Diploma Thesis

Phonon Spectrum Analysis of Diamond using Gaussian Process Regression

submitted in satisfaction of the requirements for the degree
Diplom-Ingenieurin
of the TU Wien, Faculty of Physics

Diplomarbeit

Phononenspektrumsanalyse von Diamant unter Verwendung von Gauss-Prozess-Regression

ausgeführt zum Zwecke der Erlangung des akademischen Grads
Diplom-Ingenieurin
eingereicht an der TU Wien, Fakultät für Physik

Sita Schönbauer, BSc

Matr.Nr.: 01526671

Betreuung: Univ. Prof. Mag.rer.nat. Dr.rer.nat. **Andreas Grüneis**
Institut für Theoretische Physik
Technische Universität Wien
Wiedner Hauptstraße, 1040 Wien, Österreich

Wien, im Juli 2023

S. Schönbauer

A. Grüneis

Contents

1 Abstract

This work employs Machine Learning (ML) tools to generate phonon spectra from ab initio Molecular Dynamics (MD) simulations. For this end, an ML algorithm is trained using two-body and Smooth Overlap of Atomic positions (SOAP) descriptors on the MD trajectory of 54 carbon atoms in a solid cubic diamond structure calculated employing Density Functional Theory (DFT) in a canonical ensemble, with energies and forces as targets.

The accuracy of the ML force field is studied using the following two approaches. Firstly, MD simulations using ML methods starting from the same geometry as the original DFT trajectory are computed and compared to the original runs, together with simulations where the ML algorithm was trained with smaller DFT datasets to inspect robustness against data reduction.

Secondly, the trained ML algorithm is also used to calculate forces for new systems, specifically finite displacements from equilibrium to compute the second derivatives of the energy surface from which phonon spectra can be generated. 250 datapoints were sufficient to obtain accurate phonon dispersions.

2 Kurzfassung

Diese Arbeit verwendet Machine Learning (ML) Werkzeuge um aus ab initio-Molekulardynamiksimulationen Phononenspektren zu generieren. Zu diesem Zweck wird ein ML-Algorithmus mithilfe von Zweikörper- und SOAP- (Smooth Overlap of Atomic Positions) Deskriptoren auf einer MD-Trajektorie von 54 Kohlenstoffatomen in einem kubisch flächenzentrierten Kristallgitter, die mit Dichtefunktionaltheorie (DFT) im kanonischen Ensemble berechnet wurde, trainiert, mit den Energien und Kräften als "targets".

Die Genauigkeit des ML Kraftfeldes wird mittels zweier Ansätze überprüft. Einerseits werden ML-basierte MD Simulationen gleicher Ausgangsgeometrie wie die der DFT Trajektorien berechnet und mit der DFT Simulation verglichen. Dabei werden auch ML Algorithmen mit kleiner werdenden Trainingssets mitverglichen, um die ML Methoden auf Robustheit gegenüber Datensetreduktion zu untersuchen.

Im zweiten Ansatz wird der bereits trainierte und optimierte ML Algorithmus benutzt um die Kräfte für neue Systeme mit endlichen Verschiebungen gegenüber dem Gleichgewicht zu berechnen, um aus der zweiten Ableitung der Potentialenergiefläche Phononenspektren zu generieren. 250 Datenpunkte reichten als Trainingsset aus, um genaue Phononendispersionen zu erhalten.

3 Introduction

Ab initio computational materials science aims at the analysis of large systems, where the scaling of the applied method to ever larger numbers of atoms or particles is of great relevance. The computational cost of Density Functional Theory (DFT) calculations using approximate exchange and correlation energy functionals, for example DFT-PBE, scales with $\mathcal{O}(N^3)$, where N is the number of atoms, limiting the method to about a few thousand atoms even on modern supercomputers. [1] Moreover, the computational cost involved also severely limits the time scale accessible in MD simulations at the DFT level.

However, systems beyond this size and complexity are of great interest, for example diffusion processes of defects in solids, which are crucial to understanding material properties. Machine Learning (ML) methods strive to be more computational cost effective by training surrogate models on already produced data. In the methods applied in this work, expensively produced DFT-PBE data created with the Vienna Ab Initio Simulation Package (VASP) is used to train an ML model. This can then be used to run subsequent MD simulations for different atomic displacements as demanded by `phonopy` [2] so it can then compute the corresponding phonon dispersion.

4 Theory

The Hamiltonian for an arbitrary system of electrons can be written as follows:

$$\hat{H} = \hat{T} + \hat{U}_{ee} + \hat{V}_{ext} \tag{1}$$

The first term describes the kinetic energy, the second term describes the Coulomb interaction between the electrons, and the third term represents an arbitrary external potential. Writing out the known terms

$$\hat{H} = \sum_{i=1}^{N} -\frac{\hbar^2}{2m_e}\nabla_i^2 + \frac{1}{8\pi\epsilon_0}\sum_{i\neq j}\frac{e^2}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \hat{V}_{ext} \tag{2}$$

where N is the number of electrons, $\hbar$ is Planck's constant, m_e is the electron mass, ϵ_0 the electric constant, e the electron charge, and $\boldsymbol{r}$ the electron positions, one can see that the first two terms depend only on the number of electrons, while $\hat{V}_{ext}$ is different for each investigated system.

Examining only the known terms $\hat{T}$ and $\hat{U}_{ee}$, it is clear that the interaction term provides much difficulty for solving the equation as it makes any system with more than one electron non-separable. To deal with this, we turn to the Hartree-Fock method.

4.1 The Hartree-Fock Method

The Hartree product

$$\Psi_H(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N) = \phi_1(\boldsymbol{r}_1)\phi_2(\boldsymbol{r}_2)...\phi_N(\boldsymbol{r}_N) \tag{3}$$

where Ψ represents the full wave function and ϕ_i the separated single-electron wave functions, is a very basic approximation that presumes no interaction between electrons. In this way, it ignores fundamental properties of electrons:

- Anti-symmetry - the wave function Ψ should change sign upon an electron switching positions with another:

$$\Psi(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\Psi(\boldsymbol{x}_j, \boldsymbol{x}_i) \tag{4}$$

Here $\boldsymbol{x}_i$ is a compound index $\{\boldsymbol{r}_i, \sigma_i\}$, where $\boldsymbol{r}_i$ is a spatial coordinate and σ_i is the spin coordinate. For brevity we will assume a symmetric wave function with regards to the spin coordinate and focus the following discussion on the spatial coordinate only.

However, for the Hartree product this is not necessarily the case:

$$\phi_i(\boldsymbol{r}_i)\phi_j(\boldsymbol{r}_j) \neq -\phi_i(\boldsymbol{r}_j)\phi_j(\boldsymbol{r}_i) \tag{5}$$

- Pauli exclusion principle - two electrons may not have the same state in the same orbital:

$$\Psi(\boldsymbol{r}_i, \boldsymbol{r}_i) = 0 \tag{6}$$

- Coulomb repulsion - as negatively charged particles, electrons repulse each other as described by the interaction term in the Hamiltonian (2):

$$\hat{U}_{ee} = \frac{1}{8\pi\epsilon_0} \sum_{i \neq j} \frac{e^2}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \tag{7}$$

The first and second point, present due to the Fermionic nature of electrons, can be solved by introducing the Slater-Determinant:

$$\Psi_{SD}(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\boldsymbol{r}_1) & \phi_1(\boldsymbol{r}_2) & \dots & \phi_1(\boldsymbol{r}_N) \\ \phi_2(\boldsymbol{r}_1) & \phi_2(\boldsymbol{r}_2) & \dots & \phi_2(\boldsymbol{r}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\boldsymbol{r}_1) & \phi_N(\boldsymbol{r}_2) & \dots & \phi_N(\boldsymbol{r}_N) \end{vmatrix} \tag{8}$$

As is the nature of determinants, all expressions with switched positions have opposite signs, and there are no terms where two electrons share the same state and orbital.

The last point - the Coulomb repulsion $\hat{U}_{ee}$ - can partly be accounted for in the mean-field approximation. Instead of considering all Coulomb terms for all electron pairs individually, the electrons are imagined to interact with a mean field of all electrons, which can be written as such:

$$\hat{V}_H = \frac{e^2}{4\pi\epsilon_0} \sum_j \int d^3r' \frac{|\phi_j(\boldsymbol{r}')|^2}{|\boldsymbol{r} - \boldsymbol{r}'|} \tag{9}$$

which has to be applied to all single particle states $\phi_i(\boldsymbol{r})$.$\hat{V}_H$ is also called the Hartree potential.

4.2 Observables

The central aspect to any theory or approximation in quantum mechanics is how it affects observables, the parts of the system that can feasibly be evaluated. Starting from the Schrödinger equation

$$\hat{H}\left|\Psi\right\rangle = E\left|\Psi\right\rangle \tag{10}$$

the observable of the system energy E can be found with

$$E = \left\langle\Psi\right|\hat{H}\left|\Psi\right\rangle \tag{11}$$

To find the ground state energy of the system, the variational principle can be applied

$$E_0 = min\left\langle\Psi\right|\hat{H}\left|\Psi\right\rangle \tag{12}$$

In the above equation minimization is performed over the single-particle states ϕ_i, yielding the orbitals used to construct Ψ_{SD}.

Moving back to our previous considerations and the Hartree-Fock method, one can write

$$E_0 = min\left\langle\Psi_{SD}\right|\hat{T} + \hat{U}_{ee} + \hat{V}_{ext}\left|\Psi_{SD}\right\rangle \tag{13}$$

where the interaction contribution $\left\langle\Psi_{SD}\right|\hat{U}_{ee}\left|\Psi_{SD}\right\rangle$ can be split into a Coulomb repulsion term (similar to the mean field approximation seen in (9))

$$J_{ij} = \frac{e^2}{4\pi\epsilon_0}\int d^3r \int d^3r' \frac{|\phi_j(\boldsymbol{r'})|^2|\phi_i(\boldsymbol{r})|^2}{|\boldsymbol{r}-\boldsymbol{r'}|} \tag{14}$$

and an exchange term representing the electrons' Fermionic anti-symmetry

$$K_{ij} = \frac{e^2}{4\pi\epsilon_0}\int d^3r \int d^3r' \frac{\phi_j^*(\boldsymbol{r'})\phi_i^*(\boldsymbol{r})\phi_j(\boldsymbol{r})\phi_i(\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|} \tag{15}$$

Together with the kinetic and external potential terms, one can write the system energy as

$$
\begin{aligned}
E &= \langle\Psi_{SD}|\,\hat{T}+\hat{V}_{ext}\,|\Psi_{SD}\rangle + \sum_{i>j}(J_{ij}-K_{ij}) \\
&= T + E_{ext} + \underbrace{E_H + E_X^{HF}}_{\langle\mathrm{\Psi_{SD}}|\hat{\mathrm{U}}_{\mathrm{ee}}|\mathrm{\Psi_{SD}}\rangle=\mathrm{E_{ee}}}
\end{aligned}
\tag{16}
$$

by summing over all electron pair combinations i, j.

4.3 Density Functional Theory

The basic principle of Density Functional Theory (DFT) revolves around modelling systems not as many-body problems with many-body solutions for the inquired wave function, but instead taking advantage of the fact that wave functions can be written as functionals of the electron density $n(\boldsymbol{r})$:

$$
\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, ..., \boldsymbol{r}_N) \rightarrow \Psi[n(\boldsymbol{r})] \tag{17}
$$

whereby N is the number of electrons in the system.
For normalized Ψ, $n(\boldsymbol{r})$ is defined as

$$
n(\boldsymbol{r}) = N\int d^3\boldsymbol{r}_2 ... \int d^3\boldsymbol{r}_N \Psi^*(\boldsymbol{r}, \boldsymbol{r}_2, \boldsymbol{r}_3, ..., \boldsymbol{r}_N)\Psi(\boldsymbol{r}, \boldsymbol{r}_2, \boldsymbol{r}_3, ..., \boldsymbol{r}_N) \tag{18}
$$

The core of Density Functional Theory (DFT) is that the electron density $n(\boldsymbol{r})$ is sufficient to determine the expectation values of all observables $\hat{O}$ of a system in the ground state: $\langle\Psi[n]|\,\hat{O}\,|\Psi[n]\rangle$. With this in mind, DFT attempts to make complex many-particle problems solvable.

4.3.1 The Kohn-Sham Scheme

To determine the ground state energy in the DFT framework, the first Hohenberg-Kohn Theorem [3] proves practical:

Hohenberg-Kohn 1. *For any system of interacting particles in an external potential $\hat{V}_{ext}(\boldsymbol{r})$, the potential is determined uniquely, except for a constant, by the ground state particle density $n_0(\boldsymbol{r})$.*

Aside from electron number N, the system is defined by $\hat{V}_{ext}$, which in turn is uniquely defined by $n_0(\boldsymbol{r})$.

To get to the ground state energy as searched for in (13), the second Hohenberg-Kohn Theorem can be employed:

Hohenberg-Kohn 2. *A universal functional for the energy $F[n] = \langle\Psi| \hat{T} + \hat{U}_{ee} |\Psi\rangle$ in terms of the density can be defined, valid for any $\hat{V}_{ext}$. A variational principle exists such that the global minimum value of this functional is the exact ground state energy:*

$$E_0 \leq \langle\Psi[n]| \hat{H} |\Psi[n]\rangle = F[n] + \int d^3r n(\boldsymbol{r})\hat{V}_{ext} = E[n] \tag{19}$$

The density that minimizes this functional is $n_0(\boldsymbol{r})$.

The second Hohenberg-Kohn theorem implies that the ground state energy E_0 (and density $n_0(\boldsymbol{r})$) can be found by systematic variation.

At this stage, the remaining trouble is the definition of $E[n]$. For this purpose, the Kohn-Sham scheme can be implemented, which asserts that for any interacting system there exists a single particle potential $\hat{V}_{KS}$ so that this auxiliary system is solved by the same $n_0(\boldsymbol{r})$ as the original one. [4] To this end, the interaction term $\hat{U}_{ee}$ and the external potential $\hat{V}_{ext}$ are replaced by a single particle Kohn-Sham potential $\hat{V}_{KS}$

$$\hat{V}_{KS} = \hat{V}_{ext} + \hat{V}_H + \hat{V}_{XC} \tag{20}$$

with the appropriate functionals

$$\begin{aligned} E[n] &= F[n] + E_{ext}[n] \\ &= T_{KS}[n] + E_H[n] + E_{XC}[n] + E_{ext}[n] \end{aligned} \tag{21}$$

This is similar to the considerations with the Hartree-Fock method. As such, $E_H[n]$ can be written out as

$$E_H[n] = \frac{1}{2}\frac{e^2}{4\pi\epsilon_0}\int\int d^3rd^3r'\frac{n(\boldsymbol{r})n(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} \tag{22}$$

However, despite similar purpose, the exchange-correlation functional $E_{XC}[n]$ differs from the structure listed in (15) and (16); K_{ij} does not depend on $n(\boldsymbol{r})$. Additionally, the transformation from the interacting system to the non-interacting one changed the kinetic component T to T_{KS}, yielding

$$E_{XC}[n] = T[n] - T_{KS}[n] + E_{ee}[n] - E_H[n] \tag{23}$$

for the exchange correlation functional $E_{XC}[n]$.

$\hat{V}_{XC}$ and the corresponding $E_{XC}[n]$ are both unknown; there are several options of choice for this exchange correlation term. This work uses the Perdew-Burke-Ernzerhof (PBE) functional (see also Section 4.3.2).

The final system of equations that need solving are structured as follows:

$$\left[-\frac{\hbar^2}{2m_e}\nabla_i^2 + \frac{e^2}{4\pi\epsilon_0}\sum_{i\neq j}\int d^3r'\frac{|\phi_j(\boldsymbol{r}')|^2}{|\boldsymbol{r}-\boldsymbol{r}'|} + \hat{V}_{XC} + \hat{V}_{ext,i}\right]\phi_i(\boldsymbol{r}) = E_i\phi_i(\boldsymbol{r}) \tag{24}$$

with

$$\hat{V}_{XC} = \frac{\delta E_{XC}[n]}{\delta n(\boldsymbol{r})} \tag{25}$$

4.3.2 Exchange Correlation Functionals

The methods devised by Hohenberg, Kohn, and Sham reduce the biggest difficulty in solving complex systems to finding the exchange correlation density functional for *all* systems; were the exchange correlation functional exact, it would also yield exact solutions.

As it stands, none of the currently used functionals are exact, and different ones yield differently accurate results depending on the system they are used on. One way of evaluating the different functionals is by comparing exchange correlation holes.

Exchange Correlation Holes. The exchange correlation hole represents a region around an electron where the likelihood of finding a second electron approaches zero due to exchange (antisymmetry) and correlation (Coulomb repulsion) effects.

The pair density or likelihood, of an electron at point $\boldsymbol{r}_1$ and another electron at point $\boldsymbol{r}_2$, irrespective of all other $N-2$ electrons' positions, can be written as

$$\rho(\boldsymbol{r}_1, \boldsymbol{r}_2) = N(N-1)\int d^3r_3 \cdots \int d^3r_N |\Psi(\boldsymbol{r}_1\boldsymbol{r}_2 ... \boldsymbol{r}_N)|^2 \tag{26}$$

From this one can construct the conditional probability that an electron is at position $\boldsymbol{r}_2$ if a different electron is at position $\boldsymbol{r}_1$:

$$\Omega(\boldsymbol{r}_1, \boldsymbol{r}_2) = \frac{\rho(\boldsymbol{r}_1, \boldsymbol{r}_2)}{n(\boldsymbol{r}_1)} \tag{27}$$

The difference between $\Omega(\boldsymbol{r}_1, \boldsymbol{r}_2)$ and the uncorrelated $n(\boldsymbol{r}_2)$ is the exchange correlation hole:

$$h_{XC}(\boldsymbol{r}_1, \boldsymbol{r}_2) = \Omega(\boldsymbol{r}_1, \boldsymbol{r}_2) - n(\boldsymbol{r}_2) \tag{28}$$

Since $n(\boldsymbol{r}_2)$ integrates to N and $\Omega(\boldsymbol{r}_1, \boldsymbol{r}_2)$ integrates to $N-1$, the exchange correlation hole integrates to

$$\int d^3r_2 h_{XC}(\boldsymbol{r}_1, \boldsymbol{r}_2) = -1 \tag{29}$$

We note that the Hartree-Fock method satisfies (29).

It is also of note that h_{XC} can be split into an exchange part h_X representing the electrons' Fermionic nature, and a correlation part h_C representing the Coulomb repulsion, that can be summed together directly to make h_{XC}

$$h_{XC} = h_C + h_X \tag{30}$$

$h_X(\boldsymbol{r}_1, \boldsymbol{r}_2)$ is non-positive at all points and integrates to -1, representing the single negative charge, while $h_C(\boldsymbol{r}_1, \boldsymbol{r}_2)$ can be positive or negative and integrates to 0.

The most fundamental exchange correlation density functional is the Local Density Approximation (LDA). It assumes that the system can be approximated at any point by a homogenous electron gas of the density the real system has at this particular point; additionally, it also assumes that the functional $E_{XC}[n]$ depends only on the electron density n itself, not its derivatives. In this case, $E_{XC}[n]$ can be written as

$$E_{XC}^{LDA}[n(\boldsymbol{r})] = \int d^3r n(\boldsymbol{r}) \varepsilon_{XC}[n(\boldsymbol{r})] \tag{31}$$

where $\varepsilon_{XC}[n(\boldsymbol{r})]$ represents the exchange correlation energy of singular electrons in the homogenous electron gas. LDA also satisfies equation (29), making it useful despite the gross approximation of homogenous electron density.

The Generalized Gradient Approximation (GGA) provides a more advanced exchange correlation functional. Here, the first derivatives (gradients) $\nabla n(\boldsymbol{r})$ are included in the functional, representing the locally different, non-constant environments:

$$E_{XC}^{GGA}[n(\boldsymbol{r})] = \int d^3 r n(\boldsymbol{r}) \varepsilon_{XC}[n(\boldsymbol{r}), \nabla n(\boldsymbol{r})] \tag{32}$$

Unfortunately, GGA does not satisfy (29); this can be brute-forced away by for example setting all the positive entries of h_X^{GGA} to zero, and truncating the functional to ensure (29) is upheld. [5]

The exchange correlation functional used in this work, PBE, is a GGA functional.

4.3.3 Bloch's Theorem and the Plane-Wave Basis Sets

Bloch's theorem states that the solution for the Schrödinger equation in a periodic lattice can be written as a plane wave modulated by a periodic function:

$$\Psi_{n,\boldsymbol{k}}(\boldsymbol{r}) = \exp^{-i\boldsymbol{k}\boldsymbol{r}} u_{n,\boldsymbol{k}}(\boldsymbol{r}) \tag{33}$$

where $u_{n,\boldsymbol{k}}(\boldsymbol{r})$ represents the periodic function. To solve the system of equations set up in the previous sections, the periodic functions u_n need to be expanded in a basis set φ_α with weights $c_{n,\alpha}$

$$u_n = \sum_\alpha c_{n,\alpha} \varphi_\alpha \tag{34}$$

While there are different options for this, this work focuses on the plane-wave ansatz. For this purpose, the system is represented by an arrangement of periodic unit cells, with basis vectors $[\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3]$ and reciprocal basis vectors $[\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3]$. The plane waves are structured as such

$$\varphi_\alpha(\boldsymbol{r}) = \frac{1}{\sqrt{V}} \exp^{i\boldsymbol{G}_\alpha \cdot \boldsymbol{r}} \tag{35}$$

where V is the volume of the chosen unit cell and $\boldsymbol{G}_\alpha$ is the wave vector. $\boldsymbol{G}_\alpha$ is a superposition of the reciprocal lattice defined by the unit cell

$$\boldsymbol{G}_\alpha = a_\alpha \boldsymbol{b}_1 + b_\alpha \boldsymbol{b}_2 + c_\alpha \boldsymbol{b}_3 \tag{36}$$

Only $\boldsymbol{G}_\alpha$ that satisfy the periodic boundary conditions are included for the basis set. Additionally, only the plane waves up to an energetic cutoff point are considered:

$$\frac{1}{2}\boldsymbol{G}^2 \leq E_{cut} \tag{37}$$

4.4 Phonon Dispersion

4.4.1 The Monoatomic Chain

To understand the genesis of phonon spectra, it is instructive to first examine a simple, one-dimensional monoatomic chain with N atoms, inter-particle distance a and particle mass m. The displacement of the nth atom from its resting position na can be given as

$$u_n(t) = na - x_n(t) \tag{38}$$

For small displacements, the force acting on the atoms can be approximated by a harmonic oscillator

$$H = \sum_{n=1}^{N} \frac{p_n^2}{2m} + \frac{\lambda}{2}\sum_{n=1}^{N}(u_n - u_{n-1})^2 \tag{39}$$

with $p_n = m\dot{u}$ and λ as the spring constant. This leads to the classical equation of motion:

$$m\ddot{u}_n = -\lambda(2u_n - u_{n-1} - u_{n+1}) \tag{40}$$

which leads to the solution

$$u_n = A\exp^{-i\omega t + ikna} \tag{41}$$

Where A represents the amplitude. Defining the open parameters requires boundary conditions; just like in 4.3.3, the choice falls on periodic boundary conditions

$$u_{n+N} = u_n \tag{42}$$

so that valid k fulfill the condition

$$k = \frac{2\pi}{Na}m \tag{43}$$

where m is a natural number.

To get to the phonon spectrum, one now looks to the dispersion relation between ω and k, which can be found by simple insertion in (40)

$$\begin{aligned}
-m\omega^2\cancel{A\exp^{-i\omega t}\exp^{ikna}} &= -\lambda(2\cancel{A\exp^{-i\omega t}\exp^{ikna}} \\
&\quad - \cancel{A\exp^{-i\omega t}\exp^{ikna}}\exp^{+ika} \\
&\quad - \cancel{A\exp^{-i\omega t}\exp^{ikna}}\exp^{-ika} \\
\omega^2 &= \frac{2\lambda}{m}(1-\cos ka)
\end{aligned} \tag{44}$$

$$\Rightarrow \omega = 2\sqrt{\frac{\lambda}{m}}\left|\sin\left(\frac{ka}{2}\right)\right| \tag{45}$$

This very simple phonon spectrum is plotted in Figure 1.

4.4.2 The Diatomic Chain

The diamond structure examined in this work only contains one type of atom, carbon. Therefore, instead of examining the diatomic chain of two atom types with distinct masses, a pseudo-monoatomic chain with alternating forces λ_1 and λ_2 shall serve to illuminate the concept of the optical modes.

The equations of motion for the pseudo-diatomic chain with masses m and force constants λ_1 and λ_2 and resting length a can be written as follows

$$\begin{aligned}
m\ddot{u}_n &= -\lambda_1(v_n - u_n) - \lambda_2(u_n - v_{n-1}) \\
m\ddot{v}_n &= -\lambda_1(v_n - u_n) - \lambda_2(v_n - u_{n+1})
\end{aligned} \tag{46}$$

From this, one obtains solutions

$$\begin{aligned}
u_n(t) &= A_1\exp^{-i\omega t + ikna} \\
v_n(t) &= A_2\exp^{-i\omega t + ikna}
\end{aligned} \tag{47}$$

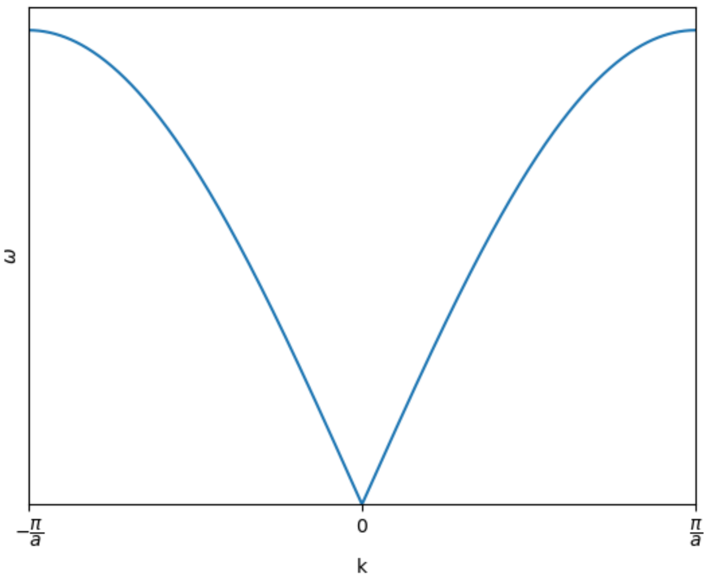


Figure 1: Phonon spectrum of a one-dimensional monoatomic chain within the first Brillouin zone.

where A_1 and A_2 describe both amplitude and phase. Insertion into (46) leads to

$$\begin{aligned}(m\omega^2 - \lambda_1 + \lambda_2)A_1 + (\lambda_1 - \lambda_2 \exp^{ika})A_2 = 0\\ (-\lambda_1 - \lambda_2 \exp^{-ika})A_1 + (m\omega^2 + \lambda_1 + \lambda_2)A_2 = 0\end{aligned} \tag{48}$$

Solving this linear system of equations gives the dispersion relation:

$$\omega^2 = \frac{\lambda_1 + \lambda_2}{m} \pm \frac{1}{m}\sqrt{\lambda_1^2 + \lambda_2^2 + 2\lambda_1\lambda_2\cos(ka)} \tag{49}$$

This dispersion relation contains two branches; an acoustic branch (+) similar to the monoatomic chain, where atoms move in phase, and an optical branch (-) where atoms move out of phase, creating optically active dipoles. The two branches can be seen in Figure 2.

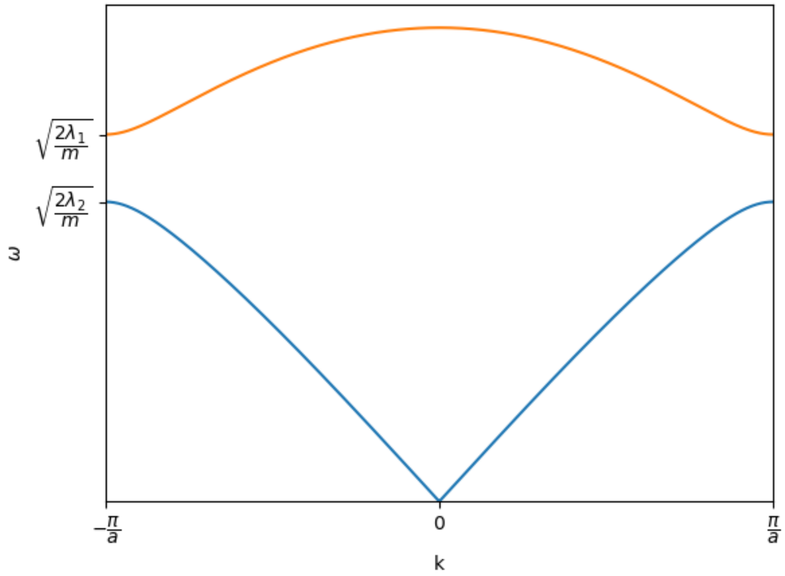


Figure 2: Phonon spectrum of a one-dimensional diatomic chain (different force constants λ, not masses) within the first Brillouin zone. Optical branch in orange and acoustic branch in blue.

4.4.3 Three-Dimensional Crystals

The basic concepts from the atomic chains can also be applied to three-dimensional crystals. The harmonic Hamiltonian can then be written as

$$H = \sum_{n=0}^{N} \frac{p_n^2}{2m} + \frac{1}{2} \sum_{K=1}^{N} \sum_{K'=1}^{N} \sum_{\xi=1}^{3} \sum_{\eta=1}^{3} \Phi_{KK'}^{\xi\eta} u_K^{\xi} u_{K'}^{\eta} \tag{50}$$

where K and K' refer to the investigated lattice point, and ξ and η to the cartesian coordinates. Φ replaces λ from before as the force constant; the forces between atoms are potentially different in all directions.

The equations of motions are then

$$m\ddot{u}_K^{\xi} = -\sum_{K'}^{N} \sum_{\eta}^{3} \Phi_{KK'}^{\xi\eta} u_{K'}^{\eta} \tag{51}$$

with solutions

$$\boldsymbol{u}_K = \boldsymbol{A}(\boldsymbol{k}, w) \exp^{-i\omega t + i\boldsymbol{k}\cdot\boldsymbol{a}_K} \tag{52}$$

Vectors are now denoted using bold fonts instead of the upper index for better readability. na from the one-dimensional solutions (41) and (47) has been replaced with the lattice vector $\boldsymbol{a}_K$.

For a three-dimensional system with N atoms in the unit cell, there are a total of $3N$ modes, 3 of them acoustic and the remaining $3N - 3$ optical. The acoustic modes consist always of two transversal and one longitudinal mode; transversal if the displacement of the atoms is perpendicular to the propagation of the wave, and longitudinal if displacement and propagation travel in the same direction.

Putting this to practice for the primitive fcc diamond structure investigated in this work, one should expect up to three acoustic branches and up to three optical branches, degenerate along high symmetry directions.

4.5 Machine Learning

This section consists mostly of summaries and adaptations of the excellent and comprehensive guide in Reference [1] and was written this way in Ref. [6].

4.5.1 Representation and Machine Learning

When using ML methods, a model is trained on value pairs of descriptors and target values; in this case, the descriptors are the atomic positions, and the target values the atomic energies and forces. Once the ML model has been trained, it can be used to predict new target values (forces and energies) based on new descriptors (positions). [7] With these forces and energies, one can make new MD simulations at a fraction of the computational cost of the original DFT calculations.

Descriptors. The most intuitive way to describe atomic positions is to simply use the cartesian coordinates of each present particle. But other representations are often more convenient, since they can be built to share certain properties with the observed system, like rotational or translational symmetry and permutational symmetry between atoms of the same species. [1]

This work uses two-body and Smooth Overlap of Atomic Positions (SOAP) descriptors. Two-body descriptors classically refer to the distances between all pairs of present particles

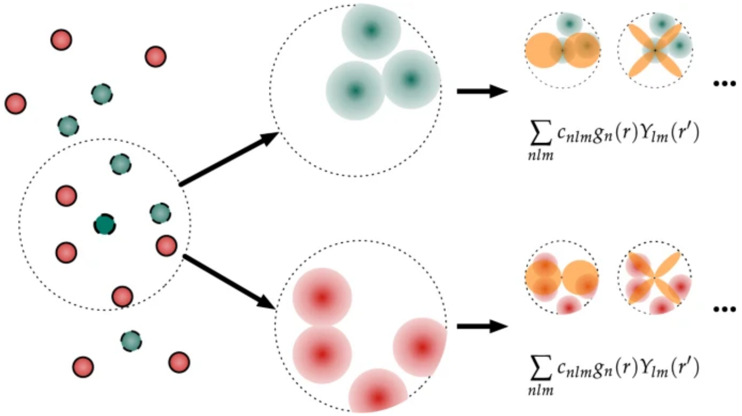
$$\boldsymbol{x} = |r - r'| \tag{53}$$

up to a certain cutoff point; $\boldsymbol{x}$ is the descriptor in question, and r and r' are the positions of two particles of the system. [1]

The SOAP descriptor bases itself on the spherical harmonic expansion of atomic coordinates. [8] This neighborhood density ρ can be written as

$$\rho(\boldsymbol{r}) = \sum_{n,l,m} c_{nlm} g_n(\boldsymbol{r}) Y_{lm}(\boldsymbol{r}) \tag{54}$$

where $\boldsymbol{r}$ is the position, c are the expansion coefficients, and g and Y are the radial and angular spherical harmonics basis functions, respectively. A visual explanation of the SOAP descriptor can be seen in Figure 3.



$\sum_{nlm} c_{nlm} g_n(r) Y_{lm}(r')$
...
$\sum_{nlm} c_{nlm} g_n(r) Y_{lm}(r')$
...

Figure 3: Visualization of the SOAP descriptor. Taken from Ref. [8].

For a more comprehensive explanation of the SOAP descriptor, please see Ref. [9].

4.5.2 Gaussian Process Regression

When trying to infer new values from existing data, there are generally two approaches: parametric fitting when there is a plausible model, and interpolation or regression based purely on data when this is not the case. [1]

Gaussian Process Regression (GPR) falls into the latter category, but unlike more common techniques, it is non-linear and can be used for high-dimensional problems like atomic environments. Inherently, GPR has no preconceived analytical notion about what physical processes may underly the investigated system, relying instead only on the fed training data. [1]

GPR can be understood from a weight-space and a function-space perspective. In the following, the weight space view is described.

Deriving GPR in Weight-Space View. In both weight- and function-space view, the idea is to approximate an unknown function, $y(x)$, with a linear combination $\tilde{y}(x)$ of weights c_m and similarity functions $k(\boldsymbol{x}, \boldsymbol{x}_m)$:

$$y(x) \sim \tilde{y}(x) = \sum_{m=1}^{M} c_m k(\boldsymbol{x}, \boldsymbol{x}_m) \tag{55}$$

The $\boldsymbol{x}$ here represent the input space where the data lives; the $\boldsymbol{x}_m$ are a subset of the input space, the so-called sparse set: Representative points within the input space deemed sufficient to solve the problem (see also Section 4.5.2).

The similarity function $k(\boldsymbol{x}, \boldsymbol{x}_m)$, also called kernel function, determines the similarity between two points $\boldsymbol{x}$ and $\boldsymbol{x}_m$. There are several options to define the kernel; Ref. [1] cites the Gaussian

$$k(\boldsymbol{x}, \boldsymbol{x}_m) = \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{x}_m|^2}{2\sigma^2}\right) \tag{56}$$

and the linear

$$k(\boldsymbol{x}, \boldsymbol{x}_m) = \boldsymbol{x} \cdot \boldsymbol{x}_m \tag{57}$$

kernels as standards, and both are applied in this work.

Here, σ (not to be confused with σ_n, which appears in equation (58) onwards) represents the hyperparameter. One can imagine k as the "overlap" between two Gaussian curves centered around $\boldsymbol{x}$ and $\boldsymbol{x}_m$. Then, σ as the standard deviation of the Gaussian curves plays an obvious and important role: when chosen too small, there will be overfitting, as two points $\boldsymbol{x}$ and $\boldsymbol{x}_m$ will only have notable levels of overlap if they are right next to each other, and when chosen too large, the fit will become inaccurate since even very far apart - and therefore presumably unrelated - $\boldsymbol{x}$ and $\boldsymbol{x}_m$ will have large overlap.

In general, kernel choice is arbitrary, but nonetheless highly relevant to the effectiveness of the model. [1]

To fit the GPR model to the data, the weights or coefficients c_m are chosen according to which minimize the loss function $\mathcal{L}$

$$\mathcal{L} = \sum_{n=1}^{N} \frac{[y_n - \tilde{y}(\boldsymbol{x}_n)]^2}{\sigma_n^2} + \sum_{m,m'}^{M} c_m k(\boldsymbol{x}_m, \boldsymbol{x}_{m'}) c_{m'} \tag{58}$$

where N is the number of data points and M is the number of representative (sparse) points; this will be relevant for sparse GPR in Section 4.5.2. Note that the c_m also appear within the structure of $\tilde{y}(\boldsymbol{x}_n)$ in equation (55). The second sum is the regularization term, in this case the Tikhonov regularization in particular; it is necessary to prevent overfitting from just the first term alone. σ_n appears here as a parameter that defines how strongly each available data point y_n and its difference to the approximated value $\tilde{y}(\boldsymbol{x}_n)$ are weighed in the fit.

We can rewrite this loss function in index form to more easily isolate the c_m

$$\mathcal{L} = (y_n - k_{nm}c_m)^T \sigma^{-1}_{nn'} (y_{n'} - k_{n'm'}c_{m'}) + c_m^T k_{mm'} c_{m'} \tag{59}$$

where $\sigma^{-1}_{nn'}$ is a diagonal matrix of the inverse of all the values of σ_n from before. Indices differentiated only by an apostrophe imply the same number of entries: N for n and n' and M for m and m'.

Deriving the loss function by the transposed weights c_m^T gives

$$\nabla_{c_m^T}\mathcal{L} = -k_{mn}\sigma^{-1}_{nn'} y_{n'} + k_{mn}\sigma^{-1}_{nn'} k_{n'm'} c_{m'} + k_{mm'} c_{m'} \tag{60}$$

Note that the k_{mn} were transposed in one step.

To minimize the loss function, we let the derivative be equal to zero

$$0 = -k_{mn}\sigma^{-1}_{nn'} y_{n'} + k_{mn}\sigma^{-1}_{nn'} k_{n'm'} c_{m'} + k_{mm'} c_{m'} \tag{61}$$

$$\Rightarrow c_{m'} = (k_{mn}\sigma^{-1}_{nn'} k_{n'm'} + k_{mm'})^{-1} k_{mn}\sigma^{-1}_{nn'} y_{n'} \tag{62}$$

Now our c_m are defined only by our data y_n, our kernel values $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ and our parameters σ_n.

Sparse GPR. Using straightforward linear algebra to solve equation (62) still scales with $\mathcal{O}(N^3)$ in terms of computational cost. [1] To avoid this, sparse GPR can be employed.

In full GPR, the indices m and n in (62) have the same number of entries, so $M = N$, reducing the equation to

$$c_n = (k_{nn'} + \sigma_{nn'})^{-1} y_{n'} \tag{63}$$

A visualization of GPR using (62) and (63) can be seen in Figure 4. In full GPR, the number of coefficients c_m necessary to predict a new location $\tilde{y}$ is N, the number of data points. In sparse GPR, this reduces to M, the number of representative points, which is by in large independent of N.

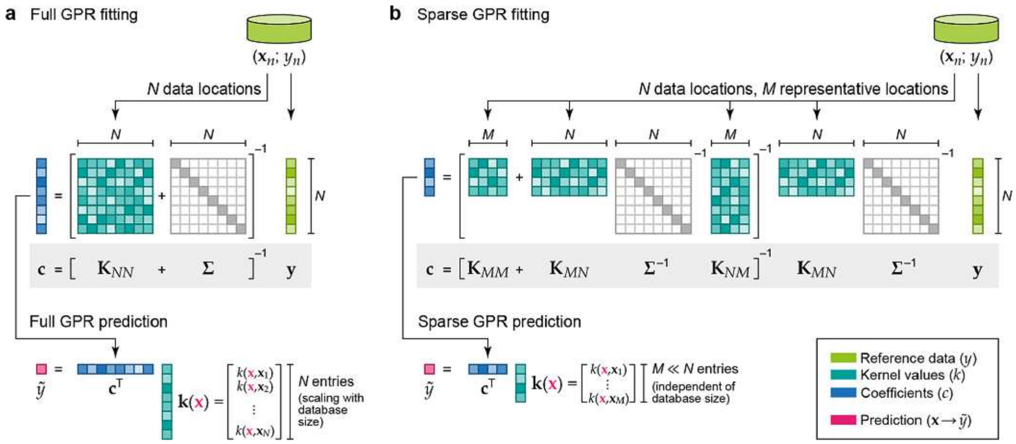

a Full GPR fitting
b Sparse GPR fitting
(x_n; y_n)
(x_n; y_n)
N data locations
N data locations, M representative locations
N
N
M
N
N
M
N
N
−1
−1
−1
−1
N
N
c = [K_NN + Σ]^−1 y
c = [K_MM + K_MN Σ^−1 K_NM]^−1 K_MN Σ^−1 y
Full GPR prediction
Sparse GPR prediction
ỹ = c^T k(x) = [k(x,x_1) k(x,x_2) ⋮ k(x,x_N)]
N entries (scaling with database size)
ỹ = c^T k(x) = [k(x,x_1) ⋮ k(x,x_M)]
M ≪ N entries (independent of database size)
Reference data (y)
Kernel values (k)
Coefficients (c)
Prediction (x → ỹ)

Figure 4: Visualization of full and sparse GPR in the matrix view. N and M are the number of entries each for n and m as used in this work. (a) Representation of full GPR with $M = N$. (b) Representation of sparse GPR with $M < N$. Taken from [1].

5 Results

5.1 Computational Workflow

A schematic depiction of how this project was handled can be found in Figure 5.

First, a calculation of a 54 atom cubic diamond structure was created using DFT with a PBE exchange correlation functional. This was used to train the `gap_fit` [10] ML model (optimized parameters used here can be found in Section 7). `gap_fit` outputs `GAP.xml` and associated files; these are used by `turbogap` [11] to run MD simulations that can be seen in Section 5.3.

Once the parameters are optimized, `phonopy` [2] is used to create a *Supercell* (see Section 4.4); this produces the file `phonopy_disp.yaml` and several POSCAR files to represent the examined displacements in the supercell (this is the *Phonopy pre-process* in Figure 5; the POSCAR files are the *Displacements*). These are then transformed into .xyz files to be used as starting points for `turbogap`, calculating the *Force constants* (see again Section 4.4) from Figure 5.

Depending on the displacements demanded by `phonopy`, there will then be one or more single-step MD simulations in .xyz format. To move on to *Phonopy post-process*, they have to be transformed into appropriate .xml files for `phonopy` to parse.

Once this is done, `phonopy` can create `FORCE_SETS`, which it can use in combination with a `band.conf` file describing the band path it should take to plot the desired phonon dispersion.

5.2 Creating an MD Trajectory with DFT

The results in this work are based on an initial MD simulation calculated using DFT methods with a PBE exchange correlation term. For this purpose, a geometry of 54 carbon atoms in a cubic diamond structure was fed into VASP to solve the Newtonian equations of motion for 1000 1-picosecond (ps) time steps. The main output file, `vasprun.xml`, containing all points of the MD trajectory (fed into descriptors) and forces (targets), was used to train the ML model.

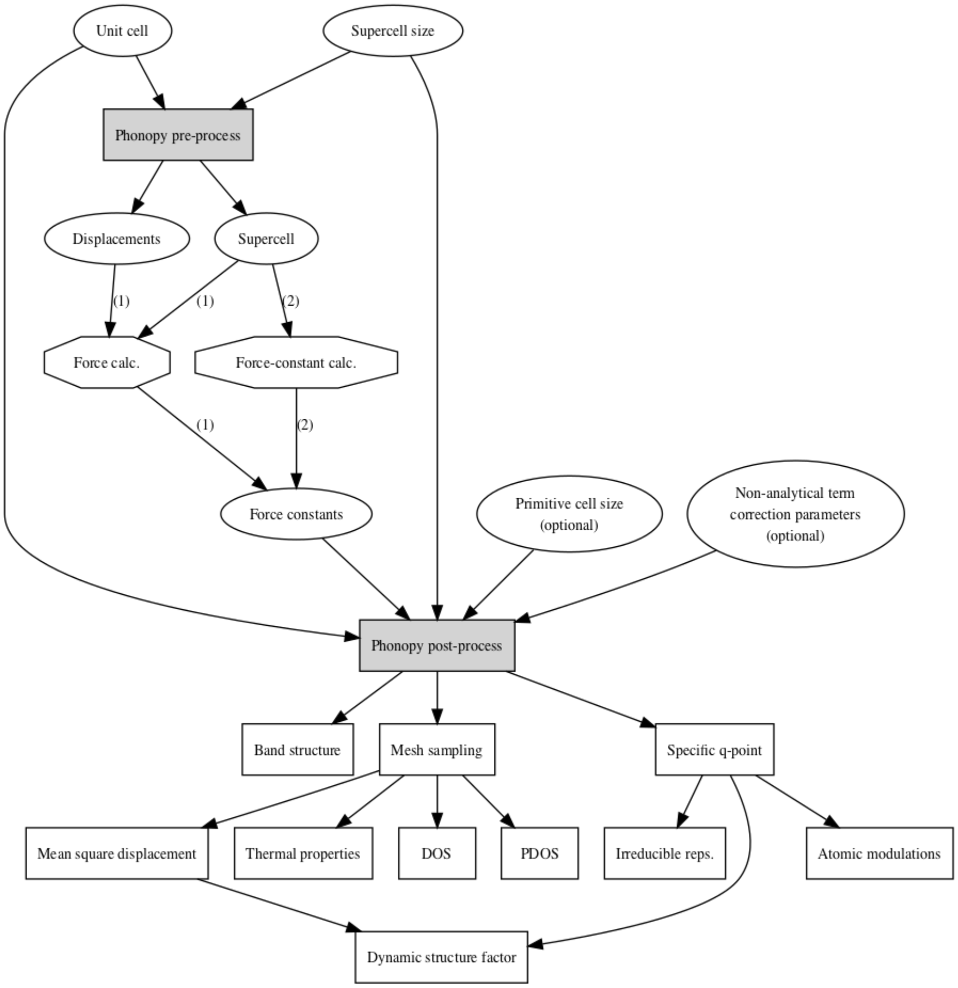


Figure 5: The `phonopy` workflow. This work starts at *Unit cell* and *Supercell size*, going through *Phonopy pre-process* to *Displacements* and *Supercell*. ML methods are applied to calculate *Force constants*. These are fed into *Phonopy post-process* to obtain *Band structure*. Taken from [12].

5.3 Training an ML Kernel

In order to expedite the process of getting to the *Force constants* as demanded by `phonopy` (see again, Figure 5), an ML kernel is trained on the data obtained in Section 5.2 using `gap_fit`. Descriptors used for training (see also 4.5.1) were a two-body descriptor using a Gaussian kernel (56) and a SOAP descriptor using a linear kernel (57). Exact values optimized for this data can be found in Section 7.

The results of this training are found in Figures 6, 7a and 7b. The simulations were conducted thrice, once using the full dataset to train the model, once using every second step (half set), and once using every fourth (quarter set) to see how dataset reduction would affect the fit. For each of them, only the first 500 of the 1000 simulated 1-ps steps are shown for better clarity.

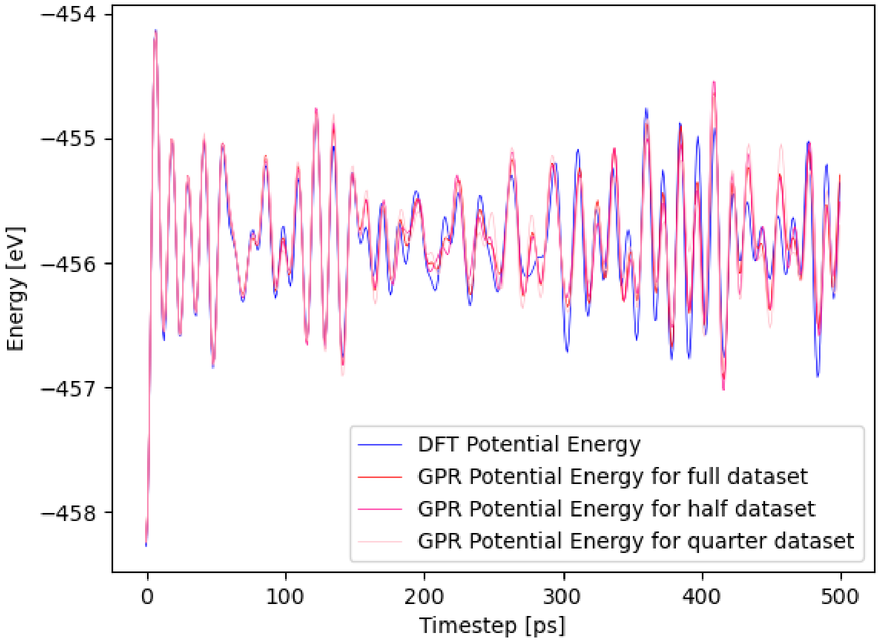
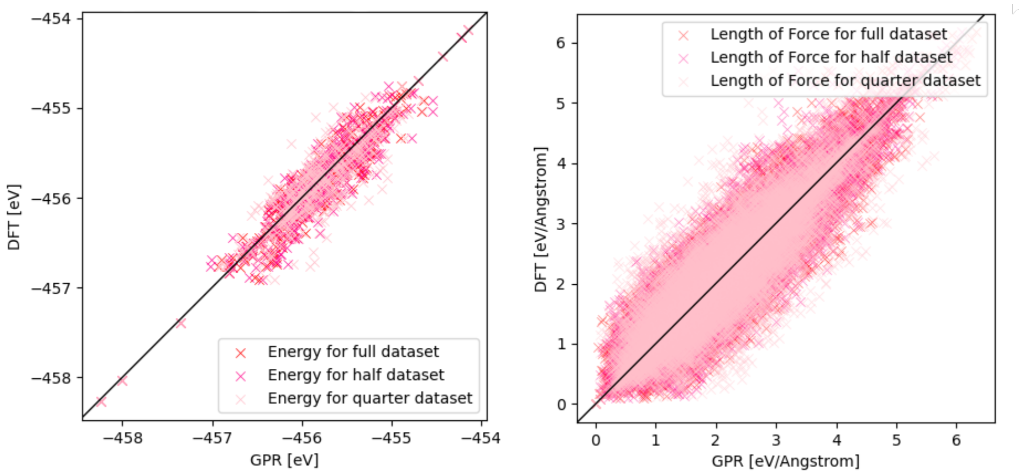


Figure 6: Comparison of different dataset MD runs with the original DFT simulation (blue line). Comparison included the full dataset (red line), half the original (magenta line), and quarter of the original set (pink line).



(a) Energies

(b) Norms of forces

Figure 7: Correlation plot comparing results from DFT versus GPR calculations for energies (a) and norm of forces (b) using the full set (red x), half the set (magenta x), and a quarter of the set (pink x).

Figure 6 shows the time evolution of the system energy for runs trained on different datasets, with the original DFT run in blue for comparison. At first, the runs are all almost identical, but they begin to diverge at around 200 steps.

Figure 7a and 7b show the correlation between the energy- and norm-of-force values generated by the DFT and GPR simulations, once again only for the first 500 steps. The root mean square errors for the energies and forces can be found in Table 1. Note that the errors in the energies are for the total system containing 54 atoms.

	Energies [eV]	Forces [eV/Å]
Full set	0.16 ± 0.1	0.029 ± 0.001
Half set	0.19 ± 0.05	0.029 ± 0.002
Quarter set	0.24 ± 0.1	0.037 ± 0.003

Table 1: Root mean square errors for energies and forces for each dataset.

It can be seen that the simulation proved robust against reductions of the dataset. Since we aim at accuracies on the scale of several meV/atom in the energies, we find that also using only the half dataset suffices. In general, reproducibility of the original system as calculated by DFT-PBE was mostly dependent on the chosen parameters, not least on the average temperature fed into `turbogap`. No matter the optimization, the system would still eventually diverge from the original DFT trajectory at about 200-300 ps.

5.4 Phonon Dispersion

In order to produce phonon spectra for the diamond system, this work makes use of the Python tool `phonopy` [2]. `phonopy` employs a supercell approach to calculate phonon spectra from a set of pre-calculated forces, see Figure 8.

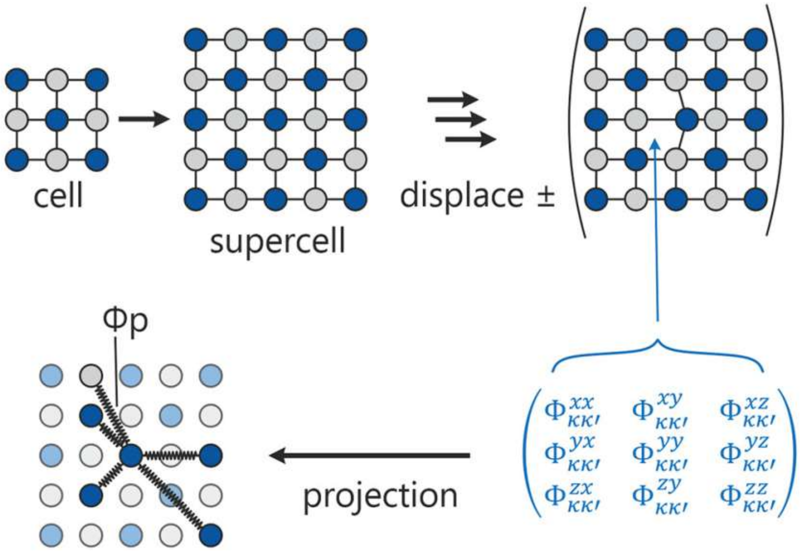

cell
supercell
displace ±
Φp
projection
$\Phi^{xx}_{\kappa\kappa'}$ $\Phi^{xy}_{\kappa\kappa'}$ $\Phi^{xz}_{\kappa\kappa'}$
$\Phi^{yx}_{\kappa\kappa'}$ $\Phi^{yy}_{\kappa\kappa'}$ $\Phi^{yz}_{\kappa\kappa'}$
$\Phi^{zx}_{\kappa\kappa'}$ $\Phi^{zy}_{\kappa\kappa'}$ $\Phi^{zz}_{\kappa\kappa'}$

Figure 8: Visual representation of the supercell approach. First, the view is expanded from a unit cell to a supercell containing many unit cells, then, one or more atoms are displaced from their resting position in the lattice. Force constants Φ on the remaining atoms can then be calculated (in this work via Gaussian Process Regression (GPR)). Image taken from Ref. [13].

The force constants Φ, generated as seen in Figure 8, are used in the harmonic expansion of the potential energy

$$V \approx \Phi_0 + \sum_{K=1}^{N} \sum_{\xi=1}^{3} \Phi_K^{\xi} u_K^{\xi} + \frac{1}{2} \sum_{K=1}^{N} \sum_{K'=1}^{N} \sum_{\xi=1}^{3} \sum_{\eta=1}^{3} \Phi_{KK'}^{\xi\eta} u_K^{\xi} u_{K'}^{\eta} + \mathcal{O}(u^3) \qquad (64)$$

where the u refer to the displacements, K and K' refer to the investigated lattice site, and ξ and η to the cartesian component. The linear term vanishes at equilibrium. Solving this harmonic oscillator provides the dispersion relation for the phonon spectrum.

This work examined an fcc diamond, with two carbon atoms (oriented in the $< 111 >$ direction) as its base. In the first Brillouin zone, this results in three acoustic modes (one longitudinal and two transversal), and $3 \cdot 2 - 3 = 3$ optical modes. That should be the maximum amount seen in the results.

Four different supercell sizes (see Figure 8) were chosen; the supercell of dimension 1 (Figure 10) proved too small to accommodate all modes, showing only one acoustic and one optical mode regardless of path. At a supercell of dimension 4 (Figure 13) the values had converged. Phonon spectra for each dimension can be found in Figure 10, 11, 12 and 13, each with comparisons of the different datasets used. Figure 9 shows a comparison plot for a spectrum of the same system calculated with DFT-PBE (blue line), the method on which `gap_fit` was trained.

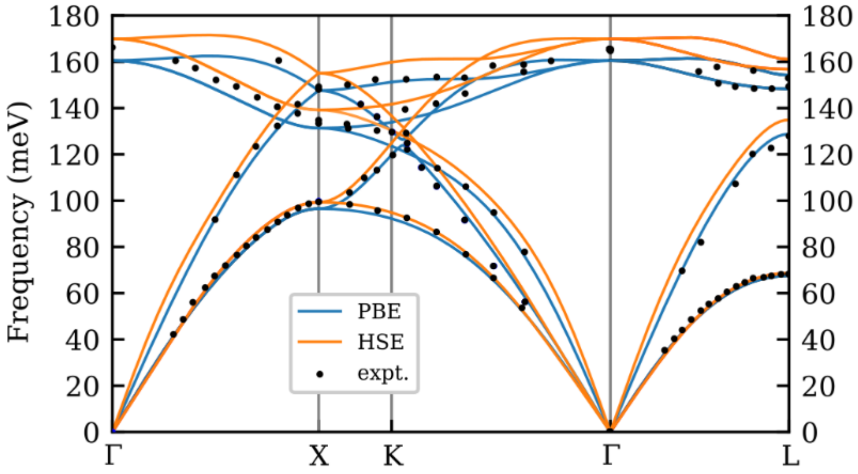


Figure 9: Phonon dispersion for cubic diamond as calculated by DFT-PBE (blue line), DFT-HSE (yellow line) and experimental values (black dots). Taken from [14].

Both in the ML and DFT calculations of the spectra, the transversal acoustic modes were degenerate along the $\Gamma \to X$ and the $\Gamma \to L$ paths, as well as one of the optical modes. Along the lower symmetry paths $X \to K$ and $K \to \Gamma$, the degeneracy is undone, showing the full set of six modes at

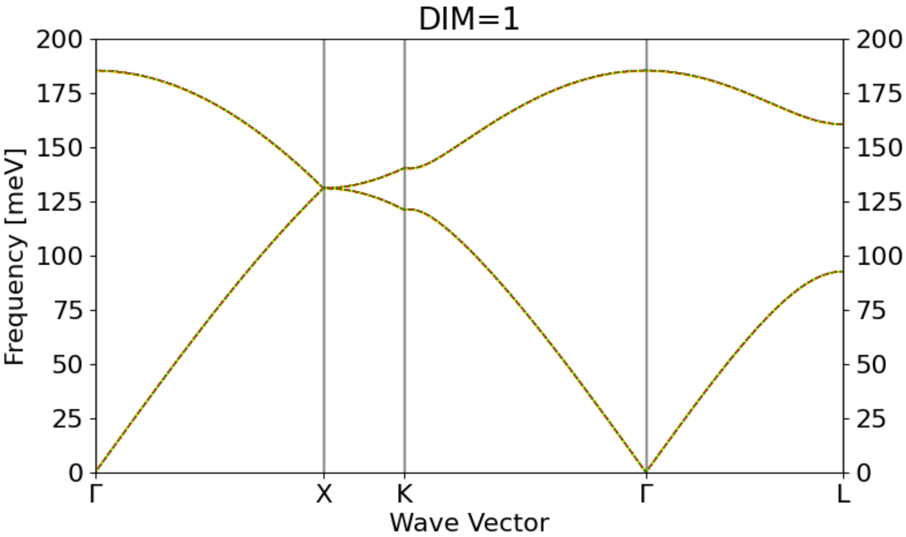


Figure 10: Phonon dispersion for cubic diamond as calculated using ML methods for a supercell with dimensions 1 1 1. Calculations were done for the full dataset (red line), half set (yellow dashed line), and the quarter set (green dotted line). This supercell proved too small to accommodate all relevant modes.

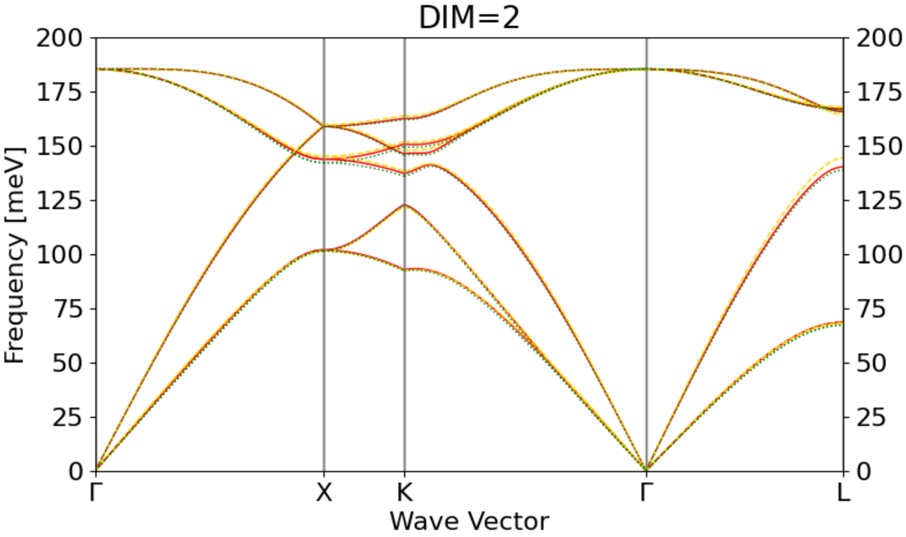


Figure 11: Phonon dispersion for cubic diamond as calculated using ML methods for a supercell with dimensions 2 2 2. Calculations were done for the full dataset (red line), half set (yellow dashed line), and the quarter set (green dotted line).

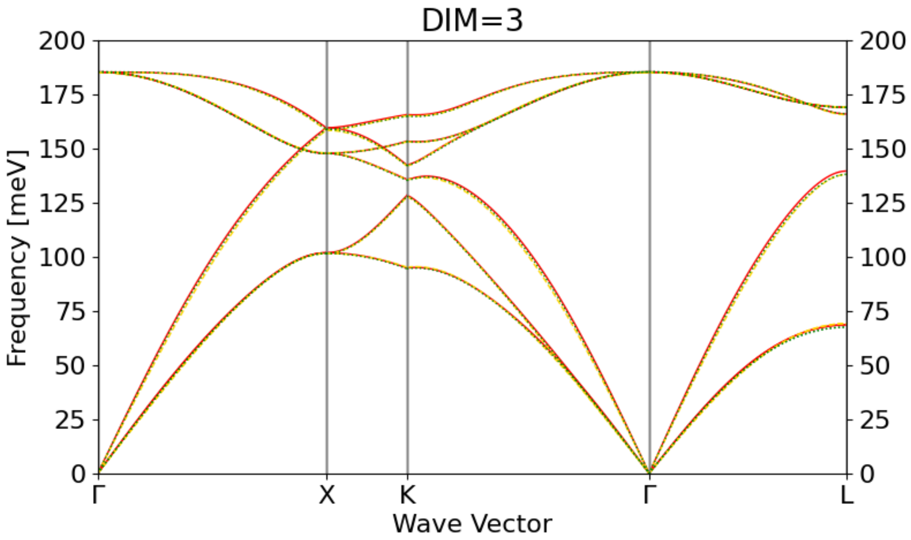


Figure 12: Phonon dispersion for cubic diamond as calculated using ML methods for a supercell with dimensions 3 3 3. Calculations were done for the full dataset (red line), half set (yellow dashed line), and the quarter set (green dotted line).

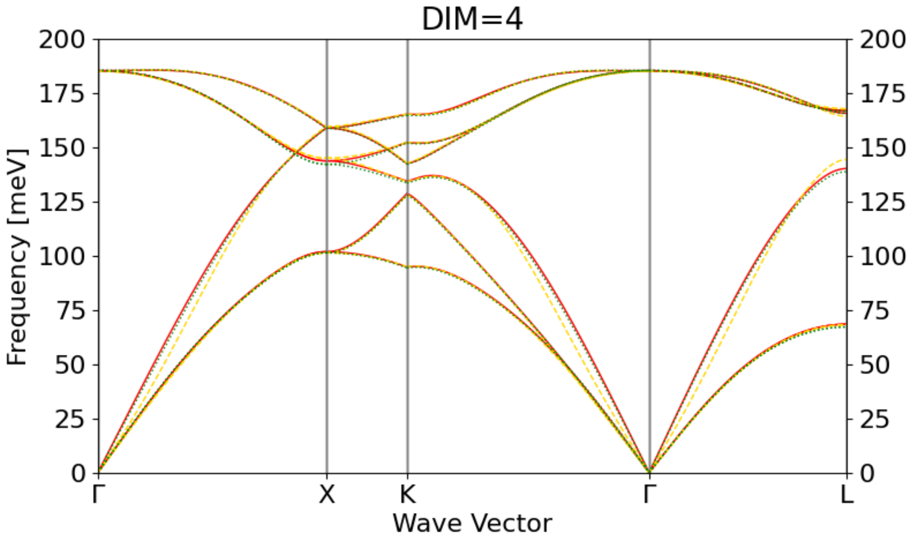


Figure 13: Phonon dispersion for cubic diamond as calculated using ML methods for a supercell with dimensions 4 4 4. Calculations were done for the full dataset (red line), half set (yellow dashed line), and the quarter set (green dotted line).

the K-point in both the DFT-PBE and the ML calculations.

None of the sets diverged strongly from each other; the largest divergence from the path can be seen in the half set (yellow dashed line) in the supercell of dimension 4 (Figure 13). The maximum frequencies of the optical modes from the ML methods were slightly higher than those of the DFT-PBE calculation, at about 185 meV compared to 170 meV for DFT-PBE.

6 Conclusion

Using ML methods to generate MD simulations and phonon spectra proved incredibly time efficient compared to conventional methods like DFT at similarly accurate results. One simulation for the full dataset including training took around 50 seconds (30 seconds training `gap_fit` + 20 seconds simulation with `turbogap`) compared to approximately 2100 seconds for the DFT run. Only fractions of seconds were added for the generation of force sets for `phonopy` once the training was already complete.

Reproduction of the original trajectory was possible up to about 200 ps, and did not change when the dataset was reduced to half, and only mildly when it was reduced to quarter the original set; reduction of the datasets also reduced training time, see Table 2. Simulation via `turbogap` took approximately equally long for each set, while training time scaled almost linearly with the set size of the training data.

The used GPR method also proved to be robust against this dataset reduction; there were no significant changes to neither the MD simulation nor the phonon spectra based on the dataset reduction. The most important factor in determining how well the MD simulation and in turn the use for phonon spectra would go, were always the parameters and descriptors chosen during the training process. They can be found in Section 7.

Overall, GPR has proven to be sufficient in terms of data pro- and reproduction of DFT examples at several orders of magnitude less of computational cost and time of DFT. Training parameter choice remains the central deciding factor for the success of the method.

	Training	Simulation (1000 steps)
Full set	33.8 sec	20.87 sec
Half set	18.08 sec	20.78 sec
Quarter set	9.58 sec	20.93 sec

Table 2: Duration of each part of the data creation using ML methods.

References

[1] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian process regression for materials and molecules," *Chemical Reviews*, vol. 121, no. 16, pp. 10073–10141, 2021. PMID: 34398616.

[2] A. Togo and I. Tanaka, "First principles phonon calculations in materials science," *Scr. Mater.*, vol. 108, pp. 1–5, Nov 2015.

[3] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964.

[4] R. M. Dreizler and E. K. U. Gross, *The Kohn-Sham Scheme*, pp. 43–74. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990.

[5] M. C. Holthausen and K. Wolfram, *A Chemist's Guide to Density Functional Theory*. Wiley-VCH, 2001.

[6] S. Schoenbauer, "Machine learning simulation of h_2o molecules using gaussian process regression," 2023.

[7] S. König, "Machine learning for ab initio simulations of molecular hydrogen crystals," bachelor's thesis, Technische Universität Wien, 2021.

[8] M. F. Langer, A. Goeßmann, and M. Rupp, "Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning," *npj Computational Materials*, vol. 8, p. 41, Mar 2022.

[9] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Phys. Rev. B*, vol. 87, p. 184115, May 2013.

[10] 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.*, vol. 104, p. 136403, Apr 2010.

[11] M. A. Caro, "Optimizing many-body atomic descriptors for enhanced computational performance of machine learning based interatomic potentials," *Phys. Rev. B*, vol. 100, p. 024112, Jul 2019.

[12] Togo, Atsushi, “Phonopy - workflow,” 2009. [Online; accessed April 17, 2023].

[13] J. Hempelmann, P. Müller, P. Konze, R. Stoffel, S. Steinberg, and R. Dronskowski, “Long-range forces in rock-salt-type tellurides and how they mirror the underlying chemical bonding,” *Advanced materials (Deerfield Beach, Fla.)*, vol. 33, p. e2100163, 09 2021.

[14] L. Razinkovas, M. W. Doherty, N. B. Manson, C. G. Van de Walle, and A. Alkauskas, “Vibrational and vibronic structure of isolated point defects: The nitrogen-vacancy center in diamond,” *Phys. Rev. B*, vol. 104, p. 045303, Jul 2021.

[15] J. Bien, Y. Xu, and M. W. Mahoney, “Cur from a sparse optimization viewpoint,” in *Advances in Neural Information Processing Systems* (J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds.), vol. 23, Curran Associates, Inc., 2010.

7 Appendix

The parameters for the two-body descriptor were chosen as follows:

```
distance_2b
cutoff=4.0
covariance_type=ard_se
delta=0.5
theta_uniform=1.0
sparse_method=uniform
add_species=T
n_sparse=20
```

`cutoff` describes to which distance apart atom pairs were considered, `covariance_type=ard_se` denotes that a Gaussian kernel (56) was used, `theta_uniform` is the hyperparamter σ as seen in (56). `delta` describes what relative portion is determined by the descriptor. 20 sparse points (see 4.5.2) were chosen at uniform distance from each other.

The parameters for the SOAP descriptor were chosen as:

```
soap_turbo
rcut_hard=4.0
rcut_soft=3.0
covariance_type=dot_product
delta=0.2
atom_sigma_r={{0.5}}
atom_sigma_t={{0.5}}
atom_sigma_r_scaling={{0.}}
atom_sigma_t_scaling={{0.}}
l_max=6
alpha_max={{6}}
amplitude_scaling={{1.}}
central_weight={{1.}}
n_species=1
species_Z={{6}}
add_species=F
n_sparse=20
scaling_mode=polynomial
basis=poly3
radial_enhancement=0
zeta=15
sparse_method=cur_points
```

`rcut_hard` and `rcut_soft` denote to what point neighbors were considered and with what transition width (see Figure 3, the gray dotted circle), `covariance_type=dot_propduct` denotes that a linear kernel (57) was used. The various `atom_sigma` describe the weights the radial (r) and angular (t) functions were given, as well as their scaling. `l_max` and `alpha_max` describe to which point the spherical harmonic expansion was continued (l and n respectively in (54)). `central_weight` describes how much weight is given to the central atom for the SOAP descriptor. `zeta` describes the power of the kernel (effectively turning a linear kernel into a polynomial one). The 20 sparse points were determined using CUR decomposition, see [15] for further reference.

The loss function parameter σ_n (see equation (58)) for the `gap_fit` program was chosen like so:

```
default_sigma={0.0001 0.0003 0.0003 0}
```

for energies, forces, stresses, and Hessians, respectively.