

Niklas Bockius, Maximilian Braun, Kay Hofmann, Friederike Schmid and Martin Hanke*

Determining extended Markov parameterizations for vector-valued generalized Langevin equations

<https://doi.org/10.1515/zna-2025-0354>

Received November 12, 2025; accepted December 3, 2025;

published online December 29, 2025

Abstract: The generalized Langevin equation is used as a model for various coarse-grained physical processes, e.g., the time evolution of the velocity of a given larger particle in an implicitly represented solvent, when the relevant time scales of the dynamics of the larger particle and the solvent particles are not strictly separated. Since this equation involves an integrated history of past velocities, considerable efforts have been made to approximate this dynamics by data-driven Markov models, where auxiliary variables are used to compensate for the memory term. In recent works we have developed two algorithms which can be used for this purpose, provided the dynamics in question are scalar processes. Here we extend these algorithms to vector-valued processes. As a physical test bed we consider an S-shaped particle sliding on a planar substrate, which gives rise to a truly two-dimensional velocity process. The two algorithms provide Markov approximations of this process with 10–20 auxiliary variables and a very accurate fit of the given autocorrelation data over the entire time interval where these data are non-negligible.

Keywords: coarse-graining; model reduction; nonsymmetric block Lanczos method; matrix-valued AAA algorithm

1 Introduction

Stochastic differential equations play a central role in many areas of physics and chemistry [1], [2], as they emerge

naturally in effective descriptions for the motion of particles or collective variables in a fluctuating surrounding environment. Among these, the generalized Langevin equation (GLE) provides a particularly flexible framework [3]–[5]. Unlike simpler stochastic models, such as the classical Langevin equation that treats fluctuations as instantaneous and uncorrelated in time, GLEs incorporate memory effects through a kernel that allows past states of the system to influence its present behavior, complemented by correlated colored noise. It arises in statistical mechanics through projection operator techniques such as the Mori–Zwanzig formalism [6], [7], which offers a formally rigorous unifying framework to connect microscopic dynamics with effective macroscopic descriptions. Motivated by this, GLEs are also frequently proposed as phenomenological models for effective dynamics in slowly relaxing environments. Applications include, among other, solvent effects in molecular dynamics, anomalous diffusion and viscoelastic behavior, chemical reaction dynamics in complex environments, and descriptions for protein dynamics [8]–[11].

When using GLEs in practice, one faces two main challenges. The first is to construct a suitable GLE for a given physical problem. In most cases, a rigorous derivation from first principles is not possible, and one instead seeks a GLE that reproduces the relevant statistical properties of representative trajectories obtained from experiments or microscopic simulations. Assuming that memory and noise are coupled to each other – usually via the so-called fluctuation dissipation relation, see below [4] – this reduces to the task of reconstructing the memory kernel from such data, a problem for which several strategies have been proposed [12]–[25].

The second challenge lies in the numerical integration of GLEs. Direct numerical evaluations of the memory terms are often computationally demanding, especially when the memory kernel decays slowly. Moreover, standard efficient integrators for Markovian equations are no longer applicable for GLEs. An elegant workaround is the Markovian embedding approach, which maps the GLE onto a system of coupled Langevin equations without memory [17], [25]–[29], provided that the kernel can be decomposed into a suitable sum of exponentials. The resulting Markovian system can be simulated at significantly reduced cost.

*Corresponding author: **Martin Hanke**, Institut für Mathematik, Johannes Gutenberg-Universität Mainz, 55099 Mainz, Germany, E-mail: hanke@math.uni-mainz.de

Niklas Bockius and Maximilian Braun, Institut für Mathematik, Johannes Gutenberg-Universität Mainz, 55099 Mainz, Germany, E-mail: nbockiu@uni-mainz.de (N. Bockius), bmaximi@uni-mainz.de (M. Braun)

Kay Hofmann and Friederike Schmid, Institut für Physik, Johannes Gutenberg-Universität Mainz, 55099 Mainz, Germany, E-mail: khofmann@uni-mainz.de (K. Hofmann), friederike.schmid@uni-mainz.de (F. Schmid)

In practical applications, however, this two-step workflow – first reconstructing a GLE from trajectory data and then mapping it to an embedded Markovian system – may be unnecessarily cumbersome and prone to error. A more direct and efficient strategy is to bypass the intermediate GLE altogether and construct the embedded Markovian representation directly from the available trajectories [22], [30]. In earlier work, three of us proposed such a procedure based on model reduction theory and the positive real lemma, and demonstrated its effectiveness in the case of one-dimensional systems [30]. The present study extends this approach to multidimensional settings, where several coupled effective degrees of freedom must be treated simultaneously.

2 Statement of the problem

In the following, we will consider GLEs designed to describe the motion of a single particle in a medium. The GLE has the form

$$M\dot{V}(t) = -\int_0^t ds \gamma(t-s)V(s) + F_R(t), \quad (2.1)$$

where M is the mass of the particle and $V = V(t) \in \mathbb{R}^d$ its velocity. The effect of the medium is encoded in the memory kernel $\gamma(t)$ and in the fluctuating force $F_R(t)$, which is assumed to be Gaussian distributed with mean zero and correlations

$$C_{F_R}(t-t') = \langle F_R(t)F_R(t')^T \rangle = k_B T \gamma(t-t') \quad (2.2)$$

($k_B T$ is the Boltzmann factor). The so-called fluctuation dissipation relation (2.2) can be derived from first principles using the Mori–Zwanzig projection operator formalism [7], and has been shown to also hold under steady-state non-equilibrium conditions for free particles with GLEs of type (2.1) [31], [32]. In the presence of external potentials, its general validity has been questioned [5], [33]. In that case, Equation (2.1) must be extended to include coupled equations of motion for the position, and it might be advisable to allow for nonlinear, possibly position-dependent memory kernels [34]. The treatment of such systems is beyond the scope of the present work.

Our intention is to determine numerically a Markovian approximation of the velocity dynamics by introducing auxiliary variables. More specifically, we are looking for matrices $A_0 \in \mathbb{R}^{N \times N}$, $K \in \mathbb{R}^{d \times d}$, and $B, C, L \in \mathbb{R}^{N \times d}$, such that the first d components Y of the stationary solution of the Langevin equation

$$d \begin{bmatrix} Y \\ Z \end{bmatrix} = \begin{bmatrix} 0 & B^T \\ -C & A_0 \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} dt + \begin{bmatrix} K \\ L \end{bmatrix} dW, \quad t > 0, \quad (2.3)$$

with a d -dimensional Brownian motion W , are close to the rescaled velocity in that the autocorrelation function C_Y satisfies

$$C_Y(v\tau) \approx Y_v := \frac{M}{k_B T} C_V(v\tau), \quad v = 0, \dots, 2n-1. \quad (2.4)$$

In other words, Y_v are given snapshots of the rescaled autocorrelation function C_V of V on an equidistant grid with time step $\tau > 0$. Take note that, due to this normalization,

$$Y_0 = \frac{M}{k_B T} \langle V(t)V(t)^T \rangle = I. \quad (2.5)$$

2.1 Mathematical approach

In order for (2.3) to have a stationary solution it is necessary that the block matrix

$$A = \begin{bmatrix} 0 & B^T \\ -C & A_0 \end{bmatrix} \in \mathbb{R}^{(N+d) \times (N+d)} \quad (2.6)$$

on the right-hand side of (2.3) is stable, i.e., that all its eigenvalues have negative real parts. In this case, cf. [2], the autocorrelation function of Y is given by

$$C_Y(t) = E^T e^{tA} \Sigma E \quad \text{for } t \geq 0, \quad (2.7)$$

where Σ is the covariance matrix associated with the stationary solution of (2.3), and

$$E = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad (2.8)$$

with a $d \times d$ identity matrix block on top of an $N \times d$ zero block. In other words, a multiplication with E^T from the left projects the solution of (2.3) onto its Y -component.

It is well-known that Σ is the unique solution of the Lyapunov equation

$$A\Sigma + \Sigma A^T = -\begin{bmatrix} KK^T & KL^T \\ LK^T & LL^T \end{bmatrix}, \quad (2.9)$$

and without loss of generality we assume that

$$\Sigma = \begin{bmatrix} I & 0 \\ 0 & \Sigma_0 \end{bmatrix} \quad (2.10)$$

with a symmetric positive semidefinite matrix block $\Sigma_0 \in \mathbb{R}^{N \times N}$ in the lower right corner. In other words, $Y(t)$ and $Z(t)$ are taken to be uncorrelated with $C_Y(0) = I = \frac{M}{k_B T} C_V(0)$. Inserting (2.10) into (2.7) we conclude that

$$C_Y(t) = E^T e^{tA} E =: \varphi(t) \quad \text{for } t \geq 0, \quad (2.11)$$

and we note that

$$C_Y(t) = C_Y(-t)^T \quad \text{for } t < 0.$$

Eliminating the auxiliary variables Z from the system (2.3) it is seen that Y obeys generalized Langevin dynamics

$$\dot{Y}(t) = -\int_0^t ds \tilde{\gamma}(t-s)Y(s) + \tilde{F}_R(t) \quad (2.12)$$

with $\langle \tilde{F}_R(t)\tilde{F}_R(0)^T \rangle = \tilde{\gamma}(t)$, where

$$\tilde{\gamma}(t) = B^T e^{tA_0} C, \quad t \geq 0, \quad (2.13)$$

cf., e.g., [35]. When necessary, this function $\tilde{\gamma}$ can be used for approximating the memory kernel of the true dynamics (2.1) of the underlying system, i.e.,

$$\gamma(t) \approx M\tilde{\gamma}(t).$$

Taking the expectation of the product of (2.12) with $Y(0)^T$, one further observes that the memory kernel $\tilde{\gamma}$ satisfies the Volterra integral equation

$$\dot{\tilde{C}}_Y(t) = -\int_0^t ds \tilde{\gamma}(t-s)C_Y(s), \quad (2.14)$$

cf., e.g., [4], [15], [36], [37]. Note that (2.14) implies that

$$\dot{\tilde{C}}_Y(0+) = 0, \quad (2.15)$$

and that

$$\dot{\tilde{C}}_Y(0+) = -\tilde{\gamma}(0) = -\langle \tilde{F}_R(0)\tilde{F}_R(0)^T \rangle \quad (2.16)$$

is symmetric and negative semidefinite. And finally, differentiating once again, we conclude that

$$\ddot{\tilde{C}}_Y(0+) - \ddot{\tilde{C}}_Y(0-) = -(\dot{\tilde{\gamma}}(0+) - \dot{\tilde{\gamma}}(0-)), \quad (2.17)$$

and the right-hand side is positive semidefinite, because $\tilde{\gamma}$ is the autocorrelation function of the random force term in (2.12), and hence, is a function of positive type, cf. [2].

Assuming A to be diagonalizable the projected matrix exponential φ of (2.11) can be written as a finite Prony series

$$\varphi(t) = \sum_{j=1}^p \Gamma_j e^{\lambda_j t}, \quad t \geq 0, \quad (2.18)$$

where the exponents λ_j are the eigenvalues of A and $\Gamma_j \in \mathbb{R}^{d \times d}$ are appropriate coefficient matrices. Accordingly, in order to satisfy (2.4) one has to select a reasonable number $m \in \mathbb{N}$ of appropriate exponents λ_j with negative real parts and associated $d \times d$ coefficient matrices Γ_j , such that

$$\varphi(\nu\tau) = \sum_{j=1}^p \Gamma_j e^{\nu\tau\lambda_j} \approx Y_\nu, \quad \nu = 0, 1, \dots, 2n-1. \quad (2.19)$$

However, not every Prony series φ of this form is admissible as an autocorrelation function, since Bochner's theorem (cf. [2]) states that the Fourier transform of every autocorrelation function must be nonnegative semidefinite for every frequency $\omega \in \mathbb{R}$.

Unfortunately, it is an unresolved problem to find simple suitable conditions on the coefficients λ_j and Γ_j of (2.19), such that the Fourier transform $\hat{\varphi}$ is nonnegative semidefinite. The good news, on the other hand, is that it often suffices to make sure that φ satisfies the two algebraic conditions which must be fulfilled by C_Y according to (2.15) and (2.16), namely that

$$\dot{\varphi}(0) = \sum_{j=1}^p \lambda_j \Gamma_j = 0 \quad \text{and} \quad \ddot{\varphi}(0) = \sum_{j=1}^p \lambda_j^2 \Gamma_j \quad \text{is symmetric,} \quad (2.20)$$

to achieve that $\hat{\varphi}$ is positive semidefinite. Take note that we also want to have

$$\varphi(0) = \sum_{j=1}^p \Gamma_j = C_Y(0) = I, \quad (2.21)$$

due to our normalization (2.5).

2.2 A physical model system

As a test case for our numerical algorithms, we consider the two-dimensional motion of a rigid S-shaped particle sliding on a planar substrate and immersed in a bath of N_L spherical particles of mass m and diameter σ . The S-shaped particle undergoes translational motion, but does not rotate. This system is deliberately chosen for its low symmetry: Not only is the diffusion tensor anisotropic, but the different velocity components are also coupled in a nontrivial manner that cannot be eliminated by a simple axis transformation. Physically, the rotation of probe particles could be prevented, e.g., by applying external orienting fields.

The state of the bath particles is described by their positions r_i and velocities v_i , and they interact with each other via a Weeks–Chandler–Anderson (WCA) potential,

$$U_{\text{WCA}}(r_{ij}) = \begin{cases} 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right) + \epsilon, & r_{ij} \leq \sqrt[6]{2}\sigma, \\ 0, & r_{ij} > \sqrt[6]{2}\sigma, \end{cases} \quad (2.22)$$

where r_{ij} is the distance between the two interacting particles. Furthermore, they are coupled to a Langevin thermostat, which mimics the friction with the substrate. Introducing the implicit substrate has the advantage of cutting off long-range hydrodynamic interactions, which decay only logarithmically in two dimensions and give rise to pronounced finite size effects and hydrodynamic backflows.

Dealing with such complications is beyond the scope of the present work. The equations of motion for the i th Langevin particle are thus

$$m\dot{v}_i = -\gamma v_i + \xi_i - \sum_{j \neq i} \nabla U_{\text{WCA}}(r_{ij}), \quad (2.23)$$

$$\dot{r}_i = v_i, \quad (2.24)$$

with γ denoting a translational friction coefficient. The random force vector ξ is defined as Gaussian white noise, with vanishing mean and autocorrelation

$$C_\xi(t) = 2k_B T \gamma \delta(t) I, \quad t \in \mathbb{R}. \quad (2.25)$$

The S-shaped probe particle is taken to be a rigid chain of N_s beads with mass m_s and diameter $d_s = \sigma$. Thus, the total mass of the S-shaped probe results in $M = N_s m_s$. The state of the S-shaped particle is described by the position R and velocity V of the center of mass. The bead particles interact with the bath particles with the same WCA potential, Equation (2.22), but they are not coupled to a thermostat, i.e., they can slide freely on the substrate. Thus, we focus on the effective dynamics arising from the interaction with the bath particles. The equations governing the motion of the S-shaped probe are

$$M\dot{V} = -\sum_{i,k} \nabla_k U_{\text{WCA}}(r_{ik}), \quad (2.26)$$

$$\dot{R} = V, \quad (2.27)$$

where the sum runs over all pairs of bath particles i and bead particles k . Since rotational motion is prevented, the S-shaped probe maintains a fixed orientation. Due to the planar substrate, the motion of the Langevin particles and the S-shaped probe is confined to a two-dimensional plane. All quantities will be given in units of the length σ , the thermal energy $k_B T$, and a “natural” time unit $t_0 = \sqrt{m\sigma^2/k_B T}$, which corresponds to the inertial time scale of bath particles. We set $\gamma = 3\pi k_B T t_0$, which represents the friction coefficient of a spherical particle in a Newtonian fluid with viscosity $\eta = 1k_B T t_0 / \sigma$ (i.e. $\gamma = 3\pi\eta\sigma$). In the WCA potential (Eq. (2.22)), we choose $\epsilon = k_B T$.

3 Methods

3.1 Simulation details

The trajectories of the molecular dynamics simulations are generated according to the Langevin equation of motion (2.23)–(2.27) modelling a canonical ensemble, as implemented in the HOOMD-blue package [38]. Each bath particle has a mass $m = 1k_B T t_0^2 / \sigma^2$. The S-shaped probe consists

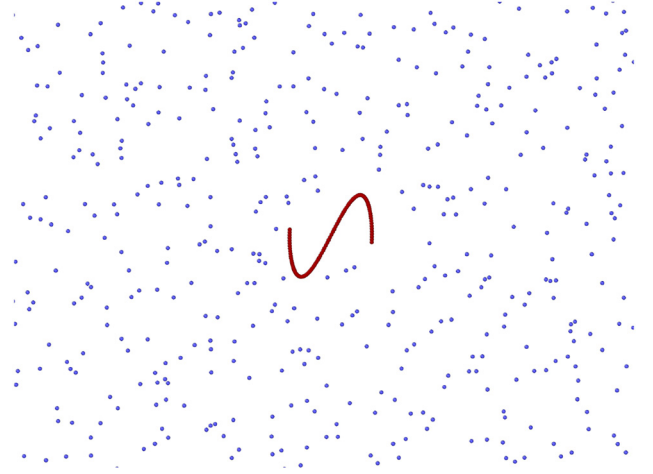


Figure 1: Snapshot of the simulated ensemble.

of $N_s = 100$ beads each with a mass of $m_s = 0.03 m$. The positions r_k of the constituent beads, which make up the particular shape displayed in Figure 1, are calculated by $r_k = R + 10 \cdot (\sin t_k, \sin 2t_k)$, where R is the position of the center of mass of the probe and t_k is evenly spaced in the interval $[-\pi + 1.5, \pi - 1.5]$. The bath consists of $N_L = 5,000$ particles in a total simulation area of size $43.113 \times 43.113 \sigma^2$ with periodic boundary conditions. Each simulation run is split in an equilibration period of 10^5 time steps and a production run of 10^8 time steps with a step size of $\Delta t = 0.001 t_0$.

3.2 A block version of Prony’s method

The first algorithm that we consider for constructing a suitable Langevin system (2.3) generalizes the method from [30] to vector-valued processes. It attempts to construct a finite Prony series (2.18) which interpolates the given time samples,

$$\varphi(\nu\tau) = Y_\nu, \quad \nu = 0, \dots, 2n - 1, \quad (3.1)$$

i.e., (2.19) holds true with equality signs.

We use these time samples to define the block Hankel matrix

$$\mathcal{H} = \begin{bmatrix} Y_0 & Y_1 & \dots & Y_n \\ Y_1 & & \ddots & \vdots \\ \vdots & \ddots & & Y_{2n-1} \\ Y_n & \dots & Y_{2n-1} & 0 \end{bmatrix} \quad (3.2)$$

and the induced bilinear form

$$[\psi, \phi]_{\mathcal{H}} = a^T \mathcal{H} b \quad (3.3)$$

for polynomials

$$\psi(x) = \sum_{i=0}^{(n+1)d-1} a_i x^i \quad \text{and} \quad \phi(x) = \sum_{i=0}^{(n+1)d-1} b_i x^i$$

of degree $(n+1)d-1$ with coefficient vectors $a = [a_0, a_1, \dots]^T$ and $b = [b_0, b_1, \dots]^T$ in $\mathbb{R}^{(n+1)d}$. In contrast to the scalar case the bilinear form (3.3) does not correspond to a moment functional, and since the matrix \mathcal{H} does not need to be symmetric, the bilinear form also fails to be symmetric in general. Instead of the symmetric Lanczos process, which is behind the scalar Prony method utilized in [30], we therefore have to resort to the nonsymmetric (block) Lanczos process, which determines two bases $\{\psi_k\}$ and $\{\phi_k\}$ of polynomials which are biorthogonal with respect to $[\cdot, \cdot]_{\mathcal{H}}$, i.e.

$$|[\psi_k, \phi_l]_{\mathcal{H}}| = \delta_{kl} \quad \text{for } k, l = 0, \dots, 2n-1.$$

To be specific, we apply the algorithm from [39] to construct these two bases, see Appendix A.1. This method proceeds similarly to the Gram–Schmidt algorithm by successively orthogonalizing certain basis polynomials of increasing degrees against the previous ones. But we note that the algorithm may fail because the bilinear form is no inner product; such a potential failure has also been a threat in the scalar case, but since our polynomial degrees are of moderate size, we have never encountered this problem in our computations.

The corresponding recursion coefficients make up the entries of a banded matrix $T \in \mathbb{R}^{dn \times dn}$ with upper and lower bandwidth d , which satisfies

$$E^T T^k E = Y_k \quad \text{for } k = 0, \dots, 2n-1, \quad (3.4)$$

with E of (2.8). Consequently, the function φ defined in (2.11), using

$$A = \frac{1}{\tau} \log T, \quad (3.5)$$

where $\log T$ denotes the matrix logarithm of T (cf. Higham [40]), solves the interpolation problem (3.1), and hence, the matrix in (3.5) is our candidate for the system matrix A in (2.6).

As in [30] the resulting Prony series may contain exponents λ with nonnegative real parts. We call the corresponding terms “spurious” exponentials since they do not converge to 0 and are thus unphysical. Since our data come from a physical model, the corresponding coefficients of the Prony series are small compared to the other ones unless the noise is too large. Eliminating these terms will therefore have a negligible impact on the values of the Prony series for $t \leq (2n-1)\tau$. We therefore remove the corresponding eigenvalues from T and A ; the technical details are postponed to Appendix A.2.

Another issue, which we have to take care of, is negative real eigenvalues of T , for otherwise A of (3.5) will not be a real-valued matrix, and the Prony series φ will not be a real-valued function. Again we already encountered this problem in the scalar case, and we fix it here in the very same way as in [30, Appendix B.2].

So far we have not discussed the two side conditions in (2.20). In the scalar case, where $d = 1$, the second condition in (2.20) is void, so that (2.20) amounts to the single scalar constraint

$$\dot{\varphi}(0) = 0. \quad (3.6)$$

In [30] this property has been achieved by taking the (scalar) function value Y_1 for the first nonzero grid point as a free parameter, and fit this parameter in an outer Newton iteration to achieve the constraint (3.6) for the solution of the interpolation problem (3.1). In the vector-valued case the situation is significantly more difficult, because (2.20) now consists of $3d^2/2 - d/2$ scalar equations; already when $d = 2$ these are five scalar equations to be fulfilled. The entries of Y_1 no longer provide enough parameters to fit, and on top of that we have experienced numerically that the higher dimensional Newton scheme is prone to deliver inappropriate Prony series.

For this reason it is with a heavy heart that we waive the side constraints (2.20) when using the above approach. Note that it follows from (2.11) that

$$\dot{\varphi}(0) = E^T A E$$

is the upper left block entry of the system matrix A . Accordingly, if $D := -\dot{\varphi}(0) \neq 0$ for the chosen Prony series, then the upper left block of the final system matrix will be nonzero. Instead of (2.3) the Langevin dynamics then assumes the form

$$d \begin{bmatrix} Y \\ Z \end{bmatrix} = \begin{bmatrix} -D & B^T \\ -C & A_0 \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} dt + \begin{bmatrix} K \\ L \end{bmatrix} dW, \quad (3.7)$$

and in order to have a stationary solution with some covariance matrix Σ as in (2.10) it must be required that $D + D^T$ is positive semidefinite with

$$D + D^T = K K^T; \quad (3.8)$$

this is a consequence of the Lyapunov equation (2.9). As shown in [35] the corresponding component Y then satisfies the generalized Langevin dynamics

$$dY(t) = \left(-DY(t) - \int_0^t ds \tilde{\gamma}(t-s)Y(s) + \tilde{F}_R(t) \right) dt + K dW(t) \quad (3.9)$$

with an extra damping and associated random force term. Further, the autocorrelation function of \tilde{F}_R now takes the form

$$\langle \tilde{F}_R(t)\tilde{F}_R(0)^T \rangle = \tilde{\gamma}(t) - B^T e^{t|A_0} LK^T.$$

Fortunately the model (3.9) is also suitable from a physical point of view, with D being associated with an instantaneous damping coefficient.

Below we will denote the above algorithm for setting up the Langevin system (3.7), respectively the Prony series φ , by Method A for short.

3.3 A method based on rational approximation

The introduction of the additional instantaneous damping coefficient in (3.9) can be avoided with a different approach, which has originally been suggested in [41] for scalar-valued stochastic processes. Here we provide an extension of this method to vector-valued generalized Langevin dynamics, later referred to as Method B.

Following [41] we introduce the generating function

$$F(z) = \sum_{\nu=0}^{\infty} Y_{\nu} z^{-\nu-1} \tag{3.10}$$

of all equidistant snapshots Y_{ν} . This function is analytic in the exterior of the unit disk. Under the assumption that (2.19) holds true with equality for all $\nu \in \mathbb{N}_0$, the generating function can be rewritten as

$$\begin{aligned} F(z) &= \sum_{\nu=0}^{\infty} \sum_{j=1}^p \Gamma_j e^{\nu\tau\lambda_j} z^{-\nu-1} = \sum_{j=1}^p \frac{\Gamma_j}{z} \sum_{\nu=0}^{\infty} (e^{\tau\lambda_j}/z)^{\nu} \\ &= \sum_{j=1}^p \frac{\Gamma_j}{z - z_j} \end{aligned} \tag{3.11}$$

with

$$z_j = e^{\tau\lambda_j}, \quad j = 1, \dots, p.$$

We therefore proceed by approximating F by a rational function with real coefficients in the exterior of the unit disk in order to use the poles of this approximation for the construction of a suitable Prony series φ . Since F is a matrix-valued function, we use a matrix-valued variant of the AAA algorithm for computing this rational approximation, cf. Appendix B.1 for details.

For every pole z of this rational approximation with $|z| < 1$, we include the exponent(s)

$$\lambda = \begin{cases} \frac{1}{\tau} \log z, & z \notin \mathbb{R}^-, \\ \frac{1}{\tau} \log |z| \pm \frac{\pi}{\tau} i, & z \in \mathbb{R}^-, \end{cases}$$

in the Prony series, whereas poles with $|z| \geq 1$ are discarded; the latter may occur as artefacts of the rational approximation. Since our rational approximation has real coefficients, the resulting exponents λ are either real or appear in complex conjugate pairs. It then remains to choose appropriate coefficient matrices $\Gamma_j \in \mathbb{C}^{d \times d}$ for the Prony series (2.18). In view of our goal (2.19) and the aforementioned side constraints (2.20) and (2.21) we consider the equality constrained least-squares problem

$$\begin{aligned} \text{minimize} \quad & \sum_{\nu=0}^{2n-1} \left\| \sum_{j=1}^p \Gamma_j e^{\lambda_j \tau \nu} - Y_{\nu} \right\|_F^2 \\ \text{subject to} \quad & \sum_{j=1}^p \Gamma_j = I, \quad \sum_{j=1}^p \lambda_j \Gamma_j = 0, \quad \text{and} \tag{3.12} \\ & \sum_{j=1}^p \lambda_j^2 (\Gamma_j - \Gamma_j^T) = 0, \end{aligned}$$

for this purpose, where $\|\cdot\|_F$ refers to the Frobenius norm. This is a well-understood problem in numerical linear algebra which can be transformed into the solution of a linear system of equations, cf. Björck [42]. Moreover, it can be shown that the matrix coefficients Γ_j corresponding to real exponents λ_j have real entries, and that the matrices Γ_j corresponding to complex conjugate pairs of exponents are also complex conjugates of each other. Accordingly, the resulting Prony series φ is a real-valued function.

To achieve the desired representation

$$\varphi(t) = E^T e^{tA} E \tag{3.13}$$

of this function, we have to assemble a matrix A with eigenvalues λ_j and suitable eigenvectors. This somewhat technical detail is worked out in Appendix B.2.

3.4 The positive real lemma

Once the system matrix A of (2.6) has been determined by either of the two approaches from Sections 3.2 or 3.3, it remains to construct the matrix blocks K and L , which steer the Brownian motion in (2.3), and to determine the associated covariance matrix Σ . Note that when inserting (2.6) and (2.10) into the Lyapunov equation (2.9), we see that the latter is equivalent to the system

$$K = 0, \quad A_0 \Sigma_0 + \Sigma_0 A_0^T = -LL^T, \quad \Sigma_0 B = C. \tag{3.14}$$

These are the so-called (singular) Lur'e equations. Likewise, if an instantaneous damping term is added to the system as in (3.9) then the corresponding equations become the (regular) Lur'e system

$$\begin{aligned} D + D^T &= KK^T, & A_0 \Sigma_0 + \Sigma_0 A_0^T &= -LL^T, \\ C - \Sigma_0 B &= LK^T. \end{aligned} \quad (3.15)$$

It is known, cf. [43], that for our input matrices A_0, B, C – and D in (3.15) – the equations (3.14) and (3.15), respectively, have a solution consisting of the matrices $K \in \mathbb{R}^{d \times d}$, $L \in \mathbb{R}^{N \times d}$, and $\Sigma_0 \in \mathbb{R}^{N \times N}$ (the latter being symmetric and positive definite) if and only if the Prony series φ (with its proper extension to the negative time axis) satisfies the requirements of Bochner’s theorem. In other words, if the Lur’e equations fail to have a solution (K, L, Σ_0) , then there is no stochastic process whose autocorrelation function coincides with the Prony series (2.18).

The numerical solution of the Lur’e equations is a bit tricky, cf. [44]. We therefore provide in Appendix C a sketch of the corresponding algorithm under the generic assumptions that

$$\dot{\varphi}(0+) = B^T C \quad \text{is positive definite} \quad (3.16a)$$

and

$$\ddot{\varphi}(0+) + \ddot{\varphi}(0+)^T = -B^T A_0 C - C^T A_0^T B \quad \text{is positive definite} \quad (3.16b)$$

in the case of the singular Lur’e equations (3.14), compare (2.16) and (2.17), and that

$$\dot{\varphi}(0+) + \dot{\varphi}(0+)^T = -(D + D^T) \quad \text{is negative definite} \quad (3.17)$$

in the regular case (3.15), respectively.

4 Numerical results

We now present numerical examples for the two-dimensional velocity (i.e., $d = 2$) of the S-shaped particle described in Section 2.2. As mentioned before our target is the autocorrelation function of the rescaled variables, cf. (2.4), which is shown as black dashed line in Figure 2.

In our first test case we use $2n = 30$ samples of these data with a grid spacing of $\tau = 0.5t_0$, highlighted as black dots in this graph; recall that t_0 is the “natural” time unit introduced in Section 2.2.

The system matrix obtained by Method A (of Section 3.2) is of size 30×30 , because $dn = 30$, too, in the two-dimensional case. Since we ignore the side constraints (2.20), the (1,1)-block $-D$ of A is nonzero; compare (3.7). This system matrix has 13 spurious eigenvalues and one negative real eigenvalue. The elimination of the spurious eigenvalues and duplication of the negative real eigenvalue

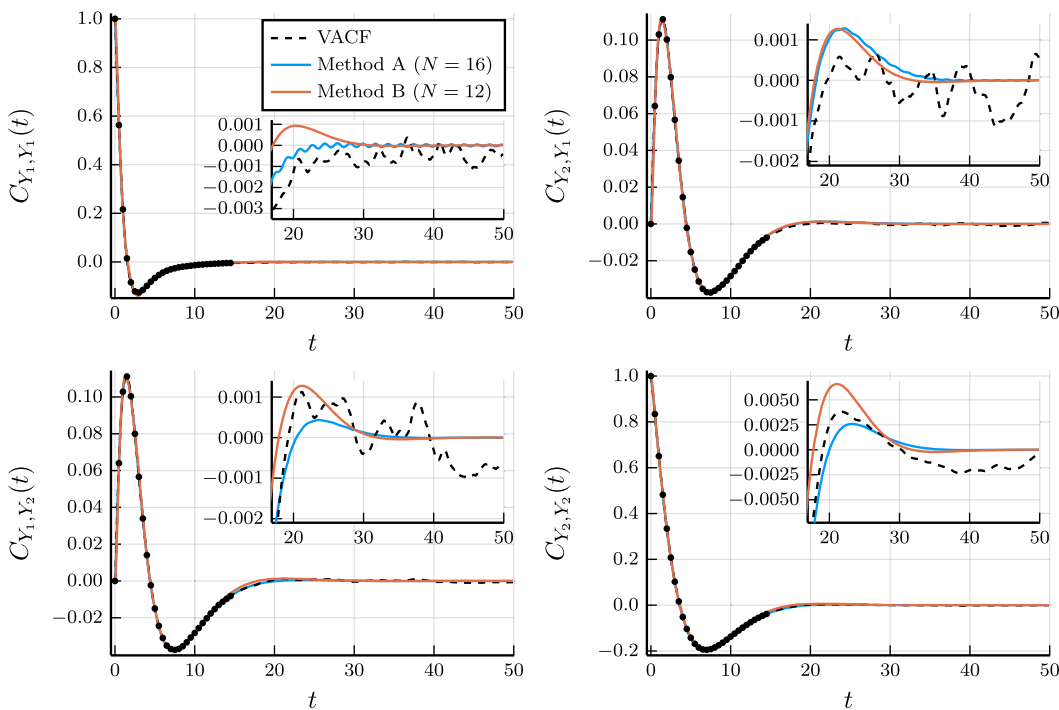


Figure 2: Rescaled velocity autocorrelation function (VACF) versus time t in units of t_0 and the two approximations: Method A is the block Prony method from Section 3.2 and Method B is the one from Section 3.3 based on rational approximation. N is the number of auxiliary variables of the respective Langevin system. Every panel shows the scalar graphs corresponding to the respective entry of the (matrix-valued) function.

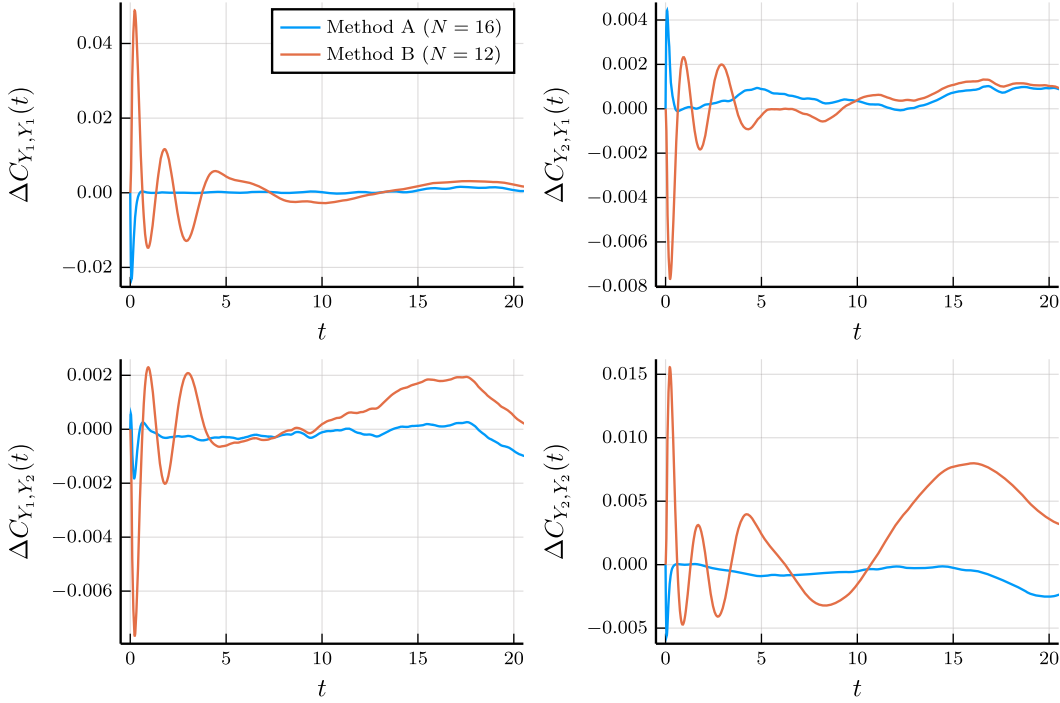


Figure 3: Differences ΔC_Y between the given velocity autocorrelation function (same units) and the two approximations: Method A is the block Prony method from Section 3.2 and Method B is the one from Section 3.3 based on rational approximation. N is the number of auxiliary variables of the respective Langevin system. Every panel shows the scalar graphs corresponding to the respective entry of the (matrix-valued) function.

results in a 18×18 system matrix A , which corresponds to $N = 18 - d = 16$ auxiliary variables. The corresponding regular Lur'e equations are solvable, which shows that the computed Prony series is a valid autocorrelation function.

Concerning Method B (from Section 3.3) the AAA algorithm determines a rational function with seven admissible poles, from which one is positive and the remaining six come in three complex conjugate pairs. The algorithm therefore proceeds with a Prony series Ansatz with $p = 7$ terms, and the computation of the corresponding matrix coefficients Γ_j is carried out as described in Section 3.3. The resulting system matrix A – constructed as in Appendix B.2 – has dimension $dp = 14$, which corresponds to $N = 12$ auxiliary variables for this system. It turns out that the singular Lur'e system (3.14) is solvable, and the solution can be found as in Appendix C, because the two conditions (3.16a) and (3.16b) are indeed satisfied. In other words, the associated Prony series is also a valid autocorrelation function.

Figure 2 displays the two constructed Prony series approximations of the autocorrelation function, the one from Method A in blue, and the one from Method B in red. Each subplot corresponds to one entry of the corresponding 2×2 matrices. In these plots the blue line is hardly visible as it is almost covered by the red one. The inset plots therefore display enlarged versions of the corresponding graphs restricted to large values of t .

Although not visible with the eye, the red line is – by construction – the graph of a function with zero derivative at the origin (in each subplot), and it turns out that it is for this reason that the most significant differences between the two approximations occur for times $t < 5t_0$. This can be seen in Figure 3, which displays the difference ΔC_Y between the two constructed Prony series $\varphi = C_Y$ and the input autocorrelation function $\frac{M}{k_B T} C_V$. It also shows that the detection and proper reconstruction of the steep descent of the autocorrelation function for $t < 1$, which is only encoded within the first few data points, poses a big challenge for both methods; accordingly, the error of both methods is largest in this initial time interval.

As can be seen from the inset plots the approximations from both methods start to deviate from the given autocorrelation data far in the extrapolated regime, i.e., for times $t \geq 20t_0$. To be specific, both approximations converge to zero in this regime, whereas the given data fail to do so because of noise. One may argue that in this regime the computed approximations provide a better picture of the true autocorrelation functions than the given MD data.

Table 1 compiles the results of further test runs for the same example, but using more input data. More precisely, each run uses data from a different grid with the same grid spacing $\tau = 0.5t_0$, but now with $2n$ data points, where n

Table 1: Approximation errors (same units as in Figure 3) of both methods (a) at the given grid points, (b) in between the grid points, and (c) for the entire time interval $[0, 50]$, for varying number of input data. Also listed are the numbers N of respective auxiliary variables.

n	Block Prony method				Rational approx. method			
	(a)	(b)	(c)	N	(a)	(b)	(c)	N
15	0.001	0.024	0.024	16	0.019	0.052	0.052	12
16	0.001	0.022	0.022	19	0.035	0.065	0.065	12
17	0.002	0.022	0.022	17	0.040	0.068	0.068	12
18	0.002	0.022	0.022	19	0.049	0.074	0.074	12
19	0.112	0.112	0.112	18	0.049	0.074	0.074	12
20	0.015	0.018	0.018	20	0.016	0.054	0.054	16
21	0.014	0.020	0.020	18	0.025	0.061	0.061	16
22	0.033	0.033	0.033	25	0.016	0.052	0.052	16
23	0.007	0.022	0.022	27	0.015	0.052	0.052	16
24	0.003	0.022	0.022	35	0.015	0.052	0.052	16
25	0.002	0.022	0.022	35	0.016	0.053	0.053	16
26	0.002	0.018	0.018	28	0.016	0.053	0.053	16
27	0.002	0.021	0.021	30	0.018	0.057	0.057	16
28	0.004	0.020	0.020	33	0.018	0.057	0.057	16
29	0.002	0.021	0.021	32	0.018	0.057	0.057	16
30	0.001	0.020	0.020	32	0.018	0.056	0.056	16

ranges up to 30. For each value of n the table lists the maximum approximation error at the respective grid points in column (a), the maximum approximation error in between these grid points in column (b), and the maximum approximation error up to $t = 50t_0$ in column (c) for each of the two methods. Each of these error numbers corresponds to the Frobenius norm of the difference ΔC_Y between the input autocorrelation function and its approximation φ at a given time.

Method A succeeds for every tested value of n , i.e. it yields approximations of the given autocorrelation function, which are admissible in the sense that they are autocorrelation functions of the Y -component of the solution of an appropriate Langevin equation (2.3). The associated approximation errors are small in all cases except for the grid corresponding to $n = 19$, where a pair of two complex conjugate spurious poles with comparably large coefficients 0.0045 has to be removed from the Prony series, which results in a larger approximation error.

Method B also yields admissible approximations for all the data sets that we have tested. For the smaller grids, i.e., for n up to 19, the amount of data has only a minor influence on the computed approximations in that the resulting Prony series are very similar for these grids. For $n = 20$ and beyond, two further (complex conjugate) exponentials (and $2d = 4$ further auxiliary variables) are chosen for a better match of the additional data points; this corresponds to one additional step of the greedy iteration in the AAA method (see Appendix B.1). For the two cases $n = 20$ and $n = 21$ this additional iteration was enforced manually, since

the approximation chosen with the default parameters of our implementation would have been too poor otherwise. For $n > 21$ the algorithm made the additional iteration by itself.

From Table 1 it can further be seen that the approximation error of Method A is usually smaller than the error of Method B by a factor of two to three, except when $n = 19$, where the error in Method A is exceptionally large. On the other hand, Method A requires a larger number of auxiliary variables.

In summary, the numerical examples show that both methods have high potential to provide excellent approximations of vector-valued stochastic processes coming from GLEs. For our particular test case Method B shows a slightly better performance, as it manages with fewer auxiliary variables, and at the same time fulfills the side constraints (2.20) and (2.21). Method A on the other hand yields better approximations errorwise, but in general needs up to twice the amount of auxiliary variables than Method B.

5 Conclusions

We have presented an extension of a recently proposed framework for constructing Markovian embeddings of GLEs directly from trajectory data to multidimensional systems. The approach bypasses the intermediate step of reconstructing a full memory kernel and instead identifies a minimal Markovian representation whose autocorrelation function matches that of the observed dynamics.

Two numerical strategies have been developed and compared. Method A uses a Prony series expansion, extending a scheme proposed in Ref. [30] for the scalar case to vector-valued stationary solutions of GLEs. Method B introduces a generating function, following Ref. [41], which is then approximated by a rational function using the AAA algorithm. Application to a two-dimensional test system – a rigid S-shaped probe particle in a solvent of Langevin particles – demonstrates that both methods can accurately reproduce velocity autocorrelation functions obtained from molecular dynamics simulations.

Approximation errors of the resulting autocorrelation functions and the number of auxiliary variables have been documented for different sizes of the input data sets to demonstrate the performance and the potential of the two methods. For our test example the block version of Prony's method (Method A of Section 3.2) yields smaller approximation errors than the method based on rational approximation (Method B of Section 3.3), whereas the latter avoids an artificial instantaneous damping term and requires somewhat fewer auxiliary variables. The comparative analysis thus suggests that Method A is preferable when accuracy is paramount, while Method B is advantageous when model parsimony is desired.

The framework is broadly applicable to the data-driven construction of reduced stochastic models in higher dimensions. In this work, we did not consider GLEs with additional deterministic forces, or other position-dependent phenomena. This is left for future work.

Acknowledgments: The research leading to this work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in the framework to the collaborative research center “Multiscale Simulation Methods for Soft-Matter Systems” (TRR 146, Subproject A3) under Project No. 233630050.

Research ethics: Not applicable.

Informed consent: Not applicable.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Use of Large Language Models, AI and Machine Learning Tools: None declared.

Conflict of interest: The authors state no conflict of interest.

Research funding: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project No. 233630050.

Data availability: The data that support the findings of this study are openly available on Github (<https://github.com/bmaximi/vector-memory>).

Appendix A. The block Lanczos approach

A.1 Outline of the Lanczos method

The algorithm from [39] computes two finite sequences of polynomials ψ_k and ϕ_k of a real variable x of degree k , respectively, with k running from zero to $(n+1)d-1$, such that

$$|[\psi_k, \phi_l]_{\mathcal{H}}| = \delta_{kl}, \quad (\text{A.1})$$

where $[\cdot, \cdot]_{\mathcal{H}}$ is the bilinear form introduced in (3.3).

These polynomials are generated recursively by first initializing

$$\psi_k = \phi_k = x^k, \quad k = 0, \dots, d-1,$$

and then, defining

$$\alpha_{k,k+d} \psi_{k+d} = x^d \psi_k - \sum_{i=0}^{k+d-1} \alpha_{k,i} \psi_i \quad (\text{A.2})$$

and

$$\beta_{k,k+d} \phi_{k+d} = x^d \phi_k - \sum_{i=0}^{k+d-1} \beta_{k,i} \phi_i \quad (\text{A.3})$$

for $k = 0, 1, 2, \dots, nd-1$, with

$$\alpha_{k,i} = \frac{[x^d \psi_k, \phi_i]_{\mathcal{H}}}{[\psi_i, \phi_i]_{\mathcal{H}}} \quad \text{and} \quad \beta_{k,i} = \frac{[\psi_i, x^d \phi_k]_{\mathcal{H}}}{[\psi_i, \phi_i]_{\mathcal{H}}} \quad (\text{A.4})$$

for $i = 0, \dots, k+d-1$. We choose the positive scaling factors $\alpha_{k,k+d}$ and $\beta_{k,k+d}$ to have the same value, and this value is chosen such that

$$|[\psi_{k+d}, \phi_{k+d}]_{\mathcal{H}}| = 1.$$

This is the point where the algorithm can fail, namely, if no such normalization is possible; but as mentioned before, we have never encountered a failure in our numerical tests. Take note that the evaluation of the bilinear form requires the coefficients of the respective polynomials, when expanded in powers of x ; compare (3.3).

Due to the block Hankel structure of the matrix \mathcal{H} the bilinear form satisfies

$$[x^d \psi, \phi]_{\mathcal{H}} = [\psi, x^d \phi]_{\mathcal{H}}$$

for any two polynomials ϕ and ψ of degree $nd-1$ or less. It therefore follows from (A.1) that

$$\alpha_{k,i} = \beta_{k,i} = 0 \quad \text{for } 0 \leq i < k-d,$$

and hence, the recursion coefficients $\alpha_{k,i}$ make up the entries of a real $nd \times nd$ matrix

$$T = \begin{bmatrix} \alpha_{0,0} & \cdots & \alpha_{0,d} & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ \alpha_{d,0} & & & & \ddots & 0 \\ 0 & \ddots & & & & \alpha_{(n-1)d-1,nd-1} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & \alpha_{nd-1,(n-1)d-1} & \cdots & \alpha_{nd-1,nd-1} \end{bmatrix} \quad (\text{A.5})$$

of upper and lower bandwidth d . This is the matrix T which satisfies (3.4); compare Freund [45] for a proof. Take note that the coefficients $\alpha_{k,i}$ with $i \geq nd$ for the last block of indices $k \geq (n-1)d$ are irrelevant for the definition of this matrix.

A.2 Spurious eigenvalues of T

The following spectral modifications of T are similar to [30]. Throughout we assume that T is diagonalizable.

To remove spurious exponentials from the Prony series, we consider the eigenvalue decompositions

$$T = X\Lambda_T X^{-1} \quad \text{and} \quad A = X\Lambda_A X^{-1}$$

of T and A with $\Lambda_T = \text{diag}(z_1, \dots, z_{nd})$ and $\Lambda_A = \text{diag}(\lambda_1, \dots, \lambda_{nd})$, where $z_k = e^{\tau\lambda_k}$. Let $q \in \mathbb{N}$ be such that $\lambda_{q+1}, \dots, \lambda_{nd}$ be all the eigenvalues of A with positive real part. Then we replace Λ_A by $\tilde{\Lambda}_A := \text{diag}(\lambda_1, \dots, \lambda_q)$ and X by a matrix \tilde{X} which is obtained by removing the last $nd - q$ columns (corresponding to eigenvectors for the spurious exponents) and certain rows from X . In principle any $nd - q$ rows can be removed from X as long as the first d rows are kept and the resulting matrix \tilde{X} is nonsingular. In the scalar case considered in [30] we always removed the last $n - q$ rows, but for $d > 1$ this choice can fail. To enhance stability we construct \tilde{X} by copying the first d rows of X , and then appending one by one the particular row of X to \tilde{X} , for which the norm of its orthogonal complement to the span of the already chosen rows divided by its norm is largest.

Appendix B. The rational approximation approach

B.1 The AAA algorithm

We choose the AAA algorithm [46] to solve the rational approximation problem. More precisely, since the function

F of (3.10) to be approximated is matrix-valued, we use a matrix-valued variant of the AAA algorithm from Gosea and Güttel [47], which goes back to Lietaert et al. [48].

We let $\mathcal{Z} = \{\zeta_l\}$ be the elements of an equiangular grid with g grid points on a circle $|\zeta| = \rho > 1$ in the complex plane, which are symmetric with respect to the real axis and include the two real points $\pm\rho$, and approximate the associated function values $F(\zeta_l)$ by

$$F_l = \sum_{v=0}^{2n-1} Y_v \zeta_l^{-v-1}.$$

Below, the matrix entries of F_l will be denoted by $F_{i,j}^l$, $i, j = 1, \dots, d$. Concerning the particular choice of the grid, we found in our numerical experiments that $g = 100$ and

$$\rho = (1/\delta)^{\frac{1}{2n}}$$

with $\delta = 10^{-6}$ is an appropriate choice for this particular example. These parameters should be adapted when other data sets are considered.

The AAA algorithm determines a rational function in barycentric form

$$r(z) = \sum_{k=1}^p \frac{w_k F_k}{z - \zeta_k} \bigg/ \sum_{k=1}^p \frac{w_k}{z - \zeta_k}, \quad (\text{B.1})$$

where $\zeta_k \in \mathcal{Z}$, $k = 1, \dots, p < g$, are the so-called support points, and w_k , $k = 1, \dots, p$, are suitable (complex) scalar weights. Under the assumption that these weights are nonzero, the function (B.1) interpolates the given values F_k at all the support points, i.e.,

$$r(\zeta_k) = F_k, \quad k = 1, \dots, p.$$

We take care (see below) that the weights associated with complex conjugate support points are also complex conjugate, so that $r(\bar{z}) = \overline{r(z)}$ for every $z \in \mathbb{C}$, and hence, r has a representation as a quotient of two polynomials with real coefficients (which albeit is never explicitly computed).

The algorithm employs a greedy iterative scheme. In each iteration a new support point $\zeta_l \in \mathcal{Z}$ is chosen, for which the Frobenius norm of the current nonlinear residual $\|F_l - r(\zeta_l)\|_F$ is maximal. We also include the complex conjugate point $\zeta_l = \bar{\zeta}_l$ to the list of support points; the two associated nonlinear residuals share the same Frobenius norm. Then the weights w_1, \dots, w_p for (B.1) are chosen such that they solve a linearized least-squares problem over the remaining points in \mathcal{Z} : with π denoting the sum in the denominator in (B.1), we minimize

$$\sum_{l=p+1}^g \left\| \pi(\zeta_l) (F_l - r(\zeta_l)) \right\|_F^2$$

$$= \sum_{l=p+1}^g \left\| \sum_{k=1}^p w_k \frac{F_l - F_k}{\zeta_l - \zeta_k} \right\|_F^2 \rightarrow \min .,$$

subject to the constraint that the weight vector $w = [w_1, \dots, w_p]^T$ has unit (Euclidean) norm. We can write this in the form

$$\|Lw\|_2 \rightarrow \min., \quad \|w\|_2 = 1, \tag{B.2}$$

with a Loewner matrix

$$L = \begin{bmatrix} \frac{F_{1,1}^{p+1} - F_{1,1}^1}{\zeta_{p+1} - \zeta_1} & \dots & \frac{F_{1,1}^{p+1} - F_{1,1}^p}{\zeta_{p+1} - \zeta_p} \\ \vdots & & \vdots \\ \frac{F_{1,1}^g - F_{1,1}^1}{\zeta_g - \zeta_1} & \dots & \frac{F_{1,1}^g - F_{1,1}^p}{\zeta_g - \zeta_p} \\ \frac{F_{2,1}^{p+1} - F_{2,1}^1}{\zeta_{p+1} - \zeta_1} & \dots & \frac{F_{2,1}^{p+1} - F_{2,1}^p}{\zeta_{p+1} - \zeta_p} \\ \vdots & & \vdots \\ \frac{F_{2,1}^g - F_{2,1}^1}{\zeta_g - \zeta_1} & \dots & \frac{F_{2,1}^g - F_{2,1}^p}{\zeta_g - \zeta_p} \\ \vdots & & \vdots \\ \frac{F_{d,d}^{p+1} - F_{d,d}^1}{\zeta_{p+1} - \zeta_1} & \dots & \frac{F_{d,d}^{p+1} - F_{d,d}^p}{\zeta_{p+1} - \zeta_p} \\ \vdots & & \vdots \\ \frac{F_{d,d}^g - F_{d,d}^1}{\zeta_g - \zeta_1} & \dots & \frac{F_{d,d}^g - F_{d,d}^p}{\zeta_g - \zeta_p} \end{bmatrix}.$$

This means that any solution of (B.2) is a unit eigenvector of the smallest eigenvalue of L^*L , and one of them (at least) has complex conjugate entries associated with any pair of complex conjugate support points. To determine one of those, one can take any nonzero vector \tilde{w} from this eigenspace and set

$$w = \begin{cases} i \tilde{w} / \|\tilde{w}\|_2 & \text{if } \tilde{w} = -\overline{P\tilde{w}}, \\ (\tilde{w} + \overline{P\tilde{w}}) / \|\tilde{w} + \overline{P\tilde{w}}\|_2 & \text{if } \tilde{w} \neq -\overline{P\tilde{w}}, \end{cases}$$

where P denotes the permutation matrix that swaps the entries corresponding to complex conjugate support points. It is not difficult to see that this vector w is an element from the very same eigenspace with the desired property.

In our implementation we initialize the greedy iterative scheme with $p = 2$ support points $\zeta_1 = \rho$ and $\zeta_2 = -\rho$, and we terminate the iteration when all nonlinear residuals $\|F_k - r(\zeta_k)\|_F$ are below a tolerance of 10^{-4} in norm and the rational approximation has at least 4 admissible poles, since the algorithm described in Section B.1 requires the poles of the approximation (B.1). These are given as the $p - 1$ finite eigenvalues z of the $(p + 1) \times (p + 1)$ generalized eigenvalue problem

$$\begin{bmatrix} 0 & w_1 & w_2 & \dots & w_m \\ 1 & \zeta_1 & & & \\ 1 & & \zeta_2 & & \\ \vdots & & & \ddots & \\ 1 & & & & \zeta_m \end{bmatrix} = z \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}.$$

Take note that two further generalized eigenvalues of this problem are infinite *per se*.

B.2 Construction of the system matrix

In the following we present one possibility to assemble the system matrix A for a given set of exponents λ_j and associated coefficient matrices Γ_j of the real-valued Prony series φ of (2.18). We start with the real Jordan canonical matrix

$$J = I \otimes \begin{bmatrix} \lambda_1 & & & & & & & \\ & \lambda_2 & & & & & & \\ & & \ddots & & & & & \\ & & & \lambda_q & & & & \\ & & & & \text{Re } \lambda_{q+1} & \text{Im } \lambda_{q+1} & & \\ & & & & -\text{Im } \lambda_{q+1} & \text{Re } \lambda_{q+1} & & \\ & & & & & & \ddots & \end{bmatrix}, \tag{B.3}$$

written as a Kronecker tensor product with a $d \times d$ identity matrix I . The leading part of the second factor in (B.3) consists of the q , say, real-valued exponents, followed by 2×2 rotation matrices for the complex conjugate pairs, where $\lambda_{q+2} = \overline{\lambda_{q+1}}$, and so on. Furthermore let

$$\Gamma_j = \sum_{i=1}^d \sigma_{(j-1)d+i} u_{(j-1)d+i} v_{(j-1)d+i}^* \tag{B.4}$$

be the singular value decompositions of the coefficient matrices Γ_j , $j = 1, \dots, p$. Then the real-valued matrices

$$U = \left[\sqrt{\sigma_1} u_1 \quad \dots \quad \sqrt{\sigma_{dq}} u_{dq} \quad \sqrt{2\sigma_{dq+1}} \text{Re } u_{dq+1} \quad \sqrt{2\sigma_{dq+1}} \text{Im } u_{dq+1} \quad \dots \right]^T$$

and

$$V = \left[\sqrt{\sigma_1} v_1 \quad \dots \quad \sqrt{\sigma_{dq}} v_{dq} \quad \sqrt{2\sigma_{dq+1}} \operatorname{Re} v_{dq+1} \quad -\sqrt{2\sigma_{dq+1}} \operatorname{Im} v_{dq+1} \quad \dots \right]^T,$$

defined in accordance with (B.3), satisfy

$$\varphi(t) = U^T e^{tJ} V, \quad t \geq 0. \quad (\text{B.5})$$

Furthermore, since $U^T V = \varphi(0) = I$ by virtue of (2.18) and (3.12), there exist nonsingular real-valued matrices X with

$$XV = E \quad \text{and} \quad X^T E = U, \quad (\text{B.6})$$

and hence, the desired representation (3.13) is achieved for

$$A = XJX^{-1}. \quad (\text{B.7})$$

A construction of such an appropriate basis transformation matrix X is presented in Appendix D.

Note that the final system matrix has dimension $dp \times dp$, unless some of the singular values σ_i in (B.4) happen to be zero. In such a rare case we eliminate the corresponding rows from U and V and the associated diagonal entries (or 2×2 block entries) from J .

Appendix C. Solving the Lur'e equations numerically

In the regular case, where $R = D + D^T$ is positive definite, solving (3.15) amounts to solving the algebraic Riccati equation

$$P\Sigma_0 + \Sigma_0 P^T + \Sigma_0 B R^{-1} B^T \Sigma_0 + C R^{-1} C^T = 0 \quad (\text{C.1})$$

for (the symmetric matrix) Σ_0 , where we have set

$$P = A_0 - C R^{-1} B^T$$

for brevity. We refer to Bini et al. [49] for an overview of possible numerical algorithms for solving (C.1); in our code we have used the algorithm suggested by Laub [50]. The left-hand side of (C.1) can be rewritten as

$$A_0 \Sigma_0 + \Sigma_0 A_0^T + (C - \Sigma_0 B) R^{-1} (C - \Sigma_0 B)^T.$$

Therefore, if we take K to be the Cholesky factor of R and let

$$L = (C - \Sigma_0 B) K^{-T},$$

then it follows that (C.1) is equivalent to

$$A_0 \Sigma_0 + \Sigma_0 A_0^T = -L L^T.$$

Accordingly, the (regular) Lur'e equation (3.15) are satisfied with this choice of Σ_0 , K , and L .

In the singular case, where $D = 0$, we let

$$S = B^T C \in \mathbb{R}^{d \times d},$$

which we assume to be symmetric and positive definite; compare (3.16a). As exemplified in Appendix D there exists a nonsingular transformation matrix $X \in \mathbb{R}^{N \times N}$, such that

$$XC = E_0 S^{1/2} \quad \text{and} \quad X^T E_0 = B S^{-1/2}, \quad (\text{C.2})$$

where $E_0 \in \mathbb{R}^{N \times d}$ is defined like E in (2.8). Then it follows from (C.2) that the solution Σ_0 of (3.14) satisfies

$$(X \Sigma_0 X^T) E_0 = X \Sigma_0 B S^{-1/2} = X C S^{-1/2} = E_0,$$

i.e., that

$$X \Sigma_0 X^T = \begin{bmatrix} I & 0 \\ 0 & \Sigma_1 \end{bmatrix} \quad (\text{C.3})$$

for some symmetric positive definite $(N-d) \times (N-d)$ matrix block Σ_1 . We further conclude from (3.14) that

$$\begin{aligned} (X A_0 X^{-1})(X \Sigma_0 X^T) + (X \Sigma_0 X^T)(X A_0 X^{-1})^T \\ = X(A_0 \Sigma_0 + \Sigma_0 A_0^T) X^T = -(X L)(X L)^T. \end{aligned} \quad (\text{C.4})$$

Decomposing the two matrices

$$X L = \begin{bmatrix} K_1 \\ L_1 \end{bmatrix} \quad \text{and} \quad X A_0 X^{-1} = \begin{bmatrix} -D_1 & B_1^T \\ -C_1 & A_1 \end{bmatrix} \quad (\text{C.5})$$

accordingly in d and $N-d$ rows and columns, it follows from (C.3) and (C.4) that

$$\begin{aligned} D_1 + D_1^T &= K_1 K_1^T, \quad A_1 \Sigma_1 + \Sigma_1 A_1^T = -L_1 L_1^T, \\ C_1 - \Sigma_1 B_1 &= L_1 K_1^T, \end{aligned} \quad (\text{C.6})$$

i.e., that Σ_1 , K_1 , and L_1 solve a *regular* Lur'e system. Since

$$\begin{aligned} D_1 &= -E_0^T (X A_0 X^{-1}) E_0 = -(E_0^T X) A_0 (X^{-1} E_0) \\ &= -S^{-1/2} B^T A_0 C S^{-1/2} \end{aligned}$$

by virtue of (C.2), $D_1 + D_1^T$ is positive definite when (3.16b) is assumed to hold. Therefore, under the given assumptions the regular Lur'e system (C.6) can be solved via an algebraic

Riccati equation as described above, and Σ_0 and L are then given by (C.3) and (C.5).

Appendix D. A useful basis transformation

For any $q > d$ let $E_q \in \mathbb{R}^{q \times d}$ be defined accordingly to E in (2.8). Furthermore, let $U, V \in \mathbb{R}^{q \times d}$ be such that

$$S = U^T V \quad (\text{D.1})$$

is symmetric positive definite. Choose a matrix $Q \in \mathbb{R}^{q \times (q-d)}$ whose columns form a basis of the orthogonal complement of the range of V ; such a basis can be obtained, for example, via a singular value decomposition of V . Then the matrix

$$X = \begin{bmatrix} S^{-1/2} U^T \\ Q^T \end{bmatrix} \in \mathbb{R}^{q \times q} \quad (\text{D.2})$$

is nonsingular, for if $Xv = 0$ for some $v \in \mathbb{R}^q$ then $Q^T v = 0$, i.e., $v = Vy$ for some $y \in \mathbb{R}^d$, and at the same time

$$0 = S^{-1/2} U^T v = S^{-1/2} U^T V y = S^{1/2} y$$

by virtue of (D.1). Since S is assumed to be positive definite, this implies that $y = 0$, and hence, $v = 0$.

Using (D.2) and (D.1) we immediately see that the matrix X provides a coordinate transformation with the two properties

$$XV = \begin{bmatrix} S^{-1/2} U^T V \\ 0 \end{bmatrix} = E_q S^{1/2} \quad (\text{D.3a})$$

and

$$X^T E_q = \begin{bmatrix} U S^{-1/2} & Q \end{bmatrix} E_q = U S^{-1/2}. \quad (\text{D.3b})$$

References

- [1] H. Risken, *The Fokker-Planck Equation*, Berlin, Heidelberg, New York, Springer, 1996.
- [2] G. A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, New York, Springer, 2014.
- [3] R. Zwanzig, *Nonequilibrium Statistical Mechanics*, New York, Oxford University Press, 2001.
- [4] V. Klippenstein, M. Tripathy, G. Jung, F. Schmid, and N. F. A. van der Vegt, "Introducing memory in coarse-grained molecular simulations," *J. Phys. Chem. B*, vol. 125, no. 19, pp. 4931–4954, 2021.
- [5] T. Schilling, "Coarse-grained modelling out of equilibrium," *Phys. Rep.*, vol. 972, pp. 1–45, 2022.
- [6] R. Zwanzig, "Memory effects in irreversible thermodynamics," *Phys. Rev.*, vol. 124, no. 4, pp. 983–992, 1961.
- [7] H. Mori, "Transport, collective motion, and Brownian motion," *Prog. Theor. Phys.*, vol. 33, no. 3, pp. 423–455, 1965.
- [8] D. Gordon, V. Krishnamurthy, and S.-H. Chung, "Generalized Langevin models of molecular dynamics simulations with applications to ion channels," *J. Chem. Phys.*, vol. 131, no. 13, 2009, Art. no. 134102.
- [9] I. Goychuk, "Viscoelastic subdiffusion: Generalized Langevin equation approach," *Adv. Chem. Phys.*, vol. 150, pp. 187–253, 2012.
- [10] H. S. Lee, S.-H. Ahn, and E. F. Darve, "The multi-dimensional generalized Langevin equation for conformational motion of proteins," *J. Chem. Phys.*, vol. 150, no. 17, 2019, Art. no. 174113.
- [11] R. R. Netz, "Derivation of the nonequilibrium generalized Langevin equation from a time-dependent many-body Hamiltonian," *Phys. Rev. E*, vol. 110, no. 1, 2024, Art. no. 014123.
- [12] A. Carof, R. Vuilleumier, and B. Rotenberg, "Two algorithms to compute projected correlation functions in molecular dynamics simulations," *J. Chem. Phys.*, vol. 140, no. 12, 2014, Art. no. 124103.
- [13] M. Chen, X. Li, and C. Liu, "Computation of the memory functions in the generalized Langevin models for collective dynamics of macromolecules," *J. Chem. Phys.*, vol. 141, no. 6, 2014, Art. no. 064112.
- [14] H. Lei, N. A. Baker, and X. Li, "Data-driven parameterization of the generalized Langevin equation," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 50, pp. 14183–14188, 2016.
- [15] H. K. Shin, C. Kim, P. Talkner, and E. K. Lee, "Brownian motion from molecular dynamics," *Chem. Phys.*, vol. 375, no. 2, pp. 316–326, 2010.
- [16] L. Ma, X. Li, and C. Liu, "The derivation and approximation of coarse-grained dynamics from Langevin simulations," *J. Chem. Phys.*, vol. 145, no. 20, 2016, Art. no. 204117.
- [17] L. Ma, X. Li, and C. Liu, "From generalized Langevin equations to Brownian dynamics and embedded Brownian dynamics," *J. Chem. Phys.*, vol. 145, no. 11, 2016, Art. no. 114102.
- [18] Z. Li, H. S. Lee, E. Darve, and G. E. Karniadakis, "Computing the non-Markovian coarse-grained interactions derived from the Mori–Zwanzig formalism in molecular systems: Application to polymer melts," *J. Chem. Phys.*, vol. 146, no. 1, 2017, Art. no. 014104.
- [19] G. Jung, M. Hanke, and F. Schmid, "Iterative reconstruction of memory kernels," *J. Chem. Theory Comput.*, vol. 13, no. 6, pp. 2481–2488, 2017.
- [20] H. Meyer, P. Pelagejcev, and T. Schilling, "Non-Markovian out-of-equilibrium dynamics: A general numerical procedure to construct time-dependent memory kernels for coarse-grained observables," *Europhys. Lett.*, vol. 128, no. 4, 2019, Art. no. 40001.
- [21] F. Grogan, H. Lei, X. Li, and N. A. Baker, "Data-driven molecular modeling with the generalized Langevin equation," *J. Comput. Phys.*, vol. 418, 2020, Art. no. 109633.
- [22] S. Wang, Z. Ma, and W. Pan, "Data-driven coarse-grained modeling of polymers in solution with structural and dynamic properties conserved," *Soft Matter*, vol. 16, no. 36, pp. 8330–8344, 2020.
- [23] M. Kerr Winter, I. Pihlajamaa, V. E. Debets, and L. M. C. Janssen, "A deep learning approach to the measurement of long-lived memory kernels from generalized Langevin dynamics," *J. Chem. Phys.*, vol. 158, no. 24, 2023, Art. no. 244115.
- [24] L. Tepper, B. Dalton, and R. R. Netz, "Accurate memory kernel extraction from discretized time-series data," *J. Chem. Theory Comput.*, vol. 20, no. 8, pp. 3061–3068, 2024.

- [25] V. Klippenstein, N. Wolf, and N. F. A. van der Vegt, “A Gauss-Newton method for iterative optimization of memory kernels for generalized Langevin thermostats in coarse-grained molecular dynamics simulations,” *J. Chem. Phys.*, vol. 160, no. 20, 2024, Art. no. 204115.
- [26] M. Ferrario and P. Grigolini, “A generalization of the Kubo–Freed relaxation theory,” *Chem. Phys. Lett.*, vol. 62, no. 1, pp. 100–106, 1979.
- [27] F. Marchesoni and P. Grigolini, “On the extension of the Kramers theory of chemical relaxation to the case of nonwhite noise,” *J. Chem. Phys.*, vol. 78, no. 10, pp. 6287–6298, 1983.
- [28] M. Ceriotti, G. Bussi, and M. Parrinello, “Langevin equation with colored noise for constant-temperature molecular dynamics simulations,” *Phys. Rev. Lett.*, vol. 102, no. 2, 2009, Art. no. 020601.
- [29] P. Siegle, I. Goychuk, P. Talkner, and P. Haenggi, “Markovian embedding of non-Markovian superdiffusion,” *Phys. Rev. E*, vol. 81, no. 1, 2010, Art. no. 011136.
- [30] N. Bockius, J. Shea, G. Jung, F. Schmid, and M. Hanke, “Model reduction techniques for the computation of extended Markov parameterizations for generalized Langevin equations,” *J. Phys.: Condens. Matter*, vol. 33, no. 21, 2021, Art. no. 214003.
- [31] G. Jung and F. Schmid, “Fluctuation-dissipation relations far from equilibrium: A case study,” *Soft Matter*, vol. 17, no. 26, pp. 6413–6425, 2021.
- [32] J. Shea, G. Jung, and F. Schmid, “Passive probe particle in an active bath: Can we tell it is out of equilibrium?,” *Soft Matter*, vol. 18, no. 36, pp. 6965–6973, 2022.
- [33] F. Glatzel and T. Schilling, “The interplay between memory and potentials of mean force: A discussion on the structure of equations of motion for coarse-grained observables,” *Europhys. Lett.*, vol. 136, no. 3, 2021, Art. no. 36001.
- [34] H. Vroylandt and P. Monmarché, “Position-dependent memory kernel in generalized Langevin equations: Theory and numerical estimation,” *J. Chem. Phys.*, vol. 156, no. 24, 2022, Art. no. 244105.
- [35] M. Ceriotti, G. Bussi, and M. Parrinello, “Colored-noise thermostats à la carte,” *J. Chem. Theory Comput.*, vol. 6, no. 4, pp. 1170–1180, 2010.
- [36] M. Hanke, “Mathematical analysis of some iterative methods for the reconstruction of memory kernels,” *ETNA*, vol. 54, pp. 483–498, 2021.
- [37] R. Kubo, “The fluctuation-dissipation theorem,” *Rep. Prog. Phys.*, vol. 29, no. 1, pp. 255–284, 1966.
- [38] J. A. Anderson, J. Glaser, and S. C. Glotzer, “Hoomd-blue: A python package for high-performance molecular dynamics and hard particle Monte Carlo simulations,” *Comput. Mater. Sci.*, vol. 173, 2020, Art. no. 109363.
- [39] R. Freund, “Computation of matrix-valued formally orthogonal polynomials and applications,” *J. Comput. Appl. Math.*, vol. 127, no. 1–2, pp. 173–199, 2001.
- [40] N. J. Higham, *Functions of Matrices: Theory and Computation, Algorithmic Differentiation*, Philadelphia, SIAM, 2008.
- [41] M. Hanke, “Stochastic modeling of stationary scalar Gaussian processes in continuous time from autocorrelation data,” *Adv. Comput. Math.*, vol. 50, no. 4, 2024, Art. no. 60.
- [42] A. Björck, *Numerical Methods for Least Squares Problems*, Philadelphia, SIAM, 2024.
- [43] B. D. O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis: A Modern Systems Theory Approach*, Englewood-Cliffs, NJ, Prentice-Hall, 1973.
- [44] Q. Wang, J. L. Speyer, and H. Weiss, “System characterization of positive real conditions,” in *29th IEEE Conference on Decision and Control*, Honolulu, USA, 1990, pp. 348–353.
- [45] R. W. Freund, “Computation of matrix Padé approximations of transfer functions via a Lanczos-type process,” in *Approximation Theory VIII*, Vol. I, C. K. Chi and L. L. Schumaker, Eds., Boston, Birkhäuser, 2001, pp. 215–222.
- [46] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, “The AAA algorithm for rational approximation,” *SIAM J. Sci. Comput.*, vol. 40, no. 3, pp. A1494–A1522, 2018.
- [47] I. V. Gosea and S. Güttel, “Algorithms for the rational approximation of matrix-valued functions,” *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. A3033–A3054, 2021.
- [48] P. Lietaert, K. Meerbergen, J. Pérez, and B. Vandereycken, “Automatic rational approximation and linearization of nonlinear eigenvalue problems,” *IMA J. Numer. Anal.*, vol. 42, no. 2, pp. 1087–1115, 2022.
- [49] D. A. Bini, B. Iannazzo, and B. Meini, *Numerical Solution of Algebraic Riccati Equations*, Philadelphia, SIAM, 2012.
- [50] J. Laub, “A Schur method for solving algebraic Riccati equations,” *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 913–921, 1979.