

Calculations of lattice vibrational mode lifetimes using *Jazz*: a Python wrapper for *LAMMPS*

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2015 Modelling Simul. Mater. Sci. Eng. 23 045002

(<http://iopscience.iop.org/0965-0393/23/4/045002>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

This content was downloaded by: daw

IP Address: 130.127.188.144

This content was downloaded on 03/04/2015 at 12:11

Please note that [terms and conditions apply](#).

Calculations of lattice vibrational mode lifetimes using *Jazz*: a Python wrapper for *LAMMPS*

Y Gao, H Wang and M S Daw

Department of Physics and Astronomy, Clemson University, Clemson, SC 29634, USA

E-mail: daw@clemson.edu

Received 24 October 2014, revised 10 February 2015

Accepted for publication 23 February 2015

Published 2 April 2015



CrossMark

Abstract

Jazz is a new python wrapper for *LAMMPS* [1], implemented to calculate the lifetimes of vibrational normal modes based on forces as calculated for any interatomic potential available in that package. The anharmonic character of the normal modes is analyzed via the Monte Carlo-based moments approximation as is described in Gao and Daw [2]. It is distributed as open-source software and can be downloaded from the website <http://jazz.sourceforge.net/>.

Keywords: vibrational mode lifetimes, moment approximation, software

(Some figures may appear in colour only in the online journal)

1. Introduction

Our knowledge of the vibrational dynamics in solids is less developed than it could be—maybe even less developed than our knowledge of the dynamics of electrons. Recent advances in experimental probes are opening up new avenues of investigation into vibrational mode lifetimes in solids. For example, mode lifetimes affect the results of measurements based on neutron scattering, IR absorption, Raman spectroscopy, and thermal transport. Aside from being a key fundamental property of solids, there is also considerable interest because of applications, particularly with regards to thermal properties. There is therefore a need for corresponding advances in theoretical techniques.

A recent theoretical innovation for calculating vibrational mode lifetimes incorporates a physical understanding of the nature of dissipation in solids through the novel combination of two modern techniques: Liouvillian operator and the recursion method [2–5]. Calculations of mode lifetimes using the new approach are much faster than previous methods because they are based on phase-space integrals which can be done by standard Monte Carlo techniques.

The lifetime of each vibrational mode is obtained as an ensemble average of specific products of mode-resolved displacements and forces.

We have implemented the new technique as a wrapper (*Jazz*) for the general MD code LAMMPS, that incorporates a wide variety of semi-empirical, interatomic potentials for solids. Because recent versions of LAMMPS now include a python interface, it is relatively easy to control LAMMPS and extract its internal variables from an external python script, or wrapper. Using that interface, *Jazz* treats LAMMPS as a blackbox source of energies and forces. The *Jazz* wrapper manages the Metropolis Monte Carlo process, displacing the atoms in a controlled way, querying LAMMPS for energies in order to make the accept/reject decision. The forces are extracted from LAMMPS and the appropriate averages obtained, to form the moment-based approximation to the vibrational lifetime of each mode.

Jazz is now included in the third-party contributions to LAMMPS and is freely available to the community. In this paper we describe the program and its uses. General information about the implementation is discussed in section 2 and an example is briefly shown in section 3.

2. Mode of implementation

We describe in this section some generalities of the technical details of the implementation. (Full details are available in the documentation provided with the package).

The three primary python modules provided in this package are

- **genvecs0** (determines normal modes from dynamical matrix)
- **calcavgs** (does the MC averaging, producing the averages in equation (6))
- **calcmoments** (combines MC averages to calculate moments and lifetimes, equations (7) and (8))

Their functions are explained in general in the following.

The first two modules interface with LAMMPS. The changes in energy required for the MC are computed brute-force, in that the total energy is computed before and after an attempted alteration of atomic positions; no attempt is made to economize based on possible locality of the interactions. This is both advantageous and disadvantageous. The disadvantage is one of speed. It is possible in local interactions to speed up significantly the calculation of the change in energy. However, this requires a different program for each type of interaction, because the detailed calculation of the change in energy due to displacing a single atom involves writing specialized code for each type of potential, which is time-consuming work. The advantage of our blackbox approach is that it will work for any potential available in LAMMPS. At this stage of our research, this flexibility was desired over whatever speed advantage might be gained for specific potentials.

2.1. Normal modes

The vibrational lifetimes are calculated for each mode, so the normal mode analysis is a necessary first step. At low temperatures, the normal modes are given by the dynamical matrix¹. These are obtained from the Hessian of the potential energy via numerical derivatives of the forces. Let $u_{i\alpha}$ be the displacement of atom i (mass m_i) from its equilibrium position along the Cartesian direction α . The dynamical matrix [6] is formed from the Hessian (matrix of second derivatives of the potential) evaluated at equilibrium:

¹For some simple crystal structures (see Maradudin [6]) symmetry restricts the normal modes to be effectively independent of temperature. However, more complex structures will have normal modes that are temperature-dependent. It is possible in these cases to construct an effective, temperature-dependent dynamical matrix from the ensemble average of the displacement–displacement correlation matrix. This extension of *Jazz* is planned.

$$D_{i\alpha,j\beta} = \frac{1}{\sqrt{m_i m_j}} \frac{\partial^2 V}{\partial u_{i\alpha} \partial u_{j\beta}} \Big|_0 \quad (1)$$

The eigenvectors of D can be labelled by index κ and correspond to normal modes with frequency ω_κ

$$\omega_\kappa^2 \eta_{i\alpha}^{(\kappa)} = \sum_{j\beta} D_{i\alpha,j\beta} \eta_{j\beta}^{(\kappa)} \quad (2)$$

The matrix Y formed from the eigenvectors

$$Y_{i\alpha,\kappa} = \frac{1}{\sqrt{m_i}} \eta_{i\alpha}^{(\kappa)} \quad (3)$$

transforms from the normal mode amplitudes to the atomic displacements

$$u_{i\alpha} = \sum_{\kappa(\text{osc})} Y_{i\alpha,\kappa} A_\kappa \quad (4)$$

where A_κ is the set of amplitudes of oscillators and $u_{i\alpha}$ is the set of atomic displacements. The sum is restricted to oscillators (excluding free modes). Likewise, the mode accelerations can be obtained from the atomic forces by

$$\ddot{A}_\kappa = \sum_{i\alpha} \frac{F_{i\alpha}}{m_i} Y_{i\alpha,\kappa} \quad (5)$$

Jazz makes extensive use of NumPy, a fundamental Python package for handling matrix-vector operations and other typical scientific computing operations. Generally in the cases to which we have applied *Jazz*, the majority of the computational time is consumed within *LAMMPS*.

2.2. Ensemble averages

The ensemble averages in *Jazz* are over positions alone, and are based on the canonical distribution function $\exp(-V/T)$, where V is the potential energy of the system and T is the temperature (expressed in energy units, as with V). An additional constraint is imposed beyond the usual canonical ensemble, which is that the center-of-mass remains fixed. This constraint makes it easier to deal with averages involving displacements, because of the existence of free modes.

Three free modes involve, for example, sliding of the system in each of three directions. Periodic boundary conditions alone do not limit this sliding. Also, in some cases, such as a cylinder that is periodic along its length but effectively free to rotate, the system will in fact end up rotated during the process of MC sampling unless something is done to avoid it. In *Jazz* we have avoided this problem by identifying certain normal modes as ‘free modes’ and the remaining as ‘oscillators’. Free modes are modes for which there is no resistance to displacement, and which therefore correspond to (effectively) zero frequency. Oscillators are modes for which there is at least linear resistance to displacement. Therefore, we can freeze out the free modes by carrying out the MC search only in the subspace of oscillators.

The calculation proceeds by carrying out the MC sampling in the subspace of normal mode amplitudes *excluding free modes*². Then the displacements, from equation (4), are added to the

²Limiting the MC steps to the subspace of oscillators guarantees that the system will not slide—or rotate if the configuration has that freedom—the motion of which can contaminate averages based on displacements.

equilibrium positions and are transmitted to *LAMMPS*, which computes the energy and forces of the modified configuration. Three averages are computed (for each mode):

$$\begin{aligned} S_{1,\kappa} &= \langle A_{\kappa}^2 \rangle \\ S_{2,\kappa} &= \langle A_{\kappa} \dot{A}_{\kappa} \rangle \\ S_{3,\kappa} &= \langle \dot{A}_{\kappa}^2 \rangle \end{aligned} \quad (6)$$

from which the moments for each mode are computed:

$$\begin{aligned} \mu_{2,\kappa} &= \frac{S_{2,\kappa}}{S_{1,\kappa}} \\ \mu_{4,\kappa} &= \frac{S_{3,\kappa}}{S_{1,\kappa}} \\ \gamma_{4,\kappa} &= \frac{S_{3,\kappa} S_{1,\kappa}}{(S_{2,\kappa})^2} \end{aligned} \quad (7)$$

It is possible to obtain S_2 exactly, by using the virial theorem, as proportional to T . However, we have found that the value of γ_4 converges more quickly with sampling size if we use the same sampling for all three sums in equation (6), rather than using the exact value for one sum and the sampled values for the other two. In particular, calculating all three S values the same way guarantees that $\gamma_4 \geq 1$ even for low temperatures where γ_4 is approaching 1.

Finally, the mode lifetimes are obtained from [2]

$$\tau_{\kappa} = \frac{1.41}{\sqrt{\mu_{2,\kappa}(\gamma_{4,\kappa} - 1)}} \quad (8)$$

2.3. Parallelization

The first versions of *Jazz* are implemented strictly in serial mode, in that *LAMMPS* calculates the energy changes and forces in serial mode, and also the MC calculations managed by *Jazz* are done serially. Because Monte Carlo is easily run in parallel processes, it is an easy matter to run multiple instances of *Jazz*, each using a different pseudo-random number seed, and then to combine the results. A measure of statistical error can also be obtained this way by sub-averaging.

However, we plan in the near future to implement a fully parallel version of *Jazz*. First, *LAMMPS* is designed to run in parallel, and for larger calculations it will be more efficient to drive the parallel part of *LAMMPS* rather than the serial mode. Also, it will be an easy matter to set up *Jazz* to manage multiple processes and combine the results.

3. Example: FCC-LJ

As an example of the both the utility of the approach in general and of *Jazz* in particular, we examine the effect of disorder on the vibrational modes of an anharmonic system. This is part of a larger study that we plan to publish in the future.

Generally we understand well how disorder (such as mass disorder in an isotopic mixture) affects vibrational modes in harmonic solids, often through a perturbative approach. Disorder generally is expected to reduce thermal conductivity by modifying the normal modes so that they become localized, all of which is understood as a harmonic effect.

However, there is noticeably less in the literature on a general understanding of how those differences can affect mode lifetimes, which intrinsically requires a nonlinear treatment.

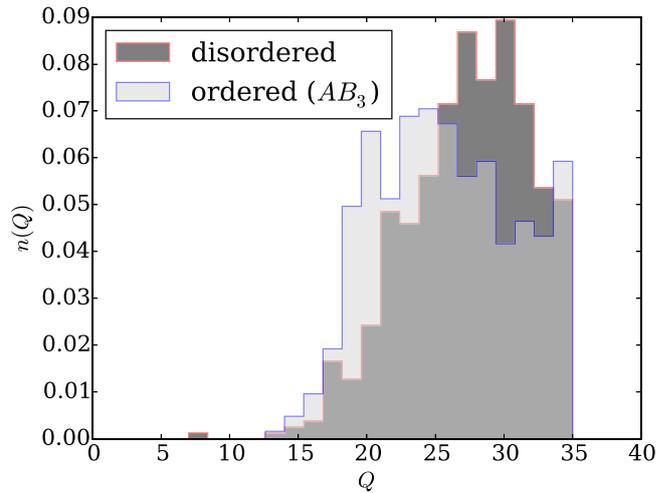


Figure 1. Disorder in a 3:1 binary alloy with significant bond-length difference causes the full set of normal modes to move to a higher quality factor.

Computer studies in the 60s by workers at Los Alamos [7] obtained surprising results in model calculations on systems with both disorder and anharmonicity, results that contradict the expectations based on perturbation theory. For example, in some range of parameters, disordered anharmonic systems can be seen to have *higher* lattice thermal conductivity than would be expected from perturbation theory. This rather surprising effect seems to have been noticed mostly by those working in low-dimensional systems, but—as far as we can determine—has not received the same attention from those working in bulk systems. One general import of the work has been to question the additivity of relaxation times that does hold in many cases but is not generally true.

We are studying then at a generic level the interplay of disorder and anharmonicity, with the well-known Lennard–Jonesium (LJ). In the LJ potential, for example, we can study binary alloys where the constituents differ based on mass, on bond strength, and/or bond length.

In doing this calculation, we set up the usual input for *LAMMPS*, by preparing files to specify atomic positions and types, with the LJ parameters for the binary alloy being specified as well. *Jazz* was then run (at a specified temperature) to analyze the vibrational mode frequencies and lifetimes. The imposition of periodic boundary conditions selects vibrational modes whose wavevectors are consonant with the periodicity. For that set of normal modes, the code calculates the normal mode eigenvectors, as well as temperature-dependent frequency and lifetime of each.

For example, we have set up cubic cells of 256 atoms in an *FCC* structure, in both ordered and disordered structures at various compositions, and examined the normal modes in these cells. For 256 atoms, there are 765 non-sliding modes, and for each we have the eigenvector, the frequency, and lifetime. One quick method of examining the characteristic of oscillators is through the Q -factor, where Q is the product of oscillator frequency and lifetime. We can examine the ensemble of modes in this cell then through a plot of the distribution of quality factors, as can be seen in figure 1. As is seen in this particular case, disorder causes the oscillators to shift to a longer lifetime. This is consistent with Payton *et al* [7], who found that in a disordered lattice, increasing anharmonicity can lead to *higher* thermal conductivity, which contradicts the usual thinking about such systems. They reasoned that while disorder

can cause localization of vibrational modes, anharmonicity causes a decay of those localized modes that effectively restores a delocalized nature.

Several examples—some simpler and some more complex than this one—are included in the package distributed with this code.

4. Availability & disclaimer

To make *Jazz* available to as many researchers as possible, it has been released as open-source software under the GNU General Public License. Thus, *Jazz* can be used free of charge, everyone can contribute to the software, extend it to his own needs and share new developments. The python source code can be downloaded from the website <http://jazz.sourceforge.net>. (*LAMMPS* is available separately from <http://lammmps.sandia.gov>).

Jazz comes with no warranty of any kind. It is a copyrighted code that is distributed free-of-charge, under the terms of the GNU Public License (GPL). This is often referred to as open-source distribution—see www.gnu.org or www.opensource.org. The legal text of the GPL is in the LICENSE file that is included in the *Jazz* distribution.

The sourceforce site includes a user discussion forum, which we suggest as the best means for resolving questions about the program and for finding out what other people may be doing with the code.

Acknowledgments

Research supported by the US Department of Energy, Office of Basic Energy Science, Division of Materials Sciences and Engineering under Award ER 46871.

References

- [1] Plimpton S 1995 *J. Comput. Phys.* **117** 1–9 (see <http://lammmps.sandia.gov>)
- [2] Gao Y and Daw M S 2014 *Modelling Simul. Mater. Sci. Eng.* **22** 075011
- [3] Dickel D and Daw M S 2010 *Comput. Mater. Sci.* **47** 698
- [4] Dickel D and Daw M S 2010 *Comput. Mater. Sci.* **49** 445
- [5] Gao Y, Dickel D, Harrison D and Daw M S 2014 *Comput. Mater. Sci.* **89** 12
- [6] Maradudin A A, Montroll E W and Weiss G H 1971 *Solid State Physics, Supplement 3: Theory of Lattice Dynamics in the Harmonic Approximation* (London: Academic)
- [7] Payton D, Rich M and Visscher W 1967 *Phys. Rev.* **160** 160