

Correlation functions

Theory of correlation functions

The theory of correlation functions is well established [108](#). We describe here the implementation of the various correlation function flavors in the GROMACS code. The definition of the autocorrelation function (ACF) $C_f(t)$ for a property $f(t)$ is:

$$C_f(t) = \langle f(\xi)f(\xi + t) \rangle_\xi \quad (438)$$

where the notation on the right hand side indicates averaging over ξ , *i.e.* over time origins. It is also possible to compute cross-correlation function from two properties $f(t)$ and $g(t)$:

$$C_{fg}(t) = \langle f(\xi)g(\xi + t) \rangle_\xi \quad (439)$$

however, in GROMACS there is no standard mechanism to do this (**note:** you can use the `xmgr` program to compute cross correlations). The integral of the correlation function over time is the correlation time τ_f :

$$\tau_f = \int_0^\infty C_f(t)dt \quad (440)$$

In practice, correlation functions are calculated based on data points with discrete time intervals Δt , so that the ACF from an MD simulation is:

$$C_f(j\Delta t) = \frac{1}{N-j} \sum_{i=0}^{N-1-j} f(i\Delta t)f((i+j)\Delta t) \quad (441)$$

where N is the number of available time frames for the calculation. The resulting ACF is obviously only available at time points with the same interval Δt . Since, for many applications, it is necessary to know the short time behavior of the ACF (*e.g.* the first 10 ps) this often means that we have to save the data with intervals much shorter than the time scale of interest. Another implication of [\(441\)](#) is that in principle we can not compute all points of the ACF with the same accuracy, since we have $N - 1$ data points for $C_f(\Delta t)$ but only 1 for $C_f((N - 1)\Delta t)$. However, if we decide to compute only an ACF of length $M\Delta t$, where $M \leq N/2$ we can compute all points with the same statistical accuracy:

$$C_f(j\Delta t) = \frac{1}{M} \sum_{i=0}^{N-1-M} f(i\Delta t) f((i+j)\Delta t) \quad (442)$$

Here of course $j < M$. M is sometimes referred to as the time lag of the correlation function. When we decide to do this, we intentionally do not use all the available points for very short time intervals ($j \ll M$), but it makes it easier to interpret the results. Another aspect that may not be neglected when computing ACFs from simulation is that usually the time origins ξ ([\(438\)](#)) are not statistically independent, which may introduce a bias in the results. This can be tested using a block-averaging procedure, where only time origins with a spacing at least the length of the time lag are included, *e.g.* using k time origins with spacing of $M\Delta t$ (where $kM \leq N$):

$$C_f(j\Delta t) = \frac{1}{k} \sum_{i=0}^{k-1} f(iM\Delta t) f((iM+j)\Delta t) \quad (443)$$

However, one needs very long simulations to get good accuracy this way, because there are many fewer points that contribute to the ACF.

Using FFT for computation of the ACF

The computational cost for calculating an ACF according to [\(441\)](#) is proportional to N^2 , which is considerable. However, this can be improved by using fast Fourier transforms to do the convolution [108](#).

Special forms of the ACF

There are some important varieties on the ACF, *e.g.* the ACF of a vector \mathbf{p} :

$$C_{\mathbf{p}}(t) = \int_0^\infty P_n(\cos \angle(\mathbf{p}(\xi), \mathbf{p}(\xi+t))) d\xi \quad (444)$$

where $P_n(x)$ is the n^{th} order Legendre polynomial. [\[1\]](#) Such correlation times can actually be obtained experimentally using *e.g.* NMR or other relaxation experiments. GROMACS can compute correlations using the 1st and 2nd order Legendre polynomial ([\(444\)](#)). This can also be used for rotational autocorrelation ([gmx rotacf](#)) and dipole autocorrelation ([gmx dipoles](#)).

In order to study torsion angle dynamics, we define a dihedral autocorrelation function as [159](#):

$$C(t) = \langle \cos(\theta(\tau) - \theta(\tau + t)) \rangle_\tau \quad (445)$$

Note that this is not a product of two functions as is generally used for correlation functions, but it may be rewritten as the sum of two products:

$$C(t) = \langle \cos(\theta(\tau)) \cos(\theta(\tau + t)) + \sin(\theta(\tau)) \sin(\theta(\tau + t)) \rangle_\tau \quad (446)$$

Some Applications

The program [gmx velacc](#) calculates the *velocity autocorrelation function*.

$$C_v(\tau) = \langle \mathbf{v}_i(\tau) \cdot \mathbf{v}_i(0) \rangle_{i \in A} \quad (447)$$

The self diffusion coefficient can be calculated using the Green-Kubo relation [108](#):

$$D_A = \frac{1}{3} \int_0^\infty \langle \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \rangle_{i \in A} dt \quad (448)$$

which is just the integral of the velocity autocorrelation function. There is a widely-held belief that the velocity ACF converges faster than the mean square displacement (sec. [Mean Square Displacement](#)), which can also be used for the computation of diffusion constants. However, Allen & Tildesley [108](#) warn us that the long-time contribution to the velocity ACF can not be ignored, so care must be taken.

Another important quantity is the dipole correlation time. The *dipole correlation function* for particles of type A is calculated as follows by [gmx dipoles](#):

$$C_\mu(\tau) = \langle \mu_i(\tau) \cdot \mu_i(0) \rangle_{i \in A} \quad (449)$$

with $\mu_i = \sum_{j \in i} \mathbf{r}_j q_j$. The dipole correlation time can be computed using [\(440\)](#). For some applications see [\(???\)](#).

The viscosity of a liquid can be related to the correlation time of the Pressure tensor \mathbf{P} [160](#), [161](#). [gmx energy](#) can compute the viscosity, but this is not very accurate [149](#), and actually the values do not converge.

[1]

$$P_0(x) = 1, P_1(x) = x, P_2(x) = (3x^2 - 1)/2$$

Copyright © 2023, GROMACS development team
Made with [Sphinx](#) and @pradyunsg's [Furo](#)