

## Notes on velocity autocorrelation function calculations after SIESTA

Andrei Postnikov ( andrei.postnikov@univ-lorraine.fr ), Shimla – Metz, Nov/Dec 2014

Velocity AutoCorrelation Function (VACF),  $C_v(\tau)$ , calculated over  $N$  steps of molecular dynamics (MD) simulation, is defined as follows:

$$C_v(\tau) = \frac{1}{N} \sum_{i=1}^N \frac{1}{t_{\max}} \sum_{t_0=1}^{t_{\max}} [\mathbf{v}_i(t_0) \cdot \mathbf{v}_i(t_0 + \tau)]. \quad (1)$$

Its Fourier transform  $G(\omega)$  taken to square yields the vibration density of states  $I(\omega)$ :

$$I(\omega) = |G(\omega)|^2, \quad G(\omega) = \int_{-\infty}^{\infty} d\tau C_v(\tau) e^{-i\omega\tau}. \quad (2)$$

The summation in Eq. (1) runs over all atoms  $N$ ; it can be split into contributions from different species and thus reveal species-related spectral features. (This is not provided in the present script but can be easily implemented).

When considering MD as a mean to study lattice dynamics, alternative to, say, frozen phonons (F $\Phi$ ), the following points deserve attention.

1. Whereas a F $\Phi$  calculation needs a fixed number of independent calculations to extract the elements of dynamical matrix (say,  $6N$  for a  $N$ -atom system, without symmetry taken into account), the MD calculation runs over a number of consecutive time steps, given by two important parameters – the time step  $\Delta t$  and the total simulation time  $t_{\text{MD}}$ . The time step has to be short enough to adequately describe the fastest vibration present in the system,

$$\Delta t \ll 1/\nu_{\max},$$

whereas the total simulation time ultimately defines the frequency resolution obtained,

$$t_{\text{MD}} \geq 1/(\Delta\nu).$$

The number of MD steps in the calculation,  $t_{\text{MD}}/\Delta t$ , does not depend on the number of atoms in the system. Therefore, starting from certain system size (typically  $\sim 10^2 - 10^3$  atoms) the MD may become totally less demanding than F $\Phi$ . However, a F $\Phi$  calculations can be done independently and in parallel whereas the MD depends on its previous history and needs to be sequential in this sense.

2. If a F $\Phi$  calculation turns out to be unsatisfactory for some reason, it usually has to be re-done from scratch. A MD calculation can be tested on a relatively short time span, and then gradually continued, whereby the accumulated history yields better frequency resolution (an increasing number of sample points spans the initially fixed frequency range). The oldest (probably poorly thermalized) parts of history can be gradually dumped, as sufficient number of “good” steps is accumulated.

3. A MD simulation can be organized at a given temperature and takes anharmonic effects into consideration (as no special hypothesis of harmonic vibrations is done, and atoms are propagated just according to the Newtonian dynamics).

A word of caution concerns the organization of MD calculations for periodic systems, that typically suggests a supercell with some “reasonably large” number of atoms. In a F $\Phi$  calculation, a “too small” supercell size will manifest itself in a “bad” (too sparse) effective  $\mathbf{q}$ -sampling over the Brillouin zone, and in “ugly” (not smooth) resulting density of modes, with otherwise no dramatic consequences. On the contrary, a too small supercell in a MD calculation, whose underlying idea

is an averaging over large ensemble, would mean imposing too artificial constraint on the atoms' movement. The resulting "lack of ergodicity" (meaning that large parts of the global phase space are excluded from the sampling) may lead to quite meaningless results. The danger is that nothing would prevent a "naive" user from staging a MD simulation with just few atoms per periodic cell, thus imitating a situation which would never happen in a realistic condensed-matter system. For *finite* systems (molecules, clusters), on the contrary, a MD simulation would fully grasp the underlying dynamics, so that a calculation on a two-atom molecule may fully make sense.

The step-by-step velocities of all atoms needed for the VACF calculation are provided by SIESTA "directly" in the (binary) .MD file. Otherwise, the velocities can be estimated from step-by-step coordinates available either in .MD\_CAR or in .ANI files. Some details concerning these files are specified below. Some other output formats may appear in the future (or stem from other codes than SIESTA). With the utmost flexibility in mind, the calculation of VACF is organized as follows:

1. The .XV file is read in, to digest information about the total number of atoms and their splitting into groups of species (needed for some MD formats, see below). The atom coordinates in .XV are neglected (that means that any .XV for the given system may be used).

2. The time step value  $\Delta t$  is asked for, since this number is not saved in MD files – but can be found in SIESTA input/output. It is the user's responsibility to pass the correct number! A possible error would result in wrong scale along the frequency axis – hopefully an easy thing to detect...

3. The extension of a MD file has to be specified (chosen from .MD, .MD\_CAR or .ANI), and a corresponding routine invoked to extract or calculate the velocities which are then temporarily written into a direct-access file according to the unique local protocol. Another solution, that of keeping in memory the velocities of *all* atoms over *all* steps seems questionable at least for particularly huge cases, but can be implemented (as an option, following a memory check) without problems.

The important input options governing the MD calculation are the following.

MD.LengthTimeStep 1.0 fs (see the discussion below about the choice of the time step).

WriteMDhistory = T → writes .MD and .MD\_CAR

The .MD file is written in iomd.f as follows (for format = F):

```
write(iupos) istep, xa(1..3,1..nat), va(1..3,1..nat)
if ( varcel ) write(iupos) cell(1..3,1..3), vcell(1..3,1..3)
```

Velocities in va are in Bohr/fs.

The .MD\_CAR file is written in md\_out.F90, having the structure of record (MD step) as follows:

- two title lines ( ---[ Label ]--- / 1.0 );
- three lines with cell parameters in Å;
- one or more lines with (30 per line) numbers of atoms of each species :

```
write(iomd,"(30i6)") natoms(:) ;
```

(Only the total number of atoms is important for the following! They need to be summed up)

- the word "Direct";
- line by line, relative cord. of atoms (in units of lattice vectors).

WriteMDXmol = T → writes .ANI

The .ANI file is written in pixmol.f, having the structure of record (MD step) as follows:

- Total number of atoms;
- Empty line;
- Line by line, chem. label of atom and its three cartesian coords (in Å).

The best way to extract and use the velocities is to get them directly from the .MD file. It can be written either with variable cell or with fixed cell, depending on the switch `MD.WriteVariableCell`. In principle it is not a good idea to run a MD simulation with variable cell (which option is anyway allowed only with `MD.TypeOfRun = CG` or `Broyden` or `FIRE`, *not* with standard MD switches `Verlet` / `Nose` / `ParrinelloRahman` / `NoseParrinelloRahman` – to check!) Anyway, as a foolproof measure first a smooth fixed-cell reading is attempted from the .MD file, then, if it fails, a variable-cell reading, and in case the latter is successful, a warning is issued but the reading proceeds. If both attempts fail, most probably the .MD file is corrupt or incompatible (written with a different word length than attempted to read).

An option to consider in case of repeated problems is to change (by hand) the switch `format = T` in the subroutine `iomd.f` of `SIESTA`, recompile, and make corresponding modifications for formatted read from the .MD file in the `read_md.f` subroutine of the present suite.

In contrast to a binary .MD file, other possible sources of MD data, the .MD\_CAR or .ANI files, are readable ASCII files, the content of which can be inspected in case of trouble. The problem is, they do not contain the velocities, but only the atomic coordinates at each MD step. To cope with this, the velocities are extracted from “inverting” the Verlet algorithm:

$$\mathbf{v}(t) = \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t - \Delta t)}{2\Delta t}.$$

To this end, the coordinates of all atoms are remembered for two MD steps back; the first two MD steps do only serve to store the coordinates, without yielding any velocity.

An at least hypothetical problem when dealing with coordinates of a periodic system is that the atoms may be propagated across the cell boundary and acquire a jump in coordinate by nearly a translation vector, thus contaminating the velocity. The case immediately prone to such danger seems to be that of a .MD\_CAR file, in which the atom coordinates are stored as relative ones, see `md_out.F90`:

```
call reclat(cell,celli,0)
  do i=1, na
    frac(:) = matmul(transpose(celli),xa(:,i))
    write(iomd,"(3f16.9)") frac(:)
  enddo
```

Therefore some heuristics is provided which accepts the coordinate steps within  $\pm 0.4$  of the translation vector and silently translates those  $> 0.6$  or  $< -0.6$ . The suspicious jumps of  $\pm[0.4 : 0.6]$  are reported into a specially created `MD_CAR.LOG` file.

More tests are needed to check this heuristics (and whether it is necessary at all). No correction is provided when extracting the coordinates from the .ANI file which contains the cartesian coordinates in Å, without the unit cell information. (However, since the lattice vectors are available from the .XV files, the check of possible translations can be implemented if necessary).

Two issues may deserve a special discussion for an unexperienced user, (i) a choice of the `MD.LengthTimeStep =  $\Delta t$`  (default value: 1 fs) and (ii) checking that a MD simulation started well and makes sense, before too many calculations are done for nothing.  $\Delta t$  must be small enough to assure a good sampling (say  $\sim 10$  steps over the period) of the fastest vibration expected in the

system. The latter can be reasonably guessed from participating chemical elements and their chemical bondings (e.g., consulting the vibration spectra of similar systems). For instance, the fastest vibrations in a water molecule are at  $\approx 3760 \text{ cm}^{-1}$ , hence

#### Frequency units conversion

1 THz = 4.136 meV = 33.356 $\text{cm}^{-1}$ ;
1 meV = 0.242 THz = 8.066 $\text{cm}^{-1}$ ;
1 $\text{cm}^{-1}$ = 0.030 THz = 0.124 meV.

$\nu=113$  THz  $\rightarrow T = 1/\nu = 8.9$  fs, so that the time step  $\Delta T=1$  fs is a very reasonable one. Note that this is one of fastest vibrations around; for a large number of cases the default choice of  $\Delta t$  would be desperately overconservative.

Take an example of metal aluminium whose fastest vibration is at  $\approx 10$  THz; this value can be taken as a reasonable guess for maximal  $\nu$  in Al alloys with heavier elements. The period of this fastest vibration is 100 fs, meaning that 100 steps in a MD simulation would be spent on sampling just a single ondulation in this system (and this the fastest vibration!), certainly a waste of calculation time. The value  $\Delta t=10$  fs (or say 7 fs, to be on a safer side) would be quite OK in this case.

A MD simulation is supposed to get atoms moving; the first thing to check after some (tens of) steps done is to ensure that this is indeed the case. A XCrySDen visualization by md2axsf from the Sies2xsf suite may give a clue, otherwise just tracing coordinates of a given atom (by hand if lack of better ideas) throughout a sequence of steps. Reasons why atoms would stay almost immobile may be numerous: an input error resulting in too large interatomic distances, the system happy at equilibrium without an additional kick, a too short time step (as discussed above) so that nothing visible happens from one step to the other.

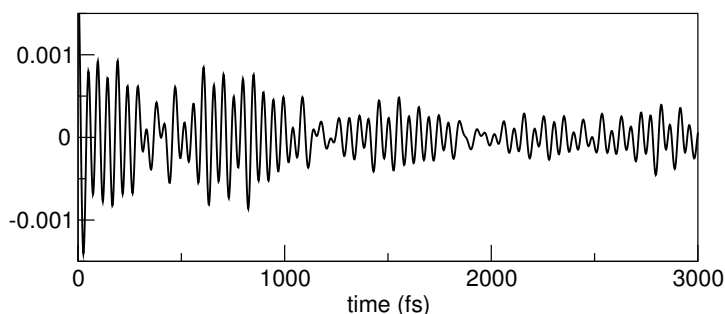
Once the `velcf` script invoked (on the command line, without parameters), it asks some questions, to be answered on the terminal prompt. Most of them are obvious; less obvious are, e.g.:

```
Cleanly read in      8623    MD steps
You may wish to retain only part of this MD history for subsequent analysis.
Specify two numbers MDfirst, MDlast - or 0 for any of them as default :
```

Here, typing "0 0 `<Enter>`" retains all MD steps; otherwise a shorter history can be selected.

```
You may wish to limit the maximal correlation length
to be calculated. This can make sense in very long runs,
where the correlations fall down much faster. Give the max. number of steps
to cutoff correlations (0 for default --> no cutoff) :
```

Here again, typing 0 `<Enter>` makes no reduction and calculates the correlations over the maximal possible number of time steps. In a long enough calculation, the correlations may fall done to essentially zero at much shorter time scale. The number of steps thus retained will give the argument span of the VACF in the time domain, written into the `.VCT` file. An example of time-resolved VACF is given on the right; it must look as a superposition of vibrations at several distinct frequencies:



From thus obtained  $C_v(\tau)$ ,  $I(\omega)$  calculated along Eq. (2) is written into the 6th column of the `.VCF` file along with some other functions indicated in the #-commented top lines of this file: frequency in  $\text{cm}^{-1}$  and THz; sin- and cos-Fourier transforms of the VACF).

