



TITLE:

Learning the constitutive relation of polymeric flows with memory

AUTHOR(S):

Seryo, Naoki; Sato, Takeshi; Molina, John J.;
Taniguchi, Takashi

CITATION:

Seryo, Naoki ...[et al]. Learning the constitutive relation of polymeric flows with memory. *Physical Review Research* 2020, 2(3): 033107.

ISSUE DATE:

2020-07-21

URL:


<http://hdl.handle.net/2433/263824>

RIGHT:

© 2020 The American Physiological Society; This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Learning the constitutive relation of polymeric flows with memory

Naoki Seryo, Takeshi Sato,* John J. Molina[†] and Takashi Taniguchi[‡]
Department of Chemical Engineering, Kyoto University, Kyoto 615-8510, Japan

 (Received 12 April 2020; accepted 26 June 2020; published 21 July 2020)

We develop a learning strategy to infer the constitutive relation for the stress of polymeric flows with memory. We make no assumptions regarding the functional form of the constitutive relations, except that they should be expressible in differential form as a function of the local stress- and strain-rate tensors. In particular, we use a Gaussian process regression to infer the constitutive relations from stress trajectories generated from small-scale (fixed strain-rate) microscopic polymer simulations. For simplicity, a Hookean dumbbell representation is used as a microscopic model, but the method itself can be generalized to incorporate more realistic descriptions. The learned constitutive relation is then used to perform macroscopic flow simulations, allowing us to update the stress distribution in the fluid in a manner that accounts for the microscopic polymer dynamics. The results using the learned constitutive relation are in excellent agreement with full multiscale simulations, which directly couple micro/macro degrees of freedom, as well as the exact analytical solution given by the Maxwell constitutive relation. We are able to fully capture the history dependence of the flow, as well as the elastic effects in the fluid. We expect the proposed learning/simulation approach to be used not only to study the dynamics of entangled polymer flows, but also for the complex dynamics of other soft-matter systems, which possess a similar hierarchy of length and timescales.

DOI: [10.1103/PhysRevResearch.2.033107](https://doi.org/10.1103/PhysRevResearch.2.033107)

I. INTRODUCTION

Polymeric materials are ubiquitous in our modern industrial societies, having transformed our food, infrastructure, and modes of transportation. Not surprisingly, the 20th century has been dubbed the “Polymer age” by Rubinstein and Colby [1]. There is a growing demand for producing more high-functioning polymeric products, and to do this in a cost-effective way. Unfortunately, there is still much we do not understand about polymer physics, particularly with regards to the fabrication method of sophisticated polymer products. At present, one of the preferred manufacturing methods for polymeric materials is polymer melt processing, where a molten polymer is extruded or molded into the desired shape, before allowing it to cool and solidify [2]. To accomplish these requirements, we need to understand not only the macroscopic flow behavior of the polymer process, but also the microscopic dynamics of the polymer chains, in order to reliably control the resultant properties of the product. However, it is not easy to understand such properties using only experimental observations, due to the hierarchy of length and timescales needed to characterize the microscopic polymer dynamics and the macroscopic flow. Computer simulations, which provide

an alternative and complimentary approach, have become an indispensable tool for studying such systems. Molecular dynamics (MD) simulations provide access to the detailed dynamics of polymer chains, but they are unable to deal with the macroscopic flow, because of the prohibitive cost. Computational fluid dynamics (CFD) simulations, with an appropriate constitutive equation for the stress tensor of the polymeric material [3], can predict the macroscopic flow behavior, but cannot provide any microscopic information on the constituent polymer chains. To address this issue and elucidate the microscopic origin of the flow problems at hand, multiscale simulation (MSS) methods, which make it possible to simultaneously consider the dynamics at both scales, have been extensively developed over the last twenty years. Within the MSS approach, the macroscopic and microscopic degrees of freedom are coupled through the stress and strain-rate tensor fields. Originally proposed by Laso and Öttinger in 1993 [4,5], with the so-called CONFFESSIT model, these types of approaches represent the state of art in polymer rheology, as they provide a rigorous connection between the microscopic and macroscopic degrees of freedom [6–16]. However, given the computational complexity, they have been limited to simple flow geometries and small system sizes, and have yet to be widely adopted within industry.

In this paper, we demonstrate how to leverage the power of machine learning techniques to accelerate the MSS to the point where they are competitive with existing macroscopic descriptions, without any significant loss of accuracy. In particular, we will show that it is possible to learn the constitutive relation from training data generated from small system size microscopic simulations. To this end, we adopt a simple microscopic description, which models the polymers as non-interacting Hookean dumbbells. We note that the proposed method is applicable to any microscopic polymer model, the

*Present address: Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan.

[†]Corresponding authors: john@cheme.kyoto-u.ac.jp

[‡]taniguch@cheme.kyoto-u.ac.jp

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Hookean dumbbell model has been chosen for its simplicity. As is well known, in the limit when the number of dumbbells goes to infinity, the time evolution equation for the stress of such an ensemble converges to an analytic expression. This makes it possible to give a stringent assessment of our proposed ML approach. We then perform simulations at fixed strain-rates, in order to measure the time evolution of the stress. This information is used as the training data for a Gaussian process (GP) regression, in order to learn the corresponding constitutive relation. A GP approach avoids over-fitting of the data and allows us to incorporate unknown and/or noisy data within a Bayesian framework, in a convenient and efficient manner [17–20]. The learned constitutive relations are then used in macroscopic flow simulations, opening the possibility of probing length and timescales that would be unreachable with standard MSS techniques. Previous work by Zhao *et al.* has used a similar approach to learn the constitutive relation of generalized Newtonian fluids [21], but as proposed, the method cannot be applied to non-Newtonian viscoelastic materials that exhibit a history dependent flow. A recent extension of this method has used GP to learn the effective viscosity and relaxation time needed to parametrize a given viscoelastic constitutive relation [22]. In both cases, however, the functional form of the constitutive equation is a fixed input of the method. Here we show that this restriction can be lifted, and that the constitutive relation itself can be learned.

Compared to full MSS (using Hookean dumbbells), we obtain speedups of around two orders of magnitude, and we expect this will only increase when more realistic (computationally expensive) polymer models are used, as the cost of performing the macroscopic flow simulations remains constant. Finally, we note that the proposed learning strategy, which is here used to learn the constitutive relation of polymeric flows from microscopic data, is not limited only to polymeric materials. In fact, we envision similar approaches being used to bridge between length and timescales in other soft-matter systems, such as colloidal dispersions or cellular tissues.

II. MULTISCALE SIMULATIONS

A. Macroscopic model

In order to consider the memory effects inherent to polymer flows we adopt a Lagrangian particle description to describe the dynamics of the fluid. This allows us to account for the convection of the polymer chains and the corresponding strain-rate history dependence on their dynamics. In particular, we will use the smooth particle hydrodynamics (SPH) method (see Appendix for details) [23]. The system is discretized into fluid particles, carrying mass, momentum, and all relevant hydrodynamic variable for the problem under consideration.

Let \mathbf{x}_i and \mathbf{v}_i be the position and velocity of particle i ; its time evolution is determined by the following equations:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i, \quad (1)$$

$$\frac{d\mathbf{v}_i}{dt} = \rho_i^{-1} [\nabla \cdot (\boldsymbol{\sigma} - p\mathbf{I})]_{\mathbf{x}_i} + \mathbf{F}(\mathbf{x}_i), \quad (2)$$

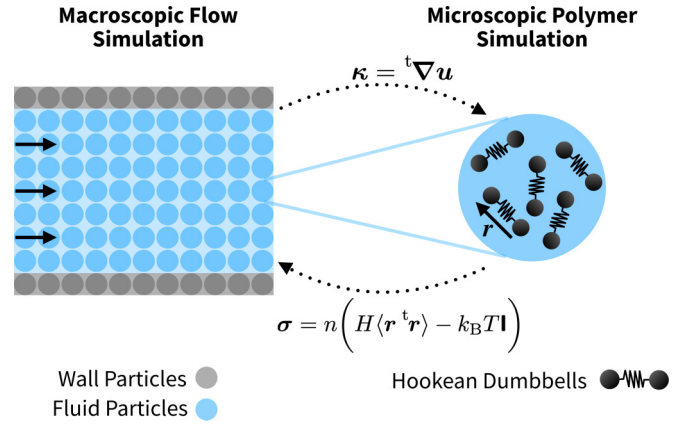


FIG. 1. Schematic representation of the MSS method used to study the micro/macro coupling of polymeric flows. The fluid is discretized into fluid particles carrying mass and momentum, as well as microscopic polymer simulators. Solving for the dynamics of the polymers, under the macroscopically obtained velocity gradient κ , allows us to compute the microscopic polymer contribution to the local stress $\boldsymbol{\sigma}$. The resultant stress distribution is then used to solve for the macroscopic flow dynamics.

where ρ_i is the density of the fluid particle and $\mathbf{F}(\mathbf{x}_i)$ is any external force acting on the particle (at position \mathbf{x}_i). The first term on the right-hand side of Eq. (2) corresponds to the forces on the particle due to internal stresses in the fluid (with p the pressure field). The stress $\boldsymbol{\sigma}$ comes from the time-dependent state of the polymer chains, i.e., the orientation and stretching of the dumbbells. The pressure p_i is defined via the following quasi-incompressible equation of state:

$$p_i = \frac{c_s^2 \rho_0}{\gamma} \left[\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right] \quad (3)$$

with c_s the speed of sound and ρ_0 the initial density. Since we are interested in low-Reynolds number flows, we use $\gamma = 1$ [24].

All that remains is to specify how $\boldsymbol{\sigma}$ is computed. The simplest approach would be to adopt a constitutive equation (e.g., Oldroyd-B), instead, within a MSS approach, we place microscopic polymer simulators inside each fluid particle, in order to directly couple the microscopic and macroscopic degrees of freedom [8,9,13–15]. The choice is then reduced to that of defining an appropriate microscopic model for the polymeric fluid.

B. Microscopic model

To describe the rheological properties of the polymeric fluid, We choose the simplest possible microscopic model, that of noninteracting Hookean dumbbells. Thus we place N_p polymer chains inside each of the N_f fluid particles, with each polymer chain represented by two point particles connected by a Hookean spring (Fig. 1). While more realistic microscopic models are known, such as the finite-extensible Hookean dumbbell model [25], the Rouse model [26], the Kremer-Grest beads-spring model [27], the Doi-Edwards reptation model [28], and the slip-link models [29–33], the basic Hookean dumbbell model offers one main advantage over the

others. In the limit when the number of dumbbells goes to infinity, the exact analytical constitutive equation for the stress is known and corresponds to that of a Maxwell viscoelastic fluid. This provides us with analytical results against which we can test our proposed learning strategy. However, the learning method we propose is not contingent on this choice, and in fact, will be most useful when considering more sophisticated polymer models, for which full MSS can become prohibitively expensive.

Since we do not consider interactions between dumbbells, the configuration of the system can be described solely in terms of the distance vector \mathbf{r} between the two beads composing the dumbbell. The dynamics of the chains are then determined by the following Langevin equation for \mathbf{r} [25,34]:

$$\frac{d\mathbf{r}}{dt} = \boldsymbol{\kappa} \cdot \mathbf{r} - \frac{2}{\zeta} H \mathbf{r} + \frac{1}{\zeta} \boldsymbol{\xi}, \quad (4)$$

$$\boldsymbol{\kappa} = {}^t \nabla \mathbf{v}, \quad (5)$$

where H is the spring constant, $\boldsymbol{\kappa}$ is the velocity gradient tensor, ζ is the friction coefficient, and $\boldsymbol{\xi}$ is a random force satisfying the fluctuation-dissipation theorem, $\langle \boldsymbol{\xi} \rangle = \mathbf{0}$ and $\langle \boldsymbol{\xi}(t) \boldsymbol{\xi}(t') \rangle = 4k_B T \zeta \mathbf{I} \delta(t - t')$, with \mathbf{I} the unit tensor. The polymer contribution to the stress tensor can be expressed in Kramers' form as [35]

$$\boldsymbol{\sigma} = n(H \langle \mathbf{r} \mathbf{r} \rangle - k_B T \mathbf{I}), \quad (6)$$

where n is the number density of dumbbells, k_B is the Boltzmann constant, and T is the temperature. From Eqs. (4)–(6), the following constitutive equation for the stress tensor can be derived:

$$\frac{d\boldsymbol{\sigma}}{dt} = \boldsymbol{\sigma} \cdot {}^t \boldsymbol{\kappa} + \boldsymbol{\kappa} \cdot {}^t \boldsymbol{\sigma} + nk_B T (\boldsymbol{\kappa} + {}^t \boldsymbol{\kappa}) - \frac{4H}{\zeta} \boldsymbol{\sigma}, \quad (7)$$

which corresponds to the upperconvected Maxwell model.

C. Nondimensionalized equations

To facilitate the simulation and analysis, we will rewrite the previous equations in nondimensional form, introducing basic units for both macroscopic and microscopic descriptions. These two sets are labeled with an uppercase (M) and lowercase (m) superscript, for macroscopic and microscopic units, respectively. At the macroscopic level, the characteristic scales are set by the length L and velocity U of the flow problem under consideration, and the corresponding fluid-advection time scale $\tau^{(M)} = L/U$. Together with the shear-viscosity of the fluid η_s , and the initial (target) density ρ_0 , we use these characteristic scales to set the units of density, time, length, and stress to be $\rho_0^{(M)} = \rho_0$, $t_0^{(M)} = \tau^{(M)}$, $l_0^{(M)} = L$, and $\sigma_0^{(M)} = \eta_s / \tau^{(M)} = \eta_s U / L$, respectively. With these units, the equations governing the macroscopic dynamics, Eqs. (1)–(3), become

$$\frac{d\tilde{\mathbf{x}}_i}{d\tilde{t}} = \tilde{\mathbf{v}}_i, \quad (8)$$

$$\text{Re} \frac{d\tilde{\mathbf{v}}_i}{d\tilde{t}} = \tilde{\rho}_i^{-1} [\tilde{\nabla} \cdot (\tilde{\boldsymbol{\sigma}} - \tilde{\rho} \mathbf{I})]_{\tilde{\mathbf{x}}_i} + \tilde{\mathbf{F}}(\tilde{\mathbf{x}}_i), \quad (9)$$

$$\tilde{\rho}_i = \text{Cs}^2 (\tilde{\rho} - 1), \quad (10)$$

where a tilde ($\tilde{\cdot}$) denotes an adimensional variable. Here, the control parameters are the Reynolds number, defined as $\text{Re} = \rho_0 U L / \eta_s = \rho_0 U^2 / \sigma_0^{(M)}$, and the dimensionless artificial sound speed $\text{Cs}^2 = c_s^2 \rho_0^{(M)} t_0^{(M)} / \eta_s$.

At the microscopic level, the characteristic scales are given by the equilibrium length of the dumbbells $l_{\text{eq}} = \sqrt{3k_B T / H}$, the dumbbell relaxation time $\tau^{(m)} = \lambda = \zeta / 4H$, and the shear viscosity of the Maxwell fluid $\eta_s \equiv nk_B T \lambda$. To facilitate comparisons between the microscopic and macroscopic models, we will use the same time and stress units for both, $t_0^{(m)} = t_0^{(M)}$ and $\sigma_0^{(m)} = \sigma_0^{(M)} = nk_B T \tau^{(m)} / \tau^{(M)}$, while the microscopic unit of length is taken to be the equilibrium dumbbell length $l_0^{(m)} = l_{\text{eq}}$. Thanks to the coupling between the macroscopic flow and the microscopic chain dynamics, the Deborah number, defined as the ratio of timescales associated to the microscopic dumbbell relaxation and macroscopic fluid advection, $\text{De} = \tau^{(m)} / \tau^{(M)}$, has appeared in the definition of the unit stress, $\sigma^{(m)} = \sigma^{(M)} = nk_B T \text{De}$. With these microscopic units, Eqs. (4)–(7) become

$$\frac{d\tilde{\mathbf{r}}}{d\tilde{t}} = \tilde{\boldsymbol{\kappa}} \cdot \tilde{\mathbf{r}} - \frac{1}{2\text{De}} \tilde{\mathbf{r}} + \sqrt{\frac{1}{3\text{De}}} \tilde{\boldsymbol{\xi}}, \quad (11)$$

$$\tilde{\boldsymbol{\sigma}} = \frac{3}{\text{De}} \left(\tilde{\mathbf{r}} \tilde{\mathbf{r}} - \frac{1}{3} \mathbf{I} \right), \quad (12)$$

$$\frac{d\tilde{\boldsymbol{\sigma}}}{d\tilde{t}} = \tilde{\boldsymbol{\sigma}} \cdot {}^t \tilde{\boldsymbol{\kappa}} + \tilde{\boldsymbol{\kappa}} \cdot {}^t \tilde{\boldsymbol{\sigma}} + \frac{1}{\text{De}} (\tilde{\boldsymbol{\kappa}} + {}^t \tilde{\boldsymbol{\kappa}}) - \frac{1}{\text{De}} \tilde{\boldsymbol{\sigma}}, \quad (13)$$

with the Deborah number De as the only control parameter. Note that $\tilde{\boldsymbol{\kappa}} = \boldsymbol{\kappa} t_0^{(m)}$ and $\tilde{\boldsymbol{\xi}} = \sqrt{t_0^{(m)}} / (4k_B T \zeta) \boldsymbol{\xi}$, with $\langle \tilde{\boldsymbol{\xi}} \rangle = \mathbf{0}$ and $\langle \tilde{\boldsymbol{\xi}}(\tilde{t}) \tilde{\boldsymbol{\xi}}(\tilde{t}') \rangle = \mathbf{I} \delta(\tilde{t} - \tilde{t}')$. Finally, we stress that, while we have chosen $\tau^{(M)}$ as unit of time, it can also be useful to use the dumbbell relaxation time $\tau^{(m)} \equiv \lambda$ as a reference. Scaled time values using both units are directly related by the Deborah number, with $\tilde{t} = t / \tau^{(M)} = (t / \lambda) \text{De}$.

In summary, the MSS method amounts to solving Eqs. (8)–(12). With the Reynolds number Re , the Deborah number De , and the dimension-less artificial sound speed Cs as the main control parameters. Unless stated otherwise, we focus on the Deborah number regime $10^{-3} \leq \text{De} \leq 10$, at low-Reynolds numbers $\text{Re} \lesssim 1$; with the sound-speed $\text{Cs} = 10$ chosen to keep the density variations within $\lesssim 3\%$ [24], while still allowing for a relatively large simulation time step $\Delta \tilde{t} \simeq 10^{-6}$. The strain-rate tensor is evaluated at the macroscopic level and used as input for the microscopic simulators, in order to evolve the configuration of the polymers (dumbbells). Then, the polymer contribution to the stress is computed and used as input to the macroscopic flow simulation, in order to update the fluid velocity, and the process is repeated (see Fig. 1). This method has been successfully used to study a polymer melt-spinning process [14], as well as flows of well-entangled polymer melts in a contraction-expansion channel (with the dumbbell model replaced by the Doi-Takimoto slip-link model) [15]. Unfortunately, while the predictive capabilities of such a bottom-up approach represent the current state-of-the-art in the field of polymer simulations, their heavy computational cost has mostly limited them to relatively simple flow-geometries in 2D. To address this issue, we propose a learning strategy based on Gaussian processes, in order to uncouple the microscopic and macroscopic degrees of freedom in the governing equations. However, our method remains multiscale, in the sense

that the microscopic model is used to generate the training data used to learn the appropriate constitutive equation, i.e., how to evolve σ . This will allow us to consider the time evolution of the stress at the macroscopic level, but in a way that satisfies the dynamics of the underlying microscopic polymer model.

III. LEARNING METHOD

A. Gaussian processes (GP)

Formally, a Gaussian process is defined as “a collection of random variables, any finite number of which have a joint Gaussian distribution” [19]. It provides a (prior) probability distribution over functions f , allowing us to use known values of the function, the so-called “training” data, to infer the values of the function at new “test” positions. Let $f(\mathbf{x})$ be a function from \mathbb{R}^D to \mathbb{R} , which is sampled at N values of the input \mathbf{x} , $\mathbf{x}_i \in \mathbb{R}^D$ ($i = 1, \dots, N$). We denote by $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ the $D \times N$ design matrix, and $\mathbf{f} = f(\mathbf{X}) = (f_1, \dots, f_N)$ the corresponding output matrix, with $f_i = f(\mathbf{x}_i)$. The f_i are considered to be correlated random variables, where the correlation between any two of them, f_i and f_j , is assumed to be a function only of the input values, \mathbf{x}_i and \mathbf{x}_j . The joint distribution for the f_i is a multivariate Gaussian, specified in terms of an average function $\mu(\mathbf{x})$ and a correlation function $k(\mathbf{x}, \mathbf{x}')$.

The probability of observing function values \mathbf{f} at \mathbf{X} , given μ and k , is

$$p(\mathbf{f}(\mathbf{X})|\mathbf{X}, \mu, k) = \frac{1}{\sqrt{(2\pi)^N \det K(\mathbf{X}, \mathbf{X})}} \times \exp\left[-\frac{1}{2} \delta \mathbf{f}(\mathbf{X}) \cdot K(\mathbf{X}, \mathbf{X})^{-1} \cdot \delta \mathbf{f}(\mathbf{X})\right], \quad (14)$$

where $\delta \mathbf{f} = \mathbf{f} - \mu$, and $K(\mathbf{X}, \mathbf{X})$ denotes the $N \times N$ correlation matrix, whose (i, j) th entry is defined as $K(\mathbf{A}, \mathbf{B})_{ij} = k(\mathbf{A}_i, \mathbf{B}_j)$. Note that a GP is uniquely defined in terms of its average and correlation functions,

$$\langle f(\mathbf{x}) \rangle = \mu(\mathbf{x}), \quad (15)$$

$$\langle \delta f(\mathbf{x}) \delta f(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}'). \quad (16)$$

Without loss of generality, and in the absence of any information about f , one can take $\mu(\mathbf{x}) = 0$, which leaves only k to be specified. Note that no assumptions have been made regarding the functional form of f , the correlation function k only determines the higher-order properties of the family of functions from which f is drawn, such as continuity, differentiability, and periodicity. Following Rasmussen and Williams [19], we also use the following shorthand notation to specify a GP:

$$\mathbf{f} \sim \mathcal{N}(\mu, K), \quad (17)$$

which should be interpreted according to Eq. (14). The fact that the function values at different positions ($f(\mathbf{x})$ and $f(\mathbf{x}')$) are correlated, with $\langle \delta f(\mathbf{x}) \delta f(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$, is what allows us to make predictions. Basically, the data for $f(\mathbf{x})$, measured at the “training” points \mathbf{x} , allow us to “learn” how the data are

correlated. This is done by inferring the hyperparameters Θ of the correlation function k , given the training data set. These hyperparameters Θ determine the precise shape of k , and thus the properties of the random functions f drawn from the GP. Once we have learned how, and to what degree, the function values are correlated with each other, we use this information, together with the known values of f at the training points \mathbf{x} , to predict the values of the function f_* at new “test” locations \mathbf{x}_* .

Assuming the N input values consist of n training points and m test points, we partition the input and output data into training and test data sets, to arrive at the following (prior) joint distribution:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{X}) \\ \mu(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right), \quad (18)$$

where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ is now the design matrix for the training points and $\mathbf{X}_* = (\mathbf{x}_{1*}, \dots, \mathbf{x}_{m*})$ that of the test points. Thus the correlation submatrices $K(\mathbf{X}, \mathbf{X})$, $K(\mathbf{X}, \mathbf{X}_*) = {}^t K(\mathbf{X}_*, \mathbf{X})$, and $K(\mathbf{X}_*, \mathbf{X}_*)$ have dimensions $n \times n$, $n \times m$, and $m \times m$, respectively. We stress the fact that Eqs. (17) and (18) are equivalent, at this point, we have simply relabeled the points as belonging to either the training or test data sets. In addition, if the training data are not known exactly, but subject to noise, we can account for this by adding the corresponding contribution to the covariance submatrix. For example, in the presence of Gaussian noise with variance σ^2 , we would simply replace $K(\mathbf{X}, \mathbf{X})$ with $K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_{n \times n}$.

The benefit of using GP comes from the Gaussian form of the distributions, since this allows us to perform most of the calculations analytically. In particular, the conditional distribution for the function values at the test points, conditioned on the training data, can be obtained from Bayes’ rule as $p(\mathbf{f}_*|\mathbf{f}) = p(\mathbf{f}_*, \mathbf{f})/p(\mathbf{f})$, and it results in yet another GP [19]

$$\begin{aligned} \mathbf{f}_*|\mathbf{f} &\sim \mathcal{N}(\mathbf{v}, \Sigma), \\ \mathbf{v} &= \mu(\mathbf{X}_*) + K(\mathbf{X}_*, \mathbf{X}) \cdot K(\mathbf{X}, \mathbf{X})^{-1} \cdot \delta \mathbf{f}(\mathbf{X}), \\ \Sigma &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X}) \cdot K(\mathbf{X}, \mathbf{X})^{-1} \cdot K(\mathbf{X}, \mathbf{X}_*). \end{aligned} \quad (19)$$

The prediction for the function values at the test points is then given by the mean \mathbf{v} , with the covariance matrix Σ providing a measure of the uncertainty at each point. Conceptually, one can interpret this prediction as the result of drawing random functions from the prior $\mathbf{f} \sim \mathcal{N}(\mu, K)$, and keeping only those that are consistent with the measured training data. The average and variance of the functions that remain will coincide with \mathbf{v} and Σ , i.e., $\mathbf{f}_* \sim \mathcal{N}(\mathbf{v}, \Sigma)$.

We have used a squared-exponential kernel for all our GP regressions. For a 1D regression problem, the kernel is defined as

$$\begin{aligned} k(x, x'; \Gamma, l) &= \Gamma^2 k_{SE}(x, x'; l), \\ k_{SE}(x, x'; l) &= \exp\left[-\frac{(x - x')^2}{2l^2}\right]. \end{aligned} \quad (20)$$

Here, $\Theta = (\Gamma, l)$ are the hyperparameters that must be inferred from the data, with Γ specifying the amplitude of the function variance and l the characteristic length scale

over which the function is varying in the x dimension. This commonly used squared-exponential kernel results in GPs that are infinitely differentiable, but other choices are possible, and might be more suitable for a given learning problem. Kernels for higher-dimensional function spaces can be defined by taking sums or products of 1D kernels [19,36]. As an example, the following are valid GP kernels for 2D functions, with input $\mathbf{x} = (x_1, x_2)$

$$k_{\text{sum}}(\mathbf{x}, \mathbf{x}'; \Gamma, l_1, l_2) = \Gamma^2(k_{\text{SE}}(x_1, x'_1; l_1) + k_{\text{SE}}(x_2, x'_2; l_2)),$$

$$k_{\text{mul}}(\mathbf{x}, \mathbf{x}'; \Gamma, l_1, l_2) = \Gamma^2 k_{\text{SE}}(x_1, x'_1; l_1) \cdot k_{\text{SE}}(x_2, x'_2; l_2).$$

In principle, we have independent sets of hyperparameters for each dimension, but for simplicity, when using additive kernels we will assume that the amplitude of the variance is equal along all dimensions, i.e., $\Gamma = \Gamma_1 = \Gamma_2$.

B. GP accelerated multiscale simulations

The idea of learning a constitutive relation for polymer flows from microscopic data is not entirely new. Indeed, previous work by Zhao *et al.* [21] has considered this precise problem, in order to perform macroscopic polymer flow simulations without having to introduce an arbitrary constitutive relation. In this work, they have assumed a generalized Newtonian model, and used simple-shear simulations to learn the apparent viscosity $\eta^{(\text{app})} = \sigma_{xy}/\dot{\gamma}$, of (inelastic) non-Newtonian fluids as a function of the shear rate $\dot{\gamma}$. When performing the macroscopic flow simulations, the maximum shear rate (computed from the second invariant of the strain-rate tensor) is taken as the local shear rate, and used within a GP regression scheme in order to obtain $\eta^{(\text{app})}$, and thus the local shear stress $\sigma_{xy} = \eta^{(\text{app})}\dot{\gamma}$. In practice, this amounts to placing a GP prior on the stress tensor itself

$$\boldsymbol{\sigma} \sim \mathcal{N}(\boldsymbol{\mu}, K). \quad (21)$$

However, as acknowledged by the authors, this excludes many interesting rheological properties of (viscoelastic) non-Newtonian fluids, as it does not allow for any type of history dependence in the flow, and relies on a separation of timescales between the microscopic and macroscopic dynamics. This history dependence, which arises from the internal stresses, is one of the most significant features of polymeric flows. Subsequent work by Zhao *et al.* [22] has considered viscoelastic flows, but the learning is restricted to parametrizing a constitutive relation with a predetermined functional form. Therefore the goal of the current work is to generalize the learning strategy, so that it can be used to model viscoelastic polymeric fluids without having to specify any constitutive relation.

Following Zhao *et al.* [21,22], we also use small-scale microscopic simulations of polymer chains at fixed strain rates to generate the training data necessary to learn the constitutive relation. However, we now place a GP prior on the time derivative of the stress tensor

$$\frac{d}{dt}\boldsymbol{\sigma} \equiv \dot{\boldsymbol{\sigma}} \sim \mathcal{N}(\boldsymbol{\mu}, K) \quad (22)$$

not on the stress tensor itself (or the effective viscosity). This difference allows us to consider the time-dependent memory effects crucial to describe the dynamics of polymer chains.

Even in the absence of polymer entanglement, this memory effect is non-negligible, thanks to the finite relaxation time of the polymer stretching and reorientation. No assumptions are made regarding the form of the constitutive equation, except for the fact that it should be expressed in differential form, as a function of the local instantaneous stress $\boldsymbol{\sigma}$ and strain-rate $\boldsymbol{\kappa}$ tensors. This includes most commonly used differential models, such as the Maxwell, Jeffreys, and Oldroyd fluids [3]. Finally, while it is possible to consider correlated output, we use a separate GP for each independent component of the stress tensor, resulting in $D(D+1)/2$ GP regressions in D dimensions.

Here, we have assumed a separation of length scales between the microscopic and macroscopic descriptions, such that at the microscopic level the system can be considered to be homogeneous, i.e., all field gradients are effectively zero. It is only at the macroscopic level where field gradient induced phenomena are incorporated. The appropriate microscopic length scale needed for this approximation to be valid will depend on the precise flow problem being studied. For the systems presented here, it is given by the characteristic size of the polymer chains. It is still possible (in principle) to apply the same type of learning strategy to microscopic models in which field gradients are not homogeneous. In such cases, the constitutive relations would be functions of the local stresses, strains, and their gradients. However, such considerations lie outside the scope of the current work.

A schematic diagram of the three-step learning strategy, consisting of data generation, learning, and prediction steps, is given in Fig. 2. For the first step, we performed fixed strain-rate simulations of the 3D microscopic model, corresponding to either simple-shear or planar elongational flow

$$\boldsymbol{\kappa}^{(\text{shear})} = \begin{pmatrix} 0 & \dot{\gamma} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ \dot{\gamma} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (23)$$

$$\boldsymbol{\kappa}^{(\text{elongational})} = \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ 0 & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (24)$$

with $\dot{\gamma}$ and $\dot{\epsilon}$ the shear and elongational flow rates, respectively. These particular flows are chosen because the learned constitutive relations will be used in simulations with an imposed 2D flow, where the z direction is assigned to be neutral. However, if one wants to apply this strategy to a more complex situation, additional applied flows, with different deformation rate tensors, will need to be used during the learning process. All simulations were started from a random and isotropic initial configuration of the polymer chains ($\langle \boldsymbol{\sigma} \rangle = 0$). The simulations were performed until a steady state was reached, such that $\dot{\boldsymbol{\sigma}} \simeq 0$. Each simulation provides us with a trajectory $(t, \boldsymbol{\sigma}(t))$, from which we can compute $\dot{\boldsymbol{\sigma}}(t)$. We randomly chose a fraction of these tuples $(\boldsymbol{\kappa}, \boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}})$ to serve as training data for the GP regression, with input $\mathbf{x} = (\boldsymbol{\kappa}, \boldsymbol{\sigma})$ and output $f(\mathbf{x}) = \dot{\boldsymbol{\sigma}}$. To compute $\dot{\boldsymbol{\sigma}}$, we use a finite-difference approximation over a characteristic timescale $\Delta t^{(c)}$. To reduce the noise in the measurements for $\boldsymbol{\sigma}$ and/or $\dot{\boldsymbol{\sigma}}$, it is recommended to apply a data smoothing operation, the details of which are given below.

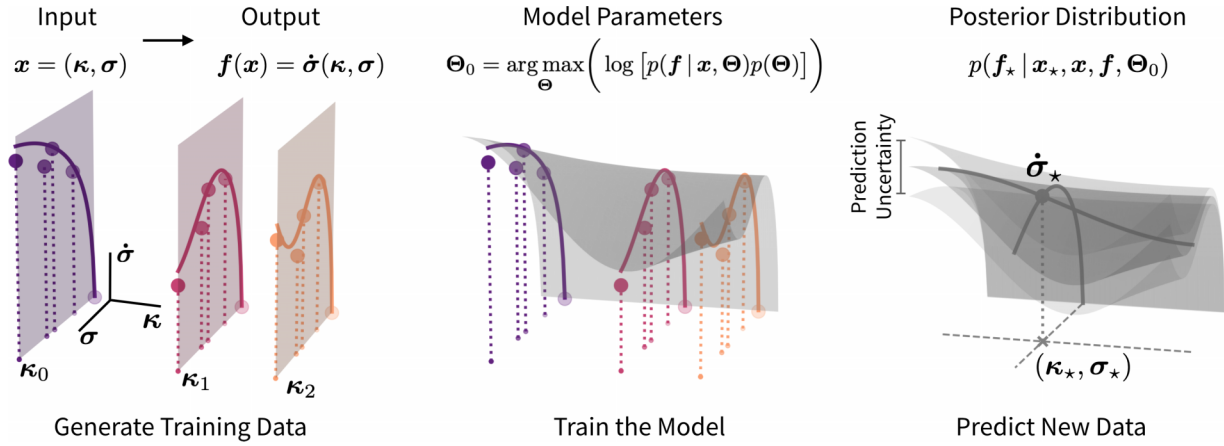


FIG. 2. Schematic representation of the GP-MSS strategy used to learn the constitutive relation of polymeric flows from microscopic data. (1) We perform small-scale microscopic simulations at fixed strain-rate κ , in order to obtain the input (κ, σ) and output $\dot{\sigma}$ training data. (2) Placing a GP prior on the constitutive relation, $\dot{\sigma} \sim \mathcal{N}(\mu, K)$, the training data are used to learn the hyper-parameters Θ (specifying the function variance and length scales) of the GP, by maximizing the posterior distribution for Θ . (3) The probability distribution for the constitutive relation at new “test” inputs (κ_*, σ_*) is then specified by a conditional GP, i.e., $\dot{\sigma}_* | \sigma \sim \mathcal{N}(\nu, \Sigma)$. We take the average value to be the best estimate, $\dot{\sigma}_* = \nu$, and use this within a macroscopic simulation, in order to update the stress at each point in the fluid. The prediction uncertainty is given by the corresponding covariance Σ , and it can be used to perform on-the-fly diagnostics [21].

The second step is to train the model, i.e., to determine the hyperparameters Θ of the kernel function $k(x, x')$, as well as any unknown uncertainties in the test data for $\dot{\sigma}$. The posterior probability distribution for the hyperparameters, given the model and measured training data, is determined from Bayes’ theorem as

$$p(\Theta | X, f) \propto p(f | X, \Theta) p(\Theta), \quad (25)$$

where the likelihood $p(f | X, \Theta)$ is given by the corresponding GP, Eq. (14), and $p(\Theta)$ is a suitably chosen prior for the hyperparameters. A full Bayesian treatment, in which we integrate out the hyperparameters and propagate the uncertainties during the simulation is possible, but for simplicity, we will consider only a pointwise solution. For the 1D problem considered below, we take the posterior average

$$\Theta_0^{(\text{avg})} = \int d\Theta \Theta p(\Theta | X, f) \quad (26)$$

estimated from Hamiltonian Monte Carlo (HMC) simulations. For the 2D problem, we instead use a stochastic gradient-based optimization method [37] to maximize the log-posterior, and find the “optimal” value

$$\Theta_0^{(\text{map})} = \arg \max_{\Theta} \log p(\Theta | X, f) \quad (27)$$

Finally, the optimized Θ_0 are used to parametrize the conditional distributions for the test data $f_* | f$, as defined in Eq. (19). In practical terms, we use the (conditional) mean $\nu = \langle \dot{\sigma}_* \rangle$ as our prediction for the constitutive relation, such that the stress of each fluid particle is updated according to

$$\sigma_i(t + \Delta t) = \sigma_i(t) + \langle \dot{\sigma}_* \rangle_i \Delta t, \quad (28)$$

where $\langle \dot{\sigma}_* \rangle_i$ is a function of the training data $x = (\kappa, \sigma)$, as well as the instantaneous (test) strain rate and stress tensors at the position of particle i , $x_* = (\kappa_*, \sigma_*) = (\kappa_i(t), \sigma_i(t))$.

We refer to this method, which uses a constitutive relation learned through a Gaussian process regression scheme, within

a macroscopic flow simulation, as a GP accelerated multi-scale simulation, or GP-MSS. We are mainly interested in learning this constitutive relation from microscopic polymer simulations, but to check the consistency of this approach, we will also consider learning from the constitutive relations themselves (e.g., from the upperconvected Maxwell model).

C. Algorithmic complexities

To understand the benefits of our proposed GP-MSS approach with respect to a standard MSS, we should consider the complexities of both algorithms. Since both methods employ the same macroscopic description for the flow, any gains or losses will be found in the calculation of the stresses. For the MSS, this is done by solving for the microscopic dynamics of the polymer chains. Assuming a coarse-grained entanglement model, the time and memory requirements will scale as $\mathcal{O}(N_f \times N_p \times z)$, with z the number of entanglement points per chain (for the case of noninteracting dumbbells we have $z = 1$). Our experience shows that $N_p \times z$ should be of the order of 10^4 – 10^5 or larger, in order to obtain reliable stress measurements. The complexity associated to the GP learning procedure is divided in two: the training step and the prediction step. The former is the most expensive of the two, scaling in time as $\mathcal{O}(n_{\text{training}}^2)$, and in memory as $\mathcal{O}(n_{\text{training}})$ [38,39]. However, we note that this procedure only needs to be done once, the resulting constitutive relation can then be used to perform arbitrarily complex flow simulations. To evaluate the performance of the GP-MSS, we focus then on the cost of making new predictions to update the stresses, as given by Eq. (28). This involves evaluating the average of the conditional GP at the test positions X_* , corresponding to the values of κ and σ for each fluid particle. From Eq. (19), we see that we must compute expressions of the form $K(X_*, X) \cdot (K(X, X)^{-1} \cdot \delta f(X))$, where the term in parenthesis depends only on the training points X . This term, which evaluates to a vector of length n_{training} , can also be precomputed. The

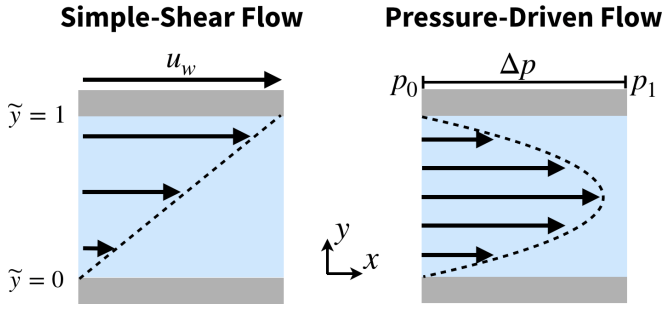


FIG. 3. Schematic representation of the two standard flow problems we have used to test our learning strategy, simple-shear flow and pressure driven flow. For the former, we consider a simplified 1D description, while for the latter we take into account the full 2D nature of the stress and strain-rate tensors.

prediction step can then be reduced to a simple matrix-vector multiplication, which scales as $\mathcal{O}(N_f \times n_{\text{training}})$. However, the memory requirements are still only $\mathcal{O}(n_{\text{training}})$. The asymptotic ratio of the GP-MSS to MSS stress calculation time is then $\mathcal{O}(n_{\text{training}}/(N_p \times z))$, which highlights the importance of generating a minimal high-quality training data set. The rule of thumb is to have the number of training points be smaller than the number of degrees of freedom in the microscopic polymer chain simulator. For the MSS simulations considered here, we need $N_p \simeq 10^5$ dumbbells to obtain reliable flow predictions. Using a random sampling protocol to generate the training data, we can achieve this same level of accuracy using $n_{\text{training}} \simeq 10^3$ training point. Although this does not account for the constant proportionality factors associated to each of the calculation costs, this two order of magnitude speedup is in agreement with the results of our simulations. The issue of generating an optimal training data set, which in this case will maximize the time saving with respect to the MSS, while keeping the same level of accuracy, is at the heart of most ML problems. It is quite likely such a protocol will have to be tailored to the specific microscopic model one wishes to study, as well as the macroscopic flow regimes to be simulated. For the systems studied here, random sampling proved to be enough, but it is not sure whether or not this will generalize to more complex setups.

IV. RESULTS

We will consider two basic flow problems in order to validate our proposed learning strategy (see Fig. 3): (1) simple-shear flow and (2) pressure driven flow. Given their symmetry, the flow in both systems is effectively one-dimensional, but a complete description of their dynamics requires that we account for all components of the stress, and their coupling to the flow. Furthermore, if this approach is to be applied to general geometries, it should be capable of learning the appropriate form of the constitutive relation for the stress without any simplifying assumptions (although it can be useful to introduce this additional information in specific cases). To show how our learning procedure can be extended as the dimensionality of the system increases, we will study (1) the simple-shear flow problem in 1D and the (2) pressure-driven flow problem in 2D.

For the simple-shear flow case, we assume that $v_y = 0$ and the strain-rate and stress tensors have only one nonzero component, i.e., the xy component. The learning problem is then 2D, since the constitutive relation is of the form $\tilde{\sigma}_{xy}(\tilde{\kappa}_{xy}, \tilde{\sigma}_{xy})$. For the pressure-driven case, we consider the full 2D nature of the stress and strain-rate dependence, such that the learning problem is now seven-dimensional, $\tilde{\sigma}_{\alpha\beta}(\tilde{\kappa}_{xx}, \tilde{\kappa}_{xy}, \tilde{\kappa}_{yx}, \tilde{\kappa}_{yy}, \tilde{\sigma}_{xx}, \tilde{\sigma}_{xy}, \tilde{\sigma}_{yy})$. For the flows, we are considering $\tilde{\kappa}_{\alpha z} = \tilde{\kappa}_{z\alpha} = 0$, and since $\text{Tr}(\tilde{\kappa}) = 0$, we have that $\tilde{\kappa}_{xx} = -\tilde{\kappa}_{yy}$. Instead of explicitly introducing such relationships into the model, we have preferred to learn them directly from the training data. As we have chosen a system of noninteracting Hookean dumbbells for our microscopic polymer model, we are able to compare our results with the exact analytical constitutive equation, Eq. (13). Thus we can check the convergences and sensitivity of our results, both in terms of the number of dumbbells N_p used in the microscopic simulations, as well as the number of training points used in the GP regression n_{training} . Finally, since the absolute value of the local stress tensor is not a physically measurable quantity, as opposed to the forces or velocities, we prefer to evaluate the accuracy of the learning strategy by comparing the error in the macroscopic predictions for the forces in the fluid and the flow velocities.

A. Simple-shear flow (2D learning)

To arrive at an effective 1D description for this simple-shear flow problem we have assumed the stress is a function of y only, and that the system starts from a relaxed state $\tilde{\sigma}_{\alpha\beta}(t=0) = 0$. The Maxwell constitutive relation, Eq. (13), then takes the following form:

$$\tilde{\sigma}_{xy}(\tilde{y}, \tilde{t}) = \frac{1}{\text{De}} [\tilde{\kappa}_{xy}(\tilde{y}, \tilde{t}) - \tilde{\sigma}_{xy}(\tilde{y}, \tilde{t})], \quad (29)$$

where $\tilde{\sigma}_{xx}(t) = \tilde{\sigma}_{yy}(t) = \tilde{\sigma}_{zz}(t) = \tilde{\sigma}_{xz}(t) = \tilde{\sigma}_{yz}(t) = 0$.

Microscopic training data were generated by performing fixed shear-rate simulations for a system of $N_p = 10^3, 10^4, \text{ and } 10^5$ noninteracting dumbbells in 3D for two different Deborah numbers, $\text{De} = 1$ and 10. In each case, we used nine different values of the adimensionalized shear rate $\tilde{\kappa}_{xy}$ within the range $[-150, 150]$, including $\tilde{\kappa}_{xy} = 0$. The time step was set to $\Delta\tilde{t} = 10^{-4}$ and the simulations were performed up to a maximum time of $\tilde{t}_{\text{max}} = 5\text{De}$. The trajectory of the stress $\tilde{\sigma}_{xy}(t)$ was then smoothed using a Gaussian filter with a width of $5 \times 10^{-3} \tau^{(m)}$, in order to reduce the noise in the estimates for $\tilde{\sigma}_{xy}$. The time derivative of the stress, obtained from the smoothed data, was computed using the following finite-difference approximation:

$$\tilde{\sigma}_{xy}(\tilde{y}, \tilde{t}) = \frac{\tilde{\sigma}_{xy}(\tilde{t}) - \tilde{\sigma}_{xy}(\tilde{t} - \Delta\tilde{t})}{\Delta\tilde{t}}. \quad (30)$$

The resulting $(\tilde{t}, \tilde{\sigma}_{xy}, \tilde{\sigma}_{xy})$ data are then partitioned into ten (possibly overlapping) randomly selected intervals over $\tilde{\sigma}_{xy}$, of width $(\max(\tilde{\sigma}_{xy}) - \min(\tilde{\sigma}_{xy}))/10$. Averaging over the points contained in each interval allows us to define a corresponding pair of input $\mathbf{x} = (\tilde{\kappa}_{xy}, \tilde{\sigma}_{xy})$ and output $f(\mathbf{x}) = \tilde{\sigma}_{xy}$ training points. The variance ϵ^2 in $\tilde{\sigma}_{xy}$, within each interval, is used as an estimate of the measurement error, and added to the corresponding covariance submatrix $K(\mathbf{X}, \mathbf{X})$, in order

to perform the GP regression [see Eq. (18)], i.e., $K(\mathbf{X}, \mathbf{X})_{ij} = k(\mathbf{X}_i, \mathbf{X}_j) + \epsilon_i^2 \delta_{ij}$.

We use a product kernel for this 2D regression problem $\mathbf{x} = (x_1, x_2) = (\tilde{\sigma}_{xy}, \tilde{\kappa}_{xy})$, such that

$$k(\mathbf{x}, \mathbf{x}') = \Gamma^2 k_{SE}(x_1, x'_1; l_1) \cdot k_{SE}(x_2, x'_2; l_2). \quad (31)$$

We thus have three hyperparameters, Γ and the two length scales for $\tilde{\sigma}_{xy}$ and $\tilde{\kappa}_{xy}$, so that $\Theta = (\Gamma, l_1, l_2)$. We use independent and uniform logarithmic priors, $p(\ln \Theta_i) \propto \text{const}$, which are scale-invariant, such that

$$p(\Theta) = p(\Gamma) \cdot p(l_1) \cdot p(l_2) \propto \frac{1}{\Gamma} \cdot \frac{1}{l_1} \cdot \frac{1}{l_2}.$$

To train the model, we performed Hamiltonian Monte Carlo simulations, using the no-U-turn sampler (NUTS), with the PYMC3 python package [40–42]. The optimal parameters Θ_0 are then taken to be the averages over the posterior distribution Θ^{avg} , and used to define the conditional GP for the constitutive relation. Figure 4 shows the results of this learning procedure for the case of $De = 10$. We are able to learn the Maxwell constitutive relation, Eq. (29), providing excellent predictions over a wide range of parameters. However, the prediction error increases the farther we get from the training points, as expected. This highlights the importance of generating a well chosen training data set, representative of the region in function space one is interested in. It is reassuring to note that even with the naive random sampling strategy outlined above, we are able to obtain precise predictions for the constitutive relation. While this is due to the simplicity of the function, we will show below that this approach also generalizes to more complicated functional forms and higher dimensions. Using this learned constitutive relation, we performed flow simulations under simple shear, at $Re = 10$, and compared the results to those obtained from standard MSS (with microscopic dumbbell simulators), as well from the Maxwell constitutive equation.

Considering the one-dimensional nature of the flow problem, one can use a simple Eulerian description instead of the Lagrangian one. Thus the system was discretized in the vertical y direction, using 128 grid points, with the velocity of the top and bottom walls set to $\tilde{v} = 1$ and 0, respectively. Starting from a quiescent fluid, a velocity wave will start at the top wall, and propagate through the channel, bouncing back and forth at the walls, before a steady-state linear velocity profile is obtained.¹ This transient regime is more pronounced at high De , as evidenced by the three kicks in the time evolution of the stress at $De = 10$, corresponding to the arrival of the wave front. For comparison, at $De = 1$ only one kick is observed [see Figs. 5(a) and 5(b)]. The excellent agreement obtained between the MSS and GP-MSS predictions for the stress is further evidence of the success in learning the constitutive relation. The stress fluctuations obtained from the MSS at long times are a consequence of the large statistical fluctuations that come from using a finite number of dumbbells.

¹See Ref. [43] for the time evolution of the velocity profile, for the simple-shear flow case, obtained from MSS and GP-MSS, as well as the exact solution given by the Maxwell constitutive relation. Results for $De = 1$ and 10 are provided.

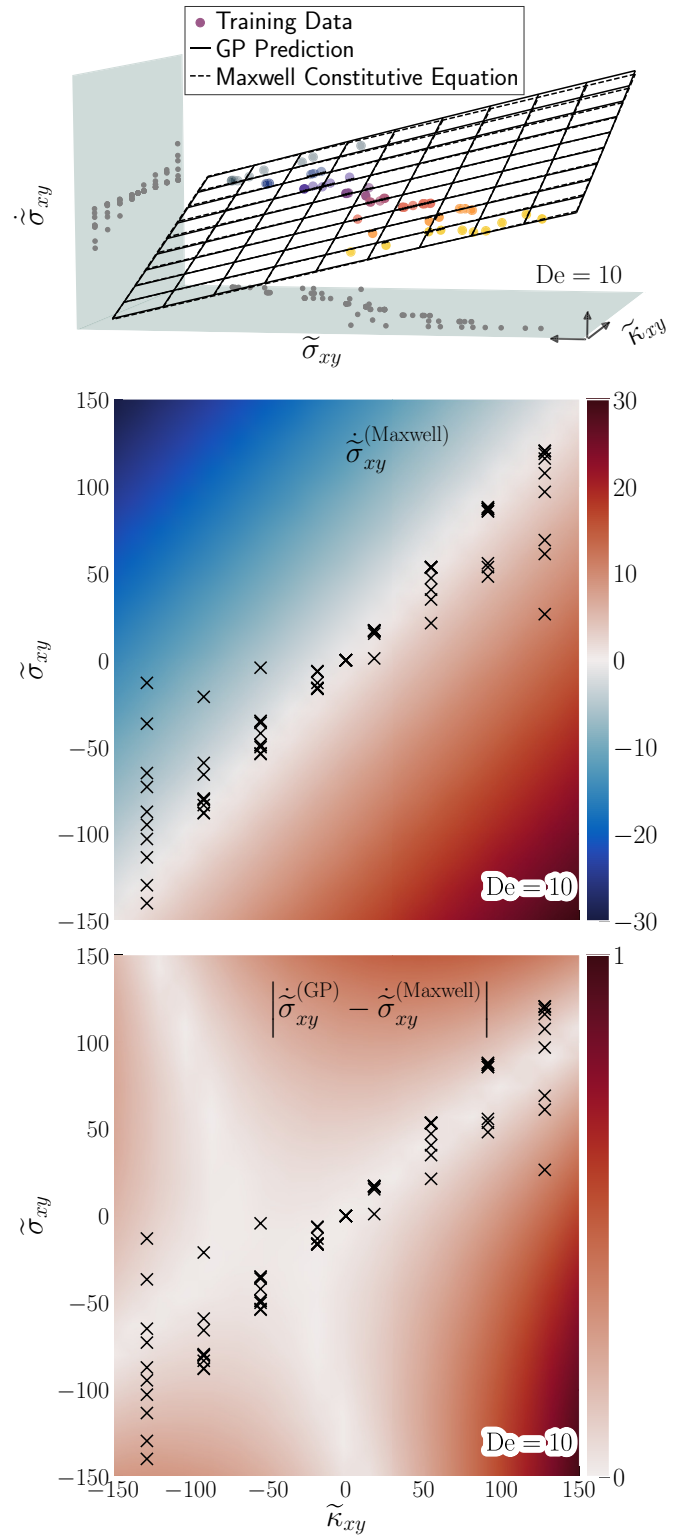


FIG. 4. The learned constitutive relation for the 1D simple shear flow problem at $De = 10$ using $N_p = 10^3$ dumbbells for the microscopic model. (Top) Training points, GP prediction, and the exact Maxwell constitutive equation ($N_p = \infty$ limit). (Middle) Color map showing the exact constitutive relation $\tilde{\sigma}_{xy}^{\text{Maxwell}}$, as a function of $\tilde{\kappa}_{xy}$ and $\tilde{\sigma}_{xy}$. (Bottom) Color map showing the absolute error between the GP prediction and the exact solution. The markers in the bottom two graphs show the location of the training data set.

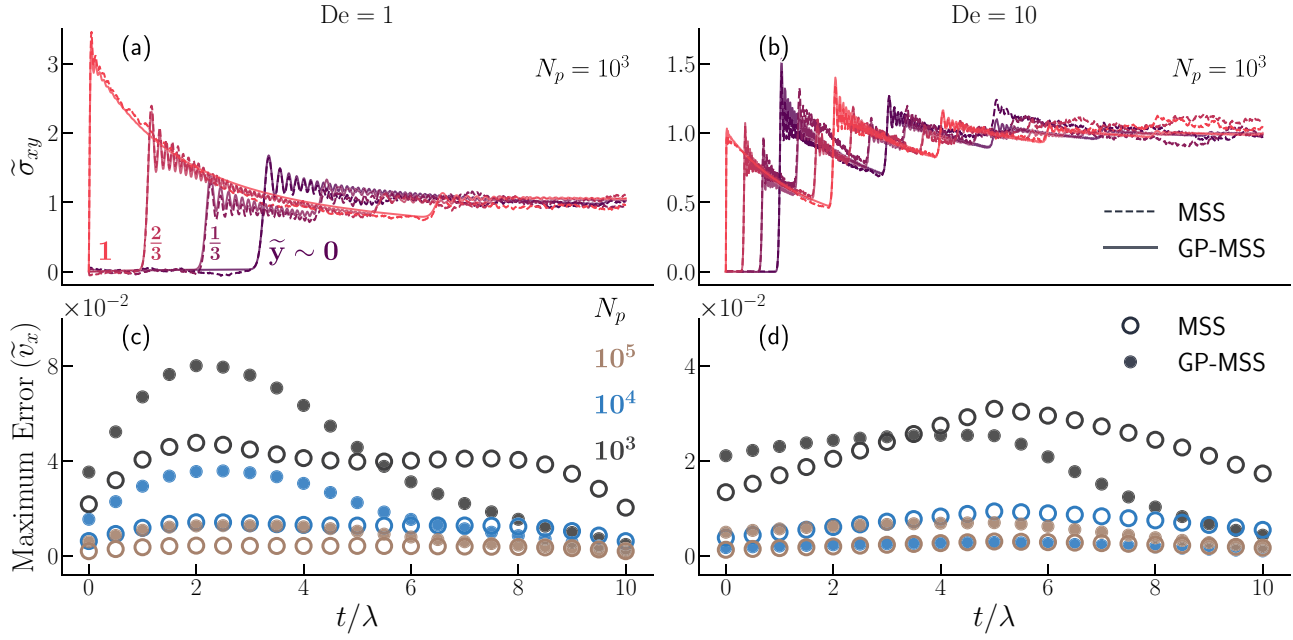


FIG. 5. [(a) and (b)] Time evolution of the stress obtained from simple-shear flow simulations ($Re = 10$) with the learned constitutive equation (GP-MSS) as well as those from the full MSS, using microscopic dumbbell simulators ($N_p = 10^3$). Dark (light) colored lines correspond to small (large) values of the channel height \tilde{y} . At $t = 0$, a velocity wave starts at $\tilde{y} = 1$ and propagates down the channel, bouncing back at the lower wall. [(c) and (d)] Maximum absolute error in the velocity, for all points in the channel, obtained from the MSS and GP-MSS predictions. The error is computed with respect to the results of the (exact) Maxwell constitutive equations, $|\tilde{v}_x - \tilde{v}_x^{(\text{Maxwell})}|$. Results for different number of dumbbells $N_p = 10^3$, 10^4 , and 10^5 used in the microscopic simulations are shown.

It is encouraging to see that the GP-MSS predictions do not show such behavior. This is because our learning strategy properly accounts for the measurement error in the training data, allowing us to infer the “true” function. To further test our ability to capture the history dependence of the flow, we have also considered an oscillatory shear flow (not shown), and obtained a similar level of agreement between MSS and GP-MSS predictions. We note that a generalized Newtonian approach, which assumes the stress σ is a function of κ , would not be able to capture this memory effect.²

Figures 5(c) and 5(d) shows the maximum absolute error in the predicted velocities, among all points in the system, as a function of the number of dumbbells N_p used in the microscopic simulations. This error is evaluated with respect to the velocities obtained from macroscopic simulations with the exact constitutive relation. Not surprisingly, given the good agreement in the stress profiles, the velocities also coincide. There is however a small offset in the transient regime

(see footnote 1), with the GP-MSS velocity wave showing a slight delay (advance) with respect to the MSS or Maxwell predictions at $De = 1$ (10). This is the main source error reported in Figs. 5(c) and 5(d), and is due to the small number of training data around this particular point in (κ, σ) space. Two aspects deserve to be highlighted. First, in all cases, we have considered, the error of the GP-MSS results is of the same order of magnitude as the MSS results. While the error can be two times larger, for times near the beginning of the start-up flow, it can also be considerably smaller, particularly at steady-state. This is most obvious for the case with small number of dumbbells $N_p = 10^3$. Second, the error decreases considerably as $N_p \rightarrow \infty$, as this decreases the statistical errors in both the MSS and the training data used to learn the constitutive relations. However, we note that the accuracy of the GP-MSS will also be affected by the quality of the training points. In the case of $De = 10$, for example, we happened to obtain slightly better results for $N_p = 10^4$ than for $N_p = 10^5$.

B. Pressure-driven flow (7D learning)

For the pressure driven-flow problem, we will consider the full 2D nature of the system. The learning problem then consists of three GP regressions, one each for $\tilde{\sigma}_{xx}$, $\tilde{\sigma}_{xy}$, and $\tilde{\sigma}_{yy}$, all of them functions in a seven-dimensional space $\mathbf{x} = (\tilde{\kappa}_{xx}, \tilde{\kappa}_{xy}, \tilde{\kappa}_{yx}, \tilde{\kappa}_{yy}, \tilde{\sigma}_{xx}, \tilde{\sigma}_{xy}, \tilde{\sigma}_{yy})$. Given the increased complexity with respect to the effective 1D flow considered previously, which resulted in a 2D learning problem, we have simplified the procedure to generate the training data. This is not directly related to the type of flow, but rather to the dimensionality of the problem. Microscopic 3D simulations for $N_p \lesssim 10^5$

²See Ref. [43] for the time evolution of the velocity profile, for oscillatory-shear flow, obtained from MSS and GP-MSS, as well as the exact solution given by the Maxwell constitutive relation. Simulations are performed by setting the velocity of the top wall to be $v_x = U \cos(\omega t)$, where the magnitude U and frequency ω of the shear flow are set by the Deborah and (squared) Womersley numbers, $De = 1$ and $Wo^2 = L^2 \rho \omega / \eta = 20$, respectively. For comparison purposes, we have also shown simulation results for a corresponding generalized Newtonian fluid, i.e., assuming a constitutive relation of the form $\sigma_{xy} = \eta^{(\text{eff})}(\dot{\gamma}) \cdot \dot{\gamma}$ (with $\dot{\gamma}$ the shear rate, and $\eta^{(\text{eff})}$ the effective viscosity), for which there is no memory effect.

noninteracting dumbbells at fixed strain-rate are performed. We considered both simple-shear and planar elongational flows within the range $|\tilde{\kappa}_{\alpha\beta}| \leq 150$. We chose 11 different values for each of $\tilde{\kappa}_{xx}$, $\tilde{\kappa}_{xy}$, and $\tilde{\kappa}_{yy}$, in addition, results for $\tilde{\kappa}_{yx}$ were obtained by taking the transpose of those at $\tilde{\kappa}_{xy}$. The simulation time step was fixed to $\Delta t = 10^{-6}$, the maximum time was set to be $t_{\max} = 10\text{De}$, and the corresponding $\sigma(t)$ trajectories were saved and used to generate the training data. For this, we randomly selected $N \simeq 10^3$ points $(\tilde{t}, \tilde{\kappa}_{\gamma\delta}, \tilde{\sigma}_{\alpha\beta})$, and included the initial state $\tilde{\sigma}_{\alpha\beta}(t=0) = 0$. Then, we use each point to define a time interval of width $0.1\tau^{(m)}$, with $\tau^{(m)}$ the polymer relaxation time. For the training data, we take the stress to be the average stress within the time interval, whereas the time derivative of the stress $\dot{\tilde{\sigma}}_{\alpha\beta}$ is taken to be the difference at the two end-points, i.e., using a coarse-grained time step of $\Delta\tilde{t}^{(c)} \simeq 0.05\text{De}$. The measurement error for $\dot{\tilde{\sigma}}_{\alpha\beta}$ cannot be ignored, but it is also not easy to evaluate in this high-dimensional space. Therefore we introduce this error $\epsilon_{\alpha\beta}$ as an additional hyperparameter that should be learned from the data. It is in such situations where the benefit of adopting a Bayesian approach pays off.

To summarize, we place a GP prior on each $\dot{\tilde{\sigma}}_{\alpha\beta}$, such that $\dot{\tilde{\sigma}}_{\alpha\beta} = f(\mathbf{x}) \sim \mathcal{N}(0, K(\mathbf{X}, \mathbf{X}))$. For the covariance matrix, we have $K(\mathbf{X}, \mathbf{X})_{ij} = k(\mathbf{X}_i, \mathbf{X}_j) + \epsilon^2\delta_{ij}$, with the measurement error assumed to be constant, equal for all \mathbf{x} . Here, we have used an additive first-order kernel, such that

$$k(\mathbf{x}, \mathbf{x}') = \Gamma^2 \sum_{\Lambda} k^{(\Lambda)}(x_{\Lambda}, x'_{\Lambda}; l_{\Lambda}) \quad (32)$$

with $\Lambda = 1, \dots, 7$ specifying one of the seven possible input components of $\mathbf{x} = (\tilde{\kappa}_{xx}, \tilde{\kappa}_{xy}, \tilde{\kappa}_{yx}, \tilde{\kappa}_{yy}, \tilde{\sigma}_{xx}, \tilde{\sigma}_{xy}, \tilde{\sigma}_{yy})$. The choice of an additive kernel is motivated by the fact that it allows for nonlocal interactions, making it less susceptible to the *curse of dimensionality* and allowing for better extrapolation in regions far way from the training data. In contrast, a multiplicative Kernel will rapidly revert to the mean away from the training data [[36], Chaps. 2.4 and 6.1]. This is not an issue for the 2D learning problem considered previously, as it was easy to generate a large enough sample of training points. For each 1D kernel $k^{(\Lambda)}$ we have one associated hyperparameter or length scale l_{Λ} , for a total of seven length scale hyperparameters. Together with Γ and the (unknown) measurement error $\epsilon_{\alpha\beta}$ associated to the training data for $\dot{\tilde{\sigma}}_{\alpha\beta}$, this results in a total of nine hyperparameters needed to learn each $\dot{\tilde{\sigma}}_{\alpha\beta}$. Note that we are assuming that the measurement error is constant, i.e., it does not depend on \mathbf{x} , although it can be different for the different components of the constitutive relation. The optimal values were obtained by maximizing the log-posterior, Eq. (25), assuming a constant prior for the hyperparameters ($p(\Theta) = \text{const}$). For this, we use ADAM,³ a stochastic gradient descent algorithm [37,44], as implemented in GPYTORCH [38,39].

This learning procedure results in three distinct functions $\dot{\tilde{\sigma}}_{xx}$, $\dot{\tilde{\sigma}}_{xy}$, and $\dot{\tilde{\sigma}}_{yy}$ of seven variables. The most interesting,

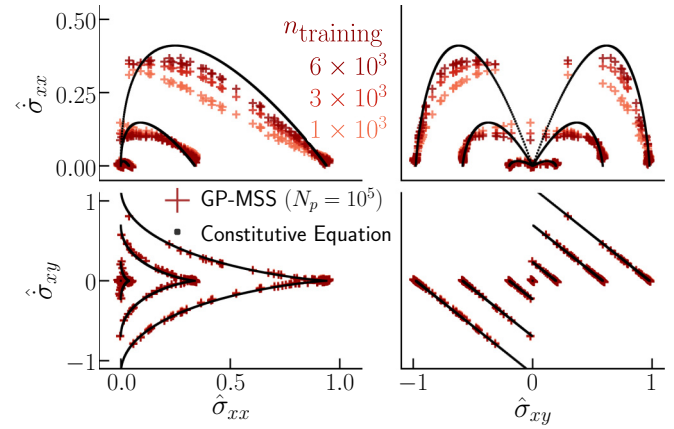


FIG. 6. GP predictions (cross symbols) for the constitutive relation $\dot{\sigma}$, learned using $n_{\text{training}} = 1, 3$ and 6×10^3 training points (generated from microscopic simulations with $N_p = 10^5$ dumbbells). Exact results (dot symbols) were obtained from the Maxwell constitutive relation, corresponding to the $N_p \rightarrow \infty$ limit. The data clearly show that the results improve as the number of training points increases, as expected. A caret ($\hat{\cdot}$) indicates data that have been scaled to lie in the range $[-1, 1]$, with the input $(\kappa_{\gamma\delta}, \sigma_{\alpha\beta})$ and output $(\dot{\sigma}_{\alpha\beta})$ scaled separately, to facilitate visualization.

nontrivial components of the constitutive relations (for the flows we are considering) are the xx and xy components, visualized in Fig. 6 for three different number of training points $n_{\text{training}} = 1, 3$ and 6×10^3 generated from microscopic simulations of $N_p = 10^5$ dumbbells.⁴ As for the 1D problem considered above, we compare our results with data generated from the exact Maxwell constitutive relation ($N_p \rightarrow \infty$), and, as expected, obtain better agreement as n_{training} increases. Finally, for comparison purposes, we have also learned the constitutive relation from training data generated from the exact solution, i.e., the Maxwell model ($N_p \rightarrow \infty$).

We used the learned constitutive relations to perform 2D simulations for the pressure driven flow problem, comparing our GP-MSS results with those of the conventional MSS and the exact Maxwell constitutive relation. We perform SPH simulations for all three cases, in a channel whose width $\tilde{L}_x = 0.6$ (along the flow direction) is about half its height $\tilde{L}_y = 1$. The fluid is discretized using 540 particles, initially arranged on a regular lattice within the system, corresponding to a 18×30 (width \times height) array of particles. The initial distance between each particle is $b = 1/30$, the smoothing length is $\tilde{h} = 1.5b$, and the cut-off length is $\tilde{R}_c = 3\tilde{h}$. We performed simulations at $\text{Re} = 10^{-2}$, for $\text{De} \lesssim 10^{-2}$, which is enough to observe elastic effects in the flow. Figure 7 shows the time evolution of the velocity at the center of the channel

³We mainly used the default parameters proposed by Kingma and Ma for machine learning problems, namely, the step size is $\alpha = 10^{-2}$, the hyperparameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We have set the maximum number of iterations to be 500.

⁴See Ref. [43] for the full constitutive relation map used in the learning procedure for the 2D pressure driven flow problem, for the case of $n_{\text{training}} = 1 \times 10^3$ points, together with data generated from the exact Maxwell constitutive relation. In addition, we also plot the trajectory data $[\kappa(t), \sigma(t), \dot{\sigma}(t)]$ obtained from GP-MSS ($\text{De} = 1 \times 10^{-2}$) for three representative points $\tilde{y} \sim 0, 1/4, 1/2$ along the channel. Simulation results using the exact constitutive relation are also given.

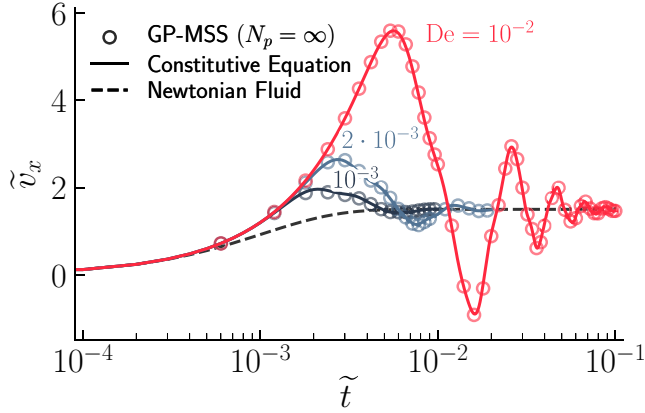


FIG. 7. Time evolution of the velocity, at $\tilde{y} = 0.5$, for the pressure-driven flow problem ($\text{Re} = 10^{-2}$). Results for three different De are shown, together with the corresponding Newtonian fluid ($\text{De} = 0$). Results obtained using the (solid line) exact Maxwell constitutive equation are compared with (open symbols) GP-MSS using a constitutive relation learned from ($n_{\text{training}} = 10^3$) noiseless training points generated from the Maxwell model.

($\tilde{y} = 0.5$) for $\text{De} = 1, 2$ and 10×10^{-3} , as obtained from GP-MSS using the constitutive relation learned from exact training data ($N_p \rightarrow \infty$), SPH simulations using the Maxwell constitutive equation, and a corresponding Newtonian fluid ($\text{De} = 0$). The default parameters for these 2D (SPH) MSS and GP-MSS are summarized in Table I.

As expected, at a steady state, the system is indistinguishable from a Newtonian fluid. However, at short times $\tilde{t} \lesssim 0.1$, effects due to the elastic energy stored in the dumbbells are visible. This is seen in the velocity oscillations around the Newtonian values, for which the overshoot can result in speeds that are more than three times the steady-state value. These elastic effects become more important as De is increased, and even results in large negative (transient) velocities at $\text{De} \gtrsim 10^{-2}$. The differences between the GP-MSS results, using the learned constitutive equation, and those from the exact constitutive equation are negligible. While

TABLE I. Default (nondimensionalized) parameter values used for the 2D MSS and GP-MSS simulations.

Parameter	Description	Value
Re	Reynolds number	10^{-2}
De	Deborah number	$10^{-3} - 10^{-1}$
C_s	artificial sound-speed	10
$\Delta \tilde{t}$	time step	10^{-6}
N_p	Number of dumbbells (MSS)	$10^3 - 10^5$
N_f	Number of fluid particles	$540 = 18 \times 30$
$(\tilde{L}_x, \tilde{L}_y)$	System size	(0.6, 1.0)
\tilde{b}	Initial particle size	$1/30$
\tilde{h}	Smoothing length	$1.5 \tilde{b}$
\tilde{R}_c	Cutoff length	$3 \tilde{h}$
\tilde{F}_e	External driving force	(12, 0)
n_{training}	GP training points	$1 \times 10^3 - 6 \times 10^3$
$\Delta \tilde{t}^{(c)}$	GP coarse-graining window	0.05De

these GP-MSS relied on a constitutive relation learned from exact noiseless data ($N_p \rightarrow \infty$), using a finite number of dumbbells N_p to generate the training data yields similar level of agreement, as will be shown below.

First, although there is a small difference between the learned constitutive relation and the exact solution (see Fig. 6), particularly for the $\tilde{\sigma}_{xx}$ component, the macroscopic predictions are in excellent agreement. This can be seen when looking at the forces in the fluid, as shown in Fig. 8 for three different positions along the height of the channel. Indeed, GP-MSS results using constitutive equations learned ($n_{\text{training}} = 10^3$) from the exact solution ($N_p \rightarrow \infty$) or from microscopic simulations ($N_p = 10^5$) are indistinguishable from each other at this scale, and they coincide with macroscopic simulation results using the Maxwell constitutive relation, although there is a small lag in the forces for the $N_p = 10^5$ MSS case. Second, we show that increasing the number of training points results in more accurate constitutive relations, and thus more reliable macroscopic flow simulations. We used the three constitutive relations of Fig. 6, generated from $n_{\text{training}} = 1, 3$, and 6×10^3 training points, to perform GP-MSS, and compared the predicted velocity profiles with the exact solution, as given by SPH simulations using the Maxwell constitutive relation.⁵ Figure 9 shows the maximum absolute error in the velocity as a function of time. Increasing the number of training points dramatically reduces the error in the simulations. Furthermore, the simulations using the constitutive relation learned on $N_p = 10^5$ dumbbells give the same level accuracy as those using the constitutive relation learned from the exact solution. All things being equal, increasing the number of training points will give better results; however, what matters is the quality of the training data set. This is the reason why the best results are obtained with the constitutive relations learned from the exact data, even though only a relatively small number of training points are used.

V. CONCLUSIONS

We have developed a learning strategy that is able to infer the constitutive relation of polymer melt flows from a small number of microscopic or coarse-grained polymer simulations. For this, we have used a Bayesian learning approach based on Gaussian process (GP) regressions. GPs provide a probability distribution over functions, allowing us to infer the most likely values, given known training data. In addition, we can estimate the uncertainty in the predictions, as well as incorporate unknown or incomplete data (e.g., measurement errors). Previous work has shown how one can use this type of approach to learn the effective viscosity of a polymer melt flow, as a function of the local shear rate

⁵See Ref. [43] for the time evolution of the velocity profile obtained from GP-MSS at $\text{De} = 10^{-2}$ (open symbols), using constitutive relations learned from $n_{\text{training}} = 1, 3, 6 \times 10^3$ points (generated from microscopic simulations with $N_p = 10^5$ dumbbells), as well as the exact solution given by the Maxwell constitutive relation (solid line). Results obtained using a constitutive relation learned from $n_{\text{training}} = 1 \times 10^3$ points generated from the exact constitutive relation are also shown (filled symbols).

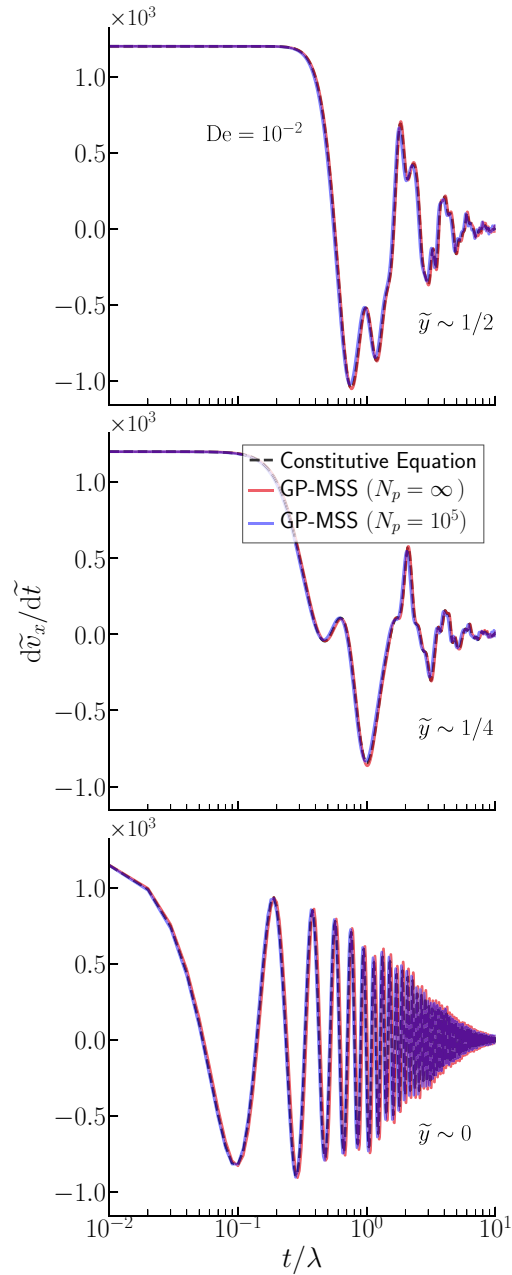


FIG. 8. Scaled force at three locations along the height of the channel, at $\tilde{y} \sim 1/2$, $1/4$, and 0 , for the pressure-driven flow problem ($Re = 10^{-2}$). Results are from GP-MSS, using a constitutive relation obtained from microscopic simulations ($N_p = 10^5$ dumbbells), as well as from the exact constitutive equation ($N_p \rightarrow \infty$), with $n_{\text{training}} = 10^3$. Exact results, obtained from simulations using the Maxwell constitutive relation are also shown.

[21], or to parametrize a viscoelastic constitutive relation [22]. Here, we demonstrate that a learning scheme that includes memory effects can be developed, which is crucial in order to describe the flow dynamics of entangled polymers in complex flow geometries. This method has great potential for polymer processing, as it will allow us to consider flow problems relevant to industrial settings. In addition, we believe that similar learning strategies can be designed for other soft-

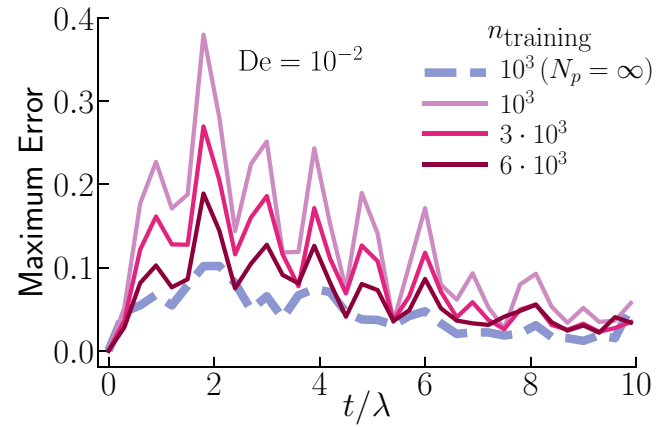


FIG. 9. Maximum absolute error in the velocity obtained from GP-MSS of the pressure-driven flow, $\max(\tilde{v} - \tilde{v}^{(\text{Maxwell})})$, using constitutive relations learned on different number of training points (generated from microscopic simulations using $N_p = 10^5$ dumbbells). Results using a constitutive relations learned from the exact solution ($N_p = \infty$) are also shown. The simulations were performed at $Re = De = 10^{-2}$.

matter systems, where the presence of multiple length and timescales gives rise to complex dynamical behavior that is expensive to simulate directly.

To validate the method, we have adopted the simplest possible microscopic polymer model, that of an ensemble of noninteracting Hookean dumbbells, since the exact constitutive equation is known in this case. This model was used in fixed strain-rate κ simulations, under simple-shear and planar elongation, to generate the required training data. For this, the time evolution of the stress $\sigma(t)$ was used to estimate the time derivative $\dot{\sigma}$. Assuming that the constitutive relation can be written in differential form, as a function of the local strain rate and stress, the goal is to learn the function $\dot{\sigma}(\kappa, \sigma)$. Thus the training data consist of input points (κ, σ) and the corresponding output $\dot{\sigma}$. We randomly selected a subset of $\simeq 10^3$ points and used them within a GP regression scheme in order to determine the optimal posterior distribution $p(\dot{\sigma}_* | \kappa, \sigma, \kappa_*, \sigma_*)$ for the constitutive equation $\dot{\sigma}_*$ at new input test points (κ_*, σ_*) , for which $\dot{\sigma}$ is not known. This conditional probability distribution is a GP, with average and covariance that are functions of both the training and test points.

We set the average $\langle \dot{\sigma}_* \rangle$, over the posterior distribution, to be our best prediction for the constitutive relation, and this function was then used within a macroscopic simulation in order to predict the flow behavior. In this way, we were able to carry out all our simulations at the macroscopic level, without having to impose any constitutive relation. Again, all we assumed was that the time derivative of the stress is a function of the local stress and strain-rate tensors. The appropriate constitutive equation is learned from a (relatively) small number of microscopic simulations. The resultant method, which we have referred to as GP-MSS, gives results that are as accurate as conventional MSS, at a fraction of the cost. With our nonoptimized PYTHON+NUMPY code, the difference in runtime between the full MSS (using 10^5 dumbbells) and the GP-MSS (with $n_{\text{training}} = 10^3$ training points) is ~ 100 times

for the 2D pressure driven flow problem. We note that the GP-MSS run only marginally slower than simulations using the Maxwell constitutive relations. Thus we get the best of both worlds, achieving run times comparable to macroscopic simulations, without sacrificing the accuracy provided by a microscopic polymer description. We expect the speedup afforded by the GP-MSS to improve dramatically when more realistic, and complex, microscopic polymer models are used. However, the reliability and efficiency of the GP-MSS will depend on the quality and size of the training data set. Thus care should be taken when devising the data generating protocol. As an added benefit to our approach, we note that it is also possible to maintain (consistent) information on the microscopic degrees of freedom, for example, by adopting a multifidelity representation [45,46]. In this case, a small number of microscopic simulators could be introduced in order to provide accurate (localized) stress measurements, which are then fused together with the approximate predictions provided by the learned constitutive relation. In future work, we will apply this learning approach to tackle the problem of entangled polymer melt flows. This will allow us to consider 3D flows in complex geometries, which have so far remained out of reach for standard MSS techniques.

ACKNOWLEDGMENTS

The authors would like to thank Ryoichi Yamamoto, Matthew Turner, and Simon K. Schnyder for fruitful discussion. This work was supported by the Japan Society for the Promotion of Science (Grants-in-Aid for Scientific Research KAKENHI No. 19H01862 and Wakate B No. 17K17825), the Ogasawara Foundation, and the SPIRITS 2020 of Kyoto University. Figures and movies were generated using MATPLOTLIB [47], a PYTHON 2D plotting library.

APPENDIX: THE SPH METHOD

1. Introduction

The smooth particle hydrodynamics (SPH) method, a particle-based method originally developed to solve astrophysics problems, provides a computationally convenient way to solve the Navier-Stokes equation in a Lagrangian framework [23]. The fluid is discretized into a number N_f of fluid particles that carry mass, momentum, and energy. Any function of the system can then be expressed as an interpolation over the (disordered) fluid particles, which serve as interpolation points. Consider a scalar quantity A and a vector quantity \mathbf{V} , which depend on position. The value of $A(\mathbf{x})$ or $\mathbf{V}(\mathbf{x})$, at a given position \mathbf{x} , which need not correspond to the position of any of the fluid particles, is given by

$$A(\mathbf{x}) = \int A(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}', \quad (\text{A1})$$

$$\mathbf{V}(\mathbf{x}) = \int \mathbf{V}(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}' \quad (\text{A2})$$

with similar expressions for higher-order tensor fields. Here, $W(\mathbf{x}, h)$ is a smoothing or interpolating kernel that should integrate to unity and tend to a delta function in the limit when the smoothing length h goes to zero. In this work, we adopt a

Gaussian interpolating kernel

$$W(\mathbf{r}, h) = \frac{1}{(h\sqrt{\pi})^D} \exp\left[-\frac{\|\mathbf{r}\|^2}{h^2}\right] \quad (\text{A3})$$

with \mathbf{r} a D -dimensional distance vector and $\|\cdot\|$ the L2-norm, i.e., $\|\mathbf{r}\| = (\sum_{i=1}^D r_i^2)^{1/2}$. We note that derivatives of these functions can be easily evaluated, as the derivative operator can be transferred to the kernel, for which analytic results can be computed in advance. Thus

$$\nabla A(\mathbf{x}) = \int d\mathbf{x}' A(\mathbf{x}')\nabla_{\mathbf{x}}W(\mathbf{x} - \mathbf{x}', h),$$

$$\nabla \cdot \mathbf{V}(\mathbf{x}) = \int d\mathbf{x}' \mathbf{V}(\mathbf{x}') \cdot \nabla_{\mathbf{x}}W(\mathbf{x} - \mathbf{x}', h).$$

Numerically, these integrals are replaced by sums over the fluid particles, i.e., the interpolating points, such that

$$A(\mathbf{x}) = \int d\mathbf{x}' A(\mathbf{x}')W(\mathbf{x} - \mathbf{x}') \rightarrow \sum_{j' \in \mathcal{R}(\mathbf{x})} \frac{m_{j'}}{\rho_{j'}} A(\mathbf{x}_{j'})W(\mathbf{x} - \mathbf{x}_{j'}) \quad (\text{A4})$$

with m_i and ρ_i the mass and density of the i th fluid particle. To reduce the computational burden, these sums have been truncated to only include fluid particles within a cutoff region $\mathcal{R}(\mathbf{x})$ centered at \mathbf{x} , such that $\|\mathbf{x}_{j'} - \mathbf{x}\| \leq R_c = 3h$. For notational simplicity, in what follows we will denote these truncated sums using a primed summation symbol. As an example, the density of each element can be computed by taking $A = \rho$ and evaluating the function at $\mathbf{x} = \mathbf{x}_i$, with $A_i \equiv A(\mathbf{x}_i)$ and $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j)$

$$\rho_i = \sum_j' m_j W_{ij} \quad (\text{A5})$$

with the mass a constant.

To solve the equations of motion for the fluid particles, Eqs. (1) and (2), we need to evaluate the forces acting on each particle, which involves computing the divergence of the stress tensor. Instead of directly discretizing $\nabla \cdot \boldsymbol{\sigma}$, the ‘‘golden rules’’ of SPH state that formulas should be rewritten to place the density inside the differential operators [23]. In this way, the forces are evaluated using the following expression:

$$\rho^{-1} \nabla \cdot (\boldsymbol{\sigma} - p\mathbf{l}) \equiv \nabla \cdot \left(\frac{\boldsymbol{\sigma} - p\mathbf{l}}{\rho} \right) + \frac{\boldsymbol{\sigma} - p\mathbf{l}}{\rho^2} \nabla \rho. \quad (\text{A6})$$

Therefore the time derivative of the fluid particle velocity \mathbf{v}_i , Eq. (2), is given by

$$\frac{d\mathbf{v}_i}{dt} = \sum_j m_j \left[\left(\frac{\boldsymbol{\sigma} - p\mathbf{l}}{\rho^2} \right)_i + \left(\frac{\boldsymbol{\sigma} - p\mathbf{l}}{\rho^2} \right)_j \right] \cdot \nabla_i W_{ij} + \mathbf{F}(\mathbf{x}_i). \quad (\text{A7})$$

Finally, the equations of motion for the SPH particles are discretized in time and integrated using the following second-order scheme:

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^n \Delta t + \frac{1}{2m_i} \left(\frac{d\mathbf{v}_i}{dt} \right)^n (\Delta t)^2$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \frac{1}{2m_i} \left[\left(\frac{d\mathbf{v}_i}{dt} \right)^{n+1} + \left(\frac{d\mathbf{v}_i}{dt} \right)^n \right] \Delta t. \quad (\text{A8})$$

2. Modified SPH

Rigid boundaries, be they fixed walls or moving particles, can be incorporated within the SPH framework by discretizing them with boundary or wall particles. However, using the standard SPH method introduced above, an unnatural flow is observed in the presence of such boundaries. To remove these artifacts, the values of the first and second derivatives of the physical quantities of interest can be used within the weighted averages [48], to arrive at the so-called modified SPH (MSPH). The appropriate expressions can be obtained by starting from the second-order Taylor expansion of the quantity of interest. For a scalar quantity $A(\mathbf{x}_j)$, we have

$$\begin{aligned} A(\mathbf{x}_j) &\simeq A(\mathbf{x}_i) + \nabla A(\mathbf{x}_i) \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ &\quad + \frac{1}{2} {}^t(\mathbf{x}_j - \mathbf{x}_i) \cdot \nabla_i \nabla_i A(\mathbf{x}_i) \cdot (\mathbf{x}_j - \mathbf{x}_i), \\ A_j &\simeq A_i + \nabla A_i \cdot \mathbf{x}_{ji} + \frac{1}{2} {}^t \mathbf{x}_{ji} \cdot \nabla \nabla A_i \cdot \mathbf{x}_{ji} \end{aligned} \quad (\text{A9})$$

with $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Multiplying both sides of Eq. (A9) by the volume element m_j/ρ_j times the smoothing function W_{ij} and summing over all particles, we obtain

$$\begin{aligned} &\sum_j' \frac{m_j}{\rho_j} A_j W_{ij} \\ &= \sum_j' \frac{m_j}{\rho_j} W_{ij} A_i + \sum_j' \frac{m_j}{\rho_j} W_{ij} \mathbf{x}_{ji} \cdot \nabla A_i \\ &\quad + \frac{1}{2} \sum_j' \frac{m_j}{\rho_j} W_{ij} {}^t \mathbf{x}_{ji} \mathbf{x}_{ji} : \nabla \nabla A_i, \end{aligned} \quad (\text{A10})$$

which relates A to its first- and second-order derivatives. To solve this equation, we then need the corresponding relations for ∇A and $\nabla \nabla A$, which are given in Eqs. (A11) and (A12)

$$\begin{aligned} \sum_j' \frac{m_j}{\rho_j} A_j \nabla_i W_{ij} &= \sum_j' \frac{m_j}{\rho_j} \nabla_i W_{ij} A_i \\ &\quad + \sum_j' \frac{m_j}{\rho_j} \nabla_i W_{ij} \mathbf{x}_{ji} \cdot \nabla A_i \\ &= \frac{1}{2} \sum_j' \frac{m_j}{\rho_j} \nabla W_{ij} {}^t \mathbf{x}_{ji} \mathbf{x}_{ji} : \nabla \nabla A_i, \end{aligned} \quad (\text{A11})$$

$$\begin{aligned} \sum_j' \frac{m_j}{\rho_j} A_j \nabla_i \nabla_i W_{ij} &= \sum_j' \frac{m_j}{\rho_j} \nabla_i \nabla_i W_{ij} A_i \\ &\quad + \sum_j' \frac{m_j}{\rho_j} \nabla_i \nabla_i W_{ij} \mathbf{x}_{ji} \cdot \nabla A_i \\ &\quad + \frac{1}{2} \sum_j' \frac{m_j}{\rho_j} \nabla_i \nabla_i W_{ij} {}^t \mathbf{x}_{ji} \mathbf{x}_{ji} : \nabla \nabla A_i. \end{aligned} \quad (\text{A12})$$

Equations (A10)–(A12) can be conveniently expressed in matrix form as follows:

$$\mathbf{t}_i = \mathbf{B}_i \cdot \mathbf{f}_i \quad (\text{A13})$$

where $\mathbf{f}_i = {}^t(A_i, \nabla A_i, \nabla \nabla A_i)$ is a vector whose entries are formed from A_i and its derivatives. In 2D this results in six

independent components, thanks to the commutativity of the partial derivatives, such that

$$\mathbf{f}_i \equiv \begin{pmatrix} A_i \\ A_{i,x} \\ A_{i,y} \\ A_{i,xx} \\ A_{i,xy} \\ A_{i,yy} \end{pmatrix}, \quad (\text{A14})$$

where commas are used to denote partial derivatives, $\partial_\alpha A_i = A_{i,\alpha}$ and $\partial_\alpha \partial_\beta A_i = A_{i,\alpha\beta}$. Likewise, the vector \mathbf{t}_i is composed using W_{ij} and its derivatives

$$\mathbf{t}_i^K = \sum_j' \frac{m_j}{\rho_j} A_j \Phi_{ij}^K, \quad (\text{A15})$$

where Φ_{ij}^K ($K = 1, \dots, 6$) are the components of $\Phi_{ij} = {}^t(W_{ij}, \nabla_i W_{ij}, \nabla_i \nabla_i W_{ij})$, given by

$$\Phi_{ij} \equiv \begin{pmatrix} W_{ij} \\ W_{ij,x} \\ W_{ij,y} \\ W_{ij,xx} \\ W_{ij,xy} \\ W_{ij,yy} \end{pmatrix}. \quad (\text{A16})$$

Finally, the \mathbf{B}_i matrix is defined as

$$\mathbf{B}_i^{KL} = \sum_j' \frac{m_j}{\rho_j} \Phi_{ij}^K \Theta_{ij}^L \quad (\text{A17})$$

with Θ_{ij}^L the components of the Θ_{ij} vector, given by

$$\Theta_{ij} = \begin{pmatrix} 1 \\ (x_i - x_j) \\ (y_i - y_j) \\ \frac{1}{2}(x_i - x_j)^2 \\ (x_i - x_j)(y_i - y_j) \\ \frac{1}{2}(y_i - y_j)^2 \end{pmatrix}. \quad (\text{A18})$$

Within the MSPH method, we use $\mathbf{f}_i = \mathbf{B}_i^{-1} \cdot \mathbf{t}_i$ to define the physical quantity A_i of particle i at position \mathbf{x}_i . Similar expressions can be defined for vector quantities and higher order tensors.

3. Boundary conditions

To enforce the no-slip boundary condition at the fluid/solid interface, we use a virtual particle method [15,49]. The virtual particles are placed by reflecting the outermost layer of wall particles with respect to the boundary. Then, the velocity vector at the positions of these virtual particles $\mathbf{v}_i^{(\text{virtual})}$ is computed using the MSPH method, and the velocities of the symmetric wall particles are set according to $\mathbf{v}_i^{(\text{wall})} = -\mathbf{v}_i^{(\text{virtual})}$. In this way, the weighted average of the particle velocities at the boundary is guaranteed to be zero.

[1] M. Rubinstein and R. H. Colby, *Polymer Physics*, 1st ed. (Oxford University Press, Oxford, 2003).

[2] National Research Council, *Polymer Science and Engineering: The Shifting Research Frontiers* (The National Academies Press, Washington, DC, 1994).

- [3] R. G. Larson, *Constitutive Equations for Polymer Melts and Solutions: Butterworths Series in Chemical Engineering* (Butterworth-Heinemann, Stoneham, 1988).
- [4] M. Laso and H. C. Öttinger, Calculation of viscoelastic flow using molecular models: The connfessit approach, *J. Non-Newtonian Fluid Mech.* **47**, 1 (1993).
- [5] H. C. Öttinger, *Beyond Equilibrium Thermodynamics*, 1st ed. (Wiley, Hoboken, New Jersey, 2005).
- [6] O. Borodin, D. Bedrov, G. D. Smith, J. Nairn, and S. Bardenhagen, Multiscale modeling of viscoelastic properties of polymer nanocomposites, *J. Polym. Sci., Part B: Polym. Phys.* **43**, 1005 (2005).
- [7] S. Yasuda and R. Yamamoto, A model for hybrid simulations of molecular dynamics and computational fluid dynamics, *Phys. Fluids* **20**, 113101 (2008).
- [8] T. Murashima and T. Taniguchi, Multiscale Lagrangian fluid dynamics simulation for polymeric fluid, *J. Polym. Sci., Part B: Polym. Phys.* **48**, 886 (2010).
- [9] T. Murashima and T. Taniguchi, Multiscale simulation of history-dependent flow in entangled polymer melts, *Europhys. Lett.* **96**, 18002 (2011).
- [10] Y. Masubuchi, T. Uneyama, and K. Saito, A multiscale simulation of polymer processing using parameter-based bridging in melt rheology, *J. Appl. Polym. Sci.* **125**, 2740 (2012).
- [11] S. Yasuda and R. Yamamoto, Synchronized Molecular-Dynamics Simulation Via Macroscopic Heat and Momentum Transfer: An Application to Polymer Lubrication, *Phys. Rev. X* **4**, 041011 (2014).
- [12] C. Wu, Multiscale modeling scheme for simulating polymeric melts: Application to poly(ethylene oxide), *Macromol. Theory Simul.* **27**, 1700066 (2018).
- [13] T. Sato, K. Takase, and T. Taniguchi, Multiscale simulation of polymer melt spinning by using the dumbbell model, *Nihon Reoroji Gakkaishi* **44**, 265 (2017).
- [14] T. Sato and T. Taniguchi, Multiscale simulations for entangled polymer melt spinning process, *J. Non-Newtonian Fluid Mech.* **241**, 34 (2017).
- [15] T. Sato, K. Harada, and T. Taniguchi, Multiscale simulations of flows of a well-entangled polymer melt in a contraction-expansion channel, *Macromolecules* **52**, 547 (2019).
- [16] Y. Mu, N. Li, L. Hang, G. Zhao, J. Gao, and Z. Niu, Investigation of the rheological behaviors of polymeric materials in the film casting process through multiscale modeling and simulation method, *Macromol. Theory Simul.* **28**, 1900001 (2019).
- [17] E. T. Jaynes, *Probability Theory: The Logic of Science*, 1st ed. (Cambridge University Press, New York, 2003).
- [18] D. S. Sivia and J. Skilling, *Data Analysis: A Bayesian Tutorial*, 2nd ed. (Oxford University Press, Oxford, 2006).
- [19] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, Cambridge, 2005).
- [20] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, 1st ed. (MIT Press, Cambridge, 2012).
- [21] L. Zhao, Z. Li, B. Caswell, J. Ouyang, and G. E. Karniadakis, Active learning of constitutive relation from mesoscopic dynamics for macroscopic modeling of non-Newtonian flows, *J. Comput. Phys.* **363**, 116 (2018).
- [22] L. Zhao, Z. Li, Z. Wang, B. Caswell, J. Ouyang, and G. E. Karniadakis, Active- and transfer-learning applied to microscale-macroscopic coupling to simulate viscoelastic flows, [arXiv:2005.04382](https://arxiv.org/abs/2005.04382).
- [23] J. Monaghan, Smoothed particle hydrodynamics, *Annu. Rev. Astron. Astrophys.* **30**, 543 (1992).
- [24] J. P. Morris, P. J. Fox, and Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, *J. Comput. Phys.* **136**, 214 (1997).
- [25] R. Byron Bird, C. F. Curtiss, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids, Vol. 2: Kinetic Theory*, 2nd ed. (Wiley-Interscience, New York, 1987).
- [26] P. E. Rouse, A theory of the linear viscoelastic properties of dilute solutions of coiling polymers, *J. Chem. Phys.* **21**, 1272 (1953).
- [27] K. Kremer and G. S. Grest, Dynamics of entangled linear polymer melts: A molecular-dynamics simulation, *J. Chem. Phys.* **92**, 5057 (1990).
- [28] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics*, 1st ed. (Oxford University Press, Oxford, 1986).
- [29] Y. Masubuchi, J. I. Takimoto, K. Koyama, G. Ianniruberto, G. Marrucci, and F. Greco, Brownian simulations of a network of reptating primitive chains, *J. Chem. Phys.* **115**, 4387 (2001).
- [30] M. Doi, J. I. Takimoto, P. G. De Gennes, R. Magerle, A. N. Semenov, D. J. Read, M. E. Cates, and X. H. Zheng, Molecular modeling of entanglement, *Philos. Trans. R. Soc., A* **361**, 641 (2003).
- [31] J. D. Schieber, J. Neergaard, and S. Gupta, A full-chain, temporary network model with sliplinks, chain-length fluctuations, chain connectivity and chain stretching, *J. Rheol.* **47**, 213 (2003).
- [32] A. E. Likhtman, Single-chain slip-link model of entangled polymers: Simultaneous description of neutron spin-echo, rheology, and diffusion, *Macromolecules* **38**, 6128 (2005).
- [33] T. Uneyama and Y. Masubuchi, Multi-chain slip-spring model for entangled polymer dynamics, *J. Chem. Phys.* **137**, 154902 (2012).
- [34] R. Byron Bird, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids, Vol. 1: Fluid Mechanics*, 2nd ed. (Wiley-Interscience, New York, 1987).
- [35] R. B. Bird and C. F. Curtiss, Molecular theory expressions for the stress tensor in flowing polymeric liquids, *J. Polym. Sci., Polym. Symp.* **73**, 187 (1985).
- [36] D. Duvenaud, Automatic model construction with Gaussian processes, Ph.D. thesis, University of Cambridge, 2014.
- [37] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [38] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, GPyTorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration, in *Advances in Neural Information Processing Systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates Inc., 2018), pp. 7587–7597.
- [39] K. A. Wang, G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson, Exact Gaussian Processes on a Million Data Points, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. D'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), pp. 14648–14659.

- [40] M. Betancourt, A conceptual introduction to hamiltonian monte carlo, [arXiv:1701.02434](https://arxiv.org/abs/1701.02434).
- [41] M. D. Hoffman and A. Gelman, The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.* **15**, 1593 (2014).
- [42] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, Probabilistic programming in Python using PyMC3, *PeerJ Comp. Sci.* **2**, e55 (2016).
- [43] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevResearch.2.033107> for movies showing the time evolution of the velocity and stress profiles obtained from the MSS, GP-MSS, and constitutive relations. SM1 and SM2 show the velocity profiles for the simple-shear flow at $De = 1$ and $De = 10$, respectively; SM3 shows the velocity profile for an oscillatory shear-flow ($De = 1$). SM4 shows the constitutive relation map used for the pressure driven flow, together with GP-MSS trajectories of the stress, and SM5 shows the corresponding velocity profiles, as a function of the number of training points used to learn the constitutive relation.
- [44] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015* (2014), pp. 1–15, <https://dblp.org/db/conf/iclr/iclr2015>.
- [45] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, *Proc. R. Soc. A* **473**, 20160751 (2017).
- [46] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, *J. Comput. Phys.* **335**, 736 (2017).
- [47] J. D. Hunter, Matplotlib: A 2D Graphics environment, *Comput. Sci. Eng.* **9**, 90 (2007).
- [48] G. M. Zhang and R. C. Batra, Modified smoothed particle hydrodynamics method and its application to transient problems, *Comput. Mech.* **34**, 137 (2004).
- [49] S. Adami, X. Y. Hu, and N. A. Adams, A generalized wall boundary condition for smoothed particle hydrodynamics, *J. Comput. Phys.* **231**, 7057 (2012).