Soft Matter

PAPER

View Article Online
View Journal | View Issue

Cite this: *Soft Matter*, 2021, **17**, 6404

Data-driven coarse-grained modeling of non-equilibrium systems†

Shu Wang, ‡ Zhan Ma ‡ and Wenxiao Pan *

Received 17th March 2021,
Accepted 25th May 2021

DOI: 10.1039/d1sm00413a

rsc.li/soft-matter-journal

Modeling a high-dimensional Hamiltonian system in reduced dimensions with respect to coarse-grained (CG) variables can greatly reduce computational cost and enable efficient bottom-up prediction of main features of the system for many applications. However, it usually experiences significantly altered dynamics due to loss of degrees of freedom upon coarse-graining. To establish CG models that can faithfully preserve dynamics, previous efforts mainly focused on equilibrium systems. In contrast, various soft matter systems are known to be out of equilibrium. Therefore, the present work concerns non-equilibrium systems and enables accurate and efficient CG modeling that preserves non-equilibrium dynamics and is generally applicable to any non-equilibrium process and any observable of interest. To this end, the dynamic equation of a CG variable is built in the form of the non-stationary generalized Langevin equation (nsGLE), where the two-time memory kernel is determined from the data of the auto-correlation function of the observable of interest. By embedding the nsGLE in an extended dynamics framework, the nsGLE can be solved efficiently to predict the non-equilibrium dynamics of the CG variable. To prove and exploit the equivalence of the nsGLE and extended dynamics, the memory kernel is parameterized in a two-time exponential expansion. A data-driven hybrid optimization process is proposed for the parameterization, which integrates the differential-evolution method with the Levenberg–Marquardt algorithm to efficiently tackle a non-convex and high-dimensional optimization problem.

1 Introduction

When we study various soft matter systems, such as polymers, biomolecules, and colloidal suspensions, full atomistic simulations based on all-atom molecular dynamics (MD) can be computationally demanding because many degrees of freedom and high-dimensional Hamiltonian equations are involved, limiting the system size and time scale accessible in the simulations. If only certain main features of the system are of interest, we can focus on a small set of observables that characterize those features. For example, considering a system of biomolecules or proteins, one is interested in the collective motion of specific groups of atoms to understand a biological mechanism.[1,2] Therefore, coarse-grained (CG) modeling can be established, which projects a high-dimensional fine-grained system onto a smaller set of variables (CG variables) and constructs the equation of motion that governs the dynamics of CG variables. By solving a reduced-dimension equation, CG modeling is computationally more efficient than full atomistic simulations and hence allows accessing larger length scales and longer-time effects in practical applications. The CG dynamic equation can be established by systematically integrating out "irrelevant" degrees of freedom such as detailed information of the molecules and/or the solvent surrounding the molecules. Along this line, most previous studies concentrated on systems at equilibrium, where the generalized Langevin equation (GLE)[3,4] has been established to describe the equilibrium dynamics of CG variables and applied in various CG models.[5–11] However, many soft matter systems are known to be out of equilibrium, *e.g.*, molecular self-assembly driven by time-dependent temperature protocols,[12–15] dynamics of DNA under an applied force,[16] and polymer dynamics in a flow.[17] Therefore, it is of more interest in practice to be able to properly describe the non-equilibrium dynamics of CG variables. To this end, the non-stationary generalized Langevin equation (nsGLE)[18–20] has recently been established as a promising mathematical framework for CG modeling of non-equilibrium systems, which provides the equation of motion for the CG variables subject to non-equilibrium processes. In principle, the nsGLE can be rigorously derived using time-dependent projection operators.[18,21] It does not require time scale separation, *i.e.*, the CG variable can be any observable of interest, whether slow

Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA. E-mail: wpan9@wisc.edu

† Electronic supplementary information (ESI) available. See DOI: 10.1039/d1sm00413a

‡ These authors contributed equally to this work.

or fast relative to the dynamics of unresolved degrees of freedom. Similarly to the GLE, the nsGLE implicitly incorporates the kinetic effects of unresolved degrees of freedom through a memory term and a fluctuating force term. However, different from the GLE, the memory kernel in the nsGLE is a two-time function to account for the dependence on the initial conditions of non-equilibrium processes. Although theoretically sound, applying the nsGLE to practical CG modeling encounters the following challenges. First, determining the memory kernel from data can involve numerical differentiation, which needs to be regularized to ensure a stable and accurate solution. Second, since the non-stationary memory kernel is a two-time function, giving an explicit form of the fluctuating force such that the memory and fluctuating force obey a fluctuation–dissipation like relation[18,19] is a challenging task. Third, directly solving the nsGLE can be computationally expensive because the convolution of memory requires the historical information of CG variables at every time step, and the fluctuating term is difficult to compute, which overshadows the benefit of CG modeling.

The present work aims to address the challenges of applying the nsGLE to CG modeling of non-equilibrium systems, and our contributions include the following. First, we propose a numerical method that can efficiently and stably determine the memory kernel from the data of the auto-correlation function of the observable of interest. For prediction beyond the time range of the data set, the memory kernel is extrapolated *via* the Gaussian process regression. Second, by approximating the memory kernel in a two-time exponential expansion, we prove the equivalence of the nsGLE and an extended Markovian process. By embedding the nsGLE in an extended dynamics framework, we provide a specific form of the fluctuating force such that the memory and fluctuating force satisfy the fluctuation–dissipation relation, and in the meantime avoid the expensive convolution of memory, rendering the CG modeling computationally efficient. Third, to parameterize the exponential expansion approximating the memory kernel, we propose a systematic approach *via* data-driven optimization. In particular, we propose a hybrid optimization process that leverages the differential-evolution and Levenberg–Marquardt algorithms for non-convex and high-dimensional optimization. Finally, for validation, we apply the proposed methodology to a representative non-equilibrium system: a star-polymer melt in a heating process. Star-polymer melts have been used as typical benchmark systems for validating CG models.[5,7,22,23] The methodology proposed in this work is applicable to CG modeling of various non-equilibrium soft matter systems, for which only data accessible in either simulations or experiments for the reference Hamiltonian system are needed.

2 Non-stationary generalized Langevin equation

Without loss of generality, we denote a CG coordinate (mass-scaled) as $\hat{\mathbf{R}}(t)$ of d dimension and the corresponding momentum as $\hat{\mathbf{P}}(t)$. The nsGLE can then be written as eqn (1) and holds for any reference time τ:[18,19]

$$\frac{\mathrm{d}\hat{\mathbf{P}}(t)}{\mathrm{d}t} = \langle \hat{\mathbf{F}}(t) \rangle - \int_{\tau}^{t} \mathrm{d}t'' \hat{K}(t'', t) \hat{\mathbf{P}}(t'') + \hat{\tilde{\mathbf{F}}}_{\tau}(\tau, t), \tag{1}$$

where the three terms in the right-hand side correspond to the mean force, a friction term, and a fluctuating force term, respectively. Note that the friction term is associated with a two-time memory kernel $\hat{K}(t'',t)$, which accounts for the dependence of non-equilibrium dynamics on the initial conditions.[18] In the original nsGLE,[18–20] $\langle \cdot \rangle$ refers to an average over ensembles given by a distribution of initial states in the phase space. In this work, we interpret eqn (1) as a stochastic process considering unknown initial states, which is an assumption that our work draws on. If we choose the normalized momentum $\mathbf{P}(t) = \dfrac{\hat{\mathbf{P}}(t) - \langle \hat{\mathbf{P}}(t) \rangle}{\sqrt{\langle |\hat{\mathbf{P}}(t)|^2 \rangle - \langle \hat{\mathbf{P}}(t) \rangle^2}}$ as the CG variable, the nsGLE can be simplified as follows:

$$\frac{\mathrm{d}\mathbf{P}(t)}{\mathrm{d}t} = -\int_{\tau}^{t} \mathrm{d}t'' K(t'', t) \mathbf{P}(t'') + \tilde{\mathbf{F}}_{\tau}(\tau, t). \tag{2}$$

Here, note that $\tilde{\mathbf{F}}_{\tau}(\tau,t)$ depends on the reference time τ. In practice, we need to specify a reference time to solve the nsGLE. Without loss of generality, we define $\tau = 0$ in this paper and hence obtain

$$\frac{\mathrm{d}\mathbf{P}(t)}{\mathrm{d}t} = -\int_{0}^{t} \mathrm{d}t'' K(t'', t) \mathbf{P}(t'') + \tilde{\mathbf{F}}(t), \tag{3}$$

where $\tilde{\mathbf{F}}(t) = \tilde{\mathbf{F}}_0(0,t)$ is the fluctuating term. The memory kernel and fluctuating force satisfy

$$K(t'', t) = \frac{\langle \tilde{\mathbf{F}}(t'') \cdot \tilde{\mathbf{F}}(t) \rangle}{\langle |\mathbf{P}(t'')|^2 \rangle}, \tag{4}$$

which is the fluctuation–dissipation relation that holds for non-stationary processes.[19] In this paper, we neglect any correlations between CG particles and between different dimensions of the CG variable, and hence the memory kernel is regarded as a scalar function.

2.1 Determination of the memory kernel

To determine the two-time memory kernel $\hat{K}(t'',t)$, we rely on the following property of the fluctuating term for any given τ_1 and τ_2:[19]

$$\tilde{\mathbf{F}}_{\tau_2}(\tau_2, t) = \tilde{\mathbf{F}}_{\tau_1}(\tau_1, t) + \int_{\tau_2}^{\tau_1} \mathrm{d}t'' K(t'', t) \mathbf{P}(t''), \tag{5}$$

which can be easily obtained by substituting $\tau = \tau_1$ and $\tau = \tau_2$ into eqn (2), respectively, and then subtracting the two resulting equations. Multiply both sides of eqn (3) by $\mathbf{P}(t')$ and simplify the resulting equation using eqn (5) with $\tau_2 = 0$ and $\tau_1 = t'$. Taking the trajectory-average and using the orthogonality of $\mathbf{P}(t')$ and $\tilde{\mathbf{F}}_{t'}(t',t)$ as $\langle \mathbf{P}(t') \cdot \tilde{\mathbf{F}}_{t'}(t',t) \rangle = 0$[18] finally lead to

$$-D(t', t) = \int_{t'}^{t} C(t', t'') K(t'', t) \mathrm{d}t'', \tag{6}$$

where $C(t',t) = \langle \mathbf{P}(t')\cdot\mathbf{P}(t)\rangle$ is the auto-correlation function of the momentum; $D(t',t) = \frac{\partial C(t',t)}{\partial t}$ is the force–momentum correlation function. The memory kernel $K(t',t)$ can then be determined by solving eqn (6) in discrete setting if we have the data of $C(t',t)$ and $D(t',t)$ at discrete time instances. In our work, since we have chosen the normalized momentum $\mathbf{P}$ as the CG variable to simplify the nsGLE, as noted in eqn (3), it is more convenient to determine $D(t',t)$ by numerical differentiation of $C(t',t)$ than to directly obtain $D(t',t)$ as the normalized force–momentum correlation function. Considering that the numerical differentiation may be sensitive to data noise and unstable, a regularization, *via* the Tikhonov regularization,[24,25] is enforced in this work, for which the regularization parameters can be determined using the quasi-optimality criterion,[26,27] as discussed in the ESI.† Once we obtain the data of $C(t',t)$ and $D(t',t)$ at discrete time instances, eqn (6) is discretized into the following linear system by applying the midpoint quadrature rule:[28]

$$\Delta t \mathbf{C}_n \mathbf{K}_n = -\mathbf{D}_n \tag{7}$$

for every discrete time $t_n = n\Delta t$, where $\mathbf{C}_n \in \mathbb{R}^{n\times n}$ with $\mathbf{C}_n^{i,j} = \frac{1}{2}(C(t'_i, t_{j-1}) + C(t'_i, t_j))$ for $i \le j$ (otherwise, $\mathbf{C}_n^{i,j} = 0$), $\mathbf{K}_n \in \mathbb{R}^n$ with $\mathbf{K}_n^i = K\left(t'_{i-\frac{1}{2}}, t_n\right)$, and $\mathbf{D}_n \in \mathbb{R}^n$ with $\mathbf{D}_n^i = D(t'_{i-1}, t_n)$. Given sufficiently smooth data of $\mathbf{C}_n$ and $\mathbf{D}_n$, eqn (7) can be solved stably. Note that the resulting linear system from discretizing the first-kind Volterra equation is not guaranteed to be well-conditioned. Upon larger data noise and condition numbers, the linear system in eqn (7) may need to be regularized, *e.g.*, *via* the Tikhonov regularization,[24,25] to ensure stable solutions.

In the literature, the non-stationary memory kernel is computed using a different approach, where eqn (6) is first integrated in terms of t. Taking the derivative with respect to t' on the resulting equation leads to a Volterra equation of the second kind, which is then numerically solved by an iterative procedure[29] or *via* matrix inversion.[30] The solution obtained is finally converted into the memory kernel by taking the derivative with respect to t. The entire process involves numerical differentiation twice, for which the data are smoothed using the Bézier surface method.[30] Our method directly solves eqn (6), requires numerical differentiation once to obtain $D(t',t)$ from $C(t',t)$, and uses the Tikhonov regularization[24,25] to stabilize the numerical differentiation and possibly an ill-conditioned linear system.

Using the method described above along with the data of $C(t',t)$ for $0 \le t' \le t \le t_{\text{data}}$, we can determine the memory kernel $K(t',t)$ up to t_{data}. In the case that we need to forecast the dynamics beyond the range of data, we can employ the Gaussian process regression (GPR) to extrapolate the memory kernel $K(t',t)$ up to t_{pred}, denoted as $K_{\text{pred}}(t',t)$. The furthest extrapolation time t_{pred} depends on the standard deviation (uncertainty level) of GPR. In particular, define $\mathbf{x} = [t',t]^T \in \mathbb{R}^2$ and $\mathbf{x}^\star = [t'^\star, t^\star]^T \in \mathbb{R}^2$ as the inputs for training and prediction, respectively, with $0 \le t' \le t \le t_{\text{data}}$ and $0 \le t'^\star \le t^\star \le t_{\text{pred}}$. The Gaussian process model is given by $K(t',t) \sim \mathcal{GP}[\mathbf{m}(\mathbf{x}),\mathbf{\Sigma}(\mathbf{x},\mathbf{x}^\star)]$ with $\mathbf{m}(\mathbf{x})$ the mean function and $\mathbf{\Sigma}(\mathbf{x},\mathbf{x}^\star)$ the covariance function. Here, the covariance function assumes a squared exponential form with the hyper-parameters determined by minimizing the negative log marginal likelihood[31] *via* the Quasi-Newton optimizer L-BFGS.[32] A key advantage of GPR is that the uncertainty bounds of prediction can be derived from the hyper-parameters, and hence a measure of the uncertainty at $t^\star \ge t_{\text{data}}$ can be defined as

$$\sigma^*(t^*) = \frac{\sum_{t'^* < t^*} |\hat{\sigma}(K_{\text{pred}}(t'^*, t^*))|^2}{\sum_{t'^* < t^*} |K_{\text{pred}}(t'^*, t^*)|^2}, \tag{8}$$

with $\hat{\sigma}(K_{\text{pred}}(t'^\star,t^\star))$ the standard deviation of GPR at $t^\star > t_{\text{data}}$ for all $(t'^\star \le t^\star)$. From it we can determine the furthest extrapolation time t_{pred} by: $\sigma^\star(t_{\text{pred}}) \le \zeta_{\text{GPR}}$, with ζ_{GPR} the desired tolerance of uncertainty in GPR.

2.2 Extended dynamics

After the memory kernel $K(t',t)$ is constructed, the next task is to specify the fluctuating force in eqn (3), and we thus obtain an effective CG description in the form of nsGLE. The resulting nsGLE can then be solved to predict the non-equilibrium dynamics of the process. However, determining the fluctuating force from the fluctuation–dissipation relation in eqn (4) is difficult. And directly solving eqn (3) requires storing historical information and is computationally expensive. To circumvent these difficulties, we first approximate the memory kernel $K(t',t)$ by an exponential expansion, noting that the memory is usually a decaying function generally with oscillations:

$$K(t',t) \approx \sum_{i=1}^{N} \alpha_i(t)\alpha_i(t') \exp\left(-\frac{a_i}{2}(t-t')\right) \times [b_i \cos(q_i(t-t')) + c_i \sin(q_i(t-t'))], \tag{9}$$

where a_i, b_i, c_i and q_i are the parameters to be determined, satisfying $a_i \ge 0$, $b_i \ge 0$ and $|c_i| \le \frac{a_i b_i}{2q_i}$; and according to desired accuracy, N can be truncated to a finite number. Rewrite eqn (9) into a matrix form:

$$K(t',t) = -\mathbf{A}_{\text{ps}}\alpha(t)\mathrm{e}^{-(t-t')\mathbf{A}_{\text{ss}}}\alpha(t')\mathbf{A}_{\text{sp}}. \tag{10}$$

Here, $\alpha(t) \in \mathbb{R}^{2N\times 2N}$ is a time-dependent parameter matrix composed of $\alpha_i(t)$; $\mathbf{A}_{\text{ps}} \in \mathbb{R}^{1\times 2N}$, $\mathbf{A}_{\text{sp}} \in \mathbb{R}^{2N\times 1}$, and $\mathbf{A}_{\text{ss}} \in \mathbb{R}^{2N\times 2N}$ are parameter matrices whose elements are comprised of the parameters: a_i, b_i, c_i and q_i in eqn (9). The specific forms of the parameter matrices are provided in the ESI.†

Given eqn (10), the following extended dynamics

$$\begin{pmatrix}\dot{P}_k(t)\\ \dot{\mathbf{S}}_k(t)\end{pmatrix} = -\begin{pmatrix}\mathbf{0} & \mathbf{A}_{\mathrm{ps}}\alpha(t)\\ \alpha(t)\mathbf{A}_{\mathrm{sp}} & \mathbf{A}_{\mathrm{ss}}\end{pmatrix}\begin{pmatrix}P_k(t)\\ \mathbf{S}_k(t)\end{pmatrix} + \begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}_{\mathrm{s}}\end{pmatrix}\begin{pmatrix}0\\ \xi(t)\end{pmatrix}, \tag{11}$$

is equivalent to the nsGLE in eqn (3) with the fluctuating force given by

$$\tilde{F}_k(t) = -\int_0^t \mathbf{A}_{\mathrm{ps}}\alpha(t)\mathrm{e}^{-(t-t')\mathbf{A}_{\mathrm{ss}}}\mathbf{B}_{\mathrm{s}}\xi(t')\mathrm{d}t' - \mathbf{A}_{\mathrm{ps}}\alpha(t)\mathrm{e}^{-t\mathbf{A}_{\mathrm{ss}}}\mathbf{S}_k(0). \tag{12}$$

Here, eqn (11) and (12) are written for each dimension (P_k) of the CG variable $\mathbf{P}(t) \in \mathbb{R}^{d\times 1}$ with $k = 1,2,\ldots,d$; $\mathbf{S}_k \in \mathbb{R}^{2N\times 1}$ is an auxiliary variable vector, whose initial state $\mathbf{S}_k(0)$ satisfies $\langle \mathbf{S}_k(0)\rangle = \mathbf{0}$ and $\langle \mathbf{S}_k(0)\mathbf{S}_k^T(0)\rangle = \frac{\langle|\mathbf{P}(t')|^2\rangle}{d}\mathbf{I}$; $\boldsymbol{\xi} \in \mathbb{R}^{2N\times 1}$ is a vector of uncorrelated Gaussian random variables with $\langle\xi_i(t)\rangle = 0$ and $\langle\xi_i(t)\xi_j(0)\rangle = \delta_{ij}\delta(t)$, where ξ_i and ξ_j denote the different elements of $\boldsymbol{\xi}$; and the matrix $\mathbf{B}_{\mathrm{s}} \in \mathbb{R}^{2N\times 2N}$ satisfies

$$\mathbf{B}_{\mathrm{s}}\mathbf{B}_{\mathrm{s}}^T = \frac{\langle|\mathbf{P}(t')|^2\rangle}{d}\left(\mathbf{A}_{\mathrm{ss}} + \mathbf{A}_{\mathrm{ss}}^T\right). \tag{13}$$

The fluctuating force given in eqn (12) satisfies the fluctuation–dissipation relation in eqn (4), as explained in the ESI,† where a detailed proof of the equivalence between eqn (11) and (3) is also provided. The nsGLE is thereby converted to the extended dynamics that avoids convolution of memory and hence is much cheaper to solve than the original nsGLE. A similar idea was proposed for solving the GLE in the context of equilibrium dynamics.[7,9,10,33,34] For non-equilibrium processes, Stella *et al.* has proposed a Markovian embedding for the nsGLE by approximating the memory kernel as a two-time exponential expansion.[35] We note two distinctions between their work and ours: (1) our expansion includes both sin and cos oscillators to represent more general oscillating behaviors of the memory kernel; (2) while the functions $\alpha(t)$ and $\alpha(t')$ can be analytically derived from the atomistic Hamiltonian for the particular problem considered by Stella *et al.*,[35] they are unknown in general and need to be determined, *e.g.*, *via* parameterization as proposed in the following section.

2.3 Parameterize the memory kernel

To determine the parameters in the expansion (eqn (9)) approximating the memory kernel, we propose a data-driven optimization process. Rewrite eqn (9) as $K(t',t) = \sum_{i=1}^{N}\alpha_i(t)\alpha_i(t')\beta_i(t-t')$ with $\beta_i((t-t')) = \exp\left(-\frac{a_i}{2}(t-t')\right)[b_i\cos(q_i(t-t')) + c_i\sin(q_i(t-t'))]$, and $\{a_i,b_i,c_i,q_i\} \in \boldsymbol{\lambda}_\beta$. Without prior knowledge, α_i can be approximated by a general polynomial as follows: $\alpha_i(t) = \sum_{j=0}^{M} p_{ij}t^j$, where M is the polynomial order, and $\{p_{ij}\} \in \boldsymbol{\lambda}_\alpha$ are the coefficients. To determine the parameters $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_\alpha,\boldsymbol{\lambda}_\beta\}$, we solve the following optimization problem:

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \Pi(\boldsymbol{\lambda}), \tag{14}$$

where $\Pi(\boldsymbol{\lambda})$ is the objective function and is defined as $\Pi(\boldsymbol{\lambda}) = \frac{\left\|K(t',t,\boldsymbol{\lambda}) - K_{\mathrm{pred}}(t',t)\right\|_1}{\left\|K_{\mathrm{pred}}(t',t)\right\|_1}$ with $\|\cdot\|_1$ the L_1 norm. Since $\Pi(\boldsymbol{\lambda})$ is generally non-convex, the optimization could easily fall into some local minima. Using different random initial guesses can alleviate this issue to some degree; however, it requires the numbers of initial guesses and independent optimization processes to increase exponentially with the dimension of $\boldsymbol{\lambda}$. Even for a moderately high-dimensional parameter space, the computational cost can be prohibitive. Thus, we employ a hybrid optimization process leveraging the differential evolution (DE)[36] and Levenberg–Marquardt (LM)[37,38] algorithms. The hybrid method is computationally efficient and can effectively avoid falling into local minima. In particular, the DE is used to narrow down appropriate initial guesses of $\boldsymbol{\lambda}$, and the LM is then used to find the nearby minimum.

Before the optimization process, we first determine the polynomial order M in the approximation of $\alpha_i(t)$. Letting $t' = t$ in eqn (9), we obtain $K(t,t) = \sum_{i=1}^{N} b_i\alpha_i^2(t)$. Since $K(t,t)$ is proportional to $\alpha_i^2(t)$, we can approximate $K(t,t)$ by a $2M$-order polynomial. Noting that any continuous function can be approximated by an expansion of Legendre polynomials,[39] we next expand $K(t,t)$ with increasing order of Legendre polynomials. To proceed, we map $t \in [0,t_{\mathrm{pred}}]$ to $\bar{t} \in [-1,1]$ by $\bar{t} = \frac{2t - t_{\mathrm{pred}}}{t_{\mathrm{pred}}}$ since Legendre polynomials are defined on the interval $[-1,1]$. $K(\bar{t},\bar{t})$ can then be expanded as

$$K(\bar{t},\bar{t}) = \sum_{k=0}^{2M}\gamma_k L_k(\bar{t}) + \sum_{k=2M+1}^{\infty}\gamma_k L_k(\bar{t}), \tag{15}$$

with $L_k(\bar{t})$ the kth-order Legendre polynomial basis. The coefficient γ_k can be determined from[39]

$$\gamma_k = \frac{2k+1}{2}\int_{-1}^{1} K(\bar{t},\bar{t})L_k(\bar{t})\mathrm{d}\bar{t}, \tag{16}$$

where the integral can be numerically evaluated by a quadrature rule, *e.g.*, the trapezoidal rule. Due to the orthonormality and completeness of the Legendre polynomials, the terms $\sum_{k=2M+1}^{\infty}\gamma_k L_k(\bar{t})$ can be neglected in eqn (15) given the tolerance of truncation error. Thereby, we can determine the polynomial order M in the approximation of $\alpha_i(t)$. After that, we proceed with the hybrid optimization process based on the DE and LM algorithms to solve the optimization problem in eqn (14).

The DE is a type of evolutionary algorithm for global optimization,[36] whose basic idea is to reduce the objective function by generating mutated vectors, usually called greedy search. Greedy search converges fast but can be trapped by some local minima. The DE overcomes this difficulty by simultaneously generating several vectors (called population),

where mutation is achieved by comparing two vectors and then adding their difference after being weighted to a third vector. The population and mutation ensure a thorough exploration over the searching space and avoid falling into a local minimum. Thus, the DE is potentially capable of solving the global optimization problems that are non-differentiable and have multiple local minima.[40] Its key specifics are provided below. First, the DE draws on a population of individuals, which are defined as the $\mathscr{D}$-dimensional parameter vectors $\boldsymbol{\lambda}_{g,k}$ for $k = 1,2,\ldots,\mathrm{NP}$, where $\mathscr{D} = N(M+5)$, NP is the population size, and g represents the generation. All individuals are randomly initialized as $\boldsymbol{\lambda}_{0,k}$ over the searching space. Unless stated otherwise, we assume a uniform probability distribution for all random samplings. Second, the following steps are iterated until the stopping criterion of optimization (based on the maximum number of iterations $g_{\max}$ or the desired tolerance) is met.

(1) Mutation: for each individual $\boldsymbol{\lambda}_{g,k}$, $k = 1,2,\ldots,\mathrm{NP}$, a mutant vector is generated according to

$$\mathbf{v}_k = \boldsymbol{\lambda}_{g,k1} + F_{\mathrm{M}}\cdot(\boldsymbol{\lambda}_{g,k2} - \boldsymbol{\lambda}_{g,k3}), \tag{17}$$

where $k_1,k_2,k_3 \in [1,\mathrm{NP}]$ are randomly chosen and distinct from each other; $F_{\mathrm{M}} > 0$ is a scaling factor that controls the mutation, which is called the mutation weight.

(2) Crossover: a trial vector $\mathbf{u}_k$ is created by

$$\mathbf{u}_k(l) = \begin{cases} \mathbf{v}_k(l), & \omega_l < \mathrm{CR} \\ \lambda_{g,k}(l), & \omega_l \geq \mathrm{CR} \end{cases}, \tag{18}$$

where $\omega_l \in [0,1]$ is a uniformly distributed random number for $l = 1,2,\ldots,\mathscr{D}$, and $\mathrm{CR} \in [0,1]$ is the preset crossover rate.

(3) Selection: to decide whether or not the trial vector $\mathbf{u}_k$ should become an individual of the next generation, it is compared with the original vector $\boldsymbol{\lambda}_{g,k}$ by a greedy selection:

$$\boldsymbol{\lambda}_{g+1,k} = \begin{cases} \mathbf{u}_k, & \Pi(\mathbf{u}_k) < \Pi(\lambda_{g,k}) \\ \lambda_{g,k}, & \Pi(\mathbf{u}_k) \geq \Pi(\lambda_{g,k}) \end{cases}, \tag{19}$$

where $\boldsymbol{\lambda}_{g+1,k}$ is the offspring of $\boldsymbol{\lambda}_{g,k}$ for the next generation. The values of the population size NP, mutation weight F_{M}, and crossover rate CR are set following Gämperle.[41]

Although the DE has demonstrated its superior performance, *e.g.* robustness and fast convergence, by numerical experiments for solving benchmark global optimization problems with dimensions up to 100,[42] its computational cost could enormously increase when refining the solution since it does not use the information of gradients. As a result, it is challenging for the DE to accurately determine the precise position of optimum with reasonable computational cost. Therefore, we propose a hybrid method combining the DE with a robust local optimization algorithm, the LM.[37] To proceed, we set a tolerance ζ_{DE} for the DE, which is larger than the desired tolerance ζ_{opt} for the entire optimization. When the minimum of the objective function for the current generation in the DE $\prod\left(\boldsymbol{\lambda}_g^*\right)$ is smaller than that of the last generation and also the preset tolerance ζ_{DE}, $\boldsymbol{\lambda}_g^*$ is taken as the starting point for the LM algorithm to search the nearby local minimum $\boldsymbol{\lambda}^*$. If $\Pi(\boldsymbol{\lambda}^*) \leq \zeta_{\mathrm{opt}}$, the optimization process is terminated, and we find the optimal parameters $\boldsymbol{\lambda}^* = \left\{\boldsymbol{\lambda}_\alpha^*, \boldsymbol{\lambda}_\beta^*\right\}$; otherwise, we return to the DE and proceed to the next generation. The detailed algorithm of the hybrid optimization is outlined in Algorithm 1.

Algorithm 1. Hybrid optimization

Require: Memory kernel data of $K_{\mathrm{pred}}(t',t)$ with $0 \leq t' \leq t \leq t_{\mathrm{pred}}$
Ensure: Optimized parameters $\boldsymbol{\lambda}^*$
for $N = 1,2,\ldots$ **do**
  Initialize NP random parameter vectors $\boldsymbol{\lambda}_{0,k}$ with $k = 1,2,\ldots,\mathrm{NP}$ and set $\zeta = \min\{\Pi(\boldsymbol{\lambda}_{0,1}),\Pi(\boldsymbol{\lambda}_{0,2}),\ldots,\Pi(\boldsymbol{\lambda}_{0,\mathrm{NP}})\}$
  for $g = 0,1,\ldots,g_{\max}$ **do**
    $\boldsymbol{\lambda}_g^* = \underset{k=1,2,\ldots,\mathrm{NP}}{\operatorname{argmin}} \{\Pi(\boldsymbol{\lambda}_{g,k})\}$
    if $\Pi\left(\boldsymbol{\lambda}_g^*\right) < \min(\zeta_{\mathrm{DE}},\zeta)$ **then**
      $\zeta = \Pi\left(\boldsymbol{\lambda}_g^*\right)$
      Search $\boldsymbol{\lambda}^*$ by the LM with $\boldsymbol{\lambda}_g^*$ as the starting point
      if $\Pi(\boldsymbol{\lambda}^*) < \zeta_{\mathrm{opt}}$ **then**
        Output $\boldsymbol{\lambda}^*$ and terminate all loops
      end if
    end if
    for $k = 1,2,\ldots,\mathrm{NP}$ **do**
      Generate a mutant vector $\mathbf{v}_k$ by eqn (17)
      Create a trial vector $\mathbf{u}_k$ by eqn (18)
      Make the next generation $\boldsymbol{\lambda}_{g+1,k}$ by eqn (19)
    end for
  end for
end for
retrun $\boldsymbol{\lambda}^*$

3 Numerical example

As a proof of principle, we applied the proposed methodology to a benchmark non-equilibrium problem: heating a star-polymer melt. The CG coordinate was chosen at the center of mass (COM) of a tagged star polymer immersed in the melt of other identical star polymers. The momentum (normalized) of the COM of the tagged star polymer was the CG variable. We analyzed the non-equilibrium dynamics of this tagged star polymer during the heating process and constructed the dynamic equation of the CG variable based on the nsGLE. The data of the reference high-dimension Hamiltonian system were obtained from MD simulations, and hence, while one set of MD simulation results were used to construct the CG dynamic equation, the other set was used for validating the CG modeling predictions. In particular, we collected the data of the two-time auto-correlation function of momentum $C(t',t) = \langle \mathbf{P}(t')\cdot\mathbf{P}(t)\rangle$ from MD simulations. The constructed nsGLE was solved *via* extended dynamics, whose predictions within or

outside the range of data set used to construct the CG model were compared with the MD simulation results. All of the simulations were performed using LAMMPS.[43]

The star-polymer melt consists of 1000 identical star polymers including the tagged one. In the MD simulations, each star polymer consists of 31 Lennard-Jones (LJ) beads, *i.e.* a core LJ bead and 10 identical arms with 3 LJ beads per arm. The core LJ bead and the LJ beads in each arm are connected by the finitely extensible non-linear elastic (FENE) bonds. The inter-atomic potential adopts the purely repulsive Weeks–Chandler–Andersen (WCA) potential given by

$$E_{\mathrm{WCA}}(r) = \begin{cases} 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} + \frac{1}{4}\right] & r \le r_{\mathrm{c}} \\ 0 & r > r_{\mathrm{c}} \end{cases}, \qquad (20)$$

where r is the distance between two LJ beads; $r_{\mathrm{c}} = 2^{1/6}\sigma$ is the cutoff distance. The FENE potential for the bonded interaction between connected LJ beads is

$$E_{\mathrm{FENE}}(r) = \begin{cases} -\frac{1}{2}k_{\mathrm{b}}r_0^2 \ln\left[1 - \left(\frac{r}{r_0}\right)^2\right] & r \le r_0 \\ \infty & r > r_0 \end{cases}, \qquad (21)$$

where $k_{\mathrm{b}} = 3000\varepsilon/\sigma^2$ is the spring constant, and $r_0 = 1.5\sigma$ is the maximum length of the FENE spring. The mass of all LJ beads was chosen to be unity. The simulation box was a periodic cubic box of length 33.8395σ. The reduced LJ units were used herein; *i.e.*, the units of mass, length, and energy are set to $m = 1$, $\sigma = 1$, and $\varepsilon = 1$, and the corresponding unit of time is $\sigma(m/\varepsilon)^{0.5} = 1$. The Nose–Hoover thermostat under the canonical ensemble (*NVT*) with the thermostat relaxation time 0.6 was employed with the time step $\Delta t = 0.001$. The MD simulation was first performed at $k_{\mathrm{B}}T = 1.0$ for 10^6 time steps to equilibrate the system and then continued with the temperature continuously increasing from $k_{\mathrm{B}}T = 1$ to $k_{\mathrm{B}}T = 2$ for 15 000 time steps from $t = 0.0$ to $t = 15.0$. With the temperature in the Nose–Hoover thermostat varying over time, the fine-grained system corresponds to a time-dependent Hamiltonian system. Note that the nsGLE in eqn (3) and the fluctuation–dissipation relation in eqn (4) are valid for CG modeling of time-dependent Hamiltonian systems.[19] From the MD simulations, 8000 independent trajectories were collected to obtain the data of $C(t',t)$ (see Fig. 1), from which the CG model was constructed. In addition, we collected 120 000 more independent trajectories to obtain the corresponding $C(t',t)$ as the reference for validating the CG modeling predictions.

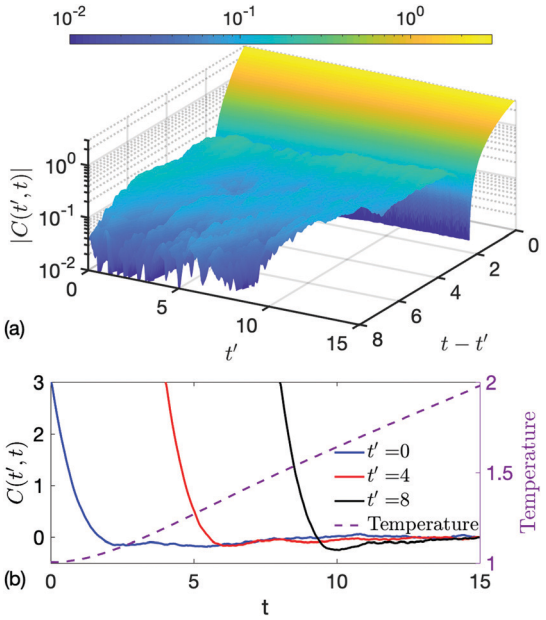

Fig. 1 Data of $C(t',t)$ obtained from the MD simulations. (a) 3-D view as a function of $t-t'$ and t', where the data are shown in absolute values. (b) Dependence of $C(t',t)$ on t at different t' (solid lines), along with the temperature ($k_{\mathrm{B}}T$) as a function of t (dashed line).

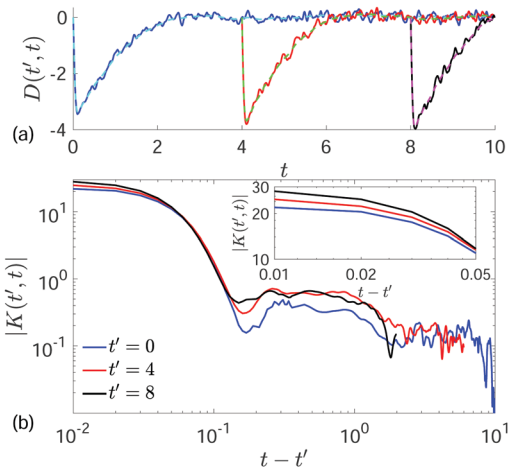

Fig. 2 (a) $D(t',t)$ obtained *via* direct numerical differentiation (solid lines) *vs. via* regularized numerical differentiation by the Tikhonov regularization (dashed lines). (b) The absolute value of $K(t',t)$ (on a logarithmic scale) computed from eqn (7) using the data of $C(t',t)$ and regularized $D(t',t)$. The zoom-in subplot provides a closer view of $|K(t',t)|$ for $0 \le t - t' \le 0.05$. All the results herein are plotted at $t' = 0$, $t' = 4$, and $t' = 8$.

3.1 Compute the memory kernel from data

To determine the memory kernel $K(t',t)$ from the data of $C(t',t)$, the force–momentum correlation function $D(t',t) = \dfrac{\partial C(t',t)}{\partial t}$ was first computed *via* numerical differentiation regularized by the Tikhonov regularization following the quasi-optimality principle (see the ESI†) with $\mu_0 = 0.1$ and $\eta = 0.99$. With the discrete data of $C(t',t)$ and $D(t',t)$ obtained, eqn (7) was directly solved for $K(t',t)$ up to t_{data}. Fig. 2 presents the resulting $D(t',t)$ and $K(t',t)$.

3.2 Extrapolate the memory kernel

The memory kernel $K(t',t)$ computed from the MD simulation data was then extrapolated by the GPR for longer time prediction beyond t_{data} until t_{pred}. For this numerical example, the following modified squared exponential function was

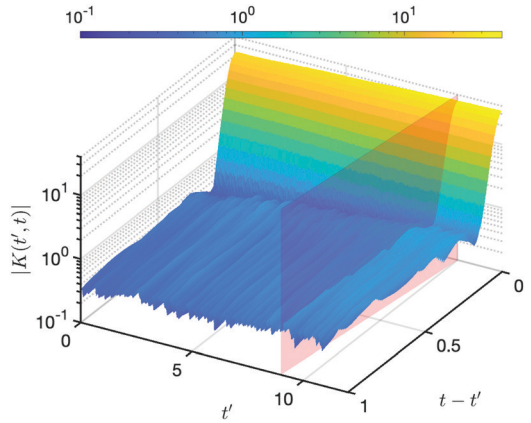

Fig. 3 Predicted memory kernel $K_{\text{pred}}(t',t)$ by the GPR model. Here, $K_{\text{pred}}(t',t)$ is divided into two parts by a red vertical flat surface: the left corresponds to the regression from the training data for $0 \le t' \le t \le t_{\text{data}} = 10$; the right corresponds to the extrapolation from the training data for $t_{\text{data}} < t' \le t \le t_{\text{pred}} = 12$.

employed as the covariance function:

$$\mathbf{\Sigma}(\mathbf{x},\mathbf{x}^*;\theta) = {\theta_{\text{f}}}^2 \exp\left[-\frac{(t-t^*)^2}{2{\theta_{l_1}}^2} - \frac{(t-t'-t^*+t'^*)^2}{2{\theta_{l_2}}^2}\right],$$

where $\theta = [\theta_{\text{f}},\theta_{l_1},\theta_{l_2}]$ are the hyper-parameters. The predicted memory kernel, K_{pred}, is shown in Fig. 3, where $t_{\text{data}} = 10$, $t_{\text{pred}} = 12$, and the tolerance of the GPR uncertainty was set to $\zeta_{\text{GPR}} = 0.008$. We further compared the GPR predictions with the test data and assessed the relative error:

$$\varepsilon_{\text{r}}(t^*) = \frac{\sum_{t'^* \le t^*} |K_{\text{pred}}(t'^*,t^*) - K(t'^*,t^*)|^2}{\sum_{t'^* \le t^*} |K(t'^*,t^*)|^2}$$

and the uncertainty level of GPR $\sigma^*(t^*)$ (defined in eqn (8)) at different prediction times $10 < t^* \le 15$. For that, $K_{\text{pred}}(t'^*,t^*)$ was predicted by GPR using the training data within $0 \le t' \le t \le 10$; the test data $K(t'^*,t^*)$ for $10 < t'^* \le t^* < 15$ was obtained by solving eqn (7) with the data of $C(t'^*,t^*)$. From the results depicted in Fig. 4, we can see that as the extrapolation time goes further, the uncertainty level $\sigma^*(t^*)$ and the relative error $\varepsilon_{\text{r}}(t^*)$ both increase with similar trends, which also supports the use of $\sigma^*(t^*)$ as a criterion to determine from data the furthest prediction time t_{pred}.

More investigations were performed for assessing the ability of GPR to extrapolate the memory kernel when using different sets of training data. Specifically, three sets of data within $0 \le t' \le t \le 2$ (Set 1), $4 \le t' \le t \le 6$ (Set 2), and $8 \le t' \le t \le 10$ (Set 3), respectively, were compared. Fig. 5 summarizes the computed uncertainty of GPR $\sigma^*(t^*)$ when using the three sets of training data for prediction of the memory kernel at different extrapolation times $t^* > t_{\text{data}}$. Although each set has the same amount of data, the prediction based on each exhibits different uncertainties at the same $t^* - t_{\text{data}}$. Using the data in earlier times resulted in a larger uncertainty in prediction. It can be due to the fact that the heating process considered in this work displayed nonlinear temperature rises in earlier times, which

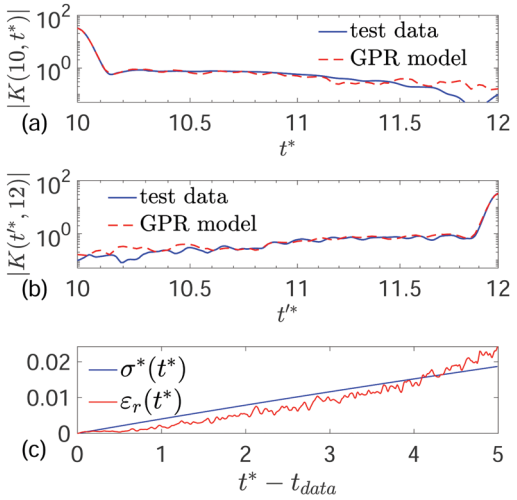

Fig. 4 (a and b) Comparison of the GPR model predictions with the test data at $t'^* = 10$ and $t^* = 12$, respectively. (c) The relative error $\varepsilon_{\text{r}}(t^*)$ and the uncertainty level $\sigma^*(t^*)$ of GPR at different extrapolation times $t^* > t_{\text{data}}$.

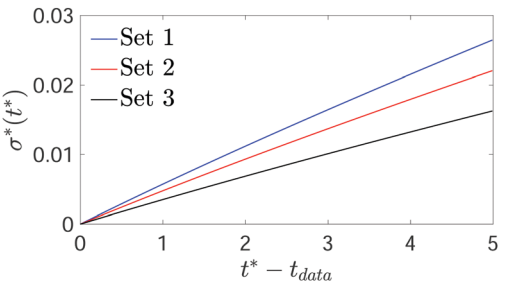

Fig. 5 Uncertainty of GPR $\sigma^*(t^*)$ in the prediction of the memory kernel for $t^* > t_{\text{data}}$ using different sets of training data.

led to more complex variations in the memory kernel as a function of time in earlier times than later. Thus, how far the memory kernel can be extrapolated depends on the choice of training data as well as the desired tolerance of uncertainty.

3.3 Parameterize the memory kernel

The parameterization of $K_{\text{pred}}(t',t)$ was achieved using the proposed hybrid optimization process, from which the approximation of $K_{\text{pred}}(t',t)$ *via* the expansion in eqn (9) was constructed with $M = 2$ and $N = 4$. The tolerance of optimization was set to $\zeta_{\text{opt}} = 0.1$, and the value of the objective function corresponding to the optimal parameters is $\Pi(\boldsymbol{\lambda}^*) = 0.092$. The values of other parameters involved in the hybrid optimization were set to NP = $10\mathscr{D}$, $F_{\text{M}} = 0.8$, CR = 0.5, $g_{\max} = 1 \times 10^6$, $\zeta_{\text{DE}} = 0.5$, and $\zeta_{\text{opt}} = 0.1$. The searching space of $\boldsymbol{\lambda}$ was set with: $p_{ij} \in [-1,+1]$, $a_i \in [0,100]$, $b_i \in [0,100]$, $q_i \in [0,100]$, and $c_i \in \left[-\frac{a_i b_i}{2q_i}, \frac{a_i b_i}{2q_i}\right]$, which is as large as physically reasonable.

The predicted and parameterized memory kernels are compared in Fig. 6. The parameterized memory kernel $K_{\text{param}}(t',t)$ with $N = 3$ is also shown. Note that the accuracy of approximating the memory kernel *via* the two-time exponential

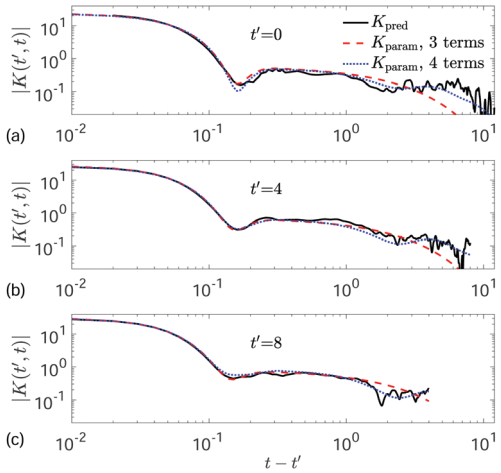

Fig. 6 The parameterized memory kernel $K_{\text{param}}(t',t)$ with N = 3 or N = 4, compared with the predicted memory kernel $K_{\text{pred}}(t',t)$ at different t' values.

expansion increases by truncating the expansion with more terms, *i.e.*, increasing N. Thus, the parameterization with larger N is more accurate.

3.4 CG modeling predictions

From the parameterized memory kernel and eqn (10), we assembled the matrices $\alpha(t)$, $\mathbf{A}$, and $\mathbf{B}_s$ for the extended dynamics. The CG modeling predictions until t_{pred} were obtained by solving eqn (11) using the implicit velocity-Verlet temporal integrator. The two-time auto-correlation function of momentum $C(t',t)$ predicted by the CG model was compared with the MD simulation results, as depicted in Fig. 7. We found good agreements. And the CG model constructed using the

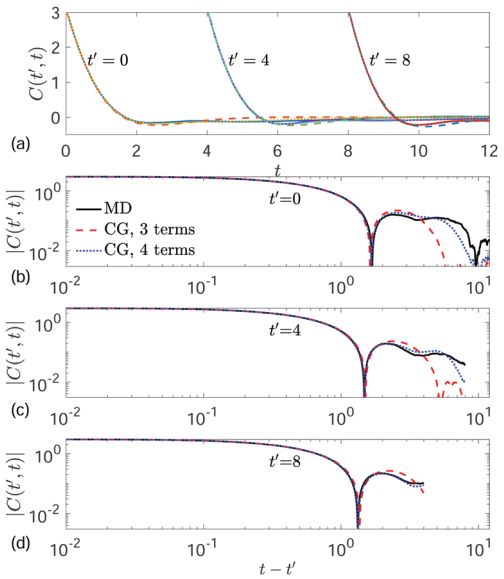

Fig. 7 $C(t',t)$ predicted by the CG model constructed from the parameterized memory kernel with N = 3 (dashed line) or N = 4 (dotted line), respectively, compared with the MD simulation results (solid line) at different t'. (a) $C(t',t)$ on a linear scale. (b–d) The absolute value of $C(t',t)$ on a logarithmic scale.

parameterized memory kernel with a larger N is more accurate. The relative error between the CG (with N = 4) prediction and the reference MD result is $\frac{\|C_{\text{CG}}(t',t)-C_{\text{MD}}(t',t)\|_1}{\|C_{\text{MD}}(t',t)\|_1}=0.03$, where $\|\cdot\|_1$ denotes the L_1 norm of discrete data; the data of $C_{\text{CG}}(t',t)$ and $C_{\text{MD}}(t',t)$ were attained from the CG and MD simulations, respectively.

4 Conclusions

We have presented a data-driven approach for constructing CG models of Hamiltonian systems in non-equilibrium dynamics. It goes beyond the existing literature in CG modeling, which mainly focuses on how to properly describe the equilibrium dynamics of CG variables. Our approach has addressed the key challenges in CG modeling of non-equilibrium systems, including how to efficiently and stably determine the non-stationary memory kernel, how to give a specific form of the fluctuating force such that the memory and fluctuating force satisfy the non-stationary fluctuation–dissipation relation, and how to efficiently solve the non-stationary dynamic equation of CG variables. Through the numerical example, we have demonstrated that the CG model can predict with desired accuracy the non-equilibrium dynamics of the observable of interest both inside and outside the regime of data used to construct the CG model. The approach only requires the data of the two-time auto-correlation function of non-equilibrium trajectory-averaged observable of interest, which can be readily obtained from simulations or experiments. We anticipate that the methodology proposed in this work can be generally applied to modeling high-dimensional non-equilibrium dynamics in reduced dimensions for various soft matter systems such as polymers, biomolecules, and colloids.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This material is based upon work supported by the Defense Established Program to Stimulate Competitive Research (DEPSCoR) Grant No. FA9550-20-1-0072 and by the National Science Foundation under Grant No. CMMI-1761068. The authors also thank the two anonymous reviewers for their insightful comments and suggestions that helped improve the manuscript.

References

1 P. C. Souza, S. Thallmair, P. Conflitti, C. Ramrez-Palacios, R. Alessandri, S. Raniolo, V. Limongelli and S. J. Marrink, *Nat. Commun.*, 2020, **11**, 1–11.
2 S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid and A. Kolinski, *Chem. Rev.*, 2016, **116**, 7898–7936.
3 H. Mori, *Prog. Theor. Phys.*, 1965, **33**, 423–455.

4 R. Zwanzig, *Phys. Rev.*, 1961, **124**, 983–992.
5 Z. Li, X. Bian, X. Li and G. E. Karniadakis, *J. Chem. Phys.*, 2015, **143**, 243128.
6 H. Lei, N. A. Baker and X. Li, *Proc. Natl. Acad. Sci. U. S. A.*, 2016, **113**, 14183–14188.
7 Z. Li, H. S. Lee, E. Darve and G. E. Karniadakis, *J. Chem. Phys.*, 2017, **146**, 014104.
8 G. Jung, M. Hanke and F. Schmid, *Soft Matter*, 2018, **14**, 9368–9382.
9 H. S. Lee, S.-H. Ahn and E. F. Darve, *J. Chem. Phys.*, 2019, **150**, 174113.
10 S. Wang, Z. Li and W. Pan, *Soft Matter*, 2019, **15**, 7567–7582.
11 S. Wang, Z. Ma and W. Pan, *Soft Matter*, 2020, **16**, 8330–8344.
12 W. M. Jacobs, A. Reinhardt and D. Frenkel, *Proc. Natl. Acad. Sci. U. S. A.*, 2015, **112**, 6313–6318.
13 M. Sajfutdinow, W. M. Jacobs, A. Reinhardt, C. Schneider and D. M. Smith, *Proc. Natl. Acad. Sci. U. S. A.*, 2018, **115**, E5877–E5886.
14 K. W. Tan, B. Jung, J. G. Werner, E. R. Rhoades, M. O. Thompson and U. Wiesner, *Science*, 2015, **349**, 54–58.
15 V. Krieger, E. Ciglia, R. Thoma, V. Vasylyeva, B. Frieg, N. de Sousa Amadeu, T. Kurz, C. Janiak, H. Gohlke and F. K. Hansen, *Chem. – Eur. J.*, 2017, **23**, 3699–3707.
16 S. R. Quake, H. Babcock and S. Chu, *Nature*, 1997, **388**, 151–154.
17 T. T. Perkins, D. E. Smith and S. Chu, *Science*, 1997, **276**, 2016–2021.
18 H. Meyer, T. Voigtmann and T. Schilling, *J. Chem. Phys.*, 2017, **147**, 214110.
19 H. Meyer, T. Voigtmann and T. Schilling, *J. Chem. Phys.*, 2019, **150**, 174118.
20 F. Glatzel and T. Schilling, *J. Chem. Phys.*, 2021, **154**, 174107.
21 H. Grabert, *Projection operator techniques in nonequilibrium statistical mechanics*, Springer, 1982.
22 P. Español and P. B. Warren, *J. Chem. Phys.*, 2017, **146**, 150901.
23 C. Hijón, P. Espanõl, E. Vanden-Eijnden and R. Delgado-Buscalioni, *Faraday Discuss.*, 2010, **144**, 301–322.
24 A. N. Tikhonov, *Sov. Math. Dokl.*, 1963, **4**, 1624–1627.
25 A. N. Tikhonov, *Sov. Math. Dokl.*, 1963, **4**, 1035–1038.
26 A. N. Tikhonov and V. B. Glasko, *USSR Comput. Math. Math. Phys.*, 1964, **4**, 236–247.
27 A. N. Tikhonov and V. B. Glasko, *USSR Comput. Math. Math. Phys.*, 1965, **5**, 93–107.
28 P. Linz, *Analytical and Numerical Methods for Volterra Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1985.
29 H. Meyer, P. Pelagejcev and T. Schilling, *Europhys. Lett.*, 2020, **128**, 40001.
30 H. Meyer, S. Wolf, G. Stock and T. Schilling, *Adv. Theory Simul.*, 2021, **4**, 2000197.
31 C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, MIT Press, Cambridge, MA, 2006, vol. 2.
32 D. C. Liu and J. Nocedal, *Math. Prog.*, 1989, **45**, 503–528.
33 M. Berkowitz, J. D. Morgan and J. A. McCammon, *J. Chem. Phys.*, 1983, **78**, 3256–3261.
34 M. Ceriotti, G. Bussi and M. Parrinello, *J. Chem. Theory Comput.*, 2010, **6**, 1170–1180.
35 L. Stella, C. D. Lorenz and L. Kantorovich, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2014, **89**, 134303.
36 R. Storn and K. Price, *J. Global Optim.*, 1997, **11**, 341–359.
37 K. Levenberg, *Quart. Appl. Math.*, 1944, **2**, 164–168.
38 C. Kanzow, N. Yamashita and M. Fukushima, *J. Comput. Appl. Math.*, 2004, **172**, 375–397.
39 M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, US Government printing office, 1970, vol. 55.
40 K. V. Price, *New Ideas in Optimization*, McGraw-Hill, London, 1999, pp. 77–106.
41 R. Gämperle, S. D. Müller and P. Koumoutsakos, *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, 2002, pp. 293–298.
42 J. Vesterstrom and R. Thomsen, *Proceedings of the 2004 Congress on Evolutionary Computation*, 2004, pp. 1980–1987.
43 S. Plimpton, *J. Comput. Phys.*, 1995, **117**, 1–19.

