technique to obtain derivatives of functions implemented as computer programs.^{25,26} It is distinct from *numerical differentiation*, where finite-difference schemes are employed, and *symbolic differentiation*, where analytical derivatives are obtained manually or via computer algebra systems, and then implemented explicitly. Instead, AD relies on the observation that complex computations can often be split up into elementary steps, for which derivatives are readily implemented. If one can track those derivatives during the computation of the forward, or “primal,” function, the chain rule allows to obtain derivatives.

For this work, two properties of AD are particularly relevant: It allows the computation of derivatives with respect to quantities that are explicitly used in the forward computation, and it can do so at the same asymptotic computational cost as the forward function. In particular, AD can obtain two quantities efficiently: Given a differentiable function $u : \mathbb{R}^N \rightarrow \mathbb{R}^M$, the Jacobian of u is defined as the $M \times N$ matrix $\partial u_i(\mathbf{x})/\partial x_j$. AD can then obtain Jacobian-vector and vector-Jacobian products, i.e., the multiplication and summation of factors over either the input or the output dimension. This corresponds to propagating derivatives from the inputs *forwards*, leading to *forward-mode AD*, or from the end result *backwards*, leading to *reverse-mode AD*. As many popular AD frameworks are primarily implemented to work with neural networks, where scalar loss functions must be differentiated with respect to many parameters, reverse-mode AD, also called “backpropagation,”⁴⁷ is more generally available. More recent frameworks implement both approaches, for instance `jax`⁴³ which is used in the present work. AD can also be leveraged to compute contractions of higher-order derivative operators.⁴⁸

III. CONSTRUCTING GRAPH MLIPs

This work considers periodic systems,⁴⁹ consisting of N atoms with atomic numbers Z_i placed in a simulation cell which is infinitely periodically tiled in space. We define

$$\begin{aligned} \mathbb{R}_{\text{sc}} &:= \{ \mathbf{r}_i : i = 1 \dots N \} && \text{positions in simulation cell} \\ \mathbb{Z} &:= \{ Z_i : i = 1 \dots N \} && \text{atomic numbers} \\ \mathbb{B} &:= \{ \mathbf{b}_a : a = 1, 2, 3 \} && \text{basis or lattice vectors} \\ \mathbf{r}_{\text{in}} &:= \mathbf{r}_i + \sum_a n_a \mathbf{b}_a && \text{(replica) position} \\ \mathbb{R}_{\text{all}} &:= \{ \mathbf{r}_{\text{in}} : \mathbf{r}_i \in \mathbb{R}_{\text{sc}}, \mathbf{n} \in \mathbb{Z}^3 \} && \text{all (bulk) positions} \\ \mathbf{r}_{ij} &:= \mathbf{r}_j - \mathbf{r}_i && \text{atom-pair vector} \\ |\mathbf{r}_{ij}^{\text{MIC}}| &:= \min_{\mathbf{n} \in \mathbb{Z}^3} \left| \mathbf{r}_j + \sum_a n_a \mathbf{b}_a - \mathbf{r}_i \right| && \text{minimum image convention} \end{aligned}$$

In this setting, a MLIP is a function that maps the structure represented by its positions, lattice vectors, and atomic numbers, $(\mathbb{R}_{\text{sc}}, \mathbb{B}, \mathbb{Z})$, to a set of atomic potential energies $U := \{U_i : i = 1, \dots, N\}$. Each U_i is a many-body function of positions; in general, we do not assume that a further decomposition into body-ordered contributions is available. The total potential energy is then obtained by summing over atomic contributions: $U = \sum_{i=1}^N U_i$. Since AD relies on the forward computation to calculate derivatives, it is sensitive to the exact implementation of this mapping. Care must be taken to construct the MLIP such that required derivatives are available.

This work considers MLIPs that scale linearly with N . Therefore, the number of atoms contributing to a given U_i must be bounded, which is achieved by introducing a cutoff radius r_c , restricting interactions to finite-sized atomic neighborhoods $\mathbb{N}(i) = \{ \mathbf{r}_j : |\mathbf{r}_{ij}| \leq r_c, \mathbf{r}_j \in \mathbb{R}_{\text{all}} \}$. To ensure translational invariance, MLIPs do not rely on neighbor positions directly, but rather on atom-pair vectors centered on i , from which atom-pair vectors between neighboring atoms can be constructed, for instance to determine angles.

The resulting structure can be seen as a graph G . The vertices V of this graph are identified with atoms, labeled with their respective atomic numbers, and connected by edges E that are labeled with atom-pair vectors if placed closer than r_c . Starting from G , MLIPs can be constructed in different ways: Local MLIPs compute a suitable representation⁵⁰ of each neighborhood, and predict U_i from that representation using a learned function, such as a neural network or a kernel machine. Such models are conceptually simple, but cannot account for effects that extend beyond r_c . Recently, semi-local models such as MPNNs^{22,29,51–58} have been introduced to tackle this shortcoming without compromising asymptotic runtime. In such models, effective longer-range interactions are built up iteratively by allowing adjacent neighborhoods to interact repeatedly. We introduce the parameter M , the *interaction depth*, to quantify how many such iterations are included. After M interactions, the energy at any given site can depend implicitly on positions within M hops on the graph, which we denote by $\mathbb{N}^M(i)$, leading to an *effective cutoff radius* $r_c^{\text{eff}} = M r_c$. Atomic potential energy contributions can therefore depend on positions up to a distance of r_c^{eff} . Since interactions are confined to neighborhoods at each iteration, the computational cost of semi-local models nevertheless remains linear in N and M .

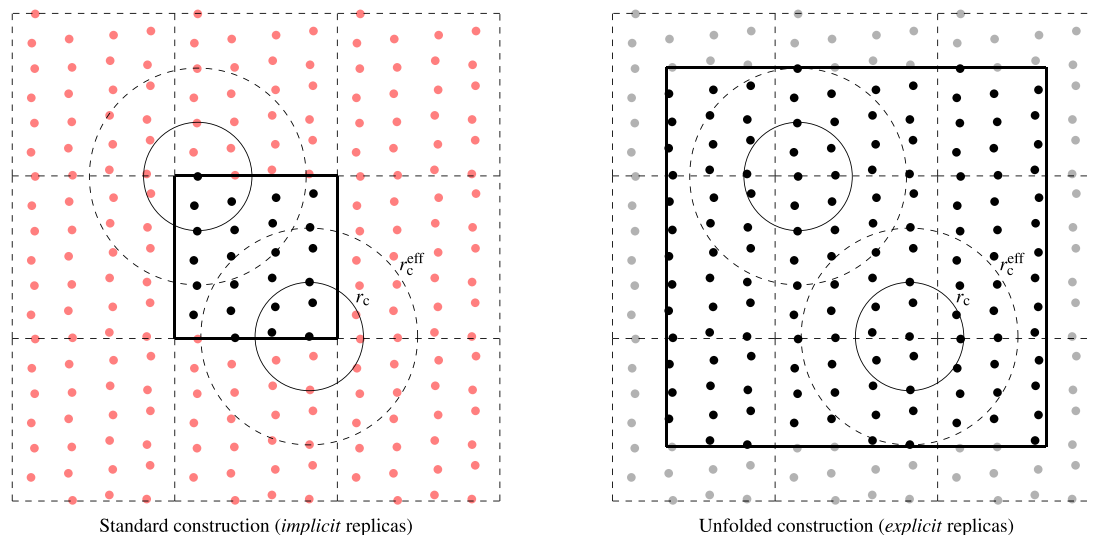


FIG. 1. Standard (left) and unfolded (right) implementation of a periodic system. The left figure shows a simulation cell (solid boundary, black circles), as well as the replicas implicitly considered through the use of the minimum image convention (dashed boundaries, red circles). For selected atoms, the cutoff radius r_c (solid circle) and the effective cutoff radius r_c^{eff} (dashed circle) after two steps of message passing is shown. The right figures shows the corresponding unfolded simulation cell, where all positions that can interact with atoms in the simulation cell have been explicitly included (solid boundaries, black circles).

We term this class of potentials, which act on sets of neighborhoods and use atom-pair vectors as input, graph-based machine-learning potentials (GLPs). Models with $M = 1$ are termed local, those with $M > 1$ semi-local. By construction, this framework does not include global interactions: GLPs are still local at thermodynamic scales. Nevertheless, the distinction between local and semi-local models is useful for the purpose of classifying the functional dependence of atomic potential energy contributions, which in turn determines the derivatives available via AD: In a local model, U_i is a function only of atom-pair vectors within its neighborhood, and derivatives of U with respect to a given \mathbf{r}_{ij} therefore uniquely map to one U_i . This is not the case for semi-local models and will be discussed further in Sec. IV.

We consider two strategies to construct GLPs: The “standard” way, which includes periodic boundary conditions via the edges in the graph, and an “unfolded” formulation, where periodicity is explicitly included via replica positions. Both concepts are illustrated in Fig. 1.

In the standard architecture, vertices in G are identified with atoms in the simulation cell, using the minimum image convention (MIC) to include periodicity. Edges E_{ij} exist between two atoms i and j in R_{sc} if they interact:

$$E_{ij}^{\text{std}} = \{ \mathbf{r}_{ij}^{\text{MIC}} : |\mathbf{r}_{ij}^{\text{MIC}}| \leq r_c \}. \quad (1)$$

We denote the graph constructed in this manner as G^{std} and the set of edges E^{std} .

Alternatively, we can first determine the total set of positions $R_{\text{unf}} \subset R_{\text{all}}$ that can interact with atoms in the simulation cell, creating an unfolded system extracted from the bulk, consisting of R_{sc} and all replicas with up to r_c^{eff} distance from the cell boundary. This construction can be performed efficiently, and adds only a number

of positions that is proportional to the surface of the simulation cell, therefore becoming increasingly negligible as N increases at constant density.⁴² However, the number of positions increases cubically with M , and this construction is therefore practical only for small M (see supplementary material).

We proceed by constructing a correspondingly modified graph G^{unf} , and compute potential energies for vertices corresponding to atoms in the simulation cell only. By construction, since the same atom-pair vectors appear in the graph, this approach then reproduces the potential energy of the standard method. A similar construction appears in spatial decomposition approaches for multi-device parallelization, discussed for instance in Ref. 59: To contain the calculation of interactions in one device, all interacting positions outside a given domain must be considered as “ghost atoms.” The number of ghost atoms corresponds to the number of additional atoms due to the unfolding. As this approach becomes unfeasible for large values of r_c^{eff} , alternative schemes must be considered, for instance propagating message-passing updates, and later gradient information, between devices.

IV. DERIVATIVES

Having constructed the forward function for a given GLP, we can compute derivatives with respect to its inputs using AD. In this section, we discuss how forces, stress, and heat flux can be computed in this manner, and demonstrate the relationship between different formulations.

A. Forces

For MD, the most relevant quantities are the forces

$$\mathbf{F}_i = -\frac{\partial U}{\partial \mathbf{r}_i} \quad (2)$$

acting on the atoms in the simulation cell. Since R_{sc} are an explicit input, they can be computed directly with AD: U is a scalar, and this therefore is a trivial Jacobian-vector product, which can be computed with the same asymptotic cost as U .

An interesting situation arises if pairwise forces are desired. Strictly speaking, in a many-body MLIP, where interactions cannot be decomposed into pairwise contributions, such quantities are not well-defined and Newton's third law is replaced by conservation of momentum, which requires $\sum_{i=1}^N \mathbf{F}_i = 0$. Nevertheless, pairwise forces with an antisymmetric structure can be defined by exploiting the construction of GLPs in terms of atom-pair vectors. In the standard formulation, U is a function of all edges,

$$U = U(\{ \mathbf{r}_{ij} : ij \in E^{\text{std}} \}). \quad (3)$$

Hence, by the chain rule,

$$\mathbf{F}_i = \sum_{j \in N(i)} \frac{\partial U}{\partial \mathbf{r}_{ij}} - \frac{\partial U}{\partial \mathbf{r}_{ji}} \quad (4)$$

$$=: \sum_{j \in N(i)} \mathbf{F}_{ij}. \quad (5)$$

The pairwise forces such defined exhibit anti-symmetry, and therefore fulfil Newton's third law. For $M = 1$, the local case, this definition reduces to a more standard form³⁷

$$\mathbf{F}_{ij} = \frac{\partial U_i}{\partial \mathbf{r}_{ij}} - \frac{\partial U_j}{\partial \mathbf{r}_{ji}}. \quad (6)$$

However, for general GLPs with $M > 1$, this definition includes a sum over all U_k that are influenced by a given edge

$$\mathbf{F}_{ij} = \sum_{k \in N^M(i)} \frac{\partial U_k}{\partial \mathbf{r}_{ij}} - \frac{\partial U_k}{\partial \mathbf{r}_{ji}}, \quad (7)$$

subverting expectations connecting local potential energies to pairwise forces. We note that this seeming contradiction is a consequence of the combination of the peculiar construction of GLPs and AD: In principle, it is always possible to define extended neighborhoods up to r_c^{eff} , obtaining U_i purely as a function of atom-pair vector originating from i . However, to construct derivatives with respect to these atom-pair vectors with AD, these extended neighborhoods have to be constructed and included explicitly, therefore negating the computational efficiency gains of a GLP architecture.

B. Stress

The definition of the (potential) stress is⁶⁰

$$\boldsymbol{\sigma} = \frac{1}{V} \frac{\partial U}{\partial \boldsymbol{\epsilon}} \Big|_{\boldsymbol{\epsilon}=0}, \quad (8)$$

with U denoting the potential energy after a strain transformation with the 3×3 symmetric tensor $\boldsymbol{\epsilon}$

$$\mathbf{r} \rightarrow (\mathbb{1} + \boldsymbol{\epsilon}) \cdot \mathbf{r}, \quad (9)$$

acting on R_{all} .

One approach, followed for instance by `schnetpack`,^{28,61} `schnetpack 2.0`,³⁰ and `nequip`,^{54,62} is to inject the strain transformation explicitly into the construction of the GLP. This can be done at different points: One can transform R_{sc} and B before constructing G , directly transform atom-pair vectors, or transform all contributing positions R_{unf} . Alternatively, as the inner derivative of inputs with respect to $\boldsymbol{\epsilon}$ is simply the input, the derivative of U with respect to inputs can be obtained with AD, and the stress can then be computed analytically from the results. This avoids modifying the forward computation of U entirely.

These approaches yield, with \otimes denoting an outer product,

$$V\boldsymbol{\sigma} = \frac{\partial U(R_{sc}, B)}{\partial \boldsymbol{\epsilon}} \quad (10)$$

$$= \frac{\partial U(E)}{\partial \boldsymbol{\epsilon}} \quad (11)$$

$$= \frac{\partial U(R_{\text{unf}})}{\partial \boldsymbol{\epsilon}} \quad (12)$$

$$= \sum_{i \in R_{sc}} \mathbf{r}_i \otimes \frac{\partial U}{\partial \mathbf{r}_i} + \sum_{b \in B} \mathbf{b} \otimes \frac{\partial U}{\partial \mathbf{b}} \quad (13)$$

$$= \sum_{ij \in E} \mathbf{r}_{ij} \otimes \frac{\partial U}{\partial \mathbf{r}_{ij}} \quad (14)$$

$$= \sum_{i \in R_{\text{unf}}} \mathbf{r}_i \otimes \frac{\partial U}{\partial \mathbf{r}_i}, \quad (15)$$

recovering previous formulations.^{31,33} Louwse and Baerends³¹ considered the calculation of the stress for periodic systems, highlighting that contributions from replica positions must be considered, either by computing the elementary definition in Eq. (10) or the inclusion of the second term in Eq. (13). Both options are considered "much effort" in that work. Thompson *et al.*³³ discuss an alternative approach based on explicitly including replica positions, obtaining a result similar to the unfolded construction used for Eq. (15).

All these forms of the stress are equivalent and can be implemented directly with AD, obtaining numerically identical results as seen in Tables I and IV. In all cases, as the scalar U is differentiated with respect to its inputs, asymptotic cost remains linear.

However, as AD computes algorithmic derivatives, which reflect the particular implementation of the energy function exactly, care must be taken to ensure that the strain transformation in Eq. (9) is applied consistently. For example, obtaining a correct stress with Eqs. (10) and (13), where derivatives are obtained before the construction of the input graph, requires that the implementation of the MIC used matches the definition in Sec. III. If the MIC is implemented using fractional coordinates,⁶³ this can be challenging. In such cases, it may be preferable to rely on approaches that compute derivatives on the basis of the graph input, such as Eqs. (11) and (14). This method is adopted, for instance, by `jax-md`.²⁷ Many other MLIP packages use offsets to implement the MIC, and can therefore use Eq. (10), for instance `schnetpack`,⁶¹ `schnetpack 2.0`,³⁰ `nequip`,⁶² and `mace`.⁶⁴ The `matgl` package,⁶⁵ on the other hand, uses Eq. (14), avoiding the introduction of a strain transformation.

TABLE I. Error in stress for Lennard-Jones argon, comparing different formulations, as well as finite differences, to analytical derivatives. Results are shown for both single and double precision arithmetic, and for $\sigma \cdot V$ in place of σ . Errors per component can be found in the supplementary material.

Equations	Single		Double	
	MAE (eV)	MAPE (%)	MAE (eV)	MAPE (%)
Fin.diff.	7.70×10^{-4}	1.04×10^{-1}	1.13×10^{-6}	1.18×10^{-4}
(10)	1.19×10^{-5}	1.79×10^{-3}	3.15×10^{-6}	3.69×10^{-4}
(11)	8.25×10^{-6}	1.27×10^{-3}	3.15×10^{-6}	3.69×10^{-4}
(12)	9.17×10^{-6}	1.36×10^{-3}	3.15×10^{-6}	3.69×10^{-4}
(13)	1.18×10^{-5}	1.79×10^{-3}	3.15×10^{-6}	3.69×10^{-4}
(14)	8.22×10^{-6}	1.27×10^{-3}	3.15×10^{-6}	3.69×10^{-4}
(15)	9.16×10^{-6}	1.37×10^{-3}	3.15×10^{-6}	3.69×10^{-4}

Additional difficulty arises if strain derivatives of atomic energies, i.e., atomic stresses

$$\sigma_i := \frac{1}{V} \frac{\partial U_i}{\partial \epsilon} \quad i \in R_{sc} \quad (16)$$

are required. Their calculation requires either one backward pass per U_i , or one forward pass for each entry in ϵ . If only reverse-mode AD is available, its evaluation therefore scales quadratically with N . Linear scaling is retained with forward mode. For GLPs with $M = 1$, linear scaling in reverse mode can be recovered by using Eq. (14): Every edge can be uniquely assigned to one U_i , and therefore the derivatives can be used to construct atomic stresses. For $M > 1$, this is not possible; similar to the observations of Sec. IV A, atomic stresses take a semi-local form.

C. Heat flux

Finally, we discuss the heat flux, which is required to compute thermal conductivities with the Green–Kubo (GK) method.^{66–68} It describes how energy flows between atoms, and has been the focus of a large body of previous work.^{32,35,37,40–42,69}

The fundamental definition of the heat flux for MLIPs was originally derived by Hardy⁷⁰ for periodic quantum systems. It reads⁴²

$$\mathbf{J}_{\text{full}} = \sum_{\substack{i \in R_{sc} \\ j \in R_{\text{all}}}} \left(\mathbf{r}_{ji} \left(\frac{\partial U_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right) + \sum_{i \in R_{sc}} E_i \mathbf{v}_i \quad (17)$$

$$=: \mathbf{J}_{\text{pot}} + \mathbf{J}_{\text{conv}}, \quad (18)$$

where \mathbf{v}_i denote velocities, m_i masses, and $E_i = U_i + 1/2m_i\mathbf{v}_i^2$ is the total energy per atom. Intuitively, the “potential” term \mathbf{J}_{pot} describes how the total instantaneous change in U_i can be attributed to interactions with other atoms, with energy flowing between them, while the second, “convective,” term \mathbf{J}_{conv} describes energy being carried by individual atoms. In the present setting, \mathbf{J}_{conv} can be computed directly, as E_i are available. \mathbf{J}_{pot} , however, presents a challenge in an AD framework: In principle, \mathbf{J}_{pot} could be computed directly, obtaining the required partial derivatives with AD. However, as \mathbf{J}_{pot} is neither a Jacobian-vector nor a vector-Jacobian product, this

requires repeated evaluations over the input or output dimension. Even when restricting $j \in R_{sc}$, which can be achieved by introducing the MIC for \mathbf{r}_{ji} (see supplementary material for details),

$$\mathbf{J}_{\text{pot}}^{\text{MIC}} = \sum_{ij \in R_{sc}} \left(\mathbf{r}_{ji}^{\text{MIC}} \left(\frac{\partial U_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right), \quad (19)$$

computational cost of a direct implementation with AD scales quadratically with N , rendering the system sizes and simulation times required for the GK method inaccessible.⁴² Based on previous work,⁴² we therefore survey approaches that restore linear scaling in the following.

For $M = 1$, edges can be uniquely assigned to atomic energy contributions as discussed for atomic stresses in Sec. IV B. In this case

$$\frac{\partial U}{\partial \mathbf{r}_{ij}} = \sum_{k \in R_{sc}} \frac{\partial U_k}{\partial \mathbf{r}_{ij}} = \frac{\partial U_i}{\partial \mathbf{r}_j} = \frac{\partial U_i}{\partial \mathbf{r}_j}, \quad (20)$$

so that

$$\mathbf{J}_{\text{pot}}^{M=1} = \sum_{ij \in E} \left(\mathbf{r}_{ji} \left(\frac{\partial U}{\partial \mathbf{r}_{ij}} \cdot \mathbf{v}_j \right) \right), \quad (21)$$

which requires a single evaluation of reverse-mode AD.

We note that the terms appearing in front of \mathbf{v}_j also appear in the stress in Eq. (14). However, for a given j , the pre-factor *cannot* be identified with the atomic stress as defined in Eq. (16) – the atomic energy being differentiated is not U_j , but U_i . The indices can only be exchanged for additive pairwise potentials; this inequivalence was recently corrected in the large-scale atomic/molecular massively parallel simulator (LAMMPS) code.⁴⁰

This approach is not applicable for $M > 1$, since the relation in Eq. (20) no longer holds, and the mapping between stress contributions and heat flux contributions becomes invalid. Therefore, Eq. (19) must be used, incurring quadratic computational cost.

To restore linear scaling, one must find a way to rewrite Eq. (19) in terms of either Jacobian-vector or vector-Jacobian products by executing the sum over i before taking the derivative with respect to j . The use of the MIC for the $\mathbf{r}_{ji}^{\text{MIC}}$ prefactors prevents this: An index-dependent offset appears in $\mathbf{r}_{ji}^{\text{MIC}} = \mathbf{r}_i - \mathbf{r}_j + \sum_a n_a^{ji} \mathbf{b}_a$. Adopting the unfolded construction avoids this issue, as replica positions are now included explicitly. The heat flux can then be split into two terms, which can be rewritten in a form suitable for AD.⁴²

Introducing auxiliary positions $\mathbf{r}_i^{\text{aux}}$, which are numerically identical to the positions \mathbf{r}_i , but not used to compute U , and defining the energy barycenter $\mathbf{B} = \sum_{i \in R_{sc}} \mathbf{r}_i^{\text{aux}} U_i$, the heat flux becomes

$$\mathbf{J}_{\text{pot}}^{M \geq 1} = \sum_{j \in R_{\text{unf}}} \frac{\partial \mathbf{B}}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j - \sum_{j \in R_{\text{unf}}} \left(\mathbf{r}_j \left(\frac{\partial U}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right). \quad (22)$$

The first term requires three reverse-mode evaluations, or one forward-mode evaluation. Correspondingly, the second term requires one reverse-mode, or three forward-mode, evaluations. Since the overhead introduced by explicitly constructing R_{unf} scales as $N^{2/3}$, overall linear scaling is restored, albeit with a pre-factor due to the higher number of positions to be considered.

To summarize, we have introduced two forms of the heat flux that can be implemented efficiently with AD: Eq. (21), which applies

TABLE II. Error in heat flux for Lennard-Jones argon, comparing different formulations to analytical derivatives. Results are shown for both single and double precision arithmetic.

Equations	Single		Double	
	MAE (eV Å/fs)	MAPE (%)	MAE (eV Å/fs)	MAPE (%)
(19)	2.81×10^{-9}	1.71×10^{-2}	1.47×10^{-10}	6.81×10^{-4}
(21)	2.84×10^{-9}	1.67×10^{-2}	1.47×10^{-10}	6.81×10^{-4}
(22)	2.44×10^{-9}	1.54×10^{-2}	1.47×10^{-10}	6.81×10^{-4}

to GLPs with $M = 1$, and Eq. (22), which applies for $M \geq 1$, but introduces some additional overhead. Both are equivalent to the general, quadratically-scaling, form given in Eq. (17), as seen in Tables II and VI.

V. EXPERIMENTS

A. Lennard-Jones argon

The stress formulas in Eqs. (10)–(15) and heat flux formulas Eqs. (19), (21), and (22) have been implemented in the `glp` package⁴⁴ using `jax`.⁴³ As a first step, we numerically verify this implementation.

To this end, the Lennard-Jones potential⁴⁵ is employed, where analytical derivatives including those required for the heat flux are readily available, and implementations are included in many packages, for example the atomic simulation environment (ASE).⁷¹ In the GLP framework, the Lennard-Jones potential can be seen as an extreme case of a $M = 1$ GLP, where U_i is composed of a sum of pair terms:

$$U_i = \frac{1}{2} \sum_{j \in N(i)} 4\epsilon \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right). \quad (23)$$

For this experiment, parameters approximating elemental argon are used.⁷² 100 randomly displaced and distorted geometries,

based on the 512-atom $8 \times 8 \times 8$ supercell of the face-centered cubic primitive cell with lattice parameter 3.72 Å and angle 60° are used. Random velocities to evaluate a finite heat flux are sampled from the Boltzmann distribution corresponding to 10 K. Additional computational details are discussed in the supplementary material.

Table I compares the stress formulations in Eqs. (10)–(15) with finite differences. We report “best-case” results for finite differences, choosing the stepsize that minimizes the error. In the table, the mean absolute error (MAE) and mean absolute percentage error (MAPE) with respect to the analytical ground truth are reported. All given formulations are found to be equivalent. In single precision arithmetic, the AD-based implementations outperform finite differences, in double precision, errors are similar.

For the heat flux, finite difference approaches are not feasible. Therefore, only AD-based implementations are shown in Table II. In the case of the Lennard-Jones potential, where $M = 1$, Eqs. (19), (21), and (22) are found to be identical.

Having confirmed the numerical correctness of the presented implementations, we then proceed with a benchmark of the computation time of Eqs. (10)–(15). Here, we construct cubic supercells of varying sizes, generating 1000 randomly displaced samples; additional details are given in the supplementary material. The results, computed on one Nvidia V100 graphics processing unit (GPU), are given in Table III: Overall, computation times per atom are similar for AD-based approaches. Computation time per atom decreases with system size, as calculations are parallelized on the GPU. In this case, stress formulations based on the unfolded construction are on par with other approaches, indicating that in this setting, computational cost is dominated by graph construction, rather than the evaluation of the potential itself. For more complex potentials, as seen in Sec. VI, this is not the case, and the introduction of additional positions yields significant computational cost. The best performance in single precision is obtained for Eq. (14), which avoids the use of a strain transformation, and re-uses gradients computed for the forces. For double precision, Eq. (15) is optimal, which avoids the use of the MIC or a strain transformation, and also re-uses gradients.

TABLE III. Time per atom to compute stress with finite differences and different AD-based formulations for Lennard-Jones argon. Results are shown for both single and double precision, and for different simulation cell sizes N .

Equations	Single			Double		
	Time per atom (s)			Time per atom (s)		
	$N = 864$	$N = 2048$	$N = 4000$	$N = 864$	$N = 2048$	$N = 4000$
F.d.	3.1×10^{-5}	1.6×10^{-5}	9.9×10^{-6}	3.8×10^{-5}	2.1×10^{-5}	1.4×10^{-5}
(10)	1.0×10^{-6}	7.1×10^{-7}	6.0×10^{-7}	1.8×10^{-6}	1.3×10^{-6}	1.1×10^{-6}
(11)	7.7×10^{-7}	5.1×10^{-7}	4.5×10^{-7}	1.3×10^{-6}	9.8×10^{-7}	8.0×10^{-7}
(12)	9.2×10^{-7}	5.4×10^{-7}	3.7×10^{-7}	1.1×10^{-6}	6.7×10^{-7}	4.9×10^{-7}
(13)	9.4×10^{-7}	6.7×10^{-7}	5.8×10^{-7}	1.6×10^{-6}	1.3×10^{-6}	1.0×10^{-6}
(14)	7.2×10^{-7}	4.5×10^{-7}	3.6×10^{-7}	1.2×10^{-6}	9.0×10^{-7}	7.1×10^{-7}
(15)	8.9×10^{-7}	5.3×10^{-7}	3.7×10^{-7}	1.0×10^{-6}	6.5×10^{-7}	4.6×10^{-7}

VI. TIN SELENIDE WITH SO3KRATES

To investigate stress and heat flux in a practical setting, we now study tin selenide (SnSe) using the state-of-the-art SO3KRATES GLP.⁴⁶ In contrast to other equivariant MLIPs, for instance NEQUIP,⁵⁴ SO3KRATES replaces shared equivariant feature representations by separated branches for invariant and equivariant information, whose information exchange is handled using an equivariant self-attention mechanism. By doing so, one can achieve data efficiency and extrapolation quality competitive to state-of-the-art GLPs at reduced time and memory complexity. As non-local interactions are not modeled in the GLP framework introduced in this work, global interactions are disabled in the SO3KRATES models used at present.

The training data for this task is obtained from a large-scale *ab initio* Green–Kubo (aiGK) benchmark study by Knoop *et al.*,^{73,74} and consists of four aiMD trajectories in the NVT ensemble at 300 K and different volumes, from which a total of 3000 timesteps are used. These simulations were performed as the first steps of the aiGK workflow, first determining the equilibrium volume at 300 K and then generating thermalized starting geometries for the long-running NVE simulations required for the GK method. For this experiment, we repurpose these thermalization trajectories as training data, aiming to replace the subsequent, computationally

expensive, aiMD with MLIP-accelerated MD. Additional details on the MLIP training can be found in the supplementary material.

A. Implementation of stress and heat flux

While no analytical derivatives are available for SO3KRATES, the implementation of the stress can be verified with finite differences, and the heat flux can be checked for consistency between

TABLE IV. Error in stress for tin selenide, comparing different formulations to finite differences, for SO3KRATES with $M = 2$. Results are shown for both single and double precision arithmetic, and for $\sigma \cdot V$ in place of σ . Results for other M , which are similar to the one shown here, can be found in the supplementary material.

Equations	Single		Double	
	MAE (eV)	MAPE (%)	MAE (eV)	MAPE (%)
(10)	1.58×10^{-2}	4.40×10^{-2}	1.45×10^{-4}	2.32×10^{-4}
(11)	1.58×10^{-2}	4.38×10^{-2}	1.45×10^{-4}	2.32×10^{-4}
(12)	1.57×10^{-2}	4.33×10^{-2}	1.45×10^{-4}	2.32×10^{-4}
(13)	1.58×10^{-2}	4.40×10^{-2}	1.45×10^{-4}	2.32×10^{-4}
(14)	1.58×10^{-2}	4.38×10^{-2}	1.45×10^{-4}	2.32×10^{-4}
(15)	1.57×10^{-2}	4.33×10^{-2}	1.45×10^{-4}	2.32×10^{-4}

TABLE V. Time per atom to compute stress with finite differences and different AD-based formulations, for tin selenide and SO3KRATES with $M = 2$. Results are shown for both single and double precision, and for different simulation cell sizes N .

Equations	Single			Double		
	Time per atom (s)			Time per atom (s)		
	$N = 864$	$N = 2048$	$N = 4000$	$N = 864$	$N = 2048$	$N = 4000$
F.d.	1.2×10^{-4}	9.5×10^{-5}	8.8×10^{-5}	2.2×10^{-4}	1.9×10^{-4}	1.8×10^{-4}
(10)	8.2×10^{-6}	7.1×10^{-6}	6.9×10^{-6}	1.6×10^{-5}	1.5×10^{-5}	1.5×10^{-5}
(11)	8.0×10^{-6}	7.0×10^{-6}	6.9×10^{-6}	1.6×10^{-5}	1.5×10^{-5}	1.5×10^{-5}
(12)	3.5×10^{-5}	2.5×10^{-5}	2.0×10^{-5}	8.4×10^{-5}	5.7×10^{-5}	4.6×10^{-5}
(13)	8.0×10^{-6}	7.1×10^{-6}	6.9×10^{-6}	1.6×10^{-5}	1.5×10^{-5}	1.5×10^{-5}
(14)	8.0×10^{-6}	7.0×10^{-6}	6.8×10^{-6}	1.6×10^{-5}	1.5×10^{-5}	1.5×10^{-5}
(15)	3.5×10^{-5}	2.5×10^{-5}	2.0×10^{-5}	8.4×10^{-5}	5.7×10^{-5}	4.6×10^{-5}

TABLE VI. Error in heat flux for tin selenide, comparing different formulations to the baseline implementation in Eq. (19) for SO3KRATES models with differing numbers of interaction steps M . Results are shown for both single and double precision arithmetic.

Equations	M	Single		Double	
		MAE (eV Å/fs)	MAPE (%)	MAE (eV Å/fs)	MAPE (%)
(21)	1	5.78×10^{-9}	9.74×10^{-4}	1.09×10^{-17}	1.73×10^{-12}
(22)	1	8.75×10^{-8}	2.65×10^{-2}	1.69×10^{-16}	4.31×10^{-11}
(21)	2	3.73×10^{-3}	5.84×10^2	3.73×10^{-3}	5.84×10^2
(22)	2	9.36×10^{-8}	1.00×10^{-2}	1.54×10^{-16}	1.60×10^{-11}
(21)	3	1.01×10^{-3}	3.27×10^2	1.01×10^{-3}	3.25×10^2
(22)	3	9.74×10^{-8}	3.04×10^{-2}	1.65×10^{-16}	2.91×10^{-11}

different implementations. Similar to Sec. V A, we use 100 randomly displaced and distorted $4 \times 8 \times 8$ supercells of the 0 K primitive cell of SnSe for this experiment, sampling velocities from the Boltzmann distribution at 10 K to evaluate a finite heat flux.

Table IV compares the stress implementations in Eqs. (10)–(15) with finite differences, confirming the equivalence of all implemented formulations. Table V shows the computation times for the stress, for SO3KRATES with $M = 2$; additional tables for $M = 1, 3$ can be found in the supplementary material. In contrast to the results for the Lennard-Jones potential, all AD-based approaches using the standard construction display similar performance, while the unfolded formulations, Eqs. (12) and (15), are significantly slower to compute, with the additional computational cost proportional to the number of unfolded positions. We conclude that in this setting, using a state-of-the-art MLIP, the computational cost of evaluating the potential dominates implementation differences in the stress, and all standard methods can be used.

Table VI compares the heat flux formulations in Eqs. (21) and (22) with the baseline in Eq. (19), implementing the quadratically-scaling “Hardy” heat flux with the MIC. For $M = 1$, all formulations are precisely equivalent. For $M > 1$, the semi-local case, only Eq. (22) is equivalent to the “Hardy” heat flux; Eq. (21) does not apply and consequently is not equivalent, displaying large deviations.

B. Equation of state and pressure

To assess the capability of SO3KRATES to predict stress- and pressure-related materials properties, we calculate energy-volume and pressure-volume curves for SnSe to obtain an equation of state (EOS) of the Vinet form.^{75,76} The experiment is performed for unit cells that were homogeneously strained up to $\pm 2\%$ starting from the fully relaxed geometry, and relaxing the internal degrees of freedom afterwards. The energy vs volume curves for SO3KRATES with $M = 1, 2, 3$ interaction steps are shown in Fig. 2 in comparison to the DFT reference using the PBEsol exchange-correlation functional

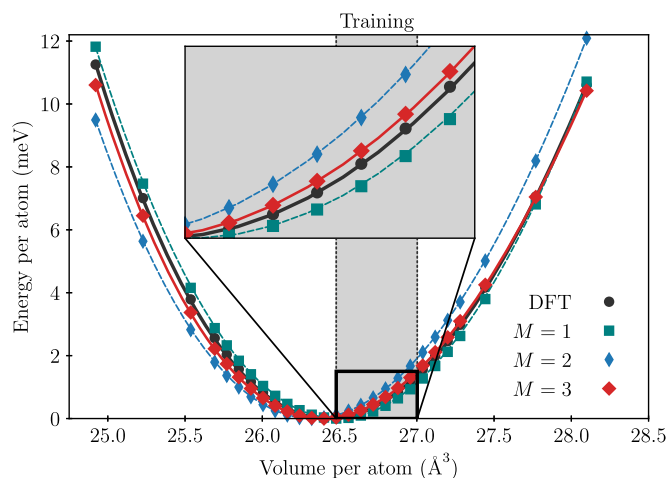


FIG. 2. EOS (energy vs volume) computed with PBEsol-DFT (black dots) compared to SO3KRATES with different numbers of interaction steps M . The connecting lines have been obtained by fitting the Vinet EOS. Inset: Zoom into the region of volumes that were covered during the training indicated by the gray shading.

TABLE VII. Cohesive properties of SnSe for PBEsol-DFT, and SO3KRATES with different numbers of interaction steps M obtained via the Vinet EOS. Best values are highlighted. Values are in good agreement with existing literature.⁷⁹

	DFT	$M = 1$	$M = 2$	$M = 3$
B_0 (eV/Å ³)	0.230	0.236	0.229	0.223
B'_0 (eV/Å ⁶)	5.576	2.830	4.815	7.118
V_0 (Å ³)	26.429	26.489	26.322	26.388
Error B_0 (%)	...	2.58	-0.47	-3.14
Error B'_0 (%)	...	-49.25	-13.64	27.66
Error V_0 (%)	...	0.22	-0.40	-0.16

with “light” default basis sets in FHI-aims.^{77,78} SO3KRATES with three interaction steps ($M = 3$) yields the best visual agreement for the energy-volume curve in Fig. 2, and the best equilibrium volume. To quantify the agreement, we evaluate the Vinet EOS and extract the cohesive properties equilibrium volume V_0 , isothermal Bulk modulus B_0 , and its volume derivative B'_0 (functional forms are given in the supplementary material). Results are listed in Table VII. For SO3KRATES with three interaction steps ($M = 3$), the predicted volume deviates by -16% from the DFT reference, and the deviation of the bulk modulus is -3.14% . A larger error is seen for the volume derivative of the bulk modulus, B'_0 , which deviates by 27.7% , indicating worse agreement further away from the training region. Overall, the agreement between DFT and SO3KRATES when predicting cohesive properties can be considered satisfactory. The energy-volume predictions are very good, and transfer even to volumes that are larger or smaller than the ones seen during training.

Finally we check the internal consistency of energy and stress predictions with SO3KRATES by fitting energy-volume and pressure-volume curves and verify that they yield identical parameters for the EOS. Results are shown in Fig. 3. As seen there, results are in perfect agreement, which verifies that stress and resulting pressure are consistent with the underlying energy function as expressed in Eq. (8).

C. Thermal conductivity

Finally, we proceed to GK calculations, following the approach outlined in our previous work.^{42,83} Finding that simulation cells

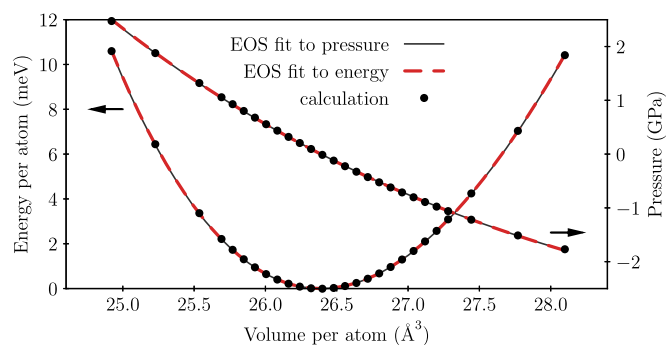


FIG. 3. Comparison of EOS obtained by fitting energy-volume (red dashed) and pressure-volume (black solid) data. The results are in perfect agreement.

TABLE VIII. Thermal conductivity κ of tin selenide at 300 K.

Source	Method	κ (W/mK)	κ_x	κ_y	κ_z
This work	SO3KRATES, $M = 1$	0.99 ± 0.10	0.53 ± 0.03	1.31 ± 0.13	1.12 ± 0.12
	SO3KRATES, $M = 2$	1.13 ± 0.07	0.48 ± 0.04	1.59 ± 0.07	1.20 ± 0.07
	SO3KRATES, $M = 3$	1.13 ± 0.10	0.56 ± 0.05	1.56 ± 0.15	1.32 ± 0.16
Brorsson <i>et al.</i> ⁸⁰	FCP	1.12	0.57	1.46	1.32
Liu <i>et al.</i> ⁸¹	MLIP	0.86 ± 0.13	0.57 ± 0.05	1.25 ± 0.24	0.76 ± 0.08
Knoop <i>et al.</i> ⁷³	DFT (extrapolated)	1.40 ± 0.39
Review by Wei <i>et al.</i> ⁸²	Experiments	0.45–1.9

with $N = 864$ atoms and a simulation duration of $t_0 = 1$ ns yield converged results (see supplementary material), we run 11 MD simulations using the `g1p` package and `SO3KRATES`, computing J at every step.

Table VIII shows the result: For $M = 2, 3$, `SO3KRATES` in excellent agreement with the work by Brorsson *et al.*,⁸⁰ which uses a force constant potential (FCP). A larger difference is observed with the work by Liu *et al.*,⁸¹ who, however, use the Perdew–Burke–Ernzerhof (PBE) exchange–correlation functional, as opposed to PBEsol, which was used for the present work. The observed thermal conductivity is consistent with experiments, as well as the size-extrapolated DFT result of Knoop *et al.*⁷³ The anisotropy of κ in SnSe is captured as well.

Overall, we find that `SO3KRATES` with more than one interaction step is able to predict the converged thermal conductivity of SnSe, using training data *only* from the thermalization step of a full aiGK workflow. With this approach, long-running equilibrium aiMD simulations, the bottleneck of the GK method, have been avoided.

VII. CONCLUSION

We explained how and demonstrated that the stress and heat flux can be computed efficiently with AD for potentials based on a graph of atom-pair vectors, which we termed GLPs, and provided example implementations in the `g1p` package.⁴⁴ In order to provide this functionality, `g1p` also implements the necessary infrastructure to generate input graphs from structures, as well as the ability to compute forces. With this, it can be used as a lightweight driver for MD. An example implementation of the velocity Verlet integrator⁸⁴ is therefore also provided. This functionality is extended in the `gkx` package,⁸⁵ which implements the GK method using GLP.

Numerical experiments for Lennard–Jones argon and tin selenide with the `SO3KRATES` GLP, verified that these quantities are computed correctly, efficiently, and consistently. The equivariant `SO3KRATES` GLP was shown to predict cohesive properties and thermal conductivity of SnSe in good agreement with DFT, other MLIPs, and experiments, confirming the practical relevance of computational access to stress and heat flux.

This work enables the use of a large class of recently developed MLIPs, those which can be described in the GLP framework, in computational materials science, and in particular for the calculation of thermal conductivities using the GK method. For GLPs implemented with `jax`, the `g1p` package is provided to enable

the calculation of stress and heat flux without requiring further implementation efforts.

SUPPLEMENTARY MATERIAL

The supplementary material contains (1) an explanation of Eq. (19), (2) additional details and results for the numerical experiments in Sec. V, (3) information about the GK workflow and convergence, and (4) explicit forms for the equation of state.

ACKNOWLEDGMENTS

M.F.L. was supported by the German Ministry for Education and Research BIFOLD program (Ref. Nos. 01IS18025A and 01IS18037A), and by the TEC1p Project (ERC Horizon 2020 Grant No. 740233). M.F.L. would like to thank Samuel Schoenholz and Niklas Schmitz for constructive discussions, Shuo Zhao for feedback on the manuscript, and acknowledges contributions by Fabian Nagel and Adam Norris. J.T.F. acknowledges support from the Federal Ministry of Education and Research (BMBF) and BIFOLD program (Ref. Nos. 01IS18025A and 01IS18037A). F.K. acknowledges support from the Swedish Research Council (VR) Program No. 2020-04630, and the Swedish e-Science Research Centre (SeRC). The computations were partially enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre (NSC) and by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at NSC partially funded by the Swedish Research Council through Grant Agreement No. 2022-06725. Xuan Gu at NSC is acknowledged for technical assistance on the Berzelius resource.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Marcel F. Langer: Conceptualization (lead); Investigation (equal); Project administration (lead); Software (lead); Visualization (equal); Writing – original draft (lead); Writing – review & editing (lead).
J. Thorben Frank: Investigation (equal); Software (supporting); Writing – original draft (supporting); Writing – review & editing

(supporting). **Florian Knoop**: Investigation (equal); Visualization (equal); Writing – original draft (supporting); Writing – review & editing (supporting).

DATA AVAILABILITY

The data and code that support the findings of this study are available at <https://doi.org/10.5281/zenodo.8406532> or <https://github.com/sirmarcel/glp-archive>. The `glp` package is available at <https://github.com/sirmarcel/glp>, the `SO3KRATES` model is implemented in `mlff` at <https://github.com/thorben-frank/mlff>. Further information can also be found at <https://marcel.science/glp>.

REFERENCES

- 1 M. E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation* (Oxford University Press, 2010).
- 2 R. Car and M. Parrinello, “Unified approach for molecular dynamics and density-functional theory,” *Phys. Rev. Lett.* **55**, 2471–2474 (1985).
- 3 A. M. Teale, T. Helgaker, A. Savin, C. Adamo, B. Aradi, A. V. Arbniznikov, P. W. Ayers, E. J. Baerends, V. Barone, P. Calaminici, E. Cancès, E. A. Carter, P. K. Chattaraj, H. Chermette, I. Ciofini, T. D. Crawford, F. De Proft, J. F. Dobson, C. Draxl, T. Frauenheim, E. Fromager, P. Fuentealba, L. Gagliardi, G. Galli, J. Gao, P. Geerlings, N. Gidopoulos, P. M. W. Gill, P. Gori-Giorgi, A. Görling, T. Gould, S. Grimme, O. Gritsenko, H. J. A. Jensen, E. R. Johnson, R. O. Jones, M. Kaupp, A. M. Köster, L. Kronik, A. I. Krylov, S. Kvaal, A. Laestadius, M. Levy, M. Lewin, S. Liu, P.-F. Loos, N. T. Maitra, F. Neese, J. P. Perdew, K. Pernal, P. Pernot, P. Piecuch, E. Rebolini, L. Reining, P. Romaniello, A. Ruzsinszky, D. R. Salahub, M. Scheffler, P. Schwerdtfeger, V. N. Staroverov, J. Sun, E. Tellgren, D. J. Tozer, S. B. Trickey, C. A. Ullrich, A. Vela, G. Vignale, T. A. Wesolowski, X. Xu, and W. Yang, “DFT exchange: Sharing perspectives on the workhorse of quantum chemistry and materials science,” *Phys. Chem. Chem. Phys.* **24**, 28700–28781 (2022).
- 4 M. A. González, “Force fields and molecular dynamics simulations,” *Éc. Thématique Soc. Fr. Neutronique* **12**, 169–200 (2011).
- 5 S. Lorenz, A. Groß, and M. Scheffler, “Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks,” *Chem. Phys. Lett.* **395**, 210–215 (2004).
- 6 G. Li, J. Hu, S.-W. Wang, P. G. Georgopoulos, J. Schoendorf, and H. Rabitz, “Random sampling-high dimensional model representation (RS-HDMR) and orthogonality of its different order component functions,” *J. Phys. Chem. A* **110**, 2474–2485 (2006).
- 7 J. Behler and M. Parrinello, “Generalized neural-network representation of high-dimensional potential-energy surfaces,” *Phys. Rev. Lett.* **98**, 146401 (2007).
- 8 A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, “Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons,” *Phys. Rev. Lett.* **104**, 136403 (2010).
- 9 S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, “Towards exact molecular dynamics simulations with machine-learned force fields,” *Nat. Commun.* **9**, 3887 (2018).
- 10 S. Chmiela, H. E. Sauceda, I. Poltavsky, K.-R. Müller, and A. Tkatchenko, “sGDML: Constructing accurate and data efficient molecular force fields using machine learning,” *Comput. Phys. Commun.* **240**, 38–45 (2019).
- 11 I. Poltavsky and A. Tkatchenko, “Machine learning force fields: Recent advances and remaining challenges,” *J. Phys. Chem. Lett.* **12**, 6551–6564 (2021).
- 12 F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, “Machine learning for molecular simulation,” *Annu. Rev. Phys. Chem.* **71**, 361–390 (2020).
- 13 O. T. Unke, D. Koner, S. Patra, S. Käser, and M. Meuwly, “High-dimensional potential energy surfaces for molecular simulations: From empiricism to machine learning,” *Mach. Learn.: Sci. Technol.* **1**, 013001 (2020).
- 14 J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko, “Combining machine learning and computational chemistry for predictive insights into chemical systems,” *Chem. Rev.* **121**, 9816–9872 (2021).
- 15 O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, “Machine learning force fields,” *Chem. Rev.* **121**, 10142–10186 (2021).
- 16 O. T. Unke, M. Stöhr, S. Gansch, T. Unterthiner, H. Maennel, S. Kashubin, D. Ahlin, M. Gastegger, L. Medrano Sandonas, A. Tkatchenko, and K.-R. Müller, “Accurate machine learned quantum-mechanical force fields for biomolecular simulations,” *arXiv:2205.08306* (2022).
- 17 G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, “Learn on the fly”: A hybrid classical and quantum-mechanical molecular dynamics simulation,” *Phys. Rev. Lett.* **93**, 175503 (2004).
- 18 R. Jinnouchi, K. Miwa, F. Karsai, G. Kresse, and R. Asahi, “On-the-fly active learning of interatomic potentials for large-scale atomistic simulations,” *J. Phys. Chem. Lett.* **11**, 6946–6955 (2020).
- 19 C. Verdi, F. Karsai, P. Liu, R. Jinnouchi, and G. Kresse, “Thermal transport and phase transitions of zirconia by on-the-fly machine-learned interatomic potentials,” *npj Comput. Mater.* **7**, 156 (2021).
- 20 C. van der Oord, M. Sachs, D. P. Kovács, C. Ortner, and G. Csányi, “Hyperactive learning for data-driven interatomic potentials,” *npj Comput. Mater.* **9**, 168 (2023).
- 21 Y. Xie, J. Vandermause, S. Ramakers, N. H. Protik, A. Johansson, and B. Kozinsky, “Uncertainty-aware molecular dynamics from Bayesian active learning for phase transformations and thermal transport in SiC,” *npj Comput. Mater.* **9**, 36 (2023).
- 22 J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia, 6–11 August 2017* (Proceedings of Machine Learning Research, 2017), pp. 1263–1272.
- 23 N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, “Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds,” in *NeurIPS 2018 Workshop on Machine Learning for Molecules and Materials* (Curran Associates, Montréal, 2018).
- 24 R. Drautz, “Atomic cluster expansion for accurate and transferable interatomic potentials,” *Phys. Rev. B* **99**, 014104 (2019).
- 25 A. Griewank and A. Walther, *Evaluating Derivatives* (Society for Industrial and Applied Mathematics, 2008).
- 26 A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: A survey,” *J. Mach. Learn. Res.* **18**, 153 (2018).
- 27 S. Schoenholz and E. D. Cubuk, “JAX, M.D. A framework for differentiable physics,” in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual* (Curran Associates, 2020), pp. 11428–11441.
- 28 K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, “SchNetPack: A deep learning toolbox for atomistic systems,” *J. Chem. Theory Comput.* **15**, 448–455 (2019).
- 29 O. T. Unke and M. Meuwly, “PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges,” *J. Chem. Theory Comput.* **15**, 3678–3693 (2019).
- 30 K. T. Schütt, S. S. P. Hessmann, N. W. A. Gebauer, J. Lederer, and M. Gastegger, “SchNetPack 2.0: A neural network toolbox for atomistic machine learning,” *J. Chem. Phys.* **158**, 144801 (2023).
- 31 M. J. Louwerse and E. J. Baerends, “Calculation of pressure in case of periodic boundary conditions,” *Chem. Phys. Lett.* **421**, 138–141 (2006).
- 32 D. Torii, T. Nakano, and T. Ohara, “Contribution of inter- and intramolecular energy transfers to heat conduction in liquids,” *J. Chem. Phys.* **128**, 044504 (2008).
- 33 A. P. Thompson, S. J. Plimpton, and W. Mattson, “General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions,” *J. Chem. Phys.* **131**, 154107 (2009).
- 34 S. G. Volz and G. Chen, “Molecular dynamics simulation of thermal conductivity of silicon nanowires,” *Appl. Phys. Lett.* **75**, 2056–2058 (1999).
- 35 Y. Chen, “Local stress and heat flux in atomistic systems involving three-body forces,” *J. Chem. Phys.* **124**, 054113 (2006).
- 36 A. Guajardo-Cuellar, D. B. Go, and M. Sen, “Evaluation of heat current formulations for equilibrium molecular dynamics calculations of thermal conductivity,” *J. Chem. Phys.* **132**, 104111 (2010).
- 37 Z. Fan, L. F. C. Pereira, H.-Q. Wang, J.-C. Zheng, D. Donadio, and A. Harju, “Force and heat current formulas for many-body potentials in molecular dynamics

- simulations with applications to thermal conductivity calculations,” *Phys. Rev. B* **92**, 094301 (2015).
- ³⁸A. Marcolongo, P. Umari, and S. Baroni, “Microscopic theory and quantum simulation of atomic heat transport,” *Nat. Phys.* **12**, 80–84 (2016).
- ³⁹C. Carbogno, R. Ramprasad, and M. Scheffler, “*Ab initio* Green-Kubo approach for the thermal conductivity of solids,” *Phys. Rev. Lett.* **118**, 175901 (2017).
- ⁴⁰P. Boone, H. Babaei, and C. E. Wilmer, “Heat flux for many-body interactions: Corrections to LAMMPS,” *J. Chem. Theory Comput.* **15**, 5579–5587 (2019).
- ⁴¹D. Surblys, H. Matsubara, G. Kikugawa, and T. Ohara, “Application of atomic stress to compute heat flux via molecular dynamics for systems with many-body interactions,” *Phys. Rev. E* **99**, 051301 (2019).
- ⁴²M. F. Langer, F. Knoop, C. Carbogno, M. Scheffler, and M. Rupp, “Heat flux for semilocal machine-learning potentials,” *Phys. Rev. B* **108**, L100302 (2023).
- ⁴³J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: Composable transformations of Python+NumPy programs,” <https://github.com/google/jax> (2018).
- ⁴⁴See <https://github.com/sirmarcel/glp/> for the glp software library.
- ⁴⁵J. E. Lennard-Jones, “On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature,” *Proc. R. Soc. London, Ser. A* **106**, 441–462 (1924).
- ⁴⁶J. Thorben Frank, O. T. Unke, and K.-R. Müller, “So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems,” in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (Curran Associates, New Orleans, LA, 2022), pp. 29400–29413.
- ⁴⁷D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature* **323**, 533–536 (1986).
- ⁴⁸N. F. Schmitz, K.-R. Müller, and S. Chmiela, “Algorithmic differentiation for automated modeling of machine learned force fields,” *J. Phys. Chem. Lett.* **13**, 10183–10189 (2022).
- ⁴⁹Non-periodic systems can be formally accommodated by setting B such that replicas lie outside the interaction cutoff radius. The glp framework supports non-periodic systems.
- ⁵⁰M. F. Langer, A. Goeßmann, and M. Rupp, “Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning,” *npj Comput. Mater.* **8**, 41 (2022).
- ⁵¹K. T. Schütt, P.-J. Kindermans, H. E. Saucedo, S. Chmiela, A. Tkatchenko, and K.-R. Müller, “SchNet: A continuous-filter convolutional neural network for modeling quantum interactions,” in *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (Curran Associates, Los Angeles, CA, 2017), pp. 992–1002.
- ⁵²K. T. Schütt, H. E. Saucedo, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet—A deep learning architecture for molecules and materials,” *J. Chem. Phys.* **148**, 241722 (2018).
- ⁵³J. Klicpera, J. Groß, and S. Günnemann, “Directional message passing for molecular graphs,” in *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia, 2020.
- ⁵⁴S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, “E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials,” *Nat. Commun.* **13**, 2453 (2022).
- ⁵⁵O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Saucedo, and K.-R. Müller, “SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects,” *Nat. Commun.* **12**, 7273 (2021).
- ⁵⁶I. Batatia, D. P. Kovács, G. N. C. Simm, C. Ortner, and G. Csányi, “MACE: Higher order equivariant message passing neural networks for fast and accurate force fields,” in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (Curran Associates, New Orleans, LA, 2022), pp. 11423–11436.
- ⁵⁷I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. C. Simm, R. Drautz, C. Ortner, B. Kozinsky, and G. Csányi, “The design space of E(3)-equivariant atom-centered interatomic potentials,” [arXiv:2205.06643](https://arxiv.org/abs/2205.06643) (2022).
- ⁵⁸A. Bochkarev, Y. Lysogorskiy, C. Ortner, G. Csányi, and R. Drautz, “Multilayer atomic cluster expansion for semilocal interactions,” *Phys. Rev. Res.* **4**, L042019 (2022).
- ⁵⁹S. Plimpton, “Fast parallel algorithms for short-range molecular dynamics,” *J. Comput. Phys.* **117**, 1–19 (1995).
- ⁶⁰F. Knuth, C. Carbogno, V. Atalla, V. Blum, and M. Scheffler, “All-electron formalism for total energy strain derivatives and stress tensor components for numeric atom-centered orbitals,” *Comput. Phys. Commun.* **190**, 33–50 (2015).
- ⁶¹See <https://github.com/atomistic-machine-learning/schnetpack/> for the SchNetPack software library.
- ⁶²See <https://github.com/mir-group/nequip> for the NequIP software library.
- ⁶³W. Smith, “The minimum image convention in non-cubic MD cells,” <https://citeseerx.ist.psu.edu/doc/10.1.1.57.1696> (1989).
- ⁶⁴See <https://github.com/ACESuit/mace> for the mace software library.
- ⁶⁵See <https://github.com/materialsvirtuallab/matgl> for the matgl software library.
- ⁶⁶M. S. Green, “Markoff random processes and the statistical mechanics of time-dependent phenomena,” *J. Chem. Phys.* **20**, 1281–1295 (1952).
- ⁶⁷R. Kubo, “Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems,” *J. Phys. Soc. Jpn.* **12**, 570–586 (1957).
- ⁶⁸R. Kubo, M. Yokota, and S. Nakajima, “Statistical-mechanical theory of irreversible processes. II. Response to thermal disturbance,” *J. Phys. Soc. Jpn.* **12**, 1203–1211 (1957).
- ⁶⁹N. C. Admal and E. B. Tadmor, “Stress and heat flux for arbitrary multibody potentials: A unified framework,” *J. Chem. Phys.* **134**, 184106 (2011).
- ⁷⁰R. J. Hardy, “Energy-flux operator for a lattice,” *Phys. Rev.* **132**, 168 (1963).
- ⁷¹A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, “The atomic simulation environment—A Python library for working with atoms,” *J. Phys.: Condens. Matter* **29**, 273002 (2017).
- ⁷²A. J. H. McGaughey and M. Kaviani, “Thermal conductivity decomposition and analysis using molecular dynamics simulations. Part I. Lennard-Jones argon,” *Int. J. Heat Mass Transfer* **47**, 1783–1798 (2004).
- ⁷³F. Knoop, T. A. R. Purcell, M. Scheffler, and C. Carbogno, “Anharmonicity in thermal insulators: An analysis from first principles,” *Phys. Rev. Lett.* **130**, 236301 (2023).
- ⁷⁴Dataset aiGK_for_anharmonic_solids on the NOMAD repository, available at <https://doi.org/10.17172/nomad/2021.11.11-1>
- ⁷⁵P. Vinet, J. Ferrante, J. H. Rose, and J. R. Smith, “Compressibility of solids,” *J. Geophys. Res.* **92**, 9319, <https://doi.org/10.1029/jb092ib09p09319> (1987).
- ⁷⁶M. Hebbache and M. Zemzemi, “*Ab initio* study of high-pressure behavior of a low compressibility metal and a hard material: Osmium and diamond,” *Phys. Rev. B* **70**, 224107 (2004).
- ⁷⁷V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, “*Ab initio* molecular simulations with numeric atom-centered orbitals,” *Comput. Phys. Commun.* **180**, 2175–2196 (2009).
- ⁷⁸J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke, “Restoring the density-gradient expansion for exchange in solids and surfaces,” *Phys. Rev. Lett.* **100**, 136406 (2008).
- ⁷⁹C. Lamuta, D. Campi, L. Pagnotta, A. Dasadia, A. Cupolillo, and A. Politano, “Determination of the mechanical properties of SnSe, a novel layered semiconductor,” *J. Phys. Chem. Solids* **116**, 306–312 (2018).
- ⁸⁰J. Brorsson, A. Hashemi, Z. Fan, E. Fransson, F. Eriksson, T. Ala-Nissila, A. V. Krasheninnikov, H.-P. Komsa, and P. Erhart, “Efficient calculation of the lattice thermal conductivity by atomistic simulations with *ab initio* accuracy,” *Adv. Theory Simul.* **5**, 2100217 (2021).

⁸¹H. Liu, X. Qian, H. Bao, C. Y. Zhao, and X. Gu, “High-temperature phonon transport properties of SnSe from machine-learning interatomic potential,” *J. Phys.: Condens. Matter* **33**, 405401 (2021).

⁸²P.-C. Wei, S. Bhattacharya, J. He, S. Neeleshwar, R. Podila, Y. Y. Chen, and A. M. Rao, “The intrinsic thermal conductivity of SnSe,” *Nature* **539**, E1–E2 (2016).

⁸³F. Knoop, M. Scheffler, and C. Carbogno, “*Ab initio* Green-Kubo simulations of heat transport in solids: Method and implementation,” *Phys. Rev. B* **107**, 224304 (2023).

⁸⁴L. Verlet, “Computer ‘experiments’ on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules,” *Phys. Rev.* **159**, 98–103 (1967).

⁸⁵See <https://github.com/sirmarcel/gkx/> for the gkx software library.