Computing memory functions from Molecular Dynamics simulations

G.R. Kneller and K. Hinsen

Centre de Biophysique Moléculaire, CNRS UPR 4301
Rue Charles Sadron, F-45071 Orléans Cedex 2, France.

The Journal of Chemical Physics, in press

Abstract

We propose a new method to compute reliable estimates for memory functions of dynamical variables from Molecular Dynamics simulations. The key point is that the dynamical variable under consideration, which we take to be the velocity of a fluid particle, is modeled as an autoregressive stochastic process. The parameters of this stochastic process can be determined from Molecular Dynamics trajectories using efficient algorithms that are well established in signal processing. The procedure is also referred to as maximum entropy method. From the autoregressive model of the velocity autocorrelation function we compute the one-sided z-transform of the discretized memory function and the memory function itself. Using liquid argon as a simple model system, we demonstrate that the autocorrelation function and its power spectrum can be approximated to almost arbitrary precision. The same is therefore true for the memory function, which is calculated within the same stochastic model.
I. INTRODUCTION

The concept of memory function equations for autocorrelation functions, as introduced by Zwanzig and Mori [1,2], has proven to be useful to describe dynamical correlations in liquids from a theoretical point of view [3,4]. Dynamical correlations are described by corresponding time correlation functions which can be obtained from spectroscopic experiments and Molecular Dynamics simulations. In his pioneering simulation work on liquid argon, Rahman has demonstrated that the velocity autocorrelation function (VACF) does not decay exponentially, which would follow from a memoryless Markovian model, but has much in common with the VACF of a damped harmonic oscillator [5]. Berne et al. have shown that such a behavior can be obtained, at least qualitatively, by assuming that the memory function instead of the VACF is an exponential [6]. Later work in the same direction, but with different empirical models for the memory function, has been performed to achieve better agreement with results obtained by computer simulation (see e.g. [7–10]) and experiment [10,11].

The abovementioned work concerns simple liquids, but the concept of memory functions is also attractive in the context of simulations of complex molecular systems. It allows to construct generalized Langevin equations for an observable dynamical process, like a chemical reaction or a proton transfer, in which the memory function represents a background medium. The latter may be a liquid or even a protein which can be modeled in some implicit but still realistic way. Even with modern computers it would be hopeless to model such processes in full detail, including a realistic environment.

A possible way to extract memory functions from computer simulations is to work with the Laplace-transformed memory function equation. Here the crucial point is to perform an inverse Laplace transform to obtain the memory function in the time domain. It has been pointed out by Egelstaff [11] that this approach, though formally correct, is errorful from a numerical point of view. Vogelsang and Hoheisel suggested to compute memory functions from simulation data by transforming the defining integral equation into a Volterra-type inte-
gral equation and solving the latter by numerical integration [12]. Other authors developed statistical mechanical models for memory functions and computed the latter numerically from these models [13,14].

In this paper we show how reliable estimates for memory functions can be obtained by assuming a model for the underlying stochastic process and not for the memory function itself. We fit an autoregressive stochastic (AR) model to the simulation data and use the latter in the one-sided z-transform of the memory function equation. Applying the inverse z-transform, the memory function can then be transformed into the time domain. The z-transform is particularly adapted to treat discrete signals, such as they are produced by Molecular Dynamics simulations. The AR model is a standard tool in spectral estimation, and the adaptation of its coefficients to a given signal is also referred to as the maximum entropy method [15]. We show that the validity of the memory function can be systematically extended to longer times, keeping the functional form for shorter times.

The paper is organized as follows: In Section II we present briefly the AR model for stochastic processes and its application to the computation of velocity autocorrelation functions and corresponding power spectra. Section III contains the central part of the paper, namely the computation of memory functions based on the AR model and the computation of corresponding transport coefficients. In Section IV we show an application to liquid argon, comparing to earlier work. The concluding remarks are presented in Section V.

II. AUTOREGRESSIVE MODEL FOR THE VACF

A. Preliminaries

In this section we give some basic definitions and relations concerning the AR model for stochastic processes. For a comprehensive introduction the reader should consult the standard literature on signal processing (see for example [15]) or the tutorial review by Makhoul [16].
The AR model of order \( P \) for a signal \( x(t) \) is defined by the relation
\[
x(t) = \sum_{n=1}^{P} a_n^{(P)} x(t - n\Delta t) + \epsilon_P(t), \tag{1}
\]
where \( \Delta t \) is the sampling step, \( a_n^{(P)} \) are real coefficients yet to determined, and \( \epsilon_P(t) \) is a Gaussian white noise process with zero mean,
\[
\langle \epsilon_P(t) \rangle = 0, \tag{2}
\]
\[
\langle \epsilon_P(t)\epsilon_P(t') \rangle = \sigma_P^2 \delta(t - t'). \tag{3}
\]
For \( P = 1 \), the time evolution of \( x(t) \) is modeled as a Markov process.

Multiplying eq. (1) by \( x(t - k\Delta t) \) and taking an ensemble average yields the Wiener-Hopf equations for the coefficients \( a_n^{(P)} \):
\[
\sum_{n=1}^{P} r(|k - n|\Delta t)a_n^{(P)} = r(k\Delta t), \quad k = 1 \ldots P. \tag{4}
\]
Here \( r(t) \) is the autocorrelation function of \( x(t) \),
\[
r(t) := \langle x(t)x(0) \rangle. \tag{5}
\]
It is assumed that \( x(t) \) is a stationary stochastic process, such that \( \langle x(t_1)x(t_2) \rangle \equiv r(t_1, t_2) = r(t_1 - t_2, 0) = r(t_2 - t_1, 0) \equiv r(|t_2 - t_1|) \). Since \( \epsilon_P(t) \) is supposed to be white noise, it is not correlated with preceding values of \( x \), and therefore \( \langle \epsilon_P(t)x(t - k\Delta t) \rangle = 0 \) for \( k = 1 \ldots P \).

Multiplication of (1) with \( x(t) \) and averaging fixes the variance of the noise,
\[
\sigma_P^2 = r(0) - \sum_{n=1}^{P} a_n^{(P)} r(n\Delta t). \tag{6}
\]
In the context of statistical physics, relation (6) can be considered as a fluctuation-dissipation theorem since it relates the amplitude \( \sigma_P \) of the fluctuations to the (decaying) autocorrelation function.

The model (1) is closely related to linear prediction of time series. In this case one approximates \( x(t) \) by a linear estimator of the form
\[
\hat{x}(t) = \sum_{n=1}^{P} a_n^{(P)} x(t - n\Delta t). \tag{7}
\]
P is then called the prediction order, and the coefficients $a_n^{(P)}$ are the corresponding predictor coefficients. *A priori* they are not related to the coefficients in the AR model (1). Using the definition of the local prediction error,

$$
\epsilon_P(t) = x(t) - \sum_{n=1}^{P} a_n^{(P)} x(t - n\Delta t),
$$

the optimal predictor coefficients are found by requiring that the mean square error is minimal,

$$
E^2 := \langle [\epsilon_P(t)]^2 \rangle = \min \{ a_n^{(P)} \}.
$$

The brackets in (9) denote again an ensemble average. A necessary condition for (9) to hold is that all derivatives of $E^2$ with respect to the predictor coefficients vanish. This yields the following system of linear equations for the predictor coefficients,

$$
0 = \left\langle \left( x(t) - \sum_{n=1}^{P} a_n^{(P)} x(t - n\Delta t) \right) x(t - k\Delta t) \right\rangle, \quad k = 1 \ldots P.
$$

Expressing the ensemble averages in terms of the autocorrelation function of $x(t)$ shows that the optimal predictor coefficients are the solution of the Wiener-Hopf equations (4). In the context of linear prediction the latter are also called the *Yule-Walker* equations.

To solve the Wiener-Hopf (Yule-Walker) equations one can use the recursive Levinson-Durban algorithm [15–18] which makes use of the fact that the matrix $r_{nk} \equiv r(|n - k|\Delta t)$ has Toeplitz structure. In numerical applications, where $x(t)$ is available as a time series of finite length, its autocorrelation function must be estimated from the available data. Depending on which estimator is used for $r(t)$, different results are to be expected for the coefficients $a_n^{(P)}$. For our application we use the Burg algorithm [19,20] which estimates the autocorrelation function implicitly and yields the predictor coefficients directly from a discrete signal of finite length. Besides being as efficient as the Levinson-Durban algorithm, it has the advantage of yielding *stable* predictors. We will come back later to this point.

In the following we will make extensive use of the one- and two-sided z-transform of discrete functions. The latter is defined by

5
\[
F(z) = \sum_{n=-\infty}^{+\infty} f(n) z^{-n}
\]

for an arbitrary discrete function \( f(n) \) (\( n \) integer). The inverse transform can be written as

\[
f(n) = \frac{1}{2\pi i} \oint_C dz \, z^{n-1} F(z),
\]

where the integration contour is to be taken within the region of convergence of \( F(z) \). In the region of convergence \( F(z) \) can be expanded in a Laurent series, and \( f(n) \) is just the coefficient corresponding to the index \(-n\). The z-transform (11) is to be distinguished from the one-sided z-transform, which can be considered as the equivalent of the Laplace transform for discrete functions,

\[
F_>(z) = \sum_{n=0}^{\infty} f(n) z^{-n}.
\]

As for Laplace transforms, one assumes that the function to be transformed is zero for negative time arguments.

In the AR model the (two-sided) z-transform of the discrete autocorrelation function \( r(n) \) has a simple “all-pole” form

\[
R^{(AR)}(z) = \frac{\sigma_p^2}{(1 - \sum_{k=1}^{P} a_k^{(P)} z^{-k})(1 - \sum_{l=1}^{P} a_l^{(P)} z^{l})}.
\]

Setting \( z = \exp[i\omega \Delta t] \) in \( R^{(AR)}(z) \) yields the power spectrum of \( x(n) \),

\[
P^{(AR)}(\omega) = R^{(AR)}(\exp[i\omega \Delta t]) = \sum_{n=-\infty}^{+\infty} r^{(AR)}(n) \exp[-in\omega \Delta t].
\]

Note that \( P^{(AR)}(\omega) \) is the discrete Fourier transform of the autocorrelation function \( r^{(AR)}(n) \).

**B. Velocity autocorrelation function**

In the following we consider the velocities of \( N \) identical particles in a simple liquid to be the dynamical quantities of interest. The information available from Molecular Dynamics (MD) simulations is a discrete trajectory of all particle velocities \( v_j(n) \equiv v_j(n \Delta t) \), where \( j \) runs over all particles and Cartesian components, \( j = 1, \ldots, 3N \), and \( n \) runs over all time
frames in the trajectory, \( n = 0, \ldots, N_t - 1 \). The autocorrelation function \( r(t) \) introduced in the previous section is here the normalized velocity autocorrelation function (VACF)

\[
\psi(t) := \frac{\langle v(t)v(0) \rangle}{\langle v^2(0) \rangle}. \tag{16}
\]

The velocity \( v(t) \) represents one of the three Cartesian velocity components of a tagged particle. From (14) we obtain

\[
\Psi^{(AR)}(z) = \frac{1}{P} \frac{-z^P \sigma_P^2}{(1 - \sum_{k=1}^P a_k^{(P)} z^{-k})(1 - \sum_{l=1}^P a_l^{(P)} z^l)}. \tag{17}
\]

According to eq. (6) \( \sigma_P^2 \) is given by

\[
\sigma_P^2 = 1 - \sum_{n=1}^P a_n^{(P)} \psi(n \Delta t), \tag{18}
\]

using that \( \psi(0) = 1 \).

In order to cast the denominator of \( \Psi^{(AR)}(z) \) into polynomial form we write

\[
\Psi^{(AR)}(z) = \frac{1}{a_P^{(P)}} \left( \frac{-z^P \sigma_P^2}{p(z)} \right) \left( \frac{1}{q(z)} \right), \tag{19}
\]

Here \( p(z) \) and \( q(z) \) are polynomials of degree \( P \) whose coefficients for \( z^P \) are equal to 1. If \( z_k \) (\( k = 1, \ldots, P \)) are the zeros of \( p(z) \), those of \( q(z) \) are simply \( z_l^{-1} \) (\( l = 1, \ldots, P \)), and we may write

\[
\Psi^{(AR)}(z) = \frac{1}{a_P^{(P)}} \left( \frac{-z^P \sigma_P^2}{\prod_{k=1}^P (z - z_k) \prod_{l=1}^P (z - z_l^{-1})} \right), \tag{20}
\]

assuming that \( p(z) \) has only simple zeros. We found that this assumption was always fulfilled in practice, but it is not necessary. In Appendix B we show that multiple zeros do not pose any problem in principle for the calculation of the VACF and the memory function. The resulting expressions are, however, less convenient for numerical evaluation.

The zeros of \( p(z) \) determine essential physical properties of the AR process, such as stability and causality. An AR process is stable and causal if all zeros of \( p(z) \) are located inside the unit circle [21],

7
\[ |z_k| < 1, \quad k = 1, \ldots, P. \]  

The above condition is fulfilled if the coefficients \( a_n^{(P)} \) are computed from the Burg algorithm [19,20]. The location of the zeros of \( p(z) \) also determines the domain of definition of \( \Psi^{(AR)}(z) \). As shown in Appendix A, \( \Psi^{(AR)}(z) \) is defined in the ring

\[ |z_k|_{\text{max}} < |z| < \frac{1}{|z_k|_{\text{max}}}. \]  

Applying the calculus of residues one obtains from (12) and (20)

\[
\psi^{(AR)}(n) = \sum_{j=1}^{P} \frac{1}{a_{P}^{(P)}} \frac{-z^{|n|+P-1} \sigma^2_{P}}{\prod_{k=1,k \neq j}^{P} (z_j - z_k) \prod_{l=1}^{P} \left( z_j - z - 1 \right)}.
\]  

The integration contour for the inverse \( z \)-transform is to be taken in the domain defined by eq. (22). In particular, one can always take the unit circle. Note that \( \psi^{(AR)}(n) \) is symmetric and decays to zero as \( |n| \) tends to infinity.

We remark that the zeros of \( p(z) \) can be reliably computed up to high orders in \( P \) by solving an eigenvalue problem for a corresponding Frobenius matrix [22]. The latter has the dimensions \( P \times P \) and contains the coefficients of \( p(z) \). It is constructed such that its characteristic polynomial is just \( p(z) \),

\[ \det(zI - \mathbf{P}) = p(z). \]  

One possible form for the Frobenius matrix \( \mathbf{P} \) is

\[
\mathbf{P} = \begin{pmatrix}
0 & 0 & \ldots & 0 & a_{P}^{(P)} \\
1 & 0 & \ldots & 0 & a_{P-1}^{(P)} \\
0 & 1 & \ldots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & a_1
\end{pmatrix}.
\]  

Finding the zeros of \( p(z) \) then amounts to finding the eigenvalues of \( \mathbf{P} \). The latter task can be performed by using standard linear algebra packages. We found that the zeros can be reliably computed up to at least \( P = 1000 \).
III. COMPUTING THE MEMORY FUNCTION

A. Discretized memory function equation and z-transform

The defining equation for the memory function reads [23]

\[
\frac{d}{dt} \psi(t) = - \int_0^t d\tau \xi(t - \tau) \psi(\tau),
\]

where \(\psi(\tau)\) is the normalized autocorrelation function given by eq. (16) and \(\xi(t - \tau)\) is the memory function for time-shifted arguments.

The first step towards a numerical computation of the memory function consists in discretizing eq. (26),

\[
\psi(n + 1) - \psi(n) = - \Delta t \sum_{k=0}^{n-1} \xi(n - k) \psi(k).
\]

Eq. (27) is now subjected to a one-sided z-transform. Using that

\[
Z_\uparrow \{ f(n + 1) - f(n) \} = z F_\uparrow(z) - zf(0)
\]

for any discrete function \(f(n)\) whose one-sided z-transform exists, one obtains from (27)

\[
\Xi_\uparrow(z) = \frac{1}{\Delta t^2} \left( \frac{z}{\Psi_\uparrow(z)} + 1 - z \right),
\]

using that \(\psi(0) = 1\). Here it has been used that the one-sided z-transform of the discrete convolution integral is just the product \(\Xi_\uparrow(z)\Psi_\uparrow(z)\).

A remark concerning the discretization scheme (27) is in place here. The discrete convolution sum is effectively a first order approximation of the convolution integral. More sophisticated approximations could be used, but they would lead to less convenient expressions upon z-transformation. Correspondingly, we have chosen a first order approximation for the differentiation on the left-hand side of (26). In this way the first order (integro-) differential equation (26) is transformed into the first order difference equation (27).
B. Memory function for the autoregressive model

To compute $\Psi_>(z)$ we use the AR model (23) for $\psi^{(AR)}(n)$ and write

$$\psi^{(AR)}(n) = \sum_{j=1}^{P} \beta_j z_j^n,$$

where the coefficients $\beta_j$ read

$$\beta_j = \frac{1}{a_P} \frac{-z_j^{P-1} \sigma_P^2}{\prod_{k=1, k \neq j}^{P} (z_j - z_k) \prod_{i=1}^{P} (z_j - z_i^{-1})}.$$  \hspace{1cm} (31)

By definition, $\Psi^{(AR)}_>(z)$ is then given by

$$\Psi^{(AR)}_>(z) = \sum_{n=0}^{\infty} \psi^{(AR)}(n) z^{-n} = \sum_{j=1}^{P} \beta_j \sum_{n=0}^{\infty} \left( \frac{z_j}{z} \right)^n.$$  \hspace{1cm} (32)

If $|z_j|_{\text{max}} < |z|$ the geometrical series converge for each $z_j$, and one obtains

$$\Psi^{(AR)}_>(z) = \sum_{j=1}^{P} \beta_j \frac{z}{z - z_j}, \quad |z| > |z_j|_{\text{max}}.$$  \hspace{1cm} (32)

Inserting now expression (32) into eq. (29) yields a rational function in $z$ for $\Xi_>(z)$:

$$\Xi_>(z) = \frac{1}{\Delta t^2} \left( \frac{1}{\sum_{j=1}^{P} \frac{1}{\beta_j z_j}} + 1 - z \right).$$  \hspace{1cm} (33)

$\Xi_>(z)$ is a rational function in $z$ whose numerator and denominator polynomial coefficients can easily be constructed algorithmically from the numerical values of the $\beta_j$. A straightforward method to obtain the discrete time series $\xi(n)$ from this rational function is to use polynomial division. In order to compute $\xi(n)$ for $0 \leq n \leq N$, $z^N \Xi_>(z)$ is divided out, yielding a polynomial in $z$ and a remainder in which we are not interested. The polynomial is then divided by $z^N$, and comparison with the definition of the z-transform shows that its coefficients are just the values of the time series $\xi(n)$. This method is applicable only to causal functions whose z-transforms are power series in $z^{-1}$.

The only difficulty with this inversion method is a progressive loss of precision with increasing $n$, which makes the inversion unstable for large $P$ and $N$. Unfortunately this fundamental problem is shared by other inversion methods. An approach based on residue
calculus similar to Eq. (23) turned out to be even less stable, as did an attempt to obtain $\xi(n)$ from a linear system of equations involving the polynomial coefficients of $\Xi_>(z)$, even using a stable linear-equation solver based on Singular Value Decomposition. Therefore we resorted to performing all calculations of memory functions in the time domain for $P > 85$ with high-precision floating-point numbers having a 150-bit mantissa.

C. Transport coefficients

The transport coefficient associated with the memory function equation (26) is the friction constant. It is defined as

$$\xi_0 = \int_0^\infty d\tau \xi(\tau).$$

(34)

The corresponding expression for the discretized memory function is then

$$\xi_0 = \sum_{n=0}^{\infty} \Delta t \xi(n) = \Delta t \Xi_>(1).$$

(35)

Inserting expression (33) for $\Xi_>(z)$ into the rhs. of eq. (35) yields

$$\xi_0 = \frac{1}{\Delta t} \sum_{j=1}^{P} \frac{1}{\beta_j \frac{1}{1-z_j}}.$$  

(36)

The coefficients $\beta_j$ are given by eq. (31). Expression (36) shows that the knowledge of the poles $z_j$ of $\Psi^{(AR)}(z)$ suffices to compute $\xi_0$. In contrast to the computation of memory functions, normal double precision arithmetic can be used to compute $\xi_0$ even for orders $P > 85$.

IV. APPLICATION TO A SIMPLE MODEL SYSTEM

To demonstrate the effectiveness of our method for numerical computation of memory functions from Molecular dynamics simulations, we have chosen liquid argon as a simple model system. For obvious reasons the theoretical work on memory functions has been restricted to simple liquids, referring often to computer simulations by Rahman of liquid...
argon [5] and rubidium [24] to validate the various models. We have simulated a system of \( N = 864 \) argon molecules in the liquid state at a temperature of 94.4 K and a pressure of 1 atm, using a time step of 10 fs. The simulation was performed in the thermodynamic \( NpT \) ensemble (average constant pressure and temperature), using the extended system method developed by Anderson and Nosé [25,26]. The total length of the simulation is 100 ps, and the velocities of all particles were stored at all 10,000 time frames. The simulation was performed using the Molecular Modeling Toolkit [27].

For each of the \( 3N \) velocity trajectories we computed a set of \( P \) coefficients \( a_n(P) \) (\( n = 1 \ldots P \)) using the Burg algorithm [19,20] with all available time frames, i.e. with \( \Delta t \) corresponding to the simulation step. The coefficients \( a_n(P) \) were then averaged over the \( 3N \) coordinates, and the averaged coefficients were used to calculate the VACF, its power spectrum or density of states (DOS), and the memory function. All these calculations were performed for orders \( P \in \{20, 50, 85, 150, 250\} \), using Python scripts that make heavy use of Numerical Python [28] and Scientific Python [29]. The high-precision calculations for obtaining the memory functions for \( P > 85 \) were based on the GMP library [30] and a Python interface for GMP [31].

To evaluate the quality of the AR model, we also computed the VACF by direct calculation for each velocity trajectory, applying the Fast Fourier Transform method with appropriate zero padding [32,33]. This VACF was then averaged over all coordinates. The corresponding power spectrum was computed by discrete Fourier transform. These calculations were performed with a prerelease version of the analysis program nMOLDYN [34].

Fig. 1 shows the velocity autocorrelation function for different orders of the AR model, as compared to the direct calculation from the MD trajectory. Fig. 2 shows the corresponding power spectra. As one would expect, the behaviour of the VACF is progressively better modeled with increasing order \( P \). We recall that \( P = 250 \) was the maximum order for which the VACFs, their power spectra (DOS) and the memory functions have been calculated.

To evaluate the influence of coarse-grained sampling we computed the VACF in the AR model using a sampling interval of five simulation steps, i.e. \( \Delta t = 50 \) fs. This approach
should allow to obtain a better representation of the long time correlations, reducing the
precision in modeling the short time correlations. Fig. 3 shows the results for \( P = 20 \) and
\( P = 50 \), and Fig. 4 the corresponding power spectra. Clearly, the agreement with the VACF
obtained by direct calculation is now extended to longer times, however not by a factor of
5. By construction, the short time behaviour is now less well sampled.

Based on the above fits of an AR model for the particle velocities in liquid argon, we
computed the corresponding memory functions, which are shown in Fig. 5. Two important
features can be seen:

1. The memory function of order \( P \) falls rapidly off to zero for \( t > P\Delta t \), where \( \Delta t \) is the
sampling step. This is, of course, a direct consequence of the ansatz for the AR model.

2. Two memory functions of different order, say \( P \) and \( P' \), are practically identical for
\( t < \min(P, P')\Delta t \), i.e. only the long time behavior changes when passing to a higher
order.

Fig. 6 shows the memory function based on the AR model compared to the exponential
model proposed by Berne et al. [6]. The corresponding VACFs are given in Fig. 7. The
exponential form of the memory function leads to a VACF which captures the oscillatory
behavior at least qualitatively. In this context it is illustrative to look at the memory
functions which some authors [35,8,36] derived numerically from Rahman’s VACF for liquid
argon [5], in order to compare them to the results from their models. Since no details of
the computation are given, one could think that the computation is straightforward and not
worth mentioning computational details. Looking at Fig. 8 shows clearly that this is not
the case. The different numerically determined memory functions exhibit even qualitative
differences. As in Fig. 6, we show for comparison the memory function obtained from an
AR model for \( P = 250 \).

We have also computed the friction constants \( \xi_0 \) corresponding to the memory functions
shown in Fig. 5. In addition we performed one calculation of \( \xi_0 \) only with \( P = 500 \), using
normal double precision calculation. Table 1 gives the values obtained from eq. (36), and,
for comparison, the corresponding values obtained by numerical integration of the memory functions. Expression (36) is to be preferred over explicit integration of the memory function, since the latter contains additional numerical errors due to the inverse z-transform and the subsequent numerical integration. For comparison we computed the friction constant also from the Stokes-Einstein relation

\[ \xi_0^{(D)} = \frac{k_B T}{m D}, \]  

(37)

where \( m \) is the mass of an argon atom and \( D \) its diffusion constant. The result is \( \xi_0^{(D)} = 7.135 \text{ ps}^{-1} \), which is close to \( \xi_0 \) obtained from the AR model for order \( P = 500 \).

Since the AR process has essentially a built-in memory of \( t = P \Delta t \), where \( \Delta t \) is the sampling interval, one expects that the poles come closer to the unit circle with increasing order \( P \). Note that each pole \( z_k \) gives a contribution proportional to \( |z_k|^n \), i.e. slow components in the signal corresponds to poles close the unit circle. Frequency information is contained in the phase of the poles. Fig. 9 shows the location of the poles \( z_k \) for the extreme cases \( P = 20 \) and \( P = 500 \). In the latter case many poles are very close to the unit circle, yielding slowly decaying contributions to the VACF. We remark that the convergence of the AR model with increasing order of \( P \) is defined by the relation [20]

\[ \langle \epsilon_P(t) \rangle = \langle \epsilon_{(P-1)}(t) \rangle \left( 1 - \left( a_P^{(P)} \right)^2 \right). \]  

(38)

Using that for a stable model the relation \( 0 \leq |a_P^{(P)}| < 1 \) holds, the mean square prediction error can be reduced by increasing the predictor order from \( P - 1 \) to \( P \) if \( a_P^{(P)} \neq 0 \). This means that there is still information in the signal that can be exploited to reduce the prediction error. Obviously no error reduction can be obtained if \( a_P^{(P)} = 0 \) beyond a certain value \( P = P_{\text{max}} \). If we consider for instance a Markov process, which is is described by an AR process of order \( P = 1 \), we have \( a_P^{(P)} = 0 \) for \( P > 1 \).

14
V. CONCLUSIONS AND DISCUSSION

We have shown that our method allows to compute reliable estimates of memory functions and corresponding transport coefficients from Molecular Dynamics simulations. The key point is to fit an autoregressive stochastic model to the dynamical quantity under consideration and to use this model in the z-transformed memory function equation. The computation of inverse z-transforms of VACFs can be achieved up to high orders in P by using the inversion formula and the theorem of residues. The computation of the VACF as well as of the friction constant within the AR model can be performed with standard double precision arithmetics. We encountered stability problems for $P > 85$ when computing the memory function from its one-sided z-transform. The inverse z-transform could be obtained by polynomial division using a package for calculations with arbitrary precision.

In contrast to models that have been used so far to describe the memory function, the AR model is not a model for the memory function itself, but for the underlying stochastic process. By increasing the prediction order $P$ the memory function can be systematically extended to longer times, without changing the short time behavior. By construction, the AR model of order $P$ considers memory effects up to time lags of $t = P \Delta t$. Clearly, algebraic long-time tails cannot be discussed within the AR model. For any finite order $P$ there is always a slowest component in the VACF which decays as $|z_k|^n$, i.e. exponentially. Nevertheless, reliable estimates for transport coefficients can be obtained if $P$ is sufficiently large. For our model system liquid argon $P = 500$ yields a friction constant which is slightly above the value obtained from the mean-square displacement and the Stokes-Einstein law, whereas it is systematically smaller for lower orders. Since the simulation time step is $\Delta t = 10 fs$, $P = 500$ corresponds to a time span of $5 ps$ for the memory function.
APPENDIX A: VACF FOR SIMPLE POLES

To find the domain of definition of $\Psi^{(AR)}(z)$ we start from eq (20) and perform formally the decomposition into partial fractions,

$$\Psi^{(AR)}(z) = \sum_{j=1}^{P} \left\{ \frac{A_j}{z - z_j} + \frac{B_j}{z - z_j^{-1}} \right\}.$$  \hspace{1cm} (A1)

We suppose that $|z_j| < 1$ ($j = 1, \ldots, P$), i.e. that the underlying AR process is stable and causal. The sum formula for the geometric series shows that $1/(z - z_j)$ can be expanded in a Laurent series if $|z| > |z_j|$, 

$$\frac{1}{z - z_j} = \frac{1}{z} \cdot \frac{1}{1 - (z_j/z)} = \frac{1}{z} \sum_{n=0}^{\infty} \left( \frac{z}{z_j} \right)^n,$$

and that $1/(z - z_j^{-1})$ can be expanded in a Taylor series for $|z| < |z_j|^{-1}$, 

$$\frac{1}{z - z_j^{-1}} = - \frac{1}{z_j^{-1}} \cdot \frac{1}{1 - (z/z_j^{-1})} = -z_j \sum_{n=0}^{\infty} (zz_j)^n.$$

Therefore $\Psi^{(AR)}(z)$ is defined in the domain

$$|z_j|_{\text{max}} < |z| < \frac{1}{|z_j|_{\text{max}}}. \hspace{1cm} (A2)$$

The above series expansions can be used to write $\Psi^{(AR)}(z)$ as a Laurent series in $z^{-n}$. According to definition (11) the coefficient corresponding to $z^{-n}$ is then just $\psi^{(AR)}(n)$. Another method to obtain $\psi^{(AR)}(n)$ is to perform the inverse z-transform

$$\psi^{(AR)}(n) = \frac{1}{2\pi i} \oint_C dz \, z^{n-1} \Psi^{(AR)}(z).$$  \hspace{1cm} (A3)

This method can be more easily generalized to the case where $\Psi^{(AR)}(z)$ has poles of higher order. The integration contour $C$ in (A3) must lie in the domain of definition. If $n \geq 1$ the function $z^{n-1} \Psi^{(AR)}(z)$ has only poles at $z = z_j$ and (A3) can be evaluated by applying the theorem of residues,

$$\psi^{(AR)}(n) = \sum_{j=1}^{P} \lim_{z \to z_j} (z - z_j) z^{n-1} \Psi^{(AR)}(z).$$  \hspace{1cm} (A4)
If \( n < 1 \), there is an additional pole of order \( |n| + 1 \) at \( z = 0 \), and in this case \( \psi^{(AR)}(n) \) is obtained by

\[
\psi^{(AR)}(n) = \sum_{j=1}^{P} \lim_{z \to z_j} (z - z_j)z^{n-1}\Psi^{(AR)}(z) + \lim_{z \to 0} \frac{1}{|n|!} \frac{d^{|n|}}{dz^{|n|}} \Psi^{(AR)}(z).
\]  

(A5)

Using the symmetry relation \( \psi^{(AR)}(n) = \psi^{(AR)}(-n) \) and that \( \Psi^{(AR)}(0) = 0 \) according to eq. (19), (A4) and (A5) may be combined into

\[
\psi^{(AR)}(n) = \begin{cases} 
\sum_{j=1}^{P} \lim_{z \to z_j} (z - z_j)z^{n-1}\Psi^{(AR)}(z), & n \geq 0, \\
\psi^{(AR)}(-n), & n < 0.
\end{cases}
\]  

(A6)

Using the form (20) for \( \Psi^{(AR)}(z) \) then yields expression (23) for \( \psi^{(AR)}(n) \).

**APPENDIX B: CONSEQUENCES OF MULTIPLE POLES**

We consider now the case where \( \Psi^{(AR)}(z) \) has \( N \) poles, \( z_1, \ldots, z_N \), each of which has a multiplicity \( m_j \), such that

\[
m_1 + m_2 + \ldots + m_N = P.
\]  

(B1)

The decomposition of \( \Psi^{(AR)}(z) \) into partial fractions then has the form

\[
\Psi^{(AR)}(z) = \sum_{j=1}^{N} \sum_{k=1}^{m_j} \left\{ \frac{A_{jk}}{(z - z_j)^{m_k}} + \frac{B_{jk}}{(z - z_j^{-1})^{m_k}} \right\}.
\]  

(B2)

This may be written in the alternative form

\[
\Psi^{(AR)}(z) = \sum_{j=1}^{N} \sum_{k=1}^{m_j} \frac{(-1)^{k-1}}{(k - 1)!} \frac{d^{k-1}}{dz^{k-1}} \left\{ \frac{A_{jk}}{z - z_j} + \frac{B_{jk}}{(z - z_j^{-1})} \right\}.
\]  

(B3)

As in the case of simple poles we assume that \( |z_j| < 1 \). Therefore each of the expressions in curly brackets is holomorphic for \( |z_j| < |z| < |z_j|^{-1} \), and the same is true for all derivatives.

The domain of definition for \( \Psi^{(AR)}(z) \) is thus the same as in the case of simple poles (see eq. (A2)). Consequently we can use the same integration contour to determine \( \psi^{(AR)}(n) \) by inverse \( z \)-transform. Applying the theorem of residues for poles of higher order yields
\[
\psi^{(AR)}(n) = \begin{cases} 
\sum_{j=1}^{N} \lim_{z \to z_j} \frac{1}{m_j - 1} \frac{d^{m_j - 1}}{dz^{m_j - 1}}(z - z_j)^{m_j} z^{n-1} \Psi^{(AR)}(z), & n \geq 0, \\
\psi^{(AR)}(-n), & n < 0.
\end{cases}
\]  
(B4)

An essential point is the calculation of the one-sided z-transform of \(\psi^{(AR)}(n)\) which can be generalized as follows: Formally the one-sided z-transform of the VACF can be written as

\[
\Psi^{(AR)}_>(z) = \sum_{n=-\infty}^{+\infty} \theta(n) \psi^{(AR)}(n) z^{-n},
\]  
(B5)

where \(\theta(n)\) is the discrete Heaviside function,

\[
\theta(n) = \begin{cases} 
1 & \text{if } n \geq 0, \\
0 & \text{if } n < 0.
\end{cases}
\]

One may now use that the z-transform of the product of two functions \(f(n)\) and \(g(n)\) is given by

\[
\sum_{n=-\infty}^{+\infty} f(n)g(n)z^{-n} = \frac{1}{2\pi i} \oint_C d\zeta \zeta^{-1} F(z/\zeta)G(\zeta).
\]

If we set \(f(n) = \theta(n)\) and \(g(n) = \psi^{(AR)}(n)\) we obtain

\[
\Psi^{(AR)}_>(z) = \frac{1}{2\pi i} \oint_C d\zeta \frac{1}{\zeta} \cdot \frac{z}{z - \zeta} \Psi^{(AR)}(\zeta), \quad |\zeta| < |z|.
\]  
(B6)

Here it has been used that \(\Theta(z) = z/(z - 1)\), where \(z\) must be taken outside the unit circle, i.e. \(|z| > 1\). Since \(\Psi^{(AR)}(\zeta)\) is defined in the ring \(|z_j|_{max} < |\zeta| < |z_j|_{max}^{-1}\), and in addition \(|\zeta| < |z|\), the poles inside the integration contour \(C\) are \(\zeta = 0\) and \(\zeta = z_j\). The pole \(\zeta = 0\) does not contribute since \(\Psi^{(AR)}(\zeta) = 0\). Applying again the theorem of residues, we obtain

\[
\Psi^{(AR)}_>(z) = \sum_{j=1}^{N} \beta_j \frac{z_j}{z - z_j},
\]  
(B7)

where \(\beta_j\) is defined as

\[
\beta_j = \frac{1}{z_j} \lim_{z \to z_j} \frac{1}{(m_j - 1)!} \frac{d^{m_j - 1}}{dz^{m_j - 1}}(z - z_j)^{m_j} \Psi^{(AR)}(z).
\]  
(B8)

In the case of simple poles we retrieve expression (31) for the \(\beta_j\). Since the functional form of \(\Psi^{(AR)}_>(z)\) is the same for simple and multiple poles, the calculation of the memory function proceeds in the same way, once the coefficients \(\beta_j\) have been determined.
REFERENCES


TABLE I. Friction coefficients $\xi_0$ computed from eq. (36) for different orders $P$ of the AR process fitted to the particle velocities. In the second line we give the friction coefficients $\xi_0^*$ computed by integration of the memory functions in Fig. 5 (see explanations in the text). The friction coefficient computed from the diffusion coefficient $D$ has the value $\xi_0^{(D)} = 7.135 \text{ ps}^{-1}$.

<table>
<thead>
<tr>
<th>Order $P$</th>
<th>20</th>
<th>50</th>
<th>85</th>
<th>150</th>
<th>250</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi_0 \text{ [ps}^{-1}]$</td>
<td>6.271</td>
<td>6.393</td>
<td>6.559</td>
<td>6.775</td>
<td>6.945</td>
<td>7.295</td>
</tr>
<tr>
<td>$\xi_0^* \text{ [ps}^{-1}]$</td>
<td>5.780</td>
<td>5.900</td>
<td>6.063</td>
<td>6.275</td>
<td>6.433</td>
<td>——</td>
</tr>
</tbody>
</table>
FIGURES

Normalized VACF

Time [ps]

Order 20
Order 50
Order 85
Order 150
Order 250
direct calculation
order 20, every 5th step
order 50, every 5th step
Normalized VACF

Time [ps]

Order 250

Exponential memory
Figure captions:

Fig. 1: Velocity autocorrelation function from direct calculation and from the autoregressive model for orders 20, 50, 85, 150 and 250.

Fig. 2: Spectra of the velocity autocorrelation functions corresponding to Fig. 1. The spectrum corresponding to the solid line has been obtained by discrete Fourier transform of the corresponding VACF in Fig. 1. All other spectra have been directly calculated from the autoregressive model (see eq. (15)).

Fig. 3: Influence of coarse-grained sampling on the calculation of the VACF within the autoregressive model for orders 20 and 50. Each 5th simulation step has been used compared to the VACFs shown in Fig. 1. The solid line corresponds again to the VACF obtained by direct calculation from the MD simulation.

Fig. 4: Spectra of the velocity autocorrelation functions corresponding to Fig. 3. The spectrum corresponding to the solid line has been obtained by discrete Fourier transform of the corresponding VACF in Fig. 3.

Fig. 5: Memory functions for orders 20, 50, 85, 150, and 250. The insert shows a zoom on the differences in the long time evolution.

Fig. 6: Memory function for prediction order 250 compared to the exponential model by Berne et al. [6] which has been obtained from the VACF of Rahman [5].

Fig. 7: VACFs corresponding to the memory functions in Fig. 6.

Fig. 8: Normalized memory functions determined from Rahman’s computer data for the VACF of liquid argon, as compared to the AR model for $P = 250$. The dotted, dashed and long-dashed line have been re-plotted from [36], [8] and [35], respectively.

Fig. 9: Localization of the poles $z_k$ of $\Psi^{(AR)}(z)$ for orders $P = 20$ (crosses) and $P = 500$ (circles). Note that all poles are localized inside the unit circle.