ON COMPUTING VIRIAL STRESS TENSOR WITH AUTOMATIC DIFFERENTIATION [*]

WUJIE WANG [†]

Abstract.

Automatic differentiation (AD) has been the key computational technique in modern machine learning to optimize model with few outputs but large number of parameters and inputs. In this paper, we report that such technique is also natural in computing gradients in molecular simulations where important dynamical observables like forces and the virial stress tensor which are gradients of the energy functions. We reformulate the process of computing forces and virials as the backward call of the forward energy computation. We show that using the source-to-source AD, fast calculation of gradient and forces can be generated at a similar cost with hand-derived derivatives. We show the calculation on potentials like Lennard-Jones potentials and Graph Neural Networks. We also showcase an example we optimize a periodic solid cells with the gradient calculated obtained from AD.

Key words. Thermodynamics, Automatic Differentiation

1. Introduction.

At the atomic level, physical processes are governed by differential equations containing many degrees of freedom. Macroscopic phenomena in matter emerge from microscopic interactions that can be simulated through numerical integration of the equations of motion. In classical simulations, these equations of motion are derived from a Hamiltonian quantity H. Examples of microscopic quantities from simulations are time series of positions, velocities, and forces on atoms and molecules. From these, a rich family of macroscopic observables can be calculated to describe the configurational and temporal correlation functions of atoms. These observables determine different properties of the simulated materials.

Recent advances have proposed many machine learning approaches. Notable examples include applying Graph Neural Nets [4] and Kernel methods [1] to parameterize potential energy functions in a data-driven way. These methods shows powerful interpolating method to learn from atomic forces and energies from the expensive electronic structure calculations of the molecules and materials. Similarly classical force field decompose the energy terms that capture classical two-body, three-body, dispersion, and electrostatic interactions. In contrast with the data-driven and machine learning methodology, the energy terms are simple functional forms derived from physical principles.

With these accelerated approaches, the dynamics of systems can be simulated at larger scale and faster speed, but no particular attention has been paid to the calculation of Virial tensor and stress given a ML potentials under periodic boundary system. The calculation of virial provides important information for bulk thermodynamic properties like compressibility and shear for solids and liquid systems. These macroscopic observables are important quantities to probe and compare to study the state and mechanical properties of matter. As previous studies haven shown, computing virial stress requires much care under periodic boundary because the relative positions of the atoms to consider the boundary of the simulation box are subtle, and highly depends on the interaction topology of the atoms involved. In this note, we show the complication can be lifted thanks to the use of automatic differentiation.

This article is organized as follows. We first introduce the basic calculation of

[*] This report is developed as a class project for the course 18.337 at MIT

[†] Department of Materials Science and Engineering , MIT

potential energy of simulated molecular systems, and how interaction topology and how the periodic boundary condition affects the computation of molecular interactions. We then introduce the thermodynamic definition and calculation of forces and virials. We derive that numerical calculation of these quantities as computing the adjoints in the language of reverse automatic differentiation and show that automatic differentiation can be used to calculate quantities like forces and virials efficiently. We perform numerical experiments on two systems: classical pairwise interactions and Graph Neural Nets with 3D coordinates as inputs.

2. Theory.

2.1. Inter-atomic Potentials. Integrating out the fast electronic degrees of freedom, classical molecular dynamics simulates the motion of atoms in their nuclear coordinates to probe the fundamental processes in life and material. The simulation solves the Hamiltonian equations of motion:

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \qquad \frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \tag{2.1}$$

where p_i and q_i are the respective momentum and position of the i^{th} particle. H is the Hamiltonian of the system and is the sum of the kinetic energy $K(p_i)$ and the potential energy $V(q_i)$. For conservative systems, it is given by the sum of kinetic energy and the potential energy. Much of the effort goes into coming up with a good approximation for $V(q_i)$, which is also referred as the potential energy surface (PES) of the system. High quality description of PES requires approximating the solution to the Schrodinger equation via the Density functional theory (DFT) which has $O(N^4)$ cost. There are many data-driven method that provides a fast function approximation to the DFT PES ground truth as a cost of accuracy. These methods include classical force fields and machine learning force fields and they have approximately $O(N^2)$ cost.

2.2. Periodic boundary condition. Simulating atoms in bulk to understand the macroscopic behavior of matter requires simulating Avogadro number of atoms, which are computationally prohibiting. It is usually sufficient to assume that all the important fluctuations and behaviors are contained within a scale of 100 Angstroms so that simulating 10^3 to 10^4 number of atoms are sufficient, and the thermodynamic description is captured by simulating an infinitely periodic images of the box. Therefore, one can extract meaningful thermodynamic properties such as pressure. Such computational techniques are named as the periodic boundary condition. Figure Figure 1 describes the concept of the periodic boundary conditions. However, handling the atomic interactions at the boundary of the box requires special care, because the atom at the boundary is also seeing atoms at the other side of the simulation box, and the distance between atom pairs are characterized with the minimum image convention which require the distances between different atoms smaller half of the box dimension, and should be shifted based on the box coordinates based on the cell matrix $\mathbf{C} = v_1 \;\; v_2 \;\; v_3$ where v_i are the basis to span a simulation box in 3 dimensions.

For example, a cubic simulation box is simply 3 by 3 matrix with length of box dimension on the diagonal of the matrix. It is important to note that the basis do not need to be orthogonal. Specially chosen basis vectors can represent different crystallography symmetry. To illustrate the calculation, let us define x_i to the atomic positions in the box. During the simulation, once an atom move beyond the designed box defined by $\mathbf{C}$, it will be shifted back opposite to the direction which the atom crosses the box boundary so that all the atoms are confined in the designated box boundary.

To carry out the calculation for the shift matrix to satisfy minimum image convention, it is usually helpful to convert the atomic coordinate in the cell coordinate system in the basis of $v_1 \;\; v_2 \;\; v_3$. We defined the reduced coordinate to be X_i which the atomic coordinates defined as a fraction of of the cell vectors

$$\mathrm{X}_i = \mathbf{C}^{-1} x_i$$

To satisfy the minimum image convention, one needs to compare the pair distance vector defined as $\mathrm{R_{ij}} = \mathrm{X}_i$ with $[\frac{1}{2}\frac{1}{2}\frac{1}{2}]$. If the any elements in $\mathrm{R_{ij}}$ is larger than $\frac{1}{2}$ or smaller than $-\frac{1}{2}$, the reduced coordinates X_i will be shifted by 1 and -1 respectively. Such shifting array $\mathrm{H_i}$ are sparse arrays with entry 1 and -1 which indicates the direction of shifting for different interaction topology. The minimum image convention defines the relative position of the atoms given specific interactions. It is important to note that such convention also applies beyond two bodies (e.g. triplet, quadruplet, etc.). However for interaction topology that requires computing the many-body effect (more than 2 atoms), the relative geometric parameters (such as distances and angles) are computed based on a reference atom which do not shift, but other atoms are shifted to satisfy the minimum image convention. Depending on the choice reference atoms of the interaction topology, an atom can be shifted differently[5].

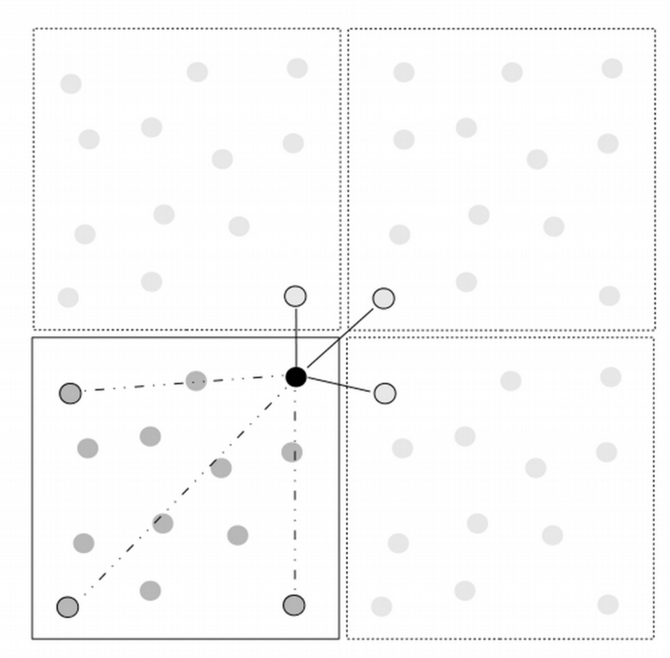

Fig. 1. *A 2D illustration to explain the minimum image convention for the periodic boundary condition. Infinite periodic replica of the simulated systems are considered where the pair interactions at the boundary of the box are kept below half of the box dimension under displacement*

2.3. Virial Calculation. Classically the Virial stress tensor are defined base on pairwise forces f_ij and relative positions r_{ija} where the third index a represents

3D coordinate basis in the x, y, z directions. The Virial stress tensor excluding the temperature dependent part is defined as the following [5]:

$$\boldsymbol{\sigma} = \sigma_{ab} = -\frac{N}{V}\sum_{i<j} f_{ija} r_{ijb} \tag{2.2}$$

i, j are atomic index from 1 to N (total number of atoms). a and b indicate the direction of orthogonal coordinate basis. $\boldsymbol{\sigma}$ is a 3 x 3 tensor indicates the change of energy under an tensorial deformation. The instantaneous pressure is defined as:

$$\frac{dE}{dV} = -\frac{1}{3}\mathbf{Tr}(\boldsymbol{\sigma})$$

And the thermodynamic pressure $P =< \frac{dE}{dV} >_T$ which are the statistical average of particle simulations under constant temperature

Here we reformulate the calculation to the ease of applying the AD calculation.The energy of the function can be formulated as function of $E(x_i, \mathrm{C})$. And the Virial stress can be expressed as:

$$\boldsymbol{\sigma} = -\mathbf{adj}(\mathbf{C})^{-1}\frac{dE}{\mathrm{dC}} \tag{2.3}$$

adj is the adjugate or the cofactor of the cell matrix. The cofactor matrix has a nice geometric meaning as the area perpendicular to the direction for each C_{ab} direction. This also just the definition of stress as average force per area along different covariant directions. This expression is rather unpopular in the literature because most virial calculation only concerns with the pair-decomposable case as defined in equation (2.2)

We show that this definition reduces to the same Virial equation (2.2) in the pair-wise decomposable case. We define the following quantities that can be derived from r: $\widetilde{r_{ija}}$ is the un-shifted inter-atomic coordinate under minimum image convention. $r_{ija} = x_i - x_j$ is the original relative inter-atomic coordinate. $\widehat{r_{ija}}$ is the reduced unshifted inter-atomic coordinate computed by $\mathbf{C}^{-1}x_i - x_j$.

We begin from our definition of the virial stress tensor

$$\begin{aligned} \frac{dE}{dC_{ab}} &= \sum_i \frac{dE}{dx_i}\frac{dx_i}{dC_{ab}} \\ &= \sum_i \sum_{j<i} \frac{dE}{d(x_i - x_j)}\frac{d(x_i - x_j)}{dC_{ab}} \\ &= \sum_i \sum_{j<i} \frac{dE}{d\widetilde{r_{ija}}}\frac{d(\widetilde{r_{ija}})}{dC_{ab}} \end{aligned} \tag{2.4}$$

We then inspect the gradient term $\frac{d(\widetilde{r_{ija}})}{dC_{ab}}$. Because we know that $\widetilde{r_{ija}}$ can be explicitly calculated as a shift array $\mathbf{H_{ija}}$ over all pair of interactions.

$$\begin{aligned} \widetilde{r_{ija}} &= \sum_b C_{ab}H_{ija} + r_{ija} \\ &= \sum_b (C_{ab}H_{ija} + C_{ab}\widehat{r_{ija}}) \end{aligned} \tag{2.5}$$

$$\frac{d\widetilde{r_{ija}}}{dC_{ab}} = H_{ija} + \widehat{r_{ija}} \tag{2.6}$$

a, b are the index for the coordinate system in the Cartesian coordinates. And we can plug in the definition of the derivative back into the last line in (2.4). For notational convenience, we neglect the index a and b, and assume the same index are summed implicitly.

$$\begin{aligned} \frac{dE}{d\mathbf{C}} &= \frac{dE}{d\widetilde{r_{ij}}}(H_{ij} + \widehat{r_{ij}}) \\ &= \mathbf{C}^{-1}\frac{dE}{d\widetilde{r_{ij}}}\mathbf{C}(H_{ij} + \widehat{r_{ij}}) \end{aligned} \tag{2.7}$$

Recall our definition and use the definition of $\mathbf{C}^{-1} = \frac{1}{\det(\mathrm{C})}\mathbf{adj}(\mathbf{C})$ and $\mathbf{det}(\mathbf{C})$ by definition of the simulation box. Moreover, $f_{ij} = -\frac{dE}{dr_{ij}}$. Plug these definition back into the equation, it is easy to show that

$$\begin{aligned} \boldsymbol{\sigma} &= \sum_{i<j} \mathbf{adj}(\mathbf{C})^{-1}\mathbf{C}^{-1}\frac{dE}{d\widetilde{r_{ij}}}\mathbf{C}(H_{ij} + \widehat{r_{ij}}) \\ &= \sum_{i<j} \mathbf{adj}(\mathbf{C})^{-1}\frac{1}{\det(\mathrm{C})}\mathbf{adj}(\mathbf{C})\frac{dE}{d\widetilde{r_{ij}}}\mathbf{C}(H_{ij} + \widehat{r_{ij}}) \\ &= \sum_{i<j} \frac{1}{V}\frac{dE}{d\widetilde{r_{ij}}}\widetilde{r_{ij}} \end{aligned} \tag{2.8}$$

2.4. Calculating Virial and Forces with Automatic Differentiation. We recover the expression shown in (2.2) from (2.3). We emphasize that re-expressing the viral in this way is particularly. Automatic differentiation is the key computational techniques in modern machine learning especially in training large neural networks. In particular, the reverse-mode AD is particularly useful to computing gradient for scalar outputs. For automatic differentiation, one only needs to define the forward computation and the vector jacobian product is performed in the reverse order to compute the gradients with respect to the inputs.

Here we emphasize the advantage of the virial definition in (2.3). **We just need to define our forward computation of energy with C and x_i as inputs, the gradient can be automatically generated, so that we only need to consider implementing the forward calculation.** The idea of AD in this case is illustrated in figure Figure 2. We apply the source-to-source reverse mode AD as implemented in Zygote.jl [2]. It is different from tracing based reverse-mode AD which have memory overhead in tracing the computational graph. Source-to-Source AD as implemented in Zygote.jl automatically generates the compiled gradient function without the cost of building/tracing the computation graph. In addition, Zygote.jl is well-optimized library of ad-joints implementation which means that the speed of gradient calculation is not necessarily slower than hand-coded gradient function.

2.5. Computing gradients for general many-body potentials wiht AD . As we mentioned before, the force and virial calculation first computes the geometrical

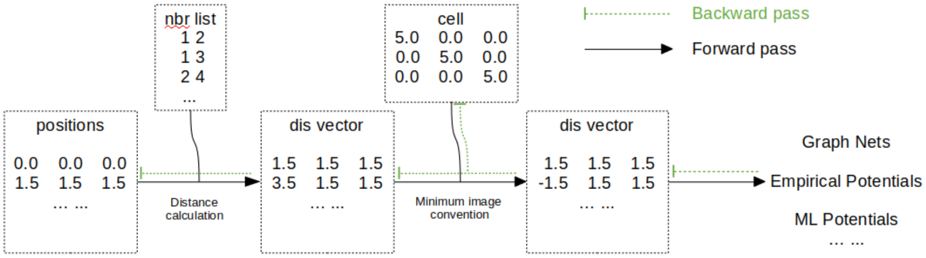

FIG. 2. *Computational graphs for the computation of atomic forces and virial stress. The green dotted line and solid black line indicates the backward and forward computation. This schematic only indicates the computation procedure for geometric features (e.g. distance, angle, etc.) are used for further energy calculations. What we want to emphasize here is that as along as we know the adjoints of the features, gradient function can be constructed back to coordinates (force) and cell matrix (Virial stress tensor). It is important to note that neighbor list(nbr list for short) are simply discrete masks for to retrieve values of arrays via indices, they do not have explicit gradients. The determination of these book-keeping quantities are done independent of the computational graph*

descriptor (e.g. distances, angles, dihedral angles) to be input into some functional form to compute the energy. That is the forward computation. For the reverse run, the gradient is computed with respect to those geometrical parameter $\xi_{ij..k}$ where the index $ij...k$ indicates the number of bodies involved in the calculation. In general, one encounter an energy function that contains multiple different topologies: an atom can participate in different interaction groups. We use k to indicate the number of particles involved in the energy calculation for particular topology. (For example, k=2 for pair interactions and k = 3 angle potentials which requires computing the angles of three atoms positions)

$$E = E^{(2)}(\xi_{ij}) + E^{(3)}(\xi_{ijk}) + ... \tag{2.9}$$

And the gradient can be computed automatically with

$$\frac{dE}{d\mathbf{C}} = \sum_k \frac{dE^{(k)}}{d\xi_{ij..k}} \frac{d\xi_{ij..k}}{d\mathbf{C}} \tag{2.10}$$

However, as pointed in Ref. [5], to correctly compute Virial stress tensor in periodic systems, one needs to carefully inspect if atoms are shifted correctly to fulfill the minimum image convention. The shifting array for H_ij is saved in the computational graph as mask. And it is usually sparse because only atoms at the boundary of box are subject to such shifting operations. In AD, the adjoint of these operations can be automatically generated with source code generation.

3. Numerical Experiments. We first introduce two type of potential energy functions we implemented for this study.

The implementation can be found in https://github.com/wwang2/thermograd.jl.

3.1. Lennard Jones Potentials. Lennard-Jones Potentials [3] are pairwise energy terms that are functions of inter-atomic distances. It is usually used to describe the dispersion forces between atoms due to fluctuating dipoles.

$$E_{LJ}(r) = 4 * \epsilon((\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}) \tag{3.1}$$

3.2. Graph Neural Nets. Graph neural network is a class of model that operates on system with graph structures with node and edges, and have been recently applied to learn inter-atomic potentials in a data-driven way. The model takes a set of atomic features, and edges features, and adjacency matrix to perform message passing to learn atomwise embeddings. In the case of learning inter-atomic potentials, the distances are used as edge features into the graph neural nets. This enjoys the benefits of constructing a energy function that is rotationally and permutational invariant for a inter-atomic potential.

The graph nets for learning inter-atomic potentials involving performing message passing with an message and update steps to obtain atom-wise embeddings $z^{(k)}$. Such operations can be performed k times to construct an function that is contains correlations among multiple atoms. One starts with an initial embedding $z^{(0)}$ and recursively perform the following operations

$$z_i^{(k)} = z_i^{k-1} + \sum_j \mathbf{update}(\mathbf{message}(z_i^{k-1}, r_{ij})) \tag{3.2}$$

The **update** and **message** function are usually parameterized with atom-wise and edge-wise multi-layer perceptrons (MLP). The final parametrized embedding $z_i^{(k)}$ is further parameterized with an atom-wise readout MLP which produces an atom-wise energy and their sum gives the total energy of the system

$$E = \sum_i \mathbf{readout}(z_i^k) \tag{3.3}$$

The ad-joints on r_{ij} is computed with the sum of partial derivatives.

$$\frac{dE}{dr_{ij}} = \sum_k \frac{\partial E}{\partial z^{(k)}} \frac{\partial z^{(k)}}{\partial r_{ij}} \tag{3.4}$$

And the virial stress tensor can be computed with chain rule which can be efficiently evaluated with AD:

$$\frac{dE}{d\mathbf{C}} = \frac{dE}{dr_{ij}} \frac{d_r ij}{d\mathbf{C}} \tag{3.5}$$

3.3. calculation of force virial for Lennard Jones system . We present calculation of the virial stress and forces with automatic differentiation on 1000 atom periodic structure of a cubic Diamond cell. The cell dimension of each side is 27.15 (Arbitrary units.). We choose $\sigma = 1.0$ and $\epsilon = 1.0$ for the benchmark calculation. To test the scalability of the implementation, we benchmark on the speed and memory allocation with different interaction cutoff which gives different number of atom pairs which are needed for the energy calculation. We choose 4 cutoffs at 3.0, 5.0, 7.0 and 9.0. The result is shown in Figure Figure 3. We compare the automatically generated Zygote gradient and the analytical gradients we implemented by hand. It shows that the implementation linearly scales the number of interaction pairs. This is what we would expect. The Zygote implementation performs similarly in terms of speed and memory allocations with hand-coded ad-joints.

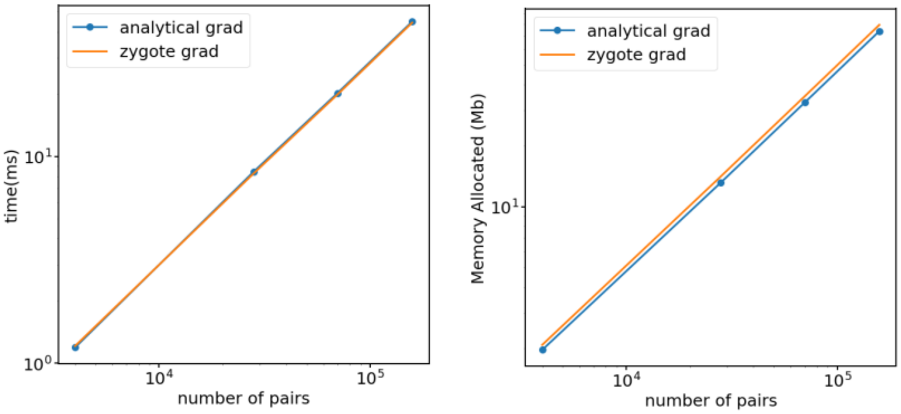

Fig. 3. *The time and memory benchmark for Lennard-Jones system of 1000 atoms.*

3.4. Graph Neural Nets. Graph Neural Nets represent a more complicated system. Similar to the Lennard Jones system, we test on a system with 1000 atoms. To determine the adjacency matrix of this periodic graph, we use moderate cutoff of 5.0, which gives 28000 interaction pairs. In the terms of graph convolutions, the depths are the number of times message-passing operations are performed on the graph. In case of parameterizing inter-atomic potentials, it is a way to explicitly construct many-body correlations to go beyond pair interactions. The operations behind message passing as described in equation (3.2). In this example we implemented the SchNet[4] model which implements a smeared distance filter to convolve over 3D space. For bench-marking purpose, we choose rather small embedding dimension of 8 and a filter dimension of 16.

One particular benchmark we are interested in, is the scalability of convolution depth, defined as k in the previous section. The main computation cost of the procedure is the message-passing step which needs to be broadcasted to compute the message function, and sum-pooled to carry out the atom-wise update step. Both the message and update functions are parametrized with multi-layer perceptrons which can be backpropated efficiently with reverse calculation. The result for the force and virial calculation versus energy calculation is presented in figure Figure 4. The computational cost for both the forward (energy) calculation and reverse calculation (force/virial) are linear as a function of convolution depth. More importantly, with reverse-mode AD, the calculation of force and energy are on the same order of magnitude. The forces and virial calculation only takes about 3 times more than the energy function. In comparison, applying forward mode or finite differencing scales with the number of atoms of the system, and is obviously not as efficient.

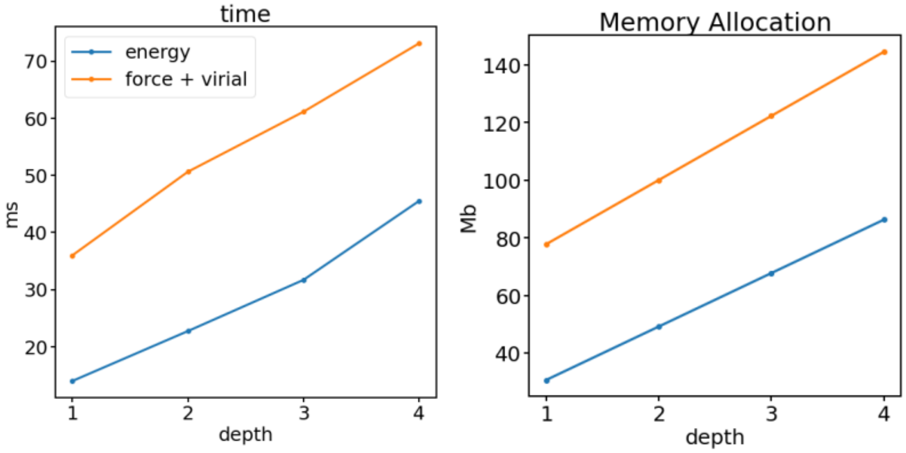

FIG. 4. *The time and memory benchmark for a GNN operating on 1000 atoms with 28000 interaction pairs*

3.5. Optimizing Crystal Cell with AD . As we show that we can compute virial efficiently. What can we do with it? In this section we present an example for crystal cell optimization with the gradient from AD.

Given the crystal symmetry of a Diamond structure, we do not optimize the position explicitly but rather re-scale the positions given the update of the crystal cell. We perform the gradient update:

$$C_{i+1} = C_i - \frac{dE}{dC_i}\eta \tag{3.6}$$

The positions are updated based on the newly obtained cell structure

$$x_i + 1 = \mathbf{C}_{i+1}\mathbf{C_i}^{-1}x_i \tag{3.7}$$

$\mathbf{C_i}^-1x_i$) gives the reduced coordinates wrapped under the cell basis. This procedure ensures the atomic positions are collinear with the deformation of cell. Out optimization converges quickly and obtain an equilibrium cell structure with a lattice constant of 16.48

4. Conclusions. In this report, we review the computational problem of calculating forces and the virial stress tensor for a periodic systems. We comment on the subtleties of handling periodic boundary conditions to compute the virial stress tensor. We generalize the virial calculations by mapping out the computational glow for virial calculation as forward-backward problem in the language of Automatic Differentiation. The formalism extends the virial calculation to arbitrary many-body inter-atomic potentials. We show that the AD calculation of forces virials on test systems are efficient and is simply done with just one backward pass. To demonstrate the use of virial stress tensor, we carry out crystal cell optimizing with gradient descent. We show that while maintaining the computational efficiency with analytical

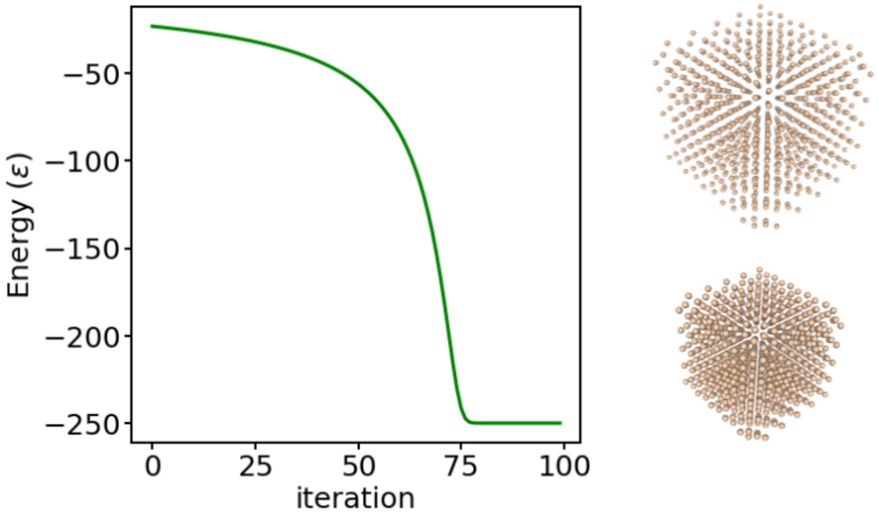

Fig. 5. *We compute the gradients on the lattice constant for a Diamond structures for 1000 atoms with pairwise Lennard-Jones potentials. TWe perform gradient descent with a update rate of 1e-2 for 100 steps. The figure also compares the initial structure (top) and optimized structure(bottom)*

gradients, source-to-source AD simplifies the process of having to derive analytical solutions which can be time-consuming.

There are several possible steps to extend the works. It would interesting to extend the gradient calculation to higher order, so that the thermodynamic derivatives like isothermal compressibility $(\frac{dP}{dV})_T$. This can be done efficiently by mixing reverse-mode and forward mode AD. Furthermore, we present in this report only applies to single point calculation, but the full thermodynamic derivatives by back-propagating the sampling procedures can directly provide ensemble averages to obtain statistics. To achieve this, the quantities of interest include $\frac{dP}{dT}$ where T is the temperature which, in MD simulations are controlled by a thermostat vraible. By differentiating through simulation trajectories, one can obtain the derivatives of Pressure (P) with respect to implicit control variables like temperature. This would be an extension of Ref. [6]

Acknowledgments. We would like to Dr. Christopher Rackauckas for helpful discussion and the teaching staff for 18.337 for a great semester.

REFERENCES

[1] A. P. Bartók and G. Csányi, *G aussian approximation potentials: A brief tutorial introduction*, International Journal of Quantum Chemistry, 115 (2015), pp. 1051–1057.

[2] M. Innes, *Don't unroll adjoint: differentiating ssa-form programs*, arXiv preprint arXiv:1810.07951, (2018).

[3] J. E. Jones, *On the determination of molecular fields.—i. from the variation of the viscosity of a gas with temperature*, Proceedings of the Royal Society of London. Series A, Containing

Papers of a Mathematical and Physical Character, 106 (1924), pp. 441–462.

[4] K. Schütt, P.-J. Kindermans, H. E. S. Felix, 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, 2017, pp. 991–1001.

[5] 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*, The Journal of chemical physics, 131 (2009), p. 154107.

[6] W. Wang, S. Axelrod, and R. Gómez-Bombarelli, *Differentiable molecular simulations for control and learning*, arXiv preprint arXiv:2003.00868, (2020).