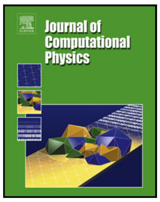
Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



A general deep learning method for computing molecular parameters of viscoelastic constitutive model by solving an inverse problem

Minghui Ye, Yuanqi Fan, Xue-Feng Yuan*

Institute for Systems Rheology, Guangzhou University, Guangzhou 510006, China

ARTICLE INFO

Article history:

Communicated by X-F. Yuan

Keywords: Machine learning, Inverse learning, Molecular constitutive model

ABSTRACT

Prediction of molecular parameters and material functions from the macroscopic viscoelastic properties of complex fluids is of great significance for molecular and formulation design in fundamental research as well as various industrial applications. A general learning method for computing molecular parameters of viscoelastic constitutive model by solving an inverse problem is proposed. The accuracy, convergence and robustness of a deep neural network (DNN) based numerical solver have been validated. The results show that the molecular parameters computed by the DNN based numerical solver not only reproduce accurately the steady viscoelastic stress of completely monodisperse linear lambda DNA solution over a wide range of shear rate at various concentrations, but also predict a power law concentration scaling with nearly same scaling exponent as those estimated from experimental results. Even for highly polydisperse polyacrylamide aqueous solution, the DNN based numerical solver could resolve its power law concentration scaling satisfactorily.

1. Introduction

Complex fluids, such as self-assembled surfactant and polymer solutions, exhibit dynamic instabilities and highly nonlinear flow effects owning to that their characteristic length and time scales are usually much larger than those of simple liquids. Similar to Newtonian fluids in high Reynolds (Re) number flow regime, complex fluids could show turbulence-like instabilities but at low Re number and high Weissenberg (Wi) number flow regime, in which elastic forces dominate over viscous forces. Such an intriguing flow phenomenon observed from micro- to macro-flows has been coined as Elastic Turbulence [1, 2, 3] and already widely exploited in many industrial applications such as significant improvement of the mass transfer (mixing) efficiency in

---

*Corresponding author: Xue-Feng Yuan
e-mail: xuefeng.yuan@gzhu.edu.cn (Xue-Feng Yuan)

the microchannel [2], great enhancement of the secondary and tertiary crude oil recovery in microporous rocks [4], and nearly four times increase of the heat transfer efficiency in microfluidic devices [5, 6]. On the contrary, the so-called Turbulent Drag Reduction phenomenon occurs at high Re number and moderate Wi number flow regime, in which the resistance of polymer solution in pipe flows could be significantly lower than that of Newtonian fluid with the same viscosity [7, 8, 9]. Thus Elastic Turbulence and Turbulent Drag Reduction could be considered as two extreme manifestations of non-linear dynamics of polymer solutions. The overriding challenge in regulating the turbulent flow effectively in engineering process relies on better understanding of strongly correlated dynamic processes with a hierarchy of the characteristic length and time scales. Molecular interactions control the equilibrium microstructure which forms on mesoscopic length scales. This in turn interacts with the hydrodynamic fields which are essentially macroscopic scale.

The essential physics of complex fluids lies in the constitutive relationship which forms a bridge between flow behaviour and microstructure evolution in flow. Doi-Edwards molecular theory on polymer dynamics [10] is a milestone breakthrough in prediction of the complex rheological behavior of entangled monodisperse polymer melts [11, 12] and makes it possible to systematically optimize the performance of polymer products from molecular to mesoscopic level. In the era of digital manufacturing, it is often required to find solutions of the inverse problem in order to not only optimize molecular and formulation design of polymeric fluids for given rheological properties, but also to identify the control variables in regulation of their multiple scale dynamic properties and even to guide the discovery of new physics [13] for construction or improvement of constitutive model with high physical fidelity. There are at least two nontrivial inverse problems. The first is to inversely solve the constitutive equation, which is often expressed in an implicit functional form with high dimensional parameter space [14], for estimation of its control variables or material functions from the known rheometric properties; The second is to construct a constitutive model based on discrete data sets on polymer properties as well as the physical principles. The present work will develop a general inverse learning method for solving the first problem and also lay down a foundation for solving the second.

Various methods have been proposed to estimate the parameters of physical models. A random search algorithm [15] used in a turbulence simulation requires a large computational resource and does not guarantee a good convergence. An alternative way [16, 17] is to directly map the outcomes of physical model to its parameters by training a deep neural network (DNN) from a series of the input parameters and the model outputs, and then to obtain the implicit molecular parameters. Raiddi et al. [18] developed the physics-informed neural networks (PINN) and successfully applied in the estimation of material parameters of Navier-Stokes equation and the unknown pressure field from some observed velocity field. The loss functions of PINN is constructed by imposing the conservation laws of fluid dynamics expressed in partial differential equations (PDEs) as the constraints of machine learning. Reyes et al. [19] extended the PINN method for learning the effective viscosity of the generalized Newtonian fluids from experimental measurements of velocity and pressure field in time-dependent three-dimensional flows. Xu et al. [20] formulated a physics constrained inverse learning method for modeling viscoelastic properties from observed displacement data. Mahmoudabadbozchelou et al. [21] proposed a multifidelity neural network (MFNN) architecture for data-driven physics-informed constitutive metamodeling of complex fluids. Their results show that the MFNN algorithms outperformed a purely data-driven classical DNNs and could capture the effects of experimental temperature, the salt concentration, as well as aging within and outside the range of training data parameters on the rheometric properties of complex fluids.

The intrinsic properties of complex fluids and the dominant factors for controlling fluid responses to external forces are often quantified experimentally under the well-defined rheometric flow either in simple shear or extensional deformation. Constitutive models could be then constructed and validated for reliable prediction of complex flows. In this paper, a DNN will be proposed and trained by a physics-informed molecular constitutive equation over its parameter space for modeling the rheometric properties of monodisperse polymer fluids. A general method is proposed to solve an inverse problem for estimating the model parameters of the molecular constitutive model. The method is then validated by computing the model parameters from a series of rheometric data for entangled completely monodisperse linear lambda DNA solution and extremely polydisperse high molecular weight polymer aqueous solutions. The rest of the article is organized as follows. The problem settings is described in Section 2. The methodology is presented in Section 3. The results and discussion are reported in Section 4. The conclusions are drawn in Section 5.

2. Problem setting

A general viscoelastic constitutive equation can be written in a form of

$$\boldsymbol{\sigma}(\mathrm{t}) = \boldsymbol{\Psi}[\boldsymbol{D}(\mathrm{t}); \boldsymbol{\Theta}], \tag{1}$$

where $\boldsymbol{\sigma}(\mathrm{t})$ is a general viscoelastic stress tensor and is a functional of strain rate tensor $\boldsymbol{D}(\mathrm{t})$ or strain history tensor at any instant of physical time t and a set of the physical model parameters $\boldsymbol{\Theta} = [\theta_1, \theta_2, ...\theta_n]$, which defines a specific complex fluid. Instead of computing the stress response $\boldsymbol{\sigma}(\mathrm{t})$ from a set of known model parameters $\boldsymbol{\Theta}$ and specified $\boldsymbol{D}(\mathrm{t})$, the inverse problem is to solve those unknown model parameters $\boldsymbol{\Theta}$ of the specified constitutive equation $\boldsymbol{\Psi}$ from a set of the known data of $\boldsymbol{\sigma}(\mathrm{t})$ and $\boldsymbol{D}(\mathrm{t})$ obtained, e.g., from rheometric experiments.

As an illustration example without loss any generality, the Rolie-Poly model [22] for modeling dynamics of entangled polymer and wormlike micellar fluids is considered here. It is simplified from the full Doi-Edwards molecular theory [10, 11] and could account for the main dynamic mechanisms of entangled polymeric fluids, including reptation, convective constraint release, chain stretch etc. The conformation tensor of polymer chain $\boldsymbol{A}(\mathrm{t})$ under a velocity field $\boldsymbol{u}$ evolves in time as

$$\frac{\mathrm{D}\boldsymbol{A}}{\mathrm{Dt}} = \boldsymbol{L} \cdot \boldsymbol{A} + \boldsymbol{A} \cdot \boldsymbol{L}^{\mathrm{T}} + \boldsymbol{f}(\boldsymbol{A}), \tag{2}$$

where $\frac{\mathrm{D}\boldsymbol{A}}{\mathrm{Dt}}$ is the material derivative, $\boldsymbol{L} = \boldsymbol{\nabla u}$ is the velocity gradient tensor and $\boldsymbol{L}^T$ is its transpose. For the non-Gaussian version of the Rolie-Poly model [23], which could account for the finite extensibility of polymer chains, the model function $\boldsymbol{f(A)}$ is defined as

$$\boldsymbol{f}(\boldsymbol{A}) = -\frac{1}{\tau_{\mathrm{D}}}(\boldsymbol{A} - \boldsymbol{I}) - \frac{2}{\tau_{\mathrm{R}}} k_{\mathrm{s}}(\lambda)\left(1 - \sqrt{\frac{3}{\operatorname{tr}\boldsymbol{A}}}\right)\left[\boldsymbol{A} + \beta\left(\frac{\operatorname{tr}\boldsymbol{A}}{3}\right)^{\delta}(\boldsymbol{A} - \boldsymbol{I})\right], \tag{3}$$

where $\boldsymbol{I}$ is the unit tensor, τ_{D} is the reptation time or the disengagement time of polymer chain, τ_{R} is the longest Rouse time or stretch time of polymer chain, β is the coefficient of the convective constraint release, $k_{\mathrm{s}}(\lambda)$ is the nonlinear spring coefficient and is approximated by

$$k_{\mathrm{s}}(\lambda) = \frac{\left(3 - \lambda^2/\lambda_{\max}^2\right)\left(1 - 1/\lambda_{\max}^2\right)}{\left(1 - \lambda^2/\lambda_{\max}^2\right)\left(3 - 1/\lambda_{\max}^2\right)}, \tag{4}$$

where $\lambda = \sqrt{\operatorname{tr}\boldsymbol{A}/3}$ is the chain stretch ratio with respect to its equilibrium conformation, $\lambda_{\max}$ is the fixed maximum stretch ratio of chain molecules. Thus, the polymeric stress could be readily calculated from its conformation tensor $\boldsymbol{A}(\mathrm{t})$ by

$$\boldsymbol{\sigma}_{\mathrm{p}}(\mathrm{t}) = G k_{\mathrm{s}}(\lambda)[\boldsymbol{A}(\mathrm{t}) - \boldsymbol{I}] = \frac{\eta_{\mathrm{p}}}{\tau_{\mathrm{D}}} k_{\mathrm{s}}(\lambda)[\boldsymbol{A}(\mathrm{t}) - \boldsymbol{I}], \tag{5}$$

where $G = \frac{\eta_{\mathrm{p}}}{\tau_{\mathrm{D}}}$ is the plateau modulus and η_{p} is the zero-shear polymer viscosity. The total stress tensor could be written as

$$\boldsymbol{\sigma}(\mathrm{t}) = 2\eta_s \boldsymbol{D}(\mathrm{t}) + \boldsymbol{\sigma}_{\mathrm{p}}(\mathrm{t}), \tag{6}$$

where the symmetric deformation rate tensor $\boldsymbol{D}(\mathrm{t}) = \frac{1}{2}\left[\boldsymbol{L} + \boldsymbol{L}^T\right]$ and η_s is the solvent viscosity. Obviously, the model parameters $\boldsymbol{\Theta} = [\eta_{\mathrm{p}}, \eta_s, \tau_{\mathrm{D}}, \tau_{\mathrm{R}}, \lambda_{\max}, \beta]$ define a specific Rolie-Poly fluid. Further details about the Rolie-Poly model could be found elsewhere, *e.g.* [22, 23].

3. Methodology

A general deep learning method for computing molecular parameters of viscoelastic constitutive model by solving inverse problems is proposed and outlined in Fig.1. It consists of two parts: 1) a DNN as an alternative representation for modeling molecular constitutive relationship; 2) a DNN based numerical solver for inversely computing the molecular parameters $\boldsymbol{\Theta}$ of constitutive model $\boldsymbol{\sigma}(\mathrm{t}) = \boldsymbol{\Psi}[\boldsymbol{D}(\mathrm{t}); \boldsymbol{\Theta}]$ from a set of known stress tensor $\boldsymbol{\sigma}$ and strain rate tensor D data using gradient descent as illustrated below.

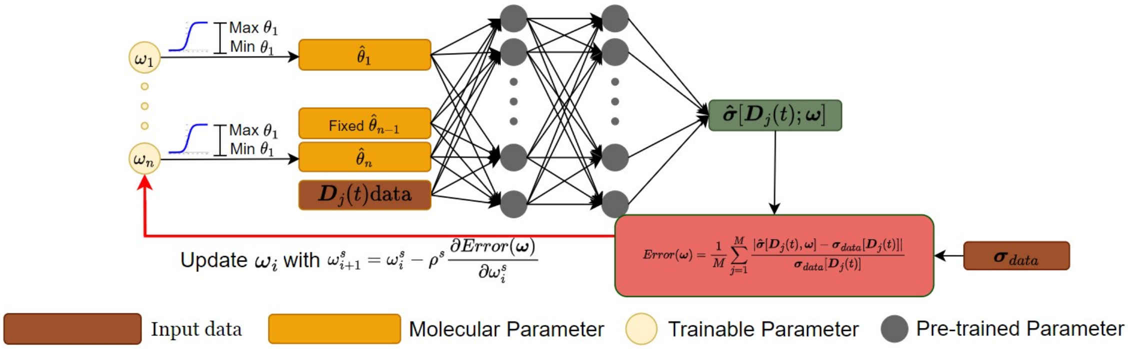


Fig. 1. A flowchart of a DNN based numerical solver for inversely computing molecular parameters.

3.1. DNN representation of molecular constitutive model

A fully-connected neural network will be firstly trained by exploring the parameter space of a given molecular constitutive model, e.g., the Rolie-Poly model presented in this work. It contains an input layer, a number of $N-1$ hidden layers and an output layer. The variables in the input layer $\mathbf{X}$ include a series of specified molecular parameters $\boldsymbol{\Theta} = [\theta_1, \theta_2, ...\theta_n]$ and strain rate tensor D. The transformation of the variables from the $(k-1)^{th}$ to the k^{th} layer can be written in the form of

$$\mathcal{L}_k\left(\hat{\mathbf{X}}^{k-1}\right) := \hat{\mathbf{y}}^k = \alpha^k(\mathbf{W}^k\hat{\mathbf{X}}^{k-1} + \mathbf{b}^k), \tag{7}$$

where k $=1,...,$N and $\hat{\mathbf{X}}^0 = \mathbf{X}$; $\mathbf{W}^k \in \mathbb{R}^{N_k \times N_{k-1}}$ are the weights of the connections between the k^{th} and the $(k-1)^{th}$ layer, $\mathbf{b}^k \in \mathbb{R}^{N_k}$ is the biases term of the k^{th} layer. α^k is a nonlinear activation function, e.g. Tanh, applied to each component of the transformed vector from each layer to make the transformation nonlinear. Note that no nonlinear activation function is used in the transformation of last layer $\mathcal{L}_N$. After the N^{th} transformation, the output of the DNN can be written as composition of $\mathcal{L}_k$ functions as,

$$\boldsymbol{\sigma}_{\Xi}(\mathbf{X}) = \mathcal{L}_N \circ \ldots \circ \mathcal{L}_k \circ \ldots \circ \mathcal{L}_1(\mathbf{X}), \tag{8}$$

where $\Xi = \left\{\mathbf{W}^k, b^k\right\}_{k=1}^{N}$ represents the trainable parameters of the DNN. It will be trained by minimising the loss function defined as:

$$Loss = \frac{1}{M}\sum_{j=0}^{M}(\boldsymbol{\sigma}[\boldsymbol{D}_j(\mathrm{t});\boldsymbol{\Theta}] - \hat{\boldsymbol{\sigma}}[\boldsymbol{D}_j(\mathrm{t});\boldsymbol{\Theta}])^2 + \xi \sum_{w_i \in \mathbf{W}} w_i^2, \tag{9}$$

where the first term is a measure of the total errors between the DNN output, $\hat{\boldsymbol{\sigma}}[\boldsymbol{D}(\mathrm{t});\boldsymbol{\Theta}]$, and the theoretical value, $\boldsymbol{\sigma}[\boldsymbol{D}(\mathrm{t});\boldsymbol{\Theta}]$, predicted by the target constitutive equation. M is a number of strain rate data points sampled in the training. The second term is L2 regularization and ξ is L2 regularization rate for weight functions to prevent over fitting. The minimization could be formulated as an optimization problem and solved by stochastic gradient descent (SGD) algorithm, within which the trainable parameters are updated iteratively as the following,

$$\Xi^{m+1} = \Xi^m - \eta_l \nabla_{\Xi^m} J^m(\Xi), \tag{10}$$

where $\eta_l > 0$ is the learning rate and m represents the m^{th} iteration. A variant of the SGD optimization algorithm, ADAM[24], will be used here.

In general, such a DNN could approximate complex functions to any desired accuracy as long as sufficiently many hidden units and data are available [25, 26]. The DNN will be trained in a sufficient accuracy to represent the target constitutive model over its molecular parameter space and strain rates under trained.

3.2. A DNN based numerical solver for inversely computing molecular parameters

Fig.1 shows a flowchart of a DNN based numerical solver for inversely computing molecular parameters. For any given set of the viscoelastic stress $\boldsymbol{\sigma}_{data}[\boldsymbol{D}_j(\mathrm{t})]$ obtained under a known deformation rate $\boldsymbol{D}_j(\mathrm{t})$ where $j \in [1, M]$ and M is a total number of the data points, initial trial parameters $\boldsymbol{\omega} = [\omega_1, \omega_2 ..., \omega_i ..., \omega_n]$ are randomly selected over their parameter space. The trial parameters $\boldsymbol{\omega}$ are then converted to the model parameter $\boldsymbol{\Theta}$ with its i^{th} parameter defined as

$$\hat{\theta}_i = (Max\ \theta_i - Min\ \theta_i) g(\omega_i) + Min\ \theta_i \tag{11}$$

where $Min\ \theta_i$ and $Max\ \theta_i$ are the minimum and maximum value of the i^{th} molecular parameter, respectively, between which the DNN has been pre-trained in a sufficient accuracy to represent the molecular constitutive model; the function g is defined as

$$g(\omega_i) = \frac{1}{1 + e^{-\omega_i}} \tag{12}$$

and is a $Sigmoid$ function with the monotonic property over its variable range of $g(\omega_i) \in [0, 1]$. The stress tensor $\hat{\boldsymbol{\sigma}}[\boldsymbol{D}_j(\mathrm{t}), \boldsymbol{\omega}]$, as a function of the known deformation rate $\boldsymbol{D}_j(\mathrm{t})$ and the trial parameters $\boldsymbol{\omega}$, is then computed by the pre-trained DNN. The outcomes are measured by an error function defined as

$$Error(\boldsymbol{\omega}) = \frac{1}{M} \sum_{j=1}^{M} \frac{|\hat{\boldsymbol{\sigma}}[\boldsymbol{D}_j(\mathrm{t}), \boldsymbol{\omega}] - \boldsymbol{\sigma}_{data}[\boldsymbol{D}_j(\mathrm{t})]|}{\boldsymbol{\sigma}_{data}[\boldsymbol{D}_j(\mathrm{t})]}. \tag{13}$$

By minimizing the error function, the inverse problem is solved by the gradient decent method, in which the gradient is computed through automatic differentiation[27][28]. In the s^{th} optimization iteration, the i^{th} parameter ω_i is updated as,

$$\omega_i^{s+1} = \omega_i^s - \rho^s \frac{\partial Error(\boldsymbol{\omega})}{\partial \omega_i^s} \tag{14}$$

where ρ^s is an iteration parameter and will rapidly decrease when the error function could not be minimized further. The above optimization iteration is repeated until a convergence criterion is satisfied as

$$|\boldsymbol{\omega}^{s+1} - \boldsymbol{\omega}^s| < \epsilon \tag{15}$$

where ϵ is set in advance as a very small value. Moreover in order to validate the independence of the solution on the initial conditions, the inverse problem could be solved by randomly selecting several different sets of the initial trial parameters $\boldsymbol{\omega}$.

4. Results and discussion

4.1. Training DNNs for modeling entangled polymer solutions

As an illustration of training DNNs for modeling entangled polymer aqueous solutions, Table 1 shows the parameter variation of the Rolie-Poly model. The solvent viscosity η_s is usually considered as a constant and in this particular case set as the viscosity of water [29, 30]. The training data were generated by sampling the model parameter space. For the input layer, each molecular parameter was sampled uniformly in either linear scale or logarithmic scale over its variation range. At least ninety data points uniformly distributed in the logarithmic scale under simple shear flow over a wide range of shear rate from 10^{-4} to 10^{4}. For each set of the sampled model parameters and shear rate, the corresponding output layer data were obtained by computing the steady viscoelastic stresses from the Rolie-Poly model. The DNN was trained through 10,000 sets of training data and subsequently validated by additional 500 sets of sampling data. The learning rate of the DNN was tuned around 0.001 to obtain a reasonably good learning convergence. The DNN with the overall lowest relative error between the viscoelastic stresses calculated by the DNNs and the original model is comprised of 5 hidden layers and 192 neurons, and will be used in the present studies here after.

Table 1. A range of parameter variation of the Rolie-Poly model

Parameter	Variation range (SI unit)
η_P	$1 \sim 6,000\ Pa.s$
λ_D	$1 \sim 5,000s$
λ_R	$1 \sim 50s$
χ_{max}	$15 \sim 20$
β	$1 \sim 25$

4.2. Validation of Convergence

In order to validate the DNN based solver for inversely computing molecular parameters, three sets of the model parameters arbitrarily sampled over the range specified in Table 1 were considered as the solutions to be found. The corresponding viscoelastic stresses over a range of shear rate could be readily calculated from the trained DNN and used as the inputs for solving the inverse problem. The initial values of the model parameters were randomly generated within the range specified in Table 1. As shown in Fig2, the loss of the DNN based solver along with the relative error of the estimated molecular parameters converge rapidly as a number of iterations is increased. The inverse problem could be solved accurately. However, the converging rate for the solutions of the model parameter η_P and λ_D is relatively slower than those of other parameters, due to their much wider variation range.

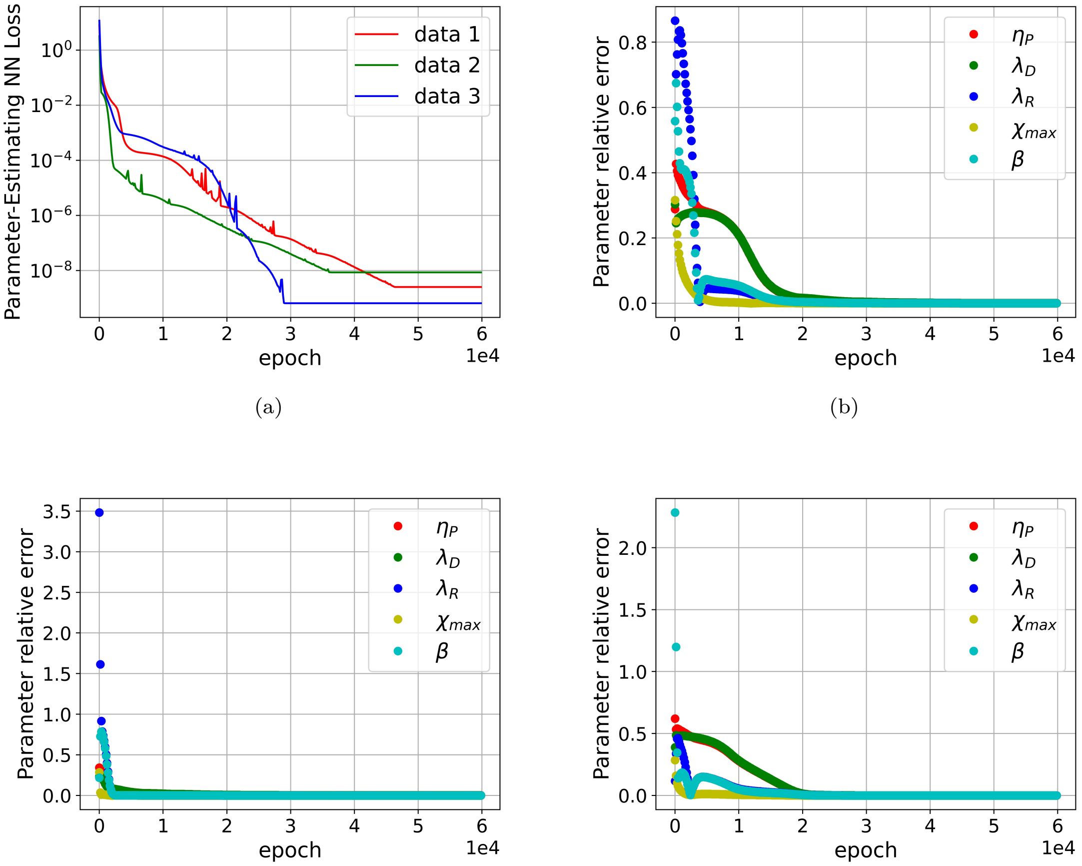
Instead of using the trained DNN, the corresponding viscoelastic stresses over a range of shear rate could be directly calculated from the Rolie-Poly model and use as the inputs for solving the inverse problem. By following the same procedure as the above for inversely computing the molecular parameters, the loss of the DNN based solver along with the relative error of the estimated molecular parameters are plotted against a number of iterations and shown in Fig3. In comparison with Fig 2, the reasonably good convergence could be achieved. As shown in Fig3(b), there is about 10% relative error for the model parameters η_P and λ_D. It could be further minimized by increasing the accuracy in the DNN approximation of the Rolie-Poly model.

4.3. Effects of input data noise

In order to evaluate the effects of stress data noise on the estimation of the molecular parameters, the white noise were superimposed into the viscoelastic stress data so that the input data deviates from their original values randomly in a certain maximum percentage. At a given noise level, eight sets of such an input noise data were generated. Each set of the noise input data was solved ten times by the DNN based solver using randomly generated initial values over the parameter range to validate the consistency of the solutions. The mean and standard deviation of each model parameter were then calculated and presented in Fig4, in which the estimated parameters are normalized by their solutions computed by using the input data without any noise. The error bars represent the standard deviation of the means, respectively. Over the noise level between 0 and 0.02, the changes of the mean values are within their standard deviation and are not statistically significant. However as the noise level increases up to 0.03, the mean solution values of the parameters η_P and λ_D are statistically different from the original one. Further increasing the noise level above 0.03, the mean solution values of the parameters η_P and λ_D are shifted even more from the original solutions and then saturated from the noise level of 0.04 onward. The standard deviation also increases slightly. Over the variation range of the noise level, the mean solution values of the parameters λ_R, χ_{max} and β are statistically indistinguishable from their original solutions. The results simply reflect the fact shown in Table 1 that the variation ranges of the parameters η_P and λ_D are about two orders magnitude higher than the those of the parameters λ_R, χ_{max} and β, hence much more sensitive to the stress data noise.

4.4. Validation by completely monodisperse entangled polymer solution

Banik et al. [29] recently reported the molecular parameters extracted from the linear viscoelastic data of completely monodisperse ultrahigh-molecular-weight linear lambda DNA solutions over various entangled concentration. In order to validate the DNN based solver for inversely computing the molecular parameters, the steady shear stress and the first normal stress difference were generated by the Rolie-Poly model using those extracted molecular parameters over a range of shear rate from 10^{-4} to 10^{4}. Taking the steady stress



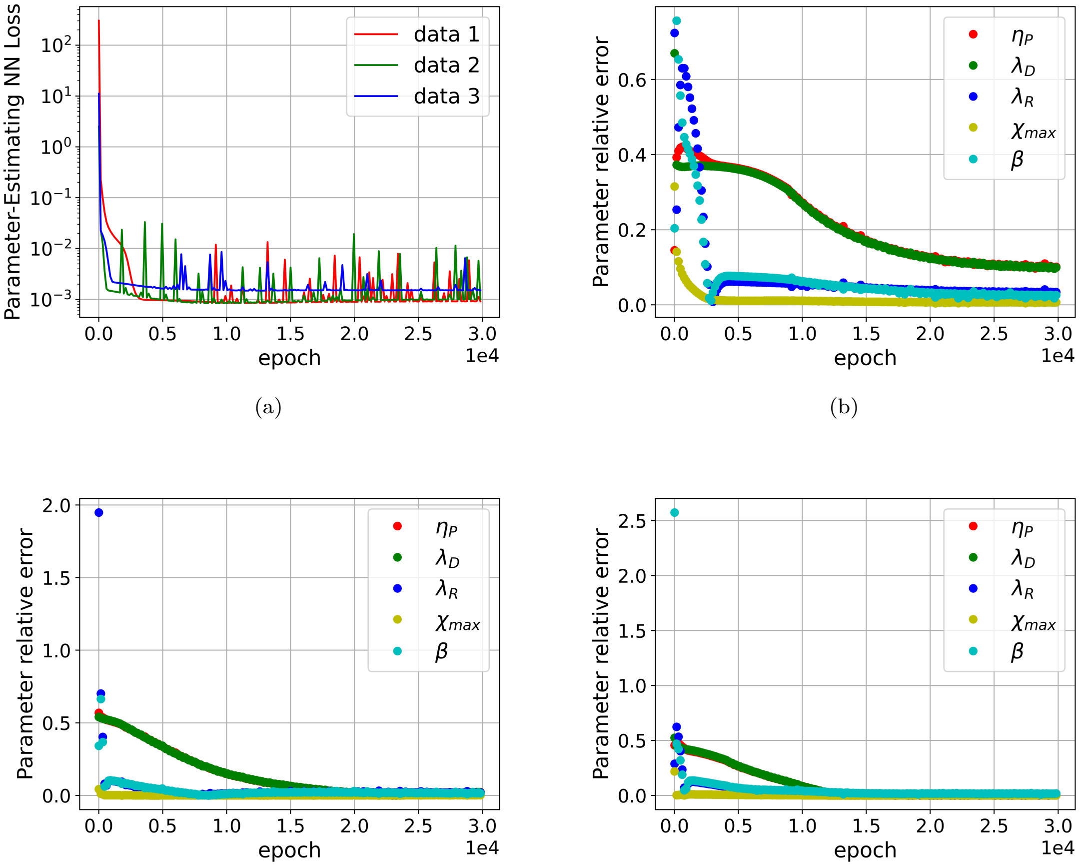
(a) (b)

(c) (d)

Fig. 2. The convergence of the DNN based solver for inversely computing molecular parameters. (a) the loss of the DNN based solver is plotted against a number of iterations; (b)(c)(d) the relative errors of the estimated molecular parameters are plotted against a number of iterations for the testing sample 1, 2, 3, respectively

Table 2. The molecular parameters inversely solved by the DNN based solver for the lambda DNA solution over various entangled concentrations [29]

Concentration	0.82 mg/mL			1.01 mg/mL			1.55 mg/mL			2.06 mg/mL		
Parameter	Extracted	DNN-2k	DNN-10k	Extracted	DNN-2k	DNN-10k	Extracted	DNN-2k	DNN-10k	Extracted	DNN-2k	DNN-10k
η_P	12.4	12.45±0.04	12.53±0.02	22.6	22.47±0.02	22.58±0.02	474.4	475.72±0.44	473.47±1.16	4811.7	4818.38±0.15	4784.20±0.37
λ_D	87.0	88.1±0.1	88.00±0.09	109.0	109.10±0.07	109.00±0.01	1006.0	996±1.4	998±3.6	4092.0	4109.50±0.01	4095.10±0.02
λ_R	11.0	10.7±0.1	10.7±0.1	14.0	12.1±0.1	13.00±0.08	21.0	21.0±0.5	20.5±0.2	29.0	31.50±0.01	31.70±0.02
χ_{max}	18.0	17±1.2	19±1.9	18.0	17±2.4	17.2±0.7	18.0	17±2.1	19±1.1	18.0	18.300±0.002	17.500±0.004
β	20.0	18.8±0.3	19.5±0.3	13.0	11.3±0.1	12.20±0.06	5.00	5.0±0.07	5.00±0.05	1.0	1.000±0.001	1.000±0.002
Error of N_1		0.0088	0.0076		0.0063	0.0025		0.0098	0.0017		0.0059	0.005
Error of sheer stress		0.0027	0.0037		0.0048	0.0017		0.0094	0.0019		0.0037	0.0035



(a) (b)

(c) (d)

Fig. 3. (a) is the Parameter-Estimating loss throughout the training process with randomly sampled data. (b)(c)(d) are the molecular parameter relative error curve for data index 1,2,3,respectively.

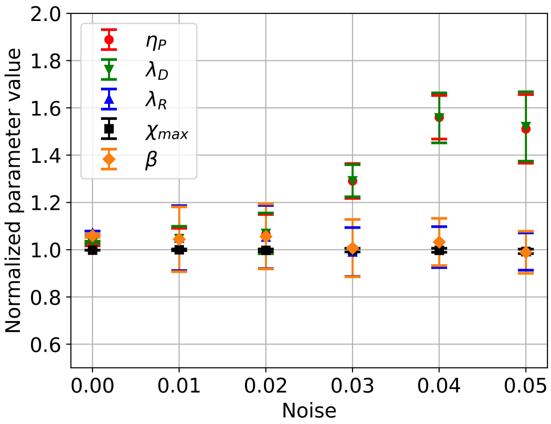


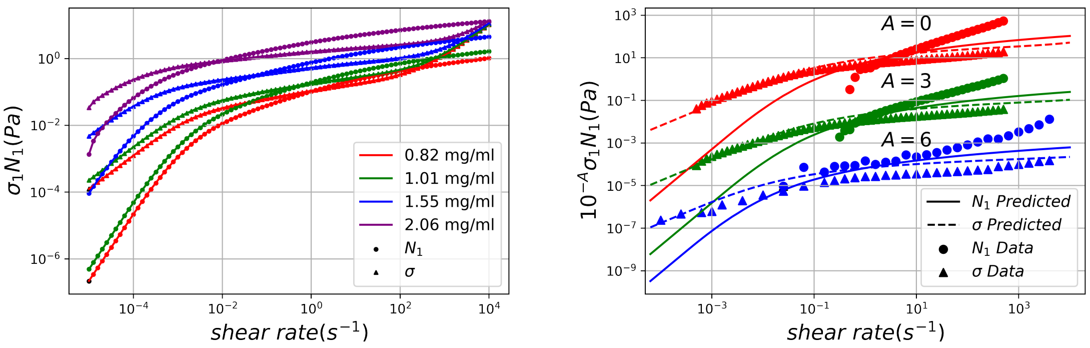
Fig. 4. A plot of the normalised mean and standard deviation of the model parameters against the noise level.

Table 3. The molecular parameters inversely solved by the DNN based solver for highly polydisperse polyacrylamide aqueous solution over semi-dilute entangled concentration region [30], where c* is the overlap concentration of polyacrylamide in aqueous solution.

Parameter Concentration	η_P	λ_D	λ_R	χ_{max}	β
48c*	65.1±0.01	3.9±0.002	49.8±0.0001	15.0±1e-05	25.5±0.0001
97c*	171.4±0.03	4.5±0.001	49.8±7e-05	15.0±1e-05	25.5±3e-05
150c*	1734.6±0.1	23.7±0.002	46.6±0.0002	16.5±8e-05	1.0±2e-05

of the DNA solution as the input data, the inverse problem was solved by the DNN based solver with the gradient decent method as an optimizer. For each concentration of the DNA sample, the model parameters were solved by the DNN based solver ten times with different set of the initial values of the Rolie-Poly model parameter randomly generated over its parameter range. As shown in Table2, the averaged solutions of the molecular parameters from ten different sets of the initial values and their standard deviations were listed along with the mean relative errors of the steady first normal stress difference and the steady shear stress over about 91 shear rates. The solutions from the DNN based solver are in a good agreement with the extracted molecular parameters. As shown in Fig5(a), the steady viscoelastic stress vs. shear rate curves calculated by the DNN with the model parameter obtained from solving the inverse problem are in excellent agreement with those predicted by the Rolie-Poly model with the originally extracted parameters. The effects of data size in the training DNN are also shown in Table2. The solutions obtained from the DNN trained with 2000 data sets are in reasonably good agreements with the results obtained from 10000 data sets. The concentration scaling analysis of η_P and λ_D shows that their power law exponents are 6.63 and 4.38, respectively. The results agree favourably with the exponents of 6.7 ± 0.3 and 4.7 ± 0.3 estimated from the linear viscoelastic data of completely monodisperse lambda DNA solutions by the time-concentration superposition method [29, 30]. Interestingly the DNN based solver not only could obtain the solutions of the model parameters accurately, but also resolve the power-law concentration scaling of the materials function of the DNA solutions satisfactorily. Note that such a concentration scaling was not explicitly built in the DNN nor in the Rolie-Poly model. It was trained implicitly into the DNN by sampling over the materials parameter space of the Rolie-Poly model.

By following the same procedure, the DNN based solver has been applied for inversely extracting the molecular parameters of the Rolie-Poly model from the experimental data of highly polydisperse and high molecular weight polyacrylamide aqueous solution over semi-dilute entangled concentration region [30]. The results are listed in Table3. The concentration scaling analysis of η_P and λ_D shows that their power law exponents are 2.74 and 1.45, respectively. The results are in a good agreement with the exponents of 2.48 ± 0.09 and 1.40 ± 0.06 estimated from the experimental data by the time-concentration superposition



(a) (b)

Fig. 5. Comparison of the original input stress vs shear rate data with the data calculated by the Rolie-Poly model with the parameters inversely solved by the DNN based solver (a) for the completely monodisperse ultrahigh-molecular-weight linear lambda DNA solutions;(b) for the highly polydisperse and high molecular weight polyacrylamide aqueous solution.

method [30]. Obviously, the DNN based solver could capture the concentration scaling of material function satisfactorily even for highly polydisperse polymer solutions. Fig5(b) shows a comparison of the shear stress and the first normal stress difference as a function of shear rate between the experimental data and the calculated results by the Rolie-Poly model using the molecular parameters obtained from the DNN based solver. Although the Rolie-Poly model gives a reasonably good fit, there are still significant deviation between the model prediction and the experimental data, especially on the first normal stress difference in the high shear rate region. The sources of the deviation are mainly from the Rolie-Poly model itself as the DNN training was with respect to monodisperse polymer solutions. At present the Rolie-Poly model with multiple modes could not yet accurately describe the nonlinear fluid flows of highly polydisperse semi-dilute entangled polymer solutions.

5. Conclusion

A general deep learning method for computing molecular parameters of viscoelastic constitutive model is proposed. It utilizes a DNN as an alternative representation for modeling molecular constitutive relationship and a DNN based numerical solver for inversely computing the molecular parameters of the pre-trained constitutive model. Without loss much generality, the method has been validated by considering the Rolie-Poly model for modeling the linear and non-linear dynamics of entangled polymer solutions in a wide range of concentrations. The results show that as long as the proposed DNN could represent the Rolie-Poly model in sufficiently high accuracy, the DNN based numerical solver would rapidly converge to its solution during inversely computing molecular parameters. The solution of the DNN based solver is robust against small white noise disturbance to the input stress data. However if the input stress significantly deviates from the original stress, the DNN based solver could readily distinguish such a difference and converge to a different solution, hence demonstrates the resolution of the numerical solver for inversely computing molecular parameters.

The molecular parameters computed by the DNN based numerical solver from the steady stress data of completely monodisperse linear lambda DNA solution not only reproduce accurately their steady stress over a wide range of shear rate at various concentrations, but also show the power law concentration scaling with nearly same scaling exponent as those estimated from the experimental results. Even for highly polydisperse polyacrylamide aqueous solution, the DNN based numerical solver also resolve the power law concentration scaling satisfactorily. However the polydispersity of polymer solutions results in significant deviation between the experimental steady stress of polyacrylamide aqueous solution and the calculated results by the Rolie-Poly model using the molecular parameters obtained from the DNN based solver. The deficits of the Rolie-Poly model could be remedied by further development of molecular constitutive model for polydisperse polymeric fluids. Such a theoretical approach could be complemented by a data driven approach through deep learning multiple scale experimental and computational data.

The proposed DNN based solver is a general method for inversely computing material function of constitutive model and in principle could be applied to other complex fluids. The physics-informed DNN captures the fundamental features of complex fluids in linear and nonlinear dynamics, it could be further trained by experimental and computer simulation data to account for other effects defined through their material functions, including temperature, pressure, molecular mass and distribution, composition, solvent quality, etc., hence to establish an highly efficient approach in optimizing molecular and formulation design of complex fluids by solving an inverse problem as well as in studying complex fluid dynamics, in complementary of the traditional approaches.

Acknowledgments

Acknowledgments: Financial support by the Ministry of Science and Technology of the People's Republic of China under the Grant No.: 2020AAA0104804 and 2020AAA0104800 are gratefully acknowledged.

References

[1] A. Groisman, V. Steinberg, Elastic turbulence in a polymer solution flow, Nature 405 (2000) 53–55.

[2] A. Groisman, V. Steinberg, Efficient mixing at low reynolds numbers using polymer additives, Nature 410 (2001) 905–908.

[3] A. Groisman, V. Steinberg, Elastic turbulence in curvilinear flows of polymer solutions, New Journal of Physics 6 (2004) 29.

[4] A. M. Howe, A. Clarke, D. Giernalczyk, Flow of concentrated viscoelastic polymer solutions in porous media: effect of mw and concentration on elastic turbulence onset in various geometries, Soft Matter 11 (2015) 6419–6431.

[5] B. Traore, C. Castelain, T. Burghelea, Efficient heat transfer in a regime of elastic turbulence, Journal of Non-Newtonian Fluid Mechanics 223 (2015) 62–76.

[6] W. M. Abed, R. D. Whalley, D. J. Dennis, R. J. Poole, Experimental investigation of the impact of elastic turbulence on heat transfer in a serpentine channel, Journal of Non-Newtonian Fluid Mechanics 231 (2016) 68–78.

[7] P. S. Virk, Drag reduction fundamentals, AIChE Journal 21 (1975) 625–656.

[8] C. M. White, M. G. Mungal, Mechanics and prediction of turbulent drag reduction with polymer additives, Annu. Rev. Fluid Mech. 40 (2008) 235–256.

[9] L. Xi, Turbulent drag reduction by polymer additives: Fundamentals and recent advances, Physics of Fluids 31 (2019) 121302.

[10] M. Doi, S. F. Edwards, S. F. Edwards, The theory of polymer dynamics, volume 73, oxford university press, 1988.

[11] R. S. Graham, A. E. Likhtman, T. C. McLeish, S. T. Milner, Microscopic theory of linear, entangled polymer chains under rapid deformation including chain stretch and convective constraint release, Journal of Rheology 47 (2003) 1171–1200.

[12] J. Bent, L. Hutchings, R. Richards, T. Gough, R. Spares, P. D. Coates, I. Grillo, O. Harlen, D. Read, R. Graham, et al., Neutron-mapping polymer flow: Scattering, flow visualization, and molecular theory, Science 301 (2003) 1691–1695.

[13] R. Iten, T. Metger, H. Wilming, L. Del Rio, R. Renner, Discovering physical concepts with neural networks, Physical review letters 124 (2020) 010508.

[14] H. C. Öttinger, Stochastic processes, polymer dynamics, and fluid mechanics, in: Stochastic Processes in Polymeric Fluids, Springer, 1996, pp. 1–15.

[15] N. Chen, A. J. Majda, A new efficient parameter estimation algorithm for high-dimensional complex nonlinear turbulent dynamical systems with partial observations, Journal of Computational Physics 397 (2019) 108836.

[16] A. M. Annaswamy, S.-H. Yu, /spl theta/-adaptive neural networks: a new approach to parameter estimation, IEEE transactions on neural networks 7 (1996) 907–918.

[17] B. M. Afkham, J. Chung, M. Chung, Learning regularization parameters of inverse problems via deep neural networks, Inverse Problems 37 (2021) 105017.

[18] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707.

[19] B. Reyes, A. A. Howard, P. Perdikaris, A. M. Tartakovsky, Learning unknown physics of non-newtonian fluids, Physical Review Fluids 6 (2021) 073301.

[20] K. Xu, A. M. Tartakovsky, J. Burghardt, E. Darve, Learning viscoelasticity models from indirect data using deep neural networks, Computer Methods in Applied Mechanics and Engineering 387 (2021) 114124.

[21] M. Mahmoudabadbozchelou, M. Caggioni, S. Shahsavari, W. H. Hartt, G. Em Karniadakis, S. Jamali, Data-driven physics-informed constitutive metamodeling of complex fluids: A multifidelity neural network (mfnn) framework, Journal of Rheology 65 (2021) 179–198.

[22] A. E. Likhtman, R. S. Graham, Simple constitutive equation for linear polymer melts derived from molecular theory: Rolie–poly equation, Journal of Non-Newtonian Fluid Mechanics 114 (2003) 1–12.

[23] K. K. Kabanemi, J.-F. Hétu, Nonequilibrium stretching dynamics of dilute and entangled linear polymers in extensional flow, Journal of non-newtonian fluid mechanics 160 (2009) 113–121.

[24] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[25] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural networks 4 (1991) 251–257.
[26] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (1989) 359–366.
[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).
[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).
[29] S. Banik, D. Kong, M. J. San Francisco, G. B. McKenna, Monodisperse lambda dna as a model to conventional polymers: A concentration-dependent scaling of the rheological properties, Macromolecules 54 (2021) 8632–8654.
[30] Y. Fan, A. Lanzaro, X.-F. Yuan, Universal concentration scaling on rheometric properties of polydisperse and high molecular weight polyacrylamide aqueous solutions, Chinese Journal of Polymer Science (2022).

Supplementary Material

Supplementary material