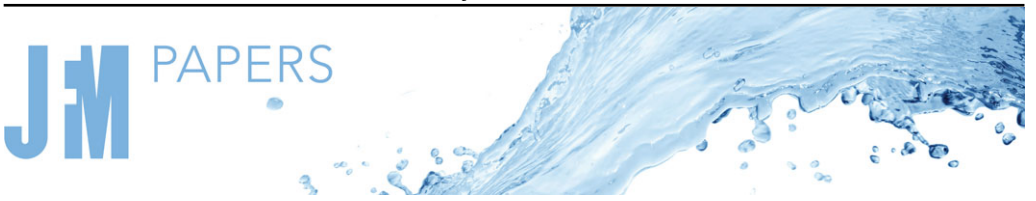
J. Fluid Mech. (2025), *vol.* 1016, A11, doi:10.1017/jfm.2025.10325



Hammering at the entropy: a GENERIC-guided approach to learning polymeric rheological constitutive equations using PINNs

David Nieto Simavilla[1], **Andrea Bonfanti**[2,3,4], **Imanol García-Beristain**[5], **Pep Español**[6] **and Marco Ellero**[4,7,8]

[1]Dept. Energía y Combustibles, Escuela Técnica Superior de Ingenieros de Minas y Energia, Universidad Politécnica de Madrid, Madrid, Spain

[2]BMW AG, Digital Campus Munich, Munich, Germany

[3]University of the Basque Country (UPV/EHU), Bilbao, Spain

[4]Basque Center for Applied Mathematics (BCAM), Bilbao, Spain

[5]Applied Mathematics Department, Engineering School of Bilbao, University of the Basque Country (UPV/EHU), Bilbao, Spain

[6]Dept. Física Fundamental, Universidad Nacional de Educación a Distancia, Madrid, Spain

[7]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

[8]Complex Fluids Research Group, Department of Chemical Engineering, Faculty of Science and Engineering, Swansea University, Swansea, UK

Corresponding author: David Nieto Simavilla, david.nsimavilla@upm.es

(Received 3 October 2024; revised 14 April 2025; accepted 27 May 2025)

We present a versatile framework that employs Physics-Informed Neural Networks (PINNs) to discover the entropic contribution that leads to the constitutive equation for the extra-stress in rheological models of dilute polymer solutions. In this framework the training of the neural network is guided by an evolution equation for the conformation tensor, which is GENERIC-compliant. We compare two training methodologies for the data-driven PINN constitutive models: one trained on data from the analytical solution of the Oldroyd-B (OB) model under steady-state rheometric flows (PINN-rheometric), and another trained on *in silico* data generated from computational fluid dynamics (CFD) simulations of complex flow around a cylinder that use the OB model (PINN-complex). The capacity of the PINN models to provide good predictions is evaluated by comparison with CFD simulations using the underlying OB model as a reference. Both models are capable of predicting flow behaviour in transient and complex conditions; however, the PINN-complex model, trained on a broader range of mixed-flow data, outperforms the PINN-rheometric model in complex flow scenarios. The geometry agnostic character of

our methodology allows us to apply the learned PINN models to flows with topologies different from those used for training.

Key words: rheology, viscoelasticity, non-Newtonian flows

1. Introduction

Rheology aims to predict the complex flows of viscoelastic fluids using balance equations, equations of state (EOS) and constitutive equations. These equations can be solved once initial and boundary conditions are provided. While balance equations are universal, EOS and constitutive equations are system specific. Therefore, models for the EOS, relating temperature and pressure to mass and energy densities, and for the constitutive equations, relating stress and microstructural conformation tensors are required. The exact functional form of these models is generally unknown (Bird *et al.* 1987; de Pablo & Schieber 2014).

Traditionally, physical models are postulated based on physical intuition and are characterized by a small number of parameters that, ideally, can identify physical constants and material properties (Dyson 2004). In particular, rheologists have dedicated huge efforts to find good models that describe complex flow properties and behaviour (Oldroyd & Wilson 1950; Thien & Tanner 1977; Larson 1988; Herrchen & Öttinger 1997; Kröger 2010). Once a model is proposed, model parameters are typically determined through simple experiments (i.e. in rheology shear or extensional steady-state viscometric flows). These experiments are designed to fit or 'learn' the model's parameters in order to achieve good predictions. Models with a minimal number of parameters enable their determination with a limited amount of data from a few simple flow experiments involving shear or extensional steady-state viscometric flows. A fundamental problem in rheology is whether viscometric flow experiments are sufficient to obtain the parameters of a model to be applied in complex flow situations. A good physical model should describe a wide range of flow situations beyond those used to fit the parameters. However, there is no guarantee that a model performing well in simple flows will excel in complex flow situations. When the model fails, it must either be improved or replaced by a new one that incorporates additional physical insight. This process often entails identifying key physical features, such as finite extensibility, constraints imposed by entanglements, etc., that better capture the system's behaviour. The systematic process of model refinement by including new physics, however, might be time-consuming, and we may also end up with something that looks very different to a good physical model with a small number of parameters (Dyson 2004).

An alternative, is to use a model with a sufficiently large number of parameters to fit the elephant (Dyson 2004). We lose the physical intuition concerning the behaviour of the model but, provided that a sufficiently large amount of data are available, we may automatize the entire process. Thanks to the universal approximation theorem (Hornik, Stinchcombe & White 1989), we may use a neural network (NN) to represent the functional form of a model with enough flexibility. The model is given by the composition of functions with very simple functional forms (like sigmoidal functions), resulting in a generic functional form depending on a large number of parameters. Using NN as the model, we may fit its parameters (train the network) in any region of the space of variables, be it the region of viscometric flows, or in arbitrary regions of complex flows. This allows us to simply retrain the NN model with new information when we require predictions for flow simulations of increasing complexity. Graphically, we hammer the function against data as a blacksmith would hammer a sheet of metal to shape it.

Traditional NNs have been used in multiscale simulation methods to retain the microscopic molecular fidelity at the macroscale (Lei, Wu & Weinan 2020). Neural networks, while powerful function approximators, often disregard established constitutive modelling principles and thermodynamic constraints. As any interpolation method, NNs will give good predictions in the vicinity of training points and will fail extrapolating to regions far from their training data (Bonfanti *et al.* 2023, 2024). Furthermore, NNs typically require large data sets to achieve a decent training (Rackauckas *et al.* 2021). In the context of rheology, data-driven approaches have been used to model nonlinear viscoelastic materials at small strains using NNs, requiring only stress and strain paths for training. These NNs can be tuned to satisfy some physical feature (i.e. convexity of the learned functional) to facilitate the handling of large data sets and noisy stress data (Rosenkranz *et al.* 2024).

Several lines of research aim at leveraging the potential advantages of machine learning methodologies in the field of rheology. Young *et al.* have employed scattering microstructure data to develop low dimensional constitutive models using a dimensionality reduction scheme (Young *et al.* 2023). Zhao and coworkers have employed yet a different machine learning methodology – Gaussian process regression – to learn constitutive models from microscopic simulations under simple shear flows. These constitutive models can then be used in macroscopic simulations (Zhao *et al.* 2018, 2021). However, an important limitation of these studies is that they set arbitrary constraints to the functional form of the learned rheological model. For example, setting the viscosity as function of shear rate alone (Zhao *et al.* 2018) or, having the microstructure description relying on a predefined FENE-P model to incorporate viscoelasticity (Zhao *et al.* 2021), thus these approaches lack generality. A more general, yet similar in the *ad hoc* choice of the functional form of the constitutive model is the work by Seryo *et al.* (2020). Gaussian process regression is also used to learn a constitutive model that introduces history-dependent viscoelasticity by considering the time derivative of the polymeric stress as a function of the flow velocity gradient and the stress (Seryo *et al.* 2020). However, as the model becomes more general the dimensionality also tends to grow. For example in Seryo *et al.* work, the derivative of the stress $\partial\boldsymbol{\tau}/\partial t(\boldsymbol{\nabla v}, \boldsymbol{\tau})$ with *a priori* six independent components for three-dimensional (3-D) problems should be learned from the velocity gradient $\boldsymbol{\nabla v}$ with nine independent components and the symmetric stress $\boldsymbol{\tau}$ with additional six independent components (i.e. mapping 15-dimensional space into six-dimensional space), making the approach difficult to apply in complex 3-D flows. A more effective, yet general, physics-informed dimensionality reduction is thus required to study complex 3-D flows.

Another alternative is to utilize Physics-Informed NNs (PINNs) (Raissi, Perdikaris & Karniadakis 2019), a novel family of NNs which are able to inherently satisfy kinematic, thermodynamic and physical constraints (Linka & Kuhl 2023). PINNs are NNs that incorporate model equations, such as partial differential equations (PDEs), directly into their structure. PINNs are currently employed to solve forward and inverse PDE problems (Raissi *et al.* 2019), fractional equations (Pang *et al.* 2019), integral–differential equations (Yuan *et al.* 2022) and stochastic PDEs (Zhang *et al.* 2019). These innovative functions act as a multitask learning framework where the NN simultaneously fits observed data and minimizes the residual of the selected PDE (Cuomo *et al.* 2022; Mahmoudabadbozchelou *et al.* 2022*b*). The introduction of governing equations in the loss function enables PINNs to offer a powerful framework for solving forward and inverse problems in fluid mechanics, where the solutions to momentum balance equations that incorporate complex constitutive models are produced as predictions of the NN (Karniadakis *et al.* 2021). However, most of the studies using PINNs have focused on predictions of specific flow simulations (i.e. using

the network to solve momentum balance equations) and then choose the right constitutive model from a selection of 'known' analytical constitutive models rather than learning 'unknown' constitutive models (Meng *et al.* 2020; Lin *et al.* 2023). For example, Thakur *et al.* used PINNs to select among several viscoelastic constitutive models (Oldroyd-B (OB), Giesekus or linear PTT) and learn the stress field from a velocity field (Thakur, Raissi & Ardekani 2022). In Saadat, Mahmoudabadbozchelou & Jamali (2022) and Mahmoudabadbozchelou *et al.* (2022*a*), the authors have proposed the rheology-informed NNs (RhINNs) for forward and inverse metamodelling of complex fluids. The RhINNs are employed to solve the constitutive models with multiple ordinary differential equations by proposing a penalization based on a thixoviscoelastoplastic model for the stress, where a few model parameters are learnt. Again, the framework used to encapsulate the physics is over-restricting for general complex fluids.

Nevertheless, PINNs present a significant advantage over traditional data-driven models by ensuring that the NN solutions adhere to fundamental physical principles. This feature is particularly beneficial in the context of rheology, where the complexity of the flow behaviour demands models that respect the underlying physics while adapting to diverse and nonlinear phenomena. In this context a number of studies have employed PINNs to advance rheological modelling. An interesting use regarding the problem discussed above is to employ PINNs to model viscoelastic materials using deep NNs to approximate rate-dependent and nonlinear constitutive relationships (Xu *et al.* 2021). One way to impose a minimal set of thermodynamics-based constraints (i.e. first and second laws of thermodynamics) on constitutive models, still retaining generality, is through the application of the so-called GENERIC (General Equations of Non-Equilibrium Reversible–Irreversible Coupling) framework (Ottinger 2005). Hernandez *et al.* proposed structure-preserving NNs (Hernández *et al.* 2021). Zhang *et al.* proposed a GENERIC formalism informed NNs (GFINNs) that adhere to the formalism symmetric degeneracy conditions. GFINNs consist of two modules, each with two components, modelled by NNs specifically designed to meet these conditions. This componentwise architecture allows flexible integration of physical information into the networks (Zhang *et al.* 2022).

In this work, we present a general approach for polymer solutions using PINNs to determine the polymeric entropy leading to the constitutive equation for the stress in rheological models. Instead of training PINNs to predict arbitrary solutions of specific flow simulations, we aim to leverage their universal approximator nature to capture the general functional relation between the eigenvalues of the conformation tensor $\boldsymbol{c}$ and the polymeric entropy for '*a priori*' unknown viscoelastic models. The approach is not only GENERIC-compliant but also significantly reduces the problem dimensionality making model learning more efficient. In fact, it only requires learning a scalar state function (the entropy) as a function of the two or three eigenvalues (depending on space dimension) of the conformation tensor $\boldsymbol{c}$. We evaluate the traditional methods using limited regions of the available conformation space (i.e. limited data from steady-state rheometric flows) to establish rheological models that then can be used to predict properties and behaviour of more complex flows. We propose two types of data sets to train our PINN models: a first one – in analogy to classical rheological calibrations – with steady-state rheometric flows (later denoted as 'PINN-rheometric'); and a second one, with data from steady-state solutions of complex flow around a cylinder (later denoted as 'PINN-complex'). We study the application of the PINN models to finite volume (FV) simulations of complex flows coupling the learned models with an OpenFOAMs RheoTool solver (Alves, Pinho & Oliveira 2001; Pimenta & Alves 2017). We find that a PINN model trained in steady-state rheometric flows data can be used to produce reasonable predictions in moderate transient

or complex flows. However, in order to reproduce complex flows more accurately, data retrieved beyond viscometric flows are required for training.

In order to evaluate the quality of the PINN model predictions, we analyse the relative errors of the entropy and the stress in the space of the eigenvalues of $\boldsymbol{c}$. We generate *in silico* conformation-tensor data according to analytical and accurate RheoTool-discretizations of an OB model in simple and complex flows. Data are provided to the PINN for training, whereas its application to complex flow is agnostic of the underlying model used to generate them, therefore providing an optimal and controlled framework for fair numerical comparisons. The PINN model presented here can effectively learn unknown forms of the polymeric entropy and integrate their GENERIC-guided NN representation into RheoTool to perform data-driven flow simulations. The only data required to train these models are the conformation tensor and velocities fields in complex flow. These kind of datasets can be obtained, for example, in experiments through combined use of particle image velocimetry and flow-induced birefringence measurements (Haward *et al.* 2021), X-ray scattering measurements (Mao, McCready & Burghardt 2014; McCready & Burghardt 2015) or, in the case of multiscale applications, from independent mesoscale polymer computations (Simavilla & Ellero 2022; Simavilla *et al.* 2023). Thus, we aim to leverage the potential of PINNs to provide rheologists with more effective, thermodynamics-guided ways to discover constitutive equations from data and, at the same time, applying them directly to fluid mechanics simulations using CFD.

2. Methods

2.1. *The GENERIC-guided approach to model constitutive equations*

In the present work, we are interested in the modelling of polymeric solutions. Within the GENERIC framework, a general polymer model can be cast into a set of PDEs including the mass and momentum balance equations

$$\boldsymbol{\nabla}\cdot\boldsymbol{v}=0, \tag{2.1}$$

$$\rho\left(\frac{\partial\boldsymbol{v}}{\partial t}+\boldsymbol{v}\cdot\nabla\boldsymbol{v}\right)-\nabla\cdot\left(\frac{\eta_s}{2}\left(\nabla\boldsymbol{v}+\nabla\boldsymbol{v}^T\right)\right)=-\nabla p+\nabla\cdot\boldsymbol{\tau}, \tag{2.2}$$

where $\boldsymbol{v}$ is the velocity vector field, p is the pressure and $\boldsymbol{\tau}$ is the non-Newtonian extra-stress term coupled with an evolution equation for the conformation tensor $\boldsymbol{c}$, which represents the microstructure generated by the polymers (Ottinger 2005). For a polymer solution undergoing affine deformation, the conformation tensor generally evolves according to (Vázquez-Quesada *et al.* 2009*b*; Simavilla *et al.* 2023)

$$\partial_t\boldsymbol{c}=-\boldsymbol{v}\cdot\boldsymbol{\nabla}\boldsymbol{c}+\boldsymbol{c}\cdot\boldsymbol{\kappa}+\boldsymbol{\kappa}^T\cdot\boldsymbol{c}+\frac{2}{\lambda_p n_p k_B T}\boldsymbol{c}\cdot\boldsymbol{\sigma}, \tag{2.3}$$

where λ_p is the polymeric characteristic relaxation time, n_p the polymer number density (i.e. the number of chains per unit volume), k_B is the Boltzmann constant and T the temperature. The first three terms in (2.3) are reversible in nature, describing the kinematic evolution of the conformation tensor under the influence of the velocity gradient $\boldsymbol{\kappa}=\nabla\boldsymbol{v}^T$. The application of (2.3) is limited to dilute polymeric solutions where the fluid can be considered a suspension of non-interacting polymers. In GENERIC, the most general formulation of the governing equations where the polymer contributions are given by the

conformation tensor takes the form (Ottinger 2005)

$$\partial_t \boldsymbol{c} = -\boldsymbol{v}\cdot\nabla\boldsymbol{c} + \boldsymbol{c}\cdot\boldsymbol{\kappa} + \boldsymbol{\kappa}^T\cdot\boldsymbol{c} + \boldsymbol{R}_2 : \rho\boldsymbol{\sigma}, \tag{2.4}$$

where $\boldsymbol{R}_2$ is a fourth-order tensor connected to the dissipation or friction matrix. In the scope of this manuscript we will focus on dilute polymeric solutions. For this kind of system, it has been shown (Vázquez-Quesada *et al.* 2009*a*), that the Wiener process leads to $\boldsymbol{R}_2 \propto \boldsymbol{c}$ and therefore (2.3). There are a number of 'entropy' models that are typically used for dilute suspensions of non-hydrodynamically interacting polymers where the evolution of the conformation tensor is well represented by this equation (Ottinger 2005). Learning general anisotropic models for concentrated systems requires determining $\boldsymbol{R}_2(\boldsymbol{c})$, which in turn necessitates incorporating additional physical information or data.

The last term in (2.3) describes the general irreversible evolution of the conformation tensor. This term is characterized by the thermodynamic force $\boldsymbol{\sigma}(\boldsymbol{c})$, defined as the derivative of the polymeric entropy density $s_p(\boldsymbol{c})$ with respect to the conformation tensor $\boldsymbol{c}$, that is

$$\frac{\boldsymbol{\sigma}}{T} = \frac{\partial s_p}{\partial \boldsymbol{c}}. \tag{2.5}$$

Finally, the momentum balance equation contains, in addition to the solvent viscous stress, a polymeric stress given by (Vázquez-Quesada *et al.* 2009*b*)

$$\boldsymbol{\tau} = -2\boldsymbol{c}\cdot\boldsymbol{\sigma} \tag{2.6}$$

which satisfies the dynamics–thermodynamics compatibility (i.e. consistency with the microstructural evolution given by (2.3)). Therefore, the knowledge of the entropy function directly provides the closure in a thermodynamic-consistent constitutive equation for a dilute polymeric suspension.

Since the entropy is invariant under rotations, it can only depend on the invariants of the conformation tensor $\boldsymbol{c}$, that we choose to be the eigenvalues c_1, c_2, c_3 and therefore, $s_p(\boldsymbol{c}) = s_p(c_1, c_2, c_3)$. Observe that the tensors $\boldsymbol{c}$ and $\boldsymbol{\sigma}$ commute (Vázquez-Quesada *et al.* 2009*b*) and diagonalize simultaneously. This implies that we may write (2.5) as

$$\frac{\sigma_\alpha}{T} = \frac{\partial s_p}{\partial c_\alpha}, \tag{2.7}$$

where $\sigma_\alpha (\alpha = 1, .., D)$ are the eigenvalues of $\boldsymbol{\sigma}$. Because of the large reduction of arguments of the entropy due to rotational symmetry, it proves convenient to express the dynamics in terms of eigenvalues and eigenvectors. The decomposition of (2.3) into eigenvalues and eigenvectors (i.e. $\boldsymbol{c} = \sum_\alpha c_\alpha \boldsymbol{u}_\alpha \boldsymbol{u}_\alpha^{\mathrm{T}}$) leads to two coupled PDEs (Vázquez-Quesada *et al.* 2009*b*),

$$0 = \partial_t c_\alpha + v_j \partial_j c_\alpha - 2c_\alpha \kappa_{\alpha\alpha} - \frac{1}{\lambda_p n_p k_B T} c_\alpha \sigma_\alpha, \tag{2.8}$$

$$0 = \partial_t \boldsymbol{u}_\alpha + v_j \partial_j \boldsymbol{u}_\alpha - \sum_\beta H_{\alpha\beta}\boldsymbol{u}_\beta, \tag{2.9}$$

where $\kappa_{\alpha\alpha} = \boldsymbol{u}_\alpha \cdot \boldsymbol{\kappa} \cdot \boldsymbol{u}_\alpha$ is the velocity gradient in the eigenbasis of the conformation tensor, and the antisymmetric matrix $H_{\alpha\beta}$ is given by

$$H_{\alpha\beta} = \frac{1}{c_\alpha - c_\beta}(c_\alpha \kappa_{\alpha\beta} + c_\beta \kappa_{\beta\alpha}). \tag{2.10}$$

The kinematics of the flow in (2.9) has been used by the authors in Simavilla *et al.* (2023) to establish the non-affine characteristics of polymer flow by introducing a mixed

derivative of the Gordon–Schowalter type. This allows for the unambiguous separation of reversible/irreversible terms in the dynamics, enabling to split non-Newtonian effects related to non-affine deformation, with irreversible effects intrinsically associated with the polymeric entropy $s_p(\mathbf{c})$, which is crucial to apply safely the model to arbitrary flows. For simplicity, non-affine motion is not considered here as it will be the subject of future refinements of the methodology. Note that the entropy appears only in (2.8) through σ_α. Equation (2.8) will be used below to construct the residuals and the loss function in the PINN model.

In this paper, we address the following problem: given that the dynamics of the polymer solution are described by (2.3) or its equivalent, (2.8), and that we have explicit data for the fields $\boldsymbol{v}(\boldsymbol{r}, t)$ and $\boldsymbol{c}(\boldsymbol{r}, t)$, our objective is to develop a Physics-Informed NN (PINN) representation of the specific entropy function $s_p(\mathbf{c})$. This PINN model should ensure that the measured fields align with the governing equation (2.3) dictated by GENERIC. It should be also noticed that our approach diverges from the usual application of PINNs where the NNs represent the fields $\boldsymbol{v}(\boldsymbol{r}, t)$ and $\boldsymbol{c}(\boldsymbol{r}, t)$ directly (Karniadakis *et al.* 2021). In contrast, in this paper, we focus solely on employing a single NN to model the functional form of the entropy. From this knowledge alone, s_p can be used to provide all the necessary stress predictions in CFD simulations (i.e. using RheoTool).

2.2. *The NN*

To demonstrate the methodology, this paper focuses on two-dimensional (2-D) flows, with the extension to 3-D flows being straightforward. We aim at representing the functional form of the entropy $s_p(\mathsf{C}) : \mathbb{R}^2 \to \mathbb{R}$ as a function of the eigenvalues $\mathsf{C} = \{c_1, c_2\} \in \mathbb{R}^2$ through a NN of the form

$$\tilde{s}_p(\mathsf{C}) = n_p k_B \tilde{s}_\theta(\mathsf{C}) \tag{2.11}$$

where n_p is the polymer number density and the dimensionless NN is

$$\tilde{s}_\theta(\mathsf{C}) = \left[W_L \cdot \phi(\cdots \phi(W_1 \cdot \phi(W_0 \mathsf{C} + b_0) + b_1) \cdots) + b_L\right](\mathsf{C} - \mathsf{C}_{eq})^2 \tag{2.12}$$

where $W_k \in \mathbb{R}^{h_k \times h_{k-1}}$ and $b_k \in \mathbb{R}^{h_k}$ denote, respectively, the weights and biases of the kth hidden layer, with $k \in (0, \cdots, L)$. The number of nodes of the kth layer is h_k. The collection of all trainable parameters of the network is identified as $\theta = \{W_k, b_k\}_{k=0}^{L}$. The activation function $\phi : \mathbb{R} \to \mathbb{R}$ is a smooth nonlinear function that is applied elementwise to a multivariate argument. The choice of the function is arbitrary and often problem-dependent, with common selections for ϕ including the hyperbolic tangent or the sine function, chosen in this work. Finally, the factor involving the equilibrium value $\mathsf{C}_{eq} = \{1, 1\}$ of the conformation tensor's eigenvalues is a common approach followed when implementing PINNs in order to impose exact satisfaction of boundary and/or initial conditions (Sukumar & Srivastava 2022; Arzani, Cassel & D'Souza 2023). In particular, we ensure that $s_p(\mathsf{C}_{eq}) = 0$ and $\sigma_\alpha(\mathsf{C}_{eq}) = \{0, 0\}$. This requirement is necessary because a non-zero entropy at the equilibrium will often result in unstable flow predictions once the NN is coupled with RheoTool. Additional information on the expressions for s_p, and therefore $\boldsymbol{\sigma}$, for simple models are discussed in Appendix A.3.

A graphical description of (2.12) is shown in figure 1, where it is also indicated how the eigenvalues σ_α are obtained from automatic differentiation of the NN entropy function with respect to C.

The final ingredient of a NN is the loss function whose minimization produces the parameters of the network. The loss function in PINNs is constructed in terms of residuals of the PDE. The definition of the residuals $e_\alpha(c_\alpha; \theta)$ follows from the

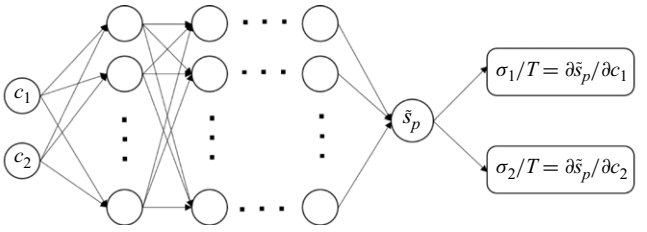


Figure 1. Sketch of the PINNs architecture.

GENERIC-consistent PDE (2.8) that already uses the NN representation in (2.11),

$$e_\alpha(c_\alpha;\theta) \equiv \partial_t c_\alpha + v_j \partial_j c_\alpha - 2c_\alpha \kappa_{\alpha\alpha} - \frac{2}{\lambda_p} c_\alpha \frac{\partial \tilde{s}_\theta}{\partial c_\alpha}, \tag{2.13}$$

where $\alpha = 1, 2$ for 2-D problems and the partial derivative of the NN $\tilde{s}_\theta$ is computed through automatic differentiation (Paszke *et al.* 2017). The residuals in (2.13) can be obtained through any dataset – produced either by simulation or experiments – for the velocity gradient and the microstructure (i.e. the conformation tensor).

The residuals can be further simplified when the model is trained exclusively on the steady state line. Therefore, when training the PINN with synthetic viscometric data (PINN-rheometric), we will use the residual

$$e_\alpha(c_\alpha;\theta) = -2c_\alpha \kappa_{\alpha\alpha} - \frac{2}{\lambda_p} c_\alpha \frac{\partial \tilde{s}_\theta}{\partial c_\alpha}. \tag{2.14}$$

With either of the residuals defined by (2.13) and (2.14), we define the loss function

$$\mathcal{L}\left(c_1^i, c_2^i, \theta\right) = \frac{1}{N} \sum_{i=1}^{N} \left(\lambda_1 ||e_1\left(c_1^i;\theta\right)||^2 + \lambda_2 ||e_2\left(c_2^i;\theta\right)||^2 \right) \tag{2.15}$$

where $\{(c_1^i, c_2^i)\}_{i=1}^N$ represents the dataset with N points used for the training of our model. The parameters λ_1 and λ_2 represent two scalars whose purpose is to balance the interplay between the two residuals. Unbalanced loss components are known to be detrimental for the training process of PINNs, which can lead to slow or unfeasible convergence (Wang, Yu & Perdikaris 2022). We established the values of the two scalars based on the heuristics proposed in Schmid *et al.* (2024), where their value is given as the proportion of the two residuals at the first iteration. For the specific models we train in our study, we notice that this approach generally yields a ratio $\lambda_1/\lambda_2 \approx 10^{-4}$. Finally, by minimizing the loss function in (2.15), we identify the optimal NN entropy function that ensures the dynamic equations in (2.13) to be consistent with the data. We train our models by the Adam optimizer (Kingma & Ba 2014), which is an enhanced first-order stochastic optimization algorithm ubiquitous in the machine learning community.

One of the main benefits of using a PINN to approach the minimization of PDE residuals is the flexibility of the training formulation. Indeed, all the physical quantities included in (2.13) can be obtained through high-fidelity numerical simulation, or experimental measurements and all those data can be introduced in the same training scheme.

2.3. *Data-driven CFD: coupling PINN with RheoTool*

The macroscopic flow simulations in this article have been performed using the RheoTool library. RheoTool is an extension for OpenFOAM (FV), a popular open-source

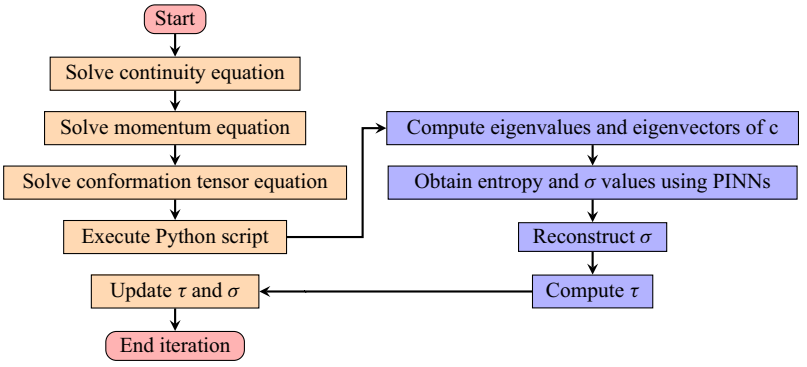


Figure 2. Flowchart of the procedure for macroscopic flow simulations using RheoTool and Python script PINN integration. Orange boxes refer to RheoTool actions. Purple boxes refer to actions in python script.

computational fluid dynamics (CFD) software, designed specifically for simulating non-Newtonian flows. The software RheoTool was developed by Pimenta *et al.* to provide advanced viscoelastic numerical methods to the OpenFOAM community (Pimenta & Alves 2017). The software RheoTool is publicly available and incorporates various constitutive models for non-Newtonian fluids including power-law, Carreau, Cross, Bingham, Herschel–Bulkley and viscoelastic models like OB, Giesekus and FENE-P, among many others.

In this paper, we use a modification of the rheoFoam uncoupled solver that incorporates a python script where the PINN model is executed using PyTorch. The RheoTool-PINN interaction is achieved using the PythonPal interface (Rodriguez & Cardiff 2022), a header-only library that provides high-level methods for OpenFOAM (C++) to Python communication. For example, it provides ready-to-use methods such as the constructor for the python script, a passToPython function that creates a NumPy array in the Python interpreter from OpenFOAM data, and the execute command to run the python script. After initiating the PyTorch library and setting required variables in the modified solver, the following steps are executed at each time iteration (see the flowchart in figure 2). First, the continuity equation (2.1) and momentum equation (2.2) are solved using a SIMPLEC type iteration. The term $(\eta_s/2)(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T)$ represents the diffusive term of the solvent stress contribution, which is solved implicitly. In contrast, $\nabla \cdot \boldsymbol{\tau}$ is handled explicitly. The momentum equation is solved using RheoTool's coupling both-sides-diffusion stabilization technique. After the SIMPLEC iteration on continuity and momentum equations is completed, the constitutive equation for the symmetric conformation tensor $\boldsymbol{c}$ is solved in a semi-implicit form, where the right-hand side of the equation is solved explicitly,

$$\partial_t \boldsymbol{c} + (\boldsymbol{v} \cdot \nabla \boldsymbol{c}) = \left(\boldsymbol{c} \cdot \boldsymbol{\kappa} + \boldsymbol{\kappa}^T \cdot \boldsymbol{c}\right) + \frac{2}{\lambda_p n_p k_B T} \boldsymbol{c} \cdot \boldsymbol{\sigma}. \tag{2.16}$$

Next, a python script is executed which involves the following steps. First, the eigenvalues $\{c_1, c_2\}$ and normalized eigenvectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ of the conformation $\boldsymbol{c}$ are computed in each cell and sorted by size, where $c_1 > c_2$. Then, $\{\sigma_1, \sigma_2\}$ values in each cell are obtained through automatic differentiation of the PINN entropy representation computed from $\{c_1, c_2\}$. As a next step, conjugate variable $\boldsymbol{\sigma}$ is reconstructed using the eigenvalues $\{\sigma_1, \sigma_2\}$ and eigenvectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ as

$$\boldsymbol{\sigma} = \sum_{\alpha} \sigma_\alpha \boldsymbol{u}_\alpha \boldsymbol{u}_\alpha^T. \tag{2.17}$$

Finally, the non-Newtonian extra-stress $\boldsymbol{\tau}$ is computed from (2.6) as

$$\boldsymbol{\tau} = -\frac{2\eta_p}{\lambda_p n_p k_B T}(\boldsymbol{c}\cdot\boldsymbol{\sigma}) \tag{2.18}$$

or equivalently

$$\boldsymbol{\tau} = -\frac{\eta_p}{\lambda}\left(\boldsymbol{c}\cdot\frac{\partial\tilde{s}_\theta}{\partial\boldsymbol{c}}\right). \tag{2.19}$$

The computed stress is then used back in the SIMPLEC iteration of RheoTool, closing the time step loop. A graphical sketch of the structure of the algorithm is shown in figure 2.

3. Numerical results

3.1. *The PINN-rheometric: training*

To validate the present methodology, we first use data produced by the analytical solution to (2.3) with (2.1), (2.2) in steady-state rheometric flows. We denominate the PINN trained with this dataset: PINN-rheometric. Since all steady-state rheometric flow solutions using the OB model fall on the same line in the c_1–c_2 space, we choose to implement our training with data for the simplest analytical solution, i.e. steady-state extensional flow characterized by extensional rate $\dot{\epsilon}$. For the OB the analytical expression for the entropy is given by (Ottinger 2005)

$$s_p(c_1, c_2) = \frac{k_B}{2}(2 - c_1 - c_2 + \ln c_1 + \ln c_2) \tag{3.1}$$

which represents the ground truth solution to target with data, whereas in extensional flow the eigenvalues of the conformation tensor c_α at a given Wi are

$$c_1 = \frac{1}{1-2Wi}, \tag{3.2}$$

$$c_2 = \frac{1}{1+2Wi}, \tag{3.3}$$

where the Weissenberg number $Wi = \dot{\epsilon}\lambda_p$. For this flow, we have computed in Appendix A.1 the velocity gradient in the eigenbasis $\kappa_{\alpha\beta}$ in (A10) and the residuals read

$$e^*_\alpha(c_\alpha;\theta) = -[Wi]_i - \frac{\partial\tilde{s}_\theta}{\partial c_\alpha} \tag{3.4}$$

where the dimensionless residual $e^*_\alpha(c_\alpha;\theta) = \lambda_p e_\alpha(c_\alpha;\theta)/2c_\alpha$, $[Wi]_i \in [0, \cdots 0.5)$ with $i = 1, \cdots, N$ where $N = 60\,000$ is the number of data points, distributed uniformly in the interval. Equations (3.2) and (3.3) show that the two eigenvalues c_1, c_2 can be parametrised with Wi. Furthermore, in the inset of figure 3, we show that the data generated from steady-state rheometric solutions of (2.3) lie on a single line in the c_1–c_2 plane. We train our models by the Adam optimizer (Kingma & Ba 2014), which is an enhanced first-order stochastic optimization algorithm ubiquitous in the machine learning community, and we limit the training to a maximum of 3×10^5 iterations.

Figure 3 shows the prediction of the entropy over the training domain (steady viscometric line). As expected, the PINN model is in very good agreement with the ground truth OB model over the training domain. Predictions for the entropy and $\boldsymbol{\sigma}$ outside the training domain are compared with the OB model in § 3.1. The inset to figure 3 shows the entropy surface $s(c_1, c_2)$ in the (c_1, c_2) domain. The surface displays the paths explored

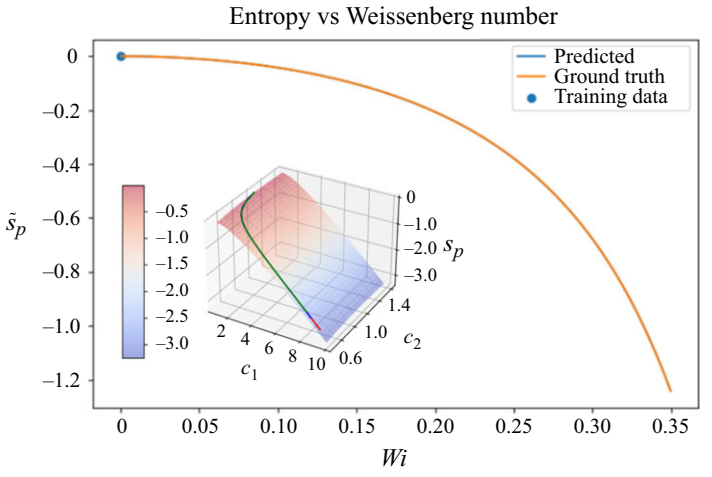


Figure 3. Entropy as a function of Wi along the steady-state viscometric line. The inset shows the whole entropy surface (3.1) as a function of c_1 and c_2. The lines over the surface correspond to steady-state rheometric flows (extensional (red), simple shear (blue) and Poiseuille (green)). These lines all coincide.

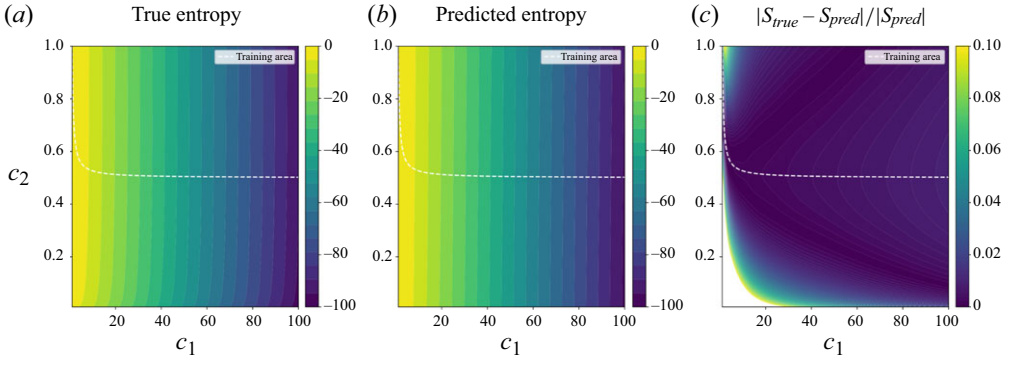


Figure 4. (*a*) True entropy. (*b*) Predicted entropy given by OB model in (3.1). (*c*) Relative error in the prediction of the entropy for the PINN-rheometric model.

by rheometric flows, with extensional flows highlighted in red, simple shear in blue and Poiseuille flows in green. They all fall on top of each other.

Figure 4 shows a comparison of the PINN-rheometric prediction and the ground-truth OB model over the entire (c_1, c_2) plane. As already shown in figure 3, predictions are excellent over the steady-state flow line that is used for the training. The quality of the prediction is reduced as the distance to the training line is increased. It is important to note that some of the areas in this graph might not be physically relevant. For example, we cannot increase c_1 keeping $c_2 = 1$ and *vice versa*. Typically, when a flow becomes more 'transient' or more 'complex' the line representing steady-state rheometric flows (figure 3) will start to widen leading to an area around said line. Notably, transient extensional and shear flows only explore the region below the steady-state rheometric line (see Appendix A.3), while complex flows explore both regions above and below.

Finally, figure 5 shows the relative error in the predictions of the eigenvalues of $\boldsymbol{\sigma}$. The relative error in σ_1 is in general much lower than in σ_2. This is a result of the low curvature of the entropy surface and the different ranges covered by $c_1 = [1, 100]$ and

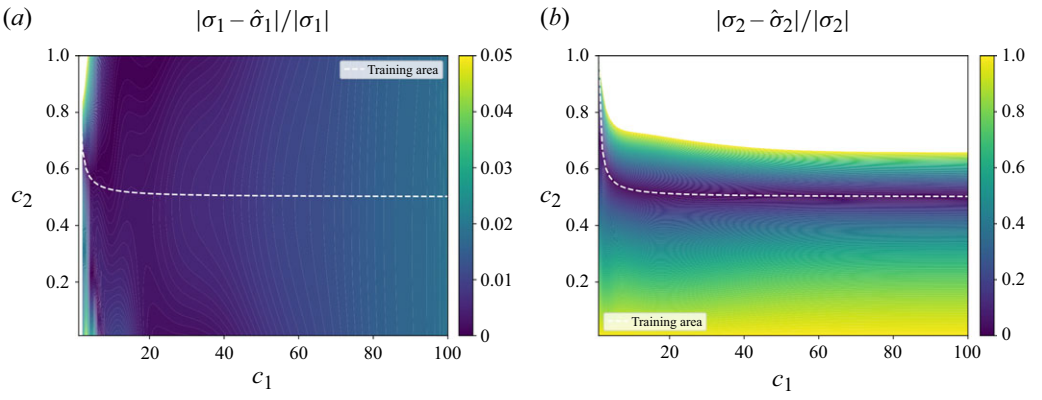


Figure 5. (*a*) Relative error in the prediction of the first eigenvalue of $\boldsymbol{\sigma}$. (*b*) Relative error in the prediction of the second eigenvalue of $\boldsymbol{\sigma}$. The eigenvalues of $\boldsymbol{\sigma}$ are determined through automatic differentiation of the PINN-rheometric model entropy in figure 4.

$c_2 = [1, 0.5]$ during training. This significant difference was limited during the training using augmentation in the computation of the loss function as described in § 2.2. Nevertheless, we can observe that, while the relative error in σ_1 is kept below 5 % over the entire studied domain, the error in σ_2 increases over 50 % at a relatively short distance from the training line. This will have important consequences in the simulation of complex flows that explore wider regions of the c_1–c_2 space.

3.2. *The PINN-rheometric: flow around a cylinder*

The validation of the RheoTool software for a flow around cylinder has already been reported by Alves *et al.* (2001) in a detailed description of the computational set-up required to simulate upper convected Maxwell and OB fluids. In this work the fluid characteristic values to replicate the cited case have been used, that is $\rho = 1$, $\eta_s = 0.59$, $\eta_p = 0.41$. A sketch of the flow geometry is presented in figure 6(*a*). Full domain length is 50 times the cylinder radius, and only half of the domain is discretized as described in Alves *et al.* (2001). The channel height is discretized with 70 cells at an expansion ratio of 3. The employed structured mesh contains 119422 points following the same discretization strategy as in their article. Figure 6(*b*) shows the mesh employed.

Fluid enters from the left-hand patch with a parabolic profile and maximum velocity $U = 1$. On the walls, no-slip conditions are used for velocity, linear extrapolation for the conformation tensor and zero-gradient for pressure. Fluid exit is achieved by setting a zero-pressure value on the right-hand patch and zero gradient for the rest of the variables. The Weissenberg number is defined consistently with the work in Alves *et al.* (2001) using the average inlet velocity ($Wi = \overline{U}\lambda/R$). In order to validate the model, simulations are run leading to a cylinder drag coefficient $C_d = 118.94$ obtained at $Wi = 0.5$, consistent with the original work ($C_d = 118.838$) (Alves *et al.* 2001; Claus & Phillips 2013), where C_d is calculated as

$$C_d \equiv \frac{1}{\eta \overline{U}} \int_s (\boldsymbol{\tau}_{tot} - p\boldsymbol{I}) \cdot \boldsymbol{n} \cdot \hat{\imath} \, \mathrm{d}S. \tag{3.5}$$

Here $\boldsymbol{\tau}_{tot}$ is the sum of both the non-Newtonian and the Newtonian viscous contributions to the stress, $\boldsymbol{I}$ is the identity tensor, $\boldsymbol{n}$ is the outward cylinder surface unit normal vector, p is the pressure and $\hat{\imath}$ is the unit vector in the x direction. In the rest of this work, the magnitudes of tensors are evaluated using the Frobenius norm, that is $\tau = \sqrt{\boldsymbol{\tau} : \boldsymbol{\tau}}$,

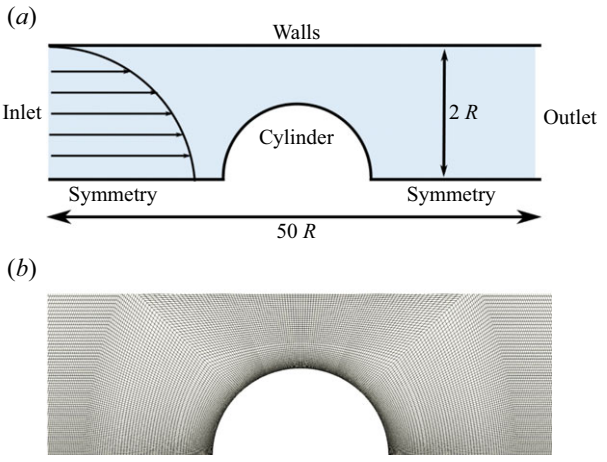


Figure 6. Sketch (*a*) and mesh example (*b*) near the cylinder in the flow around cylinder case.

whereas relative errors are computed by normalizing them by the Frobenius norm of the FV RheoTool reference solution. To prevent the relative error from diverging when the reference solution approaches zero, a threshold value is added in the denominator. Specifically, the error in the stress variable is computed with an exponential decay factor for values of τ such that $\tau/\tau_{max} < 0.05$.

Figures 7 and 8 show the simulation results for flow around a cylinder comparing predictions of the PINN-rheometric model and the ground-truth OB model. For this comparison, we monitor the fields c_1, c_2, the magnitude of the velocity and stress fields in the simulation domain. The PINN-rheometric model flow predictions errors are below 2 % for the stress and <1 % for all other fields at low $Wi = 0.067$ as shown by figure 7. However, as Wi is increased to $Wi = 0.15$ (see figure 8) significant relative errors up to 7.5 % for the stress and up to 7 % for c_1 are observed. This is a result of the large error in σ_2 reported in figure 5. The errors in the stress and σ_α are linked to the spreading of the region of the conformation space explored during simulation (see figure 9*b*).

These results can be better appreciated in figures 9(*a*) and 9(*b*). For low Wi, the conformation tensor eigenvalues are very close to the steady-state line. As Wi increases, the explored region in the c_1–c_2 plane becomes wider and separates from the steady-state line. The error in $\boldsymbol{\sigma}$ is larger in this region where the PINN model is extrapolating away from the steady viscometric training domain. As a result, the poor estimation of the stress in the simulation numerically propagates these errors, which in turn contribute to create an even wider region of the mapped conformational space (i.e. the maximum for c_1 is higher and the minimum for c_2 is lower than the ground truth OB solution).

Solutions for the stress on the symmetry plane and around the cylinder are also reported following the benchmark studied in Alves *et al.* (2001). It can be observed in figure 16 that at high Wi number the stress on top of the cylinder is increased due to high error associated with large c_1 values. From these results, it can be concluded that in order to investigate regions outside the line in the c_1–c_2 space explored in rheometric steady-state flow, one should train the network with data from non-rheometric flows (i.e. either complex or transient flows cases able to map a wide range of the conformation space). Examples of the conformational space explored in complex flow around a cylinder are presented in figure 9, whereas the cases corresponding to transient extensional and shear flows are presented in Appendix A.3.

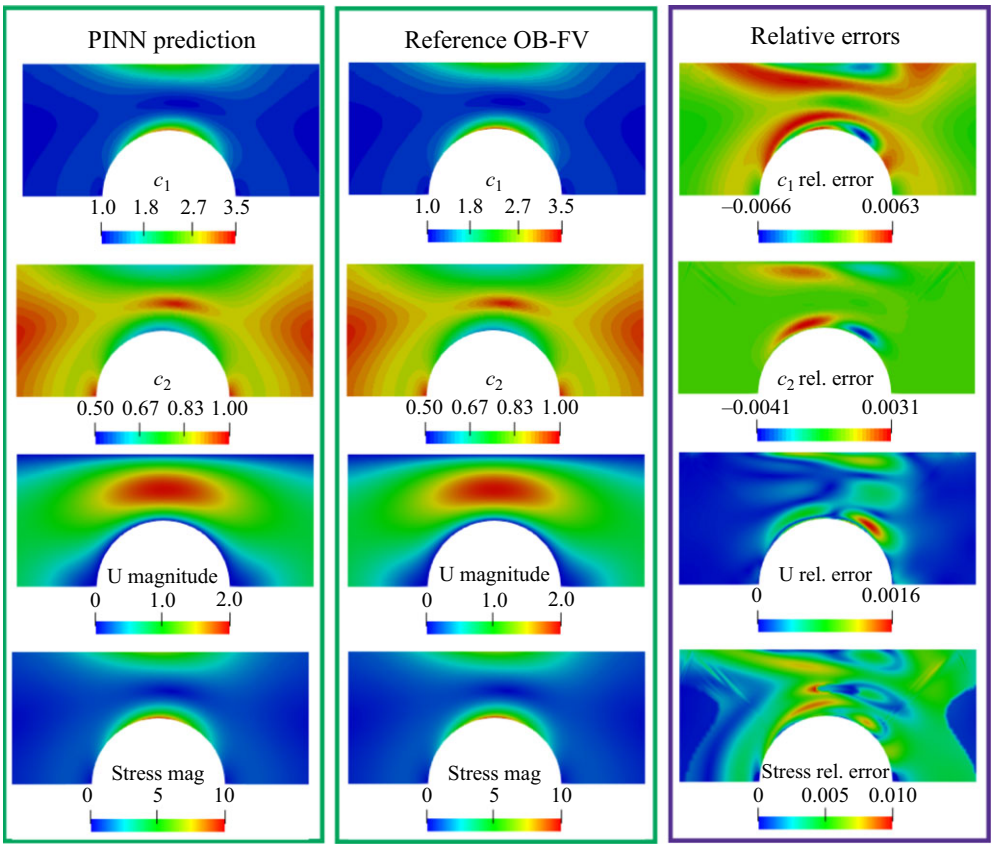


Figure 7. RheoTool simulation with steady-state rheometric training (PINN-rheometric) versus standard RheoTool simulation of the OB model at $Wi = 0.067$.

3.3. *The PINN-complex: training*

In this section, we study a PINN constitutive model trained with data from macroscopic simulations. This allows us to include the training values outside the line explored by rheometric steady-state flows. To that end, the data obtained for the steady-state flow around cylinder at $Wi = 0.45$ has been processed to evaluate the residuals in (2.13). Effectively, we mimic the procedure that would be followed in the application of our learning protocol to experimental data (i.e. with the conformation tensor measured by X-ray scattering (Mao *et al.* 2014; McCready & Burghardt 2015) and velocity field measured via particle image velocimetry (Haward *et al.* 2021)). In Appendix B, we have included the anisotropy factor and the orientation angle computed from conformation tensor components that could alternatively be used to train the PINN from this kind of experiments. Here the PINNs model is trained exclusively using the discrete *in silico* data obtained from CFD RheoTool simulations of the OB model, but notably, the PINN is agnostic of the 'true' constitutive model behind the training data.

The procedure starts by computing the eigenvalues and eigenvectors from the conformation tensor. As we work with steady-state flow the time derivative is zero for steady-state converged solution, $\kappa_{\alpha\alpha}$ is computed using (A9). The required gradients of these vectors are computed using a linear Gauss method. With all terms evaluated, the derivative of the entropy with respect to the conformation tensor is computed by (2.13).

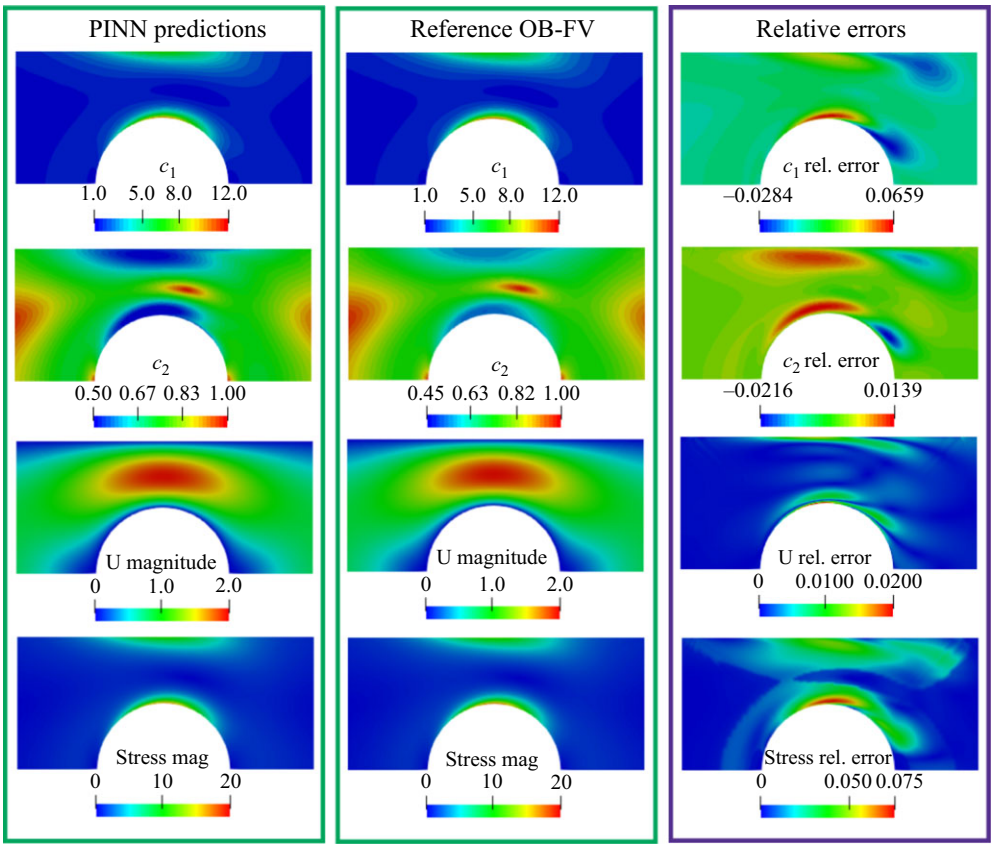


Figure 8. RheoTool simulation with steady-state rheometric training (PINN-rheometric) versus standard RheoTool simulation of the OB model at $Wi = 0.15$.

Effectively this is equivalent to computing σ_α, which is then used for the PINN-complex training with PyTorch.

In figure 10 we compare the entropy predictions of the PINN-complex model with the ground-truth solution given by the OB model. As in the PINN-rheometric model, the predictions for the entropy given by the PINN-complex are very accurate in the training area of the conformational space and a relative error is consistently below 10 % even in regions far from the training area.

Figure 11 shows the relative error achieved by the PINN-complex model for the prediction of the eigenvalues of $\boldsymbol{\sigma}$. The relative errors in σ_1 and σ_2 are again affected by the low curvature of the entropy surface due to their different training ranges $c_1 = [1, 90]$ and $c_2 = [1, 0.5]$. However, we notice that introducing training data from complex flows can largely benefit the accuracy of prediction and goodness of extrapolation for the value of σ_2.

3.4. *The PINN-complex: flow around a cylinder*

In this section we compare the accuracy of the RheoTool flow simulations using the PINN-complex model discussed in the previous section with the PINN-rheometric model and the ground truth OB solution (i.e. the OB model used to generate the training datasets). We consider here two cases: $Wi = 0.15$ where significant errors in the stress and conformation tensor fields were observed when using the PINN-rheometric model, and $Wi = 0.45$ to

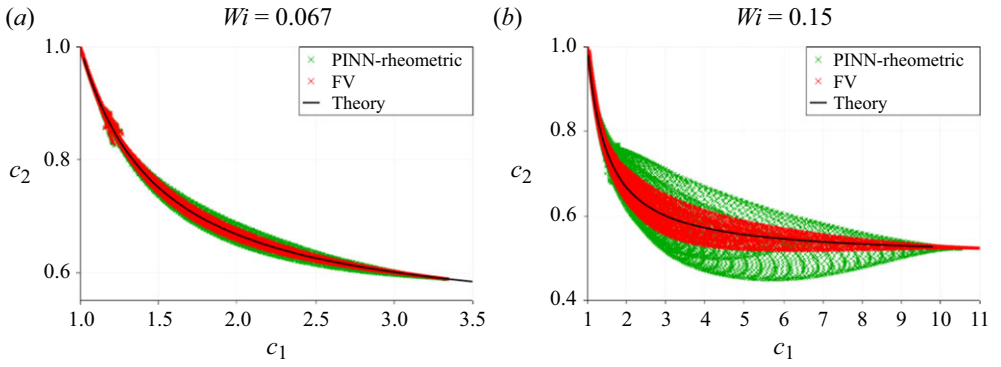


Figure 9. The c_1–c_2 region covered in simulations of flow around a cylinder with $Wi = 0.067$ (a) and $Wi = 0.15$ (b). Green crosses represent the results using the PINN-rheometric model, red crosses represent the OB implementation in RheoTool (FV) and the black solid line represent the analytical solution for steady-state rheometric flows where the training has been applied.

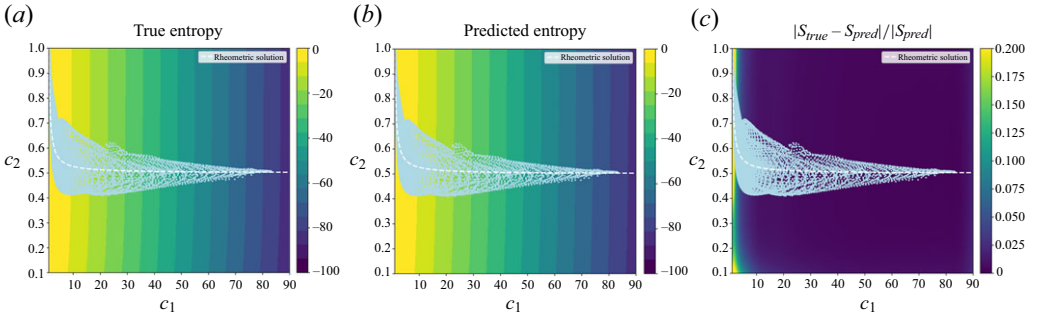


Figure 10. (a) True entropy given by OB model in (3.1). (b) Predicted entropy. (c) Relative error in the prediction of the entropy for the PINN-complex model. In all three maps, the light blue points represent the training data and the white dashed line represents the steady-state rheometric flow solution.

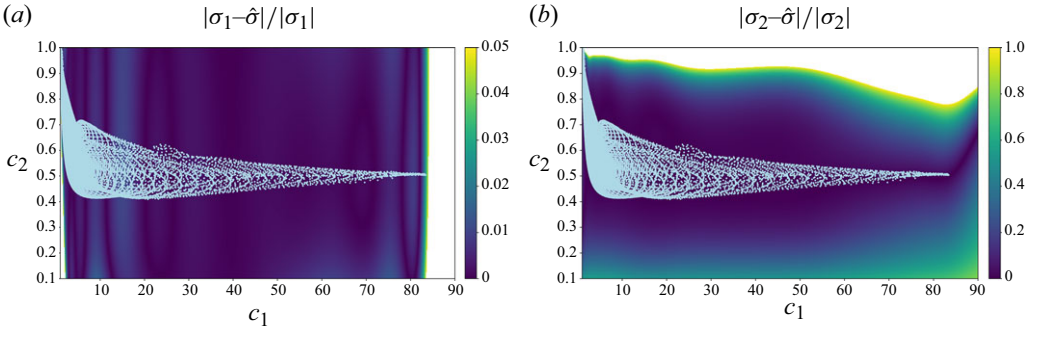


Figure 11. (a) Relative error in the prediction of the first eigenvalue of $\boldsymbol{\sigma}$. (b) Relative error in the prediction of the second eigenvalue of $\boldsymbol{\sigma}$. The eigenvalues of $\boldsymbol{\sigma}$ are determined through automatic differentiation of the PINN-complex model entropy in figure 4. The light blue points represent the training data.

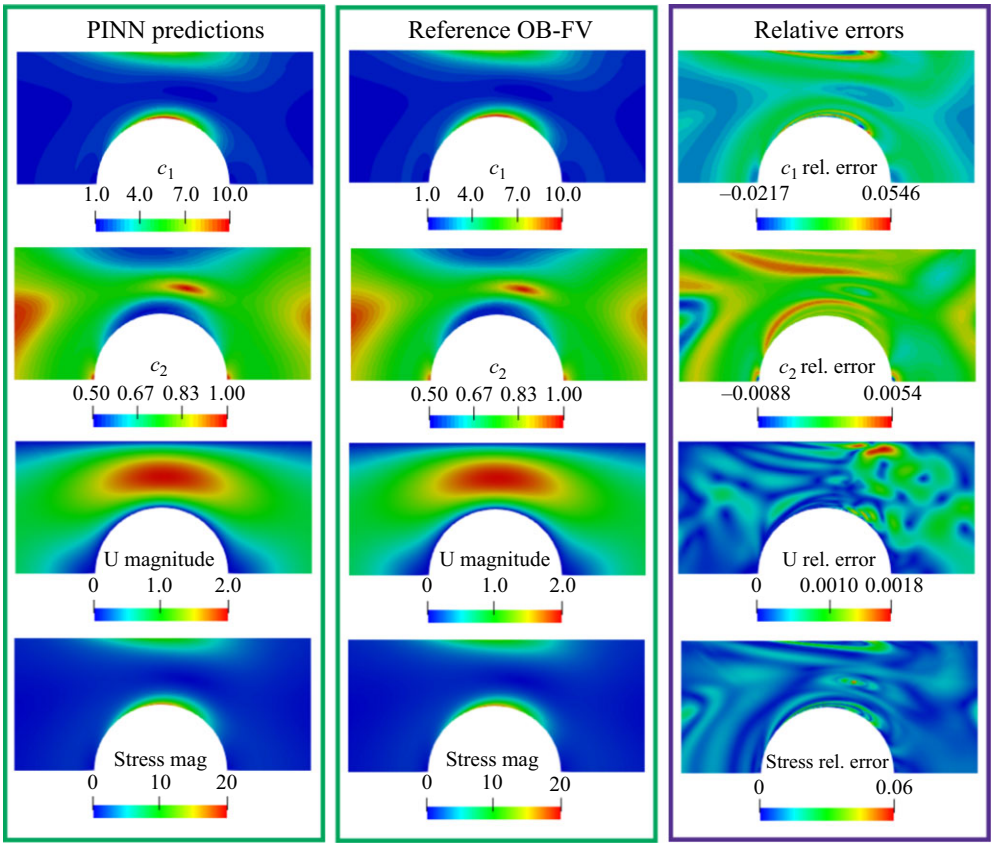


Figure 12. Comparison of RheoTool simulation results for PINN-complex and OB models at $Wi = 0.15$.

demonstrate the better accuracy of the PINN-complex model at larger Wi and away from the steady-state rheometric line.

The new RheoTool results using the PINN-complex model are reported in figures 12 and 13. Comparing these results with the previous ones in figure 8, we can clearly observe a significant overall improvement in accuracy in the predictions for the eigenvalues of the conformation tensor and the stress. In order to show the model flexibility, equivalent results but using a PINN-complex network trained with data generated using an alternative FENE-P model have been included in Appendix A.4. At $Wi = 0.15$, the relative error in c_1 shows the maximum error in the complex case located on top of the cylinder (where c_1 is also the maximum) with a value of 5.4 %, while for the PINN rheometric in figure 8 it was around 6.5 %. The error in the rest of the domain for the PINN-complex is generally below 1 %, far better than the PINN-rheometric simulation that has an error exceeding 5 %. Also the PINN-complex simulation results for c_2 are significantly better, with a relative error less than 0.5 %. In contrast, the PINN-rheometric case in figure 8 shows an error as large as 1.4 %.

The reasons behind the improvement of the constitutive modelling using complex flow data can be further analysed by comparing figures 9(*b*) and 14. Here, the regions covered in the c_1–c_2 space using the different training datasets are compared. Figure 9(*b*) reveals the widening of the c_1–c_2 space of the PINN-rheometric predictions outside of the area explored by the ground-truth RheoTool solution for the OB model. This result reveals how the larger error in the stress has a strong effect on the conformational space that is explored

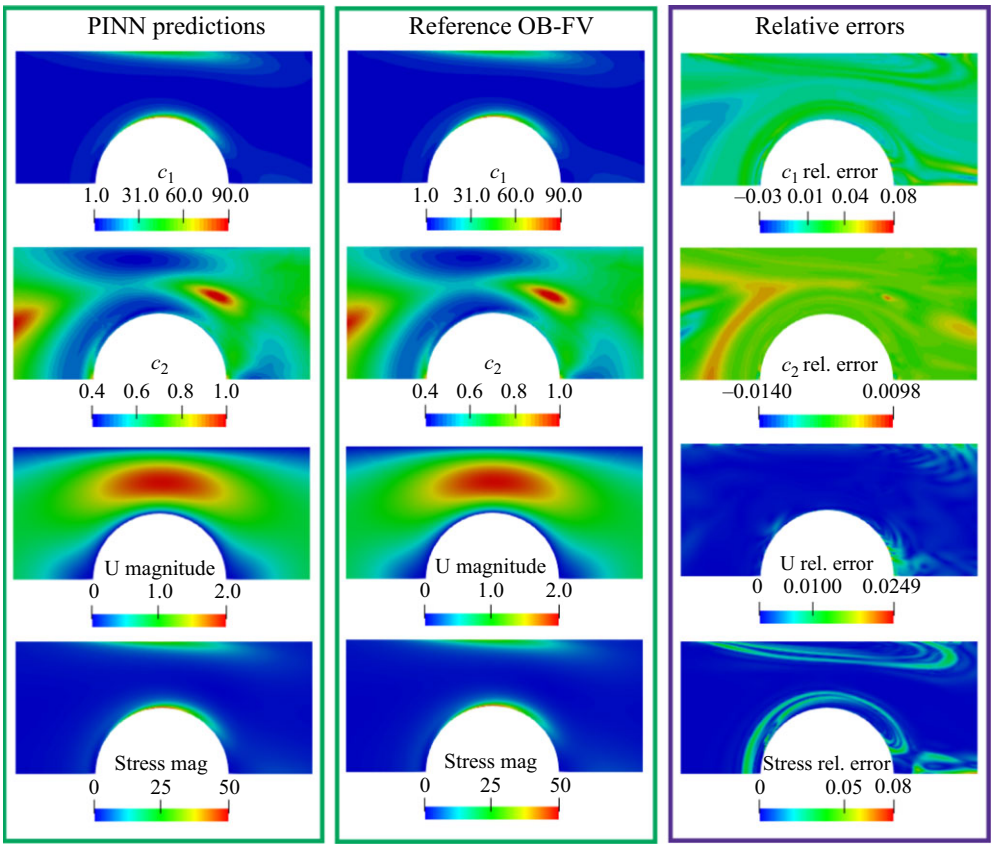


Figure 13. Comparison of RheoTool simulation results for PINN-complex and OB models at $Wi = 0.45$.

in a given simulation. In contrast, no widening of the explored area in the c_1–c_2 space is observed in figure 14 (i.e. PINN-complex simulation results for c_1 and c_2 throughout the simulation domain are in very good agreement with the OB model simulation results). The maximum error occurs at the top of the cylinder, where c_1 reaches its peak, approaching the maximum c_1 value used for the training data. This result clearly confirms that the training with the complex flow data better captures the underlying constitutive model (OB) in terms of consistent conformation space mapping for a complex flow problem. The small discrepancies between the PINN-complex model and the OB model in the c_1–c_2 map comparison in figure 14 are likely to be due to the specific PINN training approach. The PINN is trained using residuals derived from the FV solver rheoTool, which provides cell-averaged rather than pointwise values. Reconstructing pointwise residuals from these averages – via interpolation and gradient calculations using Gauss' theorem – introduces numerical errors. These errors lead to minor inconsistencies when the PINN-complex is validated against the training data. Integrating the PINN directly within the FV solver could reduce these errors, enhancing both accuracy and the method's applicability.

We must remark that it is important not to mistake the region of the c_1–c_2 space explored by a specific flow simulation with the flow geometry used in said simulation. In principle, the same (or very similar) regions in the c_1–c_2 space can be explored through completely different geometries. In figure 15, we present examples of the regions explored by different flows at a nominal $Wi = 0.2$: cross-slot, flow around a cylinder and contraction flow. Note that the Wi number is defined differently in all three flows, so the comparison between the

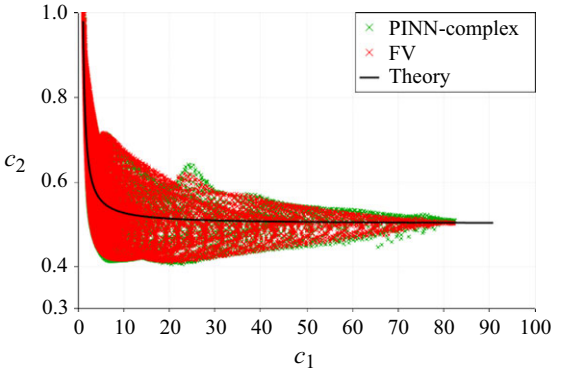


Figure 14. The c_1–c_2 region covered in simulation of flow around a cylinder with $Wi = 0.45$. Green crosses represent the results using the PINN-complex model, red crosses represent the OB implementation in RheoTool (FV), whereas the black solid line represent the analytical solution for steady-state rheometric flow.

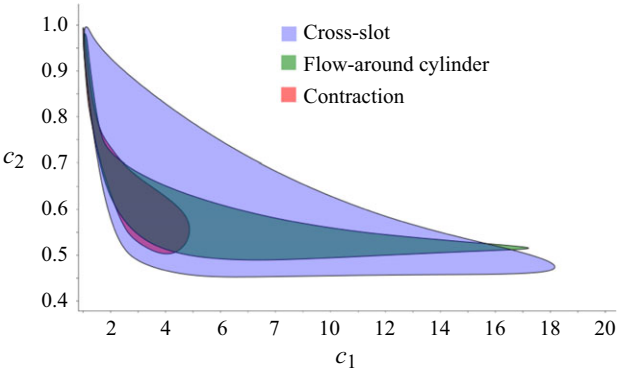


Figure 15. The c_1–c_2 region covered in simulation of cross-slot, flow around a cylinder and contraction flow with $Wi = 0.2$.

three ought to be qualitative. To obtain a good NN model of the entropy (or stress) in the region where the three flows overlap, we can train a NN with either of the flows. Assuming the same number of similarly distributed data points, the resulting model network from all three geometries should offer equivalent predictions. From this figure one might conclude that cross-slot geometry is better suited to explore wide areas of the c_1–c_2 space. An interesting approach will be to train the network in the regime of elastic instabilities. Elastic turbulence could possibly offer a better (or at least alternative) calibration of the data-driven PINN-complex approach (Shaqfeh & Khomami 2021).

Finally, in figure 16 we examine the stress values along the channel midplane and on the cylinder surface. The figure presents numerical results using three approaches: PINN-complex, PINN-rheometric and the reference FV OB solution computed with the standard RheoTool. As expected, for $Wi = 0.067$ (red) both models, PINN-rheometric and PINN-complex, provide a good representation of the OB model reference solution. This is explained by the complex flow simulation staying close to the steady-state rheometric flow line where PINN-rheometric is trained. At $Wi = 0.15$ (black), the PINN-complex model shows significantly improved results compared with the PINN-rheometric model. This improvement is due to more accurate predictions of the PINN-complex model as the space

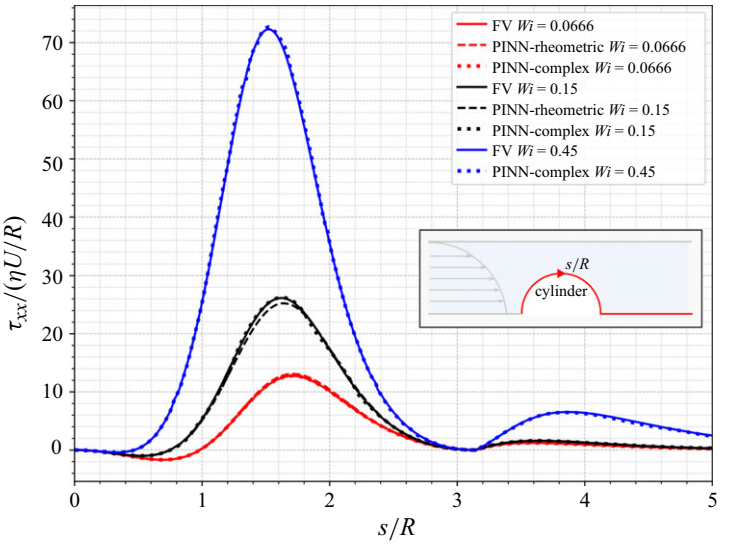


Figure 16. Comparison of stress on cylinder and symmetry plane. Inset shows the line over which the stress is computed.

explored in the c_1–c_2 map starts to widen. At larger $Wi = 0.45$ (blue), only PINN-complex is able to offer meaningful predictions.

3.5. *The PINN-complex: flow around an array of cylinders*

One important advantage of the data-driven procedure to model the constitutive equation for the stress using PINNs presented in this work is that the resulting PINN models are geometry-agnostic. This means that generalization to new geometries is in principle straightforward since it does not require extra training. In order to assess this feature of the model, in this section we applied the PINN-complex model to a new geometry set-up: the flow around a periodic array of cylinders (PAC) (Ellero & Adams 2011; Simavilla & Ellero 2022). In this case a single cylinder is simulated within the unit cell and periodic boundary conditions on the left/right faces are activated, making therefore possible for the cylinders to interact hydrodynamically. This will be important, since for significantly close cylinders, a topological change of the flow occurs, therefore enabling the assessment of the PINN accuracy in a different flow scenario. Two different separation lengths between cylinders L are used for our analysis: $L = 3R$ and $L = 2.5R$. In the latter case a topological change with a flow separation instability in the space between cylinders occurs. Figure 17 shows the geometry and mesh details. In figure 18(a) the region explored by the simulation results of the PAC flow for PINN-complex and RheoTool of OB model (FV) are shown for $Wi = 0.35$. At this Wi, the explored area in the $c_1 - c_2$ space for the new PAC geometry is 'mostly' within the training region (PINN hull) that is represented by the dashed line in figure 18(a). The limitation to $Wi = 0.35$ is motivated by the small region near $c_1 = 5$ and $c_2 = 0.8$ where a significant number of points are outside of the PINN training hull. This effect is accentuated as the distance between the cylinders is reduced.

Figure 19 shows a comparison of the different field (c_1, c_2, U, τ) in simulations for the PAC flow with $L = 3R$. The PINN-complex predictions demonstrate good accuracy, as indicated by the relative errors in c_1 and c_2 remaining below 2 %. The relative errors in stress and velocity are also small. However, it is important to note that the relative error for U is extremely sensitive near stagnation lines.

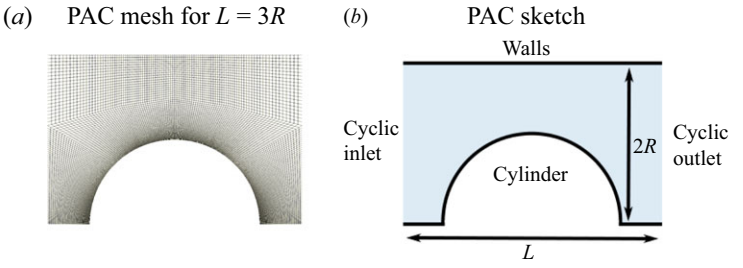


Figure 17. Periodic array of cylinders (PAC) test case geometry representation.

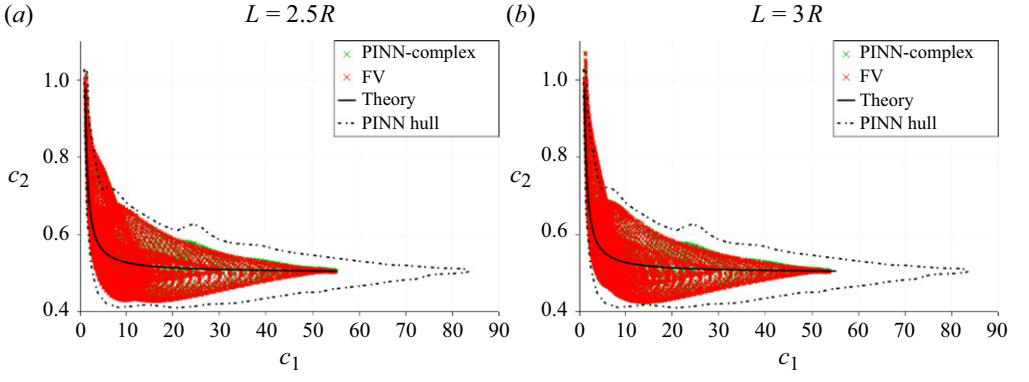


Figure 18. The c_1–c_2 region covered in a PAC flow at $Wi = 0.35$ at (*a*) $L = 2.5\ R$ and (*b*) $L = 3\ R$ for the PINN-complex model (green crosses). Red crosses represent the OB implementation in RheoTool (FV), and the black solid line represent the analytical solution for steady-state rheometric flows. The convex hull of the PINN-complex training range in a flow around a single cylinder is also shown with a dashed line.

The same analysis is performed in the case of closely interacting cylinders for $L = 2.5R$. In figure 20, we analyse the relative error with the PINN-complex applied to this new flow scenario. In this situation, a topological change of the flow occurs due to the recirculation area generated in the region between cylinders, as observed by the new streamlines.

The relative errors found in this flow type are of the same order of magnitude as for the $L = 3R$ case and follow a similar distribution over the fields. The only noteworthy difference is the increase in relative error up to 0.3 % in the c_2 field within the recirculation area, adjacent to the cyclic inlet/outlet. Note that, the plot ranges for c_2 are kept the same as the $L = 3R$ case to facilitate a direct comparison.

The excellent solution of the PINN-complex model verifies the capacity of the model to be geometry agnostic when the explored c_1–c_2 space is contained within the training-hull as shown in figure 18.

4. Conclusions

We have developed a GENERIC compliant framework that employs PINNs to learn the polymeric contribution to the entropy, which determines the rheological constitutive model. In this approach, the NN training is guided by a general evolution equation of the conformation tensor in dilute polymeric solutions. The model captures the polymeric contribution to the entropy as a function of the eigenvalues of the conformation

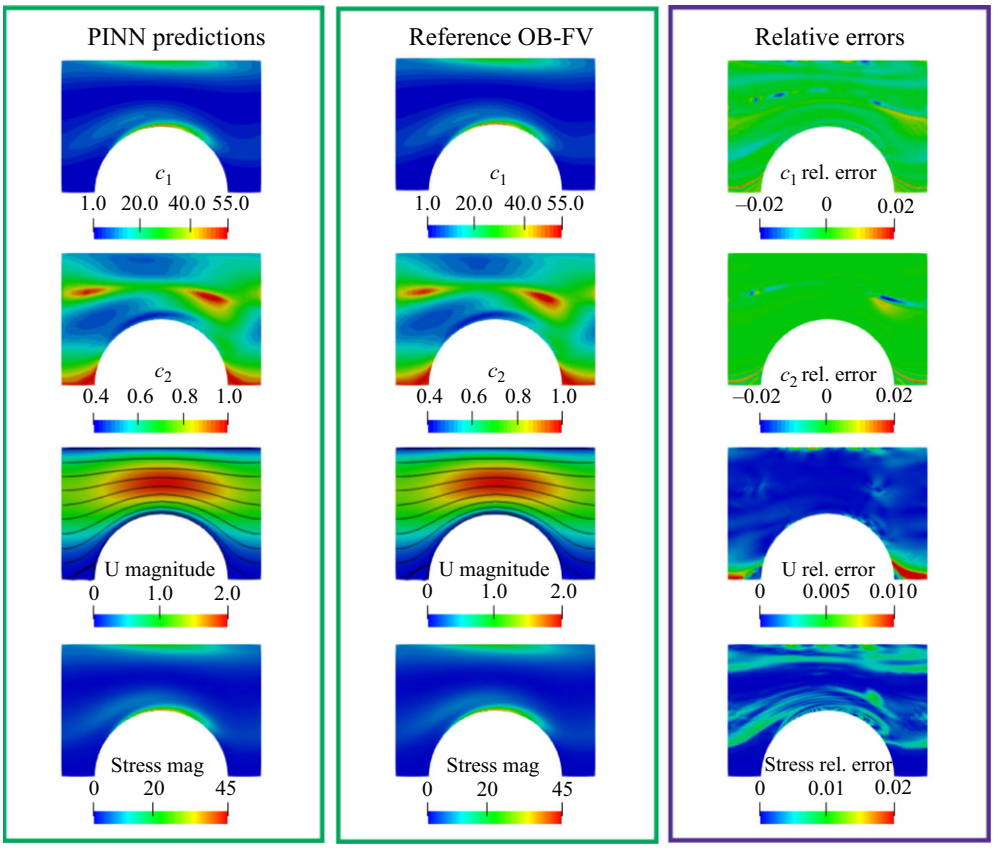


Figure 19. Comparison of RheoTool simulation with PINN-complex and OB models at $Wi = 0.35$ in a PAC for $L = 3R$.

tensor. This functional approximator can then be easily differentiated using automatic differentiation to compute the stress contributions required in CFD simulations.

Our results compare two different methods for training the PINN constitutive models. The first model (PINN-rheometric) is trained with data generated from the analytical solution of the OB model with the fluid subjected to steady-state rheometric flows. The PINN-rheometric model is physics-informed, and its training relies exclusively in the minimization of the residuals derived from the evolution equation for $\boldsymbol{c}$(2.3). The second model (PINN-complex) is trained with in-silico data generated directly from CFD simulations of complex flow around a cylinder that use OB or FENE-P as constitutive models. Both PINN models are capable of predicting flow behaviour in transient and complex flow conditions that explore regions of the conformational space that are not too far from the domain covered by the training data. However, the PINN-complex model outperforms the PINN-rheometric model in complex flow simulations. Our results highlight the importance of the regions used to train both models: the wider region used to train the PINN-complex model results in better predictions than the narrow region (a line) used to train the PINN-rheometric model. Furthermore, we apply the PINN-complex model to demonstrate a key advantage of our method being geometry agnostic by performing simulations of flow around a periodic array of cylinders in the conditions in which the flow topology is affected by the hydrodynamic interactions in between cylinders. We are able to capture this phenomenon as well as the underlying OB model does. Another key advantage of the

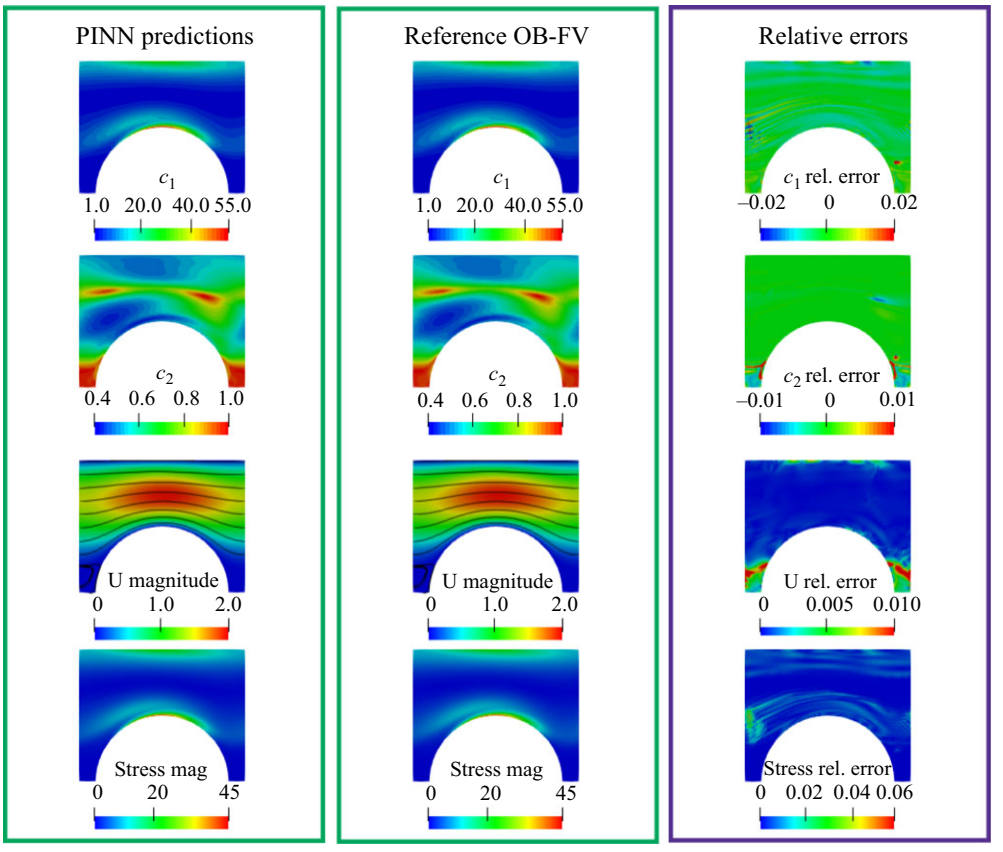


Figure 20. Comparison of RheoTool simulation with PINN-complex and OB models at $Wi = 0.35$ in a PAC for $L = 2.5R$.

presented methodology is that it bypasses the need for learning tensorial functionals that depend on the velocity gradients (nine variables) – that are flow specific – together with the conformation tensor (six variables). Instead, our method focuses on learning a scalar function – the entropy – that depends uniquely on the evolution of the eigenvalues of the conformation tensor (three variables). This results in better efficiency due to the significant reduction in the dimensionality of the learning problem. The PINNs powered modelling framework presented in this manuscript has the potential to change the way constitutive models are created by leveraging data, thermodynamics and physical knowledge and could be in principle applied to other complex fluids, such as suspensions, provided that a GENERIC framework is available to provide thermodynamically consistent constraints.

Funding. This research is supported by the Basque Government through the BERC 2022-2025 program, the ELKARTEK 2022 and 2024 programs (KAIROS project: grant KK-2022/00052 and ELASTBAT: KK-2024/00091). The research is also partially funded by the Spanish State Research Agency through BCAM Severo Ochoa excellence accreditation CEX2021-0011 42-S/MICIN/AEI/10.13039/501100011033, and through projects PID2020-117080RB-C55 (‘Microscopic foundations of soft-matter experiments: computational nano-hydrodynamics’ and acronym ‘Compu-Nano-Hydro’) and PID2020-117080RB-C54 (‘Coarse-Graining theory and experimental techniques for multiscale biological systems’) funded by AEI – MICIN.

Declaration of interests. The authors report no conflict of interest.

Data availability statement. A curated version of the code, training data and CFD simulation results is available at https://github.com/BCAM-CFD/GENERICFoam-PINN

Appendices

In these appendices, we present supporting results concerning some of the known constitutive models discussed in the main text of the manuscript: OB and FENE-P models.

Appendix A. The OB and FENE models

For a system with chains modelled as Hookean dumbbells, the entropy per unit volume as a function of the conformation tensor is given by OB model

$$s_p(\boldsymbol{c}) = \frac{k_B}{2}\left(\mathrm{tr}(\boldsymbol{I}-\boldsymbol{c}) + \ln(\det(\boldsymbol{c}))\right). \tag{A1}$$

The entropy can be expressed in terms of the eigenvalues of the conformation tensor c_1 and c_2,

$$s_p(c_1, c_2) = \frac{k_B}{2}\left(2 - c_1 - c_2 + \ln c_1 + \ln c_2\right). \tag{A2}$$

If instead the chains are modelled with finite extensible springs of the FENE type, the entropy is (Ottinger 2005)

$$s_p(\boldsymbol{c}) = \frac{k_B}{2}\left(b \ln \phi(\boldsymbol{c}) + \ln\det \boldsymbol{c}\right), \tag{A3}$$

$$\phi(\boldsymbol{c}) = \frac{b+D}{b} - \frac{1}{b}\mathrm{tr}(\boldsymbol{c}) = \frac{b+2}{b} - \frac{1}{b}(c_1+c_2), \tag{A4}$$

where dimensionality is given by $D=2$. Finally, b is the finite extensibility parameter related to the spring constant. Note that as $b\to\infty$, the FENE model converges to the OB model.

The thermodynamic force $\boldsymbol{\sigma}$ defined in (2.5) for the OB model it is given by

$$\boldsymbol{\sigma} = k_B T\left(\boldsymbol{c}^{-1} - \mathbf{I}\right); \tag{A5}$$

while for FENE

$$\boldsymbol{\sigma} = k_B T\left(\boldsymbol{c}^{-1} - \frac{\boldsymbol{I}}{\phi(\boldsymbol{c})}\right). \tag{A6}$$

At equilibrium, the entropy is maximal, implying $\boldsymbol{\sigma}=0$. This occurs in both models for $\boldsymbol{c}_{eq} = \mathbf{1}$. The corresponding entropy value at equilibrium is zero.

A.1. *Steady-state extensional flow*

For steady-state extensional flow, the velocity gradient takes the form in 2-D,

$$\boldsymbol{\kappa} = \begin{bmatrix} \dot{\varepsilon} & 0 \\ 0 & -\dot{\varepsilon} \end{bmatrix} \tag{A7}$$

and the conformation tensor is independent of space and time and takes the form

$$\boldsymbol{c} = \begin{bmatrix} \frac{1}{1-2Wi} & 0 \\ 0 & \frac{1}{1+2Wi} \end{bmatrix}. \tag{A8}$$

In order to construct the residual (2.14), we need to express the velocity gradient in the eigenbasis $\boldsymbol{u}_i = \boldsymbol{v}_i/\|\boldsymbol{v}_i\|$ of the conformation tensor. The components in this basis are as follows:

$$\kappa_{\alpha\beta} \equiv \boldsymbol{u}_\alpha^T \cdot \boldsymbol{\kappa} \cdot \boldsymbol{u}_\beta. \tag{A9}$$

Note that you can multiply e_c by λ_p so that everything is given by Wi. As a result, $\kappa_{\alpha\beta}$ are given in terms of the Wi number,

$$\lambda_p \kappa_{11} = Wi, \tag{A10}$$

$$\lambda_p \kappa_{22} = -Wi, \tag{A11}$$

$$\lambda_p \kappa_{12} = \lambda_p \kappa_{21} = 0 \tag{A12}$$

and the residual equations become

$$\tilde{e}_c^1 = -2c_1 \lambda_p \kappa_{11} - c_1 \tilde{\sigma}_1, \tag{A13}$$

$$\tilde{e}_c^2 = -2c_2 \lambda_p \kappa_{22} - c_2 \tilde{\sigma}_2. \tag{A14}$$

$$\tag{A15}$$

A.2. *Steady-state shear flow*

In this flow the velocity gradient $\boldsymbol{\kappa}$, the conformation tensor $\boldsymbol{c}$ and the thermodynamic forces $\boldsymbol{\sigma}$ are

$$\boldsymbol{\kappa} = \begin{bmatrix} 0 & \dot{\gamma} \\ 0 & 0 \end{bmatrix}, \tag{A16}$$

$$\boldsymbol{c} = \begin{bmatrix} 1 + 2Wi^2 & Wi \\ \mathrm{W}i & 1 \end{bmatrix} \tag{A17}$$

and

$$\frac{\boldsymbol{\sigma}}{k_B T} = \boldsymbol{c}^{-1} - \boldsymbol{I} = \frac{1}{1 + Wi^2} \begin{bmatrix} -Wi^2 & -Wi \\ -\mathrm{W}i & Wi^2 \end{bmatrix}. \tag{A18}$$

The eigenvalues and non-normalized eigenvectors of the conformation tensor are

$$c_1 = 1 + Wi^2 + Wi\sqrt{1 + Wi^2}, \tag{A19}$$

$$c_2 = 1 + Wi^2 - Wi\sqrt{1 + Wi^2}, \tag{A20}$$

$$\boldsymbol{v}_1 = \begin{bmatrix} Wi + \sqrt{Wi^2 + 1} \\ 1 \end{bmatrix}, \tag{A21}$$

$$\boldsymbol{v}_2 = \begin{bmatrix} Wi - \sqrt{Wi^2 + 1} \\ 1 \end{bmatrix} \tag{A22}$$

while the eigenvalues of $\boldsymbol{\sigma}$ are

$$\frac{\sigma_1}{k_B T} = +\frac{Wi}{\left(1 + Wi^2\right)^{1/2}}, \tag{A23}$$

$$\frac{\sigma_2}{k_B T} = -\frac{Wi}{\left(1 + Wi^2\right)^{1/2}}. \tag{A24}$$

A.3. *Transient solutions to start-up extensional and shear flows*

Start-up uniaxial extension

The velocity/deformation vector is generally given by $v = (\dot{\varepsilon}x, -1/2\dot{\varepsilon}y, -1/2\dot{\varepsilon}z)$ for $t > 0$. However, for our simplified 2-D model, we are looking at planar extension. In that case, $v = (\dot{\varepsilon}x, -\dot{\varepsilon}y)$ and the velocity gradient is given by

$$\nabla v = \begin{bmatrix} \dot{\varepsilon} & 0 \\ 0 & -\dot{\varepsilon} \end{bmatrix}. \tag{A25}$$

The evolution of the conformation tensor using the OB model for start-up extensional flow (i.e. at time $t = 0$ stretching begins at strain rate $\dot{\varepsilon}$) is

$$c_{xx} = \frac{1}{2Wi - 1}\{2Wi \exp\left[(2Wi - 1)t/\lambda_p\right] - 1\}, \tag{A26}$$

$$c_{yy} = \frac{1}{2Wi + 1}\{2Wi \exp\left[-(2Wi + 1)t/\lambda_p\right] + 1\}, \tag{A27}$$

$$c_{xy} = 0. \tag{A28}$$

Note that these expressions are limited to $Wi < 0.5$. Furthermore, these expressions have the right limits: for $t = 0$ we have $c_{xx} = c_{yy} = 1$ and for $t \to \infty$ we arrive to the steady state extensional flow solution,

$$c_{xx} = \frac{1}{1 - 2Wi}, \tag{A29}$$

$$c_{yy} = \frac{1}{1 + 2Wi}. \tag{A30}$$

Start-up shear flow.

The OB start-up shear flow (i.e. walls start to move at $t = 0$ with shear rate $\dot{\gamma} = V_{wall}/H$) for $t > 0$. The velocity gradient is given by

$$\nabla \boldsymbol{v} = \begin{bmatrix} 0 & \dot{\gamma} \\ 0 & 0 \end{bmatrix}. \tag{A31}$$

The analytical solution for the evolution of the conformation tensor is given by

$$c_{xx} = 1 + 2Wi^2 \left(1 - \frac{t}{\lambda_p} \exp\left[-t/\tau_p\right] - \exp\left[-t/\lambda_p\right]\right), \tag{A32}$$

$$c_{yy} = 1, \tag{A33}$$

$$c_{xy} = Wi \left(1 - \exp\left[-t/\lambda_p\right]\right). \tag{A34}$$

Here again, the expressions have the right limits for $t = 0$ and for $t \to \infty$ with the steady-state solution

$$c_{xx} = 1 + 2Wi^2, \tag{A35}$$

$$c_{yy} = 1, \tag{A36}$$

$$c_{xy} = Wi. \tag{A37}$$

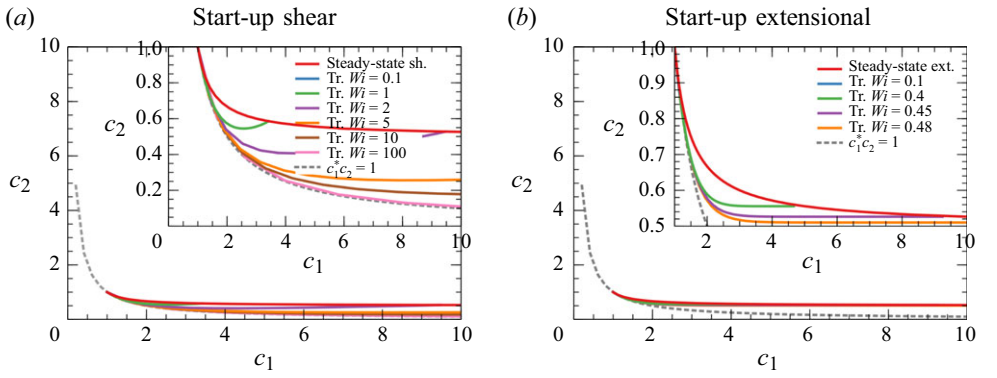


Figure 21. Start-up and steady-state shear and extensional flow analytic solutions for the OB model.

For shear flow,

$$\lambda_p \kappa_{11} = \frac{Wi^2 + Wi\sqrt{Wi^2+1}}{\|v_1\|^2}, \tag{A38}$$

$$\lambda_p \kappa_{22} = \frac{Wi^2 - Wi\sqrt{Wi^2+1}}{\|v_2\|^2}, \tag{A39}$$

$$\lambda_p \kappa_{12} = \frac{Wi^2 - Wi\sqrt{Wi^2+1}}{\|v_1\|\|v_2\|}, \tag{A40}$$

$$\lambda_p \kappa_{21} = \frac{Wi^2 + Wi\sqrt{Wi^2+1}}{\|v_1\|\|v_2\|} \tag{A41}$$

where we have used $\boldsymbol{u}_i = \boldsymbol{v}_i/\|v_i\|$, where $\boldsymbol{v}_i$ are the eigenvectors in (A21) and

$$\|v_1\| = \sqrt{2[1 + Wi^2 + Wi\sqrt{Wi^2+1}]}, \tag{A42}$$

$$\|v_2\| = \sqrt{2[1 + Wi^2 - Wi\sqrt{Wi^2+1}]}. \tag{A43}$$

Analytic solutions for shear and extensional steady-state and start-up flows are shown in figure 21.

A.4. *PINN-complex applied to FENE-P model at Wi = 0.5*

Using the same geometry in § 3.4 of the manuscript, we have validated the training of a PINN-complex network with data from a simulation of flow around a cylinder with RheoTool FENE-P implementation with $b = 100$ (see figure 22). The training was performed with data from a FV simulation at $Wi = 0.6$ and the validation of the model is down at $Wi = 0.5$.

It is noteworthy that in the c_1–c_2 map, we observe better agreement between the region covered by the model PINN-complex (FENE) and RheoTool implementation. When comparing figure 23 with the analogous figure 14 for OB, we observe that for FENE the regions explored by PINN-complex and FV in the c_1–c_2 map are more consistent for FENE. This is explained by a contraction of the c_1–c_2 map as a direct result of the finite extensibility.

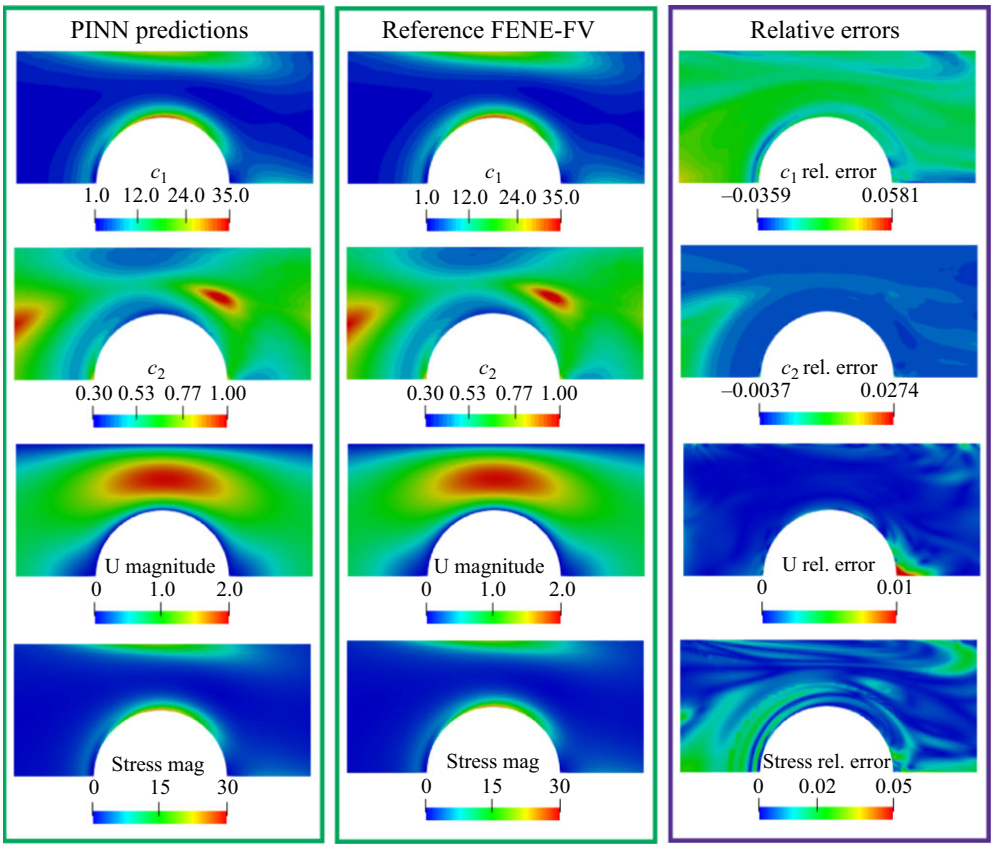


Figure 22. Comparison of PINN-complex (FENE) and RheoTool simulation results at $Wi = 0.5$. The stress relative error has been plotted with exponential decay when the magnitude of the true value is lower than 5 %.

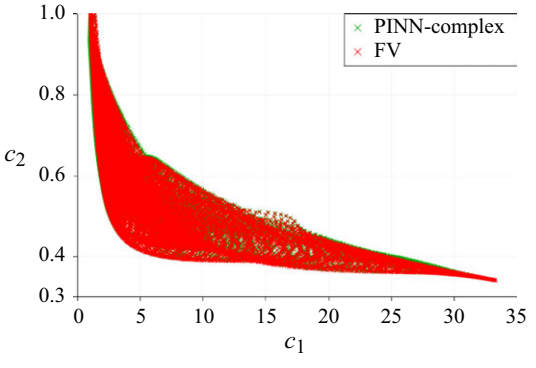


Figure 23. The c_1–c_2 region covered in simulations of flow around a cylinder with $Wi = 0.5$. Green crosses represent the results using the PINN-rheometric model, and red crosses represent the implementation of FENE-P in RheoTool (FV).

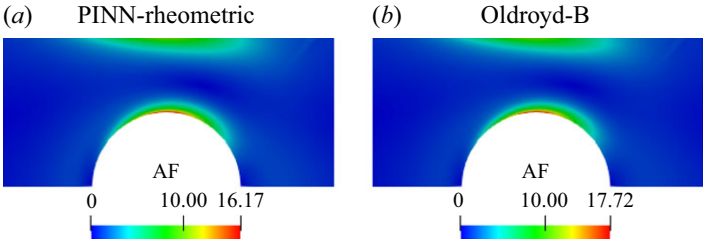


Figure 24. Anisotropy factor (AF) for flow around a cylinder at $Wi = 0.2$.

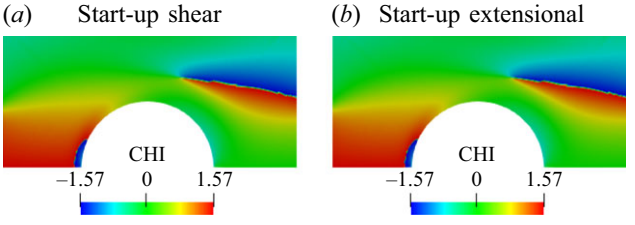


Figure 25. Orientation angle (χ) for flow around a cylinder at $Wi = 0.2$.

Appendix B. Anisotropy factor and orientation angle

The two typical metrics used in X-ray scattering are the anisotropy factor and the orientation angle as follows (Mao *et al.* 2014; McCready & Burghardt 2015):

$$AF = \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2} = \sqrt{\mathrm{tr}\boldsymbol{c}^2 - 4\det \boldsymbol{c}} = \sqrt{(c_1 + c_2)^2 - 4c_1c_2}, \tag{B1}$$

$$\chi = \frac{1}{2}\arctan\left(\frac{2c_{12}}{c_{11} - c_{22}}\right). \tag{B2}$$

These are computed from a measurement like the scattering vector $\boldsymbol{q}$ in X-ray scattering (or the polarization vector $\boldsymbol{p}$ in birefringence). In order to compare such measurements with the conformation tensor, we are assuming that these two properties are proportional to the end-to-end vector $\boldsymbol{r}$ of the polymer strands and therefore the second moments of these (i.e. $\langle \boldsymbol{qq} \rangle$ or $\boldsymbol{n} = \langle \boldsymbol{pp} \rangle$ are proportional to the conformation tensor $\boldsymbol{c} = \langle \boldsymbol{rr} \rangle$). Figures 24 and 25 present a comparison of these two metrics for PINN-rheometric and the OB model for flow around a cylinder at $Wi = 0.2$. The orientation angle has been computed with the function arctan2 since it tracks the quadrants.

References

Alves, M.A., Pinho, F.T. & Oliveira, P.J. 2001 The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *J. Non-Newtonian Fluid Mech.* **97** (2), 207–232.

Arzani, A., Cassel, K.W. & D'Souza, R.M. 2023 Theory-guided physics-informed neural networks for boundary layer problems with singular perturbation. *J. Comput. Phys.* **473**, 111768.

Bird, R.B., Bird, R.B., Armstrong, R.C. & Hassager, O. 1987 Dynamics of Polymeric Liquids. In *Volume 1: Fluid Mechanics*. Wiley.

Bonfanti, A., Bruno, G. & Cipriani, C. 2024 The challenges of the nonlinear regime for physics-informed neural networks. arXiv:2402.03864.

Bonfanti, A., Santana, R., Ellero, M. & Gholami, B. 2023 On the generalization of PINNs outside the training domain and the hyperparameters influencing it. arXiv:2302.07557.

Claus, S. & Phillips, T.N. 2013 Viscoelastic flow around a confined cylinder using spectral/hp element methods. *J. Non-Newtonian Fluid Mech.* **200**, 131–146.

CUOMO, S., DI COLA, V.S., GIAMPAOLO, F., ROZZA, G., RAISSI, M. & PICCIALLI, F. 2022 Scientific machine learning through physics–informed neural networks: where we are and what's next. *J. Sci. Comput.* **92** (3), 88.

DYSON, F. 2004 A meeting with Enrico Fermi. *Nature* **427** (6972), 297–297.

ELLERO, M. & ADAMS, N.A. 2011 SPH simulations of flow around a periodic array of cylinders confined in a channel. *Intl J. Numer. Meth. Engng* **86** (8), 1027–1040.

HAWARD, S.J., HOPKINS, C.C., VARCHANIS, S. & SHEN, A.Q. 2021 Bifurcations in flows of complex fluids around microfluidic cylinders. *Lab Chip* **21** (21), 4041–4059.

HERNÁNDEZ, Q., BADÍAS, A., GONZÁLEZ, D., CHINESTA, F. & CUETO, E.D. 2021 Structure-preserving neural networks. *J. Comput. Phys.* **426**, 109950.

HERRCHEN, M. & ÖTTINGER, H.C. 1997 A detailed comparison of various FENE dumbbell models. *J. Non-Newtonian Fluid Mech.* **68** (1), 17–42.

HORNIK, K., STINCHCOMBE, M. & WHITE, H. 1989 Multilayer feedforward networks are universal approximators. *Neural Networks* **2** (5), 359–366.

KARNIADAKIS, G.E., KEVREKIDIS, I.G., LU, L., PERDIKARIS, P., WANG, S. & YANG, L. 2021 Physics-informed machine learning. *Nat. Rev. Phys.* **3** (6), 422–440.

KINGMA, D.P. & BA, J. 2014 Adam: a method for stochastic optimization. arXiv:1412.6980.

KRÖGER, M. 2010 *Models for Polymeric and Anisotropic Liquids*. Springer.

LARSON, R.G. 1988 *Constitutive Equations for Polymer Melts and Solutions*. Butterworths.

LEI, H., WU, L. & E, W. 2020 Machine-learning-based non-Newtonian fluid model with molecular fidelity. *Phys. Rev. E* **102** (4), 043309.

LIN, B., MAO, Z., WANG, Z. & KARNIADAKIS, G.E. 2023 Operator learning enhanced physics-informed neural networks for solving partial differential equations characterized by sharp solutions. arXiv:2310.19590.

LINKA, K. & KUHL, E. 2023 A new family of Constitutive Artificial Neural Networks towards automated model discovery. *Comput. Meth. Appl. Mech. Engng* **403**, 115731.

MAHMOUDABADBOZCHELOU, M., KAMANI, K.M., ROGERS, S.A. & JAMALI, S. 2022*a* Digital rheometer twins: learning the hidden rheology of complex fluids through rheology-informed graph neural networks. *Proc. Natl Acad. Sci. USA* **119** (20), e2202234119.

MAHMOUDABADBOZCHELOU, M., KARNIADAKIS, G.E. & JAMALI, S. 2021*b* nn-PINNs: Non-Newtonian physics-informed neural networks for complex fluid modeling. *Soft Matt.* **18** (1), 172–185.

MAO, R., MCCREADY, E.M. & BURGHARDT, W.R. 2014 Structural response of an ordered block copolymer melt to uniaxial extensional flow. *Soft Matt.* **10** (33), 6198–6207.

MCCREADY, E.M. & BURGHARDT, W.R. 2015 Structural response of a prealigned cylindrical block copolymer melt to extensional flow. *J. Rheol.* **59** (4), 935–956.

MENG, X., LI, Z., ZHANG, D. & KARNIADAKIS, G.E. 2020 PPINN: Parareal physics-informed neural network for time-dependent PDEs. *Comput. Meth. Appl. Mech. Engng* **370**, 113250.

NIETO SIMAVILLA, D. & ELLERO, M. 2022 Mesoscopic simulations of inertial drag enhancement and polymer migration in viscoelastic solutions flowing around a confined array of cylinders. *J. Non-Newtonian Fluid Mech.* **305**, 104811.

NIETO SIMAVILLA, D., ESPAÑOL, P. & ELLERO, M. 2023 Non-affine motion and selection of slip coefficient in constitutive modelling of polymeric solutions using a mixed derivative. *J. Rheol.* **67** (1), 253–267.

OLDROYD, J.G. & WILSON, A.H. 1950 On the formulation of rheological equations of state. *Proc. R. Soc. Lond. A: Math. Phys. Sci.* **200**, 523–541.

OTTINGER, H.C. 2005 *Complex fluids*. John Wiley Sons, Inc.

DE PABLO, J.J. & SCHIEBER, J.D. 2014 *Molecular Engineering Thermodynamics*. Cambridge University Press.

PANG, G., LU, L. & KARNIADAKIS, G.E. 2019 fPINNs: Fractional physics-informed neural networks. *SIAM J. Sci. Comput.* **41** (4), A2603–A2626.

PASZKE, A., GROSS, S., CHINTALA, S., CHANAN, G., YANG, E., DEVITO, Z., LIN, Z., DESMAISON, A., ANTIGA, L. & LERER, A. 2017 Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*. Long Beach California, USA. Available at: https://openreview.net/forum?id=BJJsrmfCZ.

PIMENTA, F. & ALVES, M.A. 2017 Stabilization of an open-source finite-volume solver for viscoelastic fluid flows. *J. Non-Newtonian Fluid Mech.* **239**, 85–104.

RACKAUCKAS, C., MA, Y., MARTENSEN, J., WARNER, C., ZUBOV, K., SUPEKAR, R., SKINNER, D., RAMADHAN, A. & EDELMAN, A. 2021 Universal differential equations for scientific machine learning. arXiv:2001.04385.

RAISSI, M., PERDIKARIS, P. & KARNIADAKIS, G.E. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.

RODRIGUEZ, S. & CARDIFF, P. 2022 A general approach for running Python codes in OpenFOAM using an embedded PYBIND11 Python Interpreter. *OpenFOAM® J.* **2**, 166–182.

ROSENKRANZ, M., KALINA, K.A., BRUMMUND, J., SUN, W. & KÄSTNER, M. 2024 Viscoelasticty with physics-augmented neural networks: model formulation and training methods without prescribed internal variables. *Comput. Mech.* Available at: arXiv:2401.14270.

SAADAT, M., MAHMOUDABADBOZCHELOU, M. & JAMALI, S. 2022 Data-driven selection of constitutive models via rheology-informed neural networks (RhINNs). *Rheol. Acta* **61** (10), 721–732.

SCHMID, J.D., BAUERSCHMIDT, P., GURBUZ, C., ESER, M. & MARBURG, S. 2024 Physics-informed neural networks for acoustic boundary admittance estimation. *Mech. Syst. Signal Process.* **215**, 111405.

SERYO, N., SATO, T., MOLINA, J.J. & TANIGUCHI, T. 2020 Learning the constitutive relation of polymeric flows with memory. *Phys. Rev. Res.* **2** (3), 033107.

SHAQFEH, E.S.G. & KHOMAMI, B. 2021 The Oldroyd-B fluid in elastic instabilities, turbulence and particle suspensions. *J. Non-Newtonian Fluid Mech.* **298**, 104672.

SUKUMAR, N. & SRIVASTAVA, A. 2022 Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Comput. Meth. Appl. Mech. Engng* **389**, 114333.

THAKUR, S., RAISSI, M. & ARDEKANI, A.M. 2022 Viscoelasticnet: a physics informed neural network framework for stress discovery and model selection. arXiv:2209.06972.

THIEN, N.P. & TANNER, R.I. 1977 A new constitutive equation derived from network theory. *J. Non-Newtonian Fluid Mech.* **2** (4), 353–365.

VÁZQUEZ-QUESADA, A., ELLERO, M. & ESPAÑOL, P. 2009*a* Consistent scaling of thermal fluctuations in smoothed dissipative particle dynamics. *J. Chem. Phys.* **130** (3), 034901.

VÁZQUEZ-QUESADA, A., ELLERO, M. & ESPAÑOL, P. 2009*b* Smoothed particle hydrodynamic model for viscoelastic fluids with thermal fluctuations. *Phys. Rev. E* **79** (5), 056707.

WANG, S., YU, X. & PERDIKARIS, P. 2022 When and why PINNs fail to train: a neural tangent kernel perspective. *J. Comput. Phys.* **449**, 110768.

XU, K., TARTAKOVSKY, A.M., BURGHARDT, J. & DARVE, E. 2021 Learning viscoelasticity models from indirect data using deep neural networks. *Comput. Meth. Appl. Mech. Engng* **387**, 114124.

YOUNG, C.D., CORONA, P.T., DATTA, A., HELGESON, M.E. & GRAHAM, M.D. 2023 Scattering-Informed Microstructure Prediction during Lagrangian Evolution (SIMPLE)—a data-driven framework for modeling complex fluids in flow. *Rheol. Acta* **62** (10), 587–604.

YUAN, L., NI, Y.-Q., DENG, X.-Y., HAO, S. 2022 A-PINN: auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *J. Comput. Phys.* **462**, 111260.

ZHANG, D., LU, L., GUO, L. & KARNIADAKIS, G.E. 2019 Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *J. Comput. Phys.* **397**, 108850.

ZHANG, Z., SHIN, Y. & EM KARNIADAKIS, G. 2022 GFINNs: GENERIC formalism informed neural networks for deterministic and stochastic dynamical systems. *Phil. Trans. R. Soc. Lond. A Math. Phys. Engng Sci.* **380** (2229), 20210207.

ZHAO, L., LI, Z., CASWELL, B., OUYANG, J. & KARNIADAKIS, G.E. 2018 Active learning of constitutive relation from mesoscopic dynamics for macroscopic modeling of non-Newtonian flows. *J. Comput. Phys.* **363**, 116–127.

ZHAO, L., LI, Z., WANG, Z., CASWELL, B., OUYANG, J. & KARNIADAKIS, G.E. 2021 Active- and transfer-learning applied to microscale-macroscale coupling to simulate viscoelastic flows. *J. Comput. Phys.* **427**, 110069.