jh210017-MDH

Development of physics informed machine learning for soft matter: polymer flows and beyond

John Molina (Kyoto University)

Abstract

Soft Matter systems, among which we find colloidal dispersions, foams, cellular tissues, and polymeric materials, are ubiquitous in life sciences and industry, and have become indispensable for modern technologies. Unfortunately, due to the coupled hierarchy of length and time scales involved, predicting their macroscopic material properties remains an incredibly challenging task, even if we have complete understanding of the underlying microscopic components. Recently, computer simulations, and multi-scale simulations (MSS) in particular, have been successfully used to simulate these systems (e.g., MSS of polymer melt-spinning processes), however the computational cost remains prohibitive. Therefore, the goal of this project is to develop physics-informed machine learning (ML) methods that allow us to accelerate/replace currently used numerical methods, without any considerable loss of accuracy or predictive capabilities. We have mainly focused on developing a method to learn the constitutive relation of polymer melt flows from microscopic training data. These constitutive relations can then be used within macroscopic flow simulations, for one to two orders of magnitude speedup. We have also investigated other soft-matter problems that can benefit from similar ML accelerators, such as flows at low Reynolds number, the dynamics of growing cellular tissues, or the control of smart agents in complex environments.

1 Basic Information

1.1 Collaborating JHPCN Centers

Tokyo, Nagoya

1.2 Research Areas

- Very large-scale numerical computation
- Very large-scale data processing

1.3 Roles of Project Members

- John Molina (Kyoto University): Development of Machine Learning methods to simulate soft matter systems.
- Hayato Shiba (The University of Tokyo): Guidance and support for accelerating cpu/gpu code.
- Ryoichi Yamamoto (Kyoto University): Guidance and support for soft matter modeling.
- Takashi Taniguchi (Kyoto University): Development of multi-scale simulation methods for polymer flows.
- Takashi Shimokawabe (The University of Tokyo):Guidance and support for accelerating gpu code.
- Takeshi Sato (Kyoto University): Development of multi-scale simulation methods for polymer flows.
- Souta Miyamoto (Kyoto University): Development of machine-learning and multi-scale simulation methods for polymer flows.
- Yan Xu (Kyoto University): Development of multi-scale simulation methods for polymer flows.

- Takumi Ueda (Kyoto University): Development of multi-scale simulation methods for polymer flows.
- Simon Schnyder (Kyoto University): Development of simulation methods for cellular tissues.
- Jintao Li (Kyoto University): Development of simulation methods for cellular tissues.
- Krongtum Sankaewtong (Kyoto University): Development of machine-learning methods to control active systems.

2 Purpose and significance of Research

Soft Matter systems, which include colloidal dispersions, polymeric materials, and cellular tissues, are characterized by a coupled hierarchy of length- and time-scales, which provides them with a variety of interesting material properties. Unfortunately, this makes them incredibly challenging to study. Thus, the goal of this work has been to develop physics-informed ML methods to efficiently and accurately simulate the dynamics of soft materials. Due to their significance for material science, the main focus of this project has been to accelerate polymer flow predictions.

Polymeric materials are typically produced using polymer melt processing, during which the molten polymer is allowed to flow, in order to mold it to the desired shape. Unfortunately, we do not yet fully understand how to explain the coupling between the microscopic polymer chain dynamics and the macroscopic flow, which makes it incredibly difficult to simulate such processes. The traditional approaches used to tackle such problems are: (1) MSS to directly couple the micro/macro degrees of freedom or (2) a fully macroscopic description using a given constitutive relation. The former is incredibly expensive, while the latter usually lacks a clear theoretical foundation. Thus, we have proposed to develop a ML method to directly learn the constitutive relation for the stress of entangled polymeric materials with memory. This allows us to drastically reduce the calculation time compared with state-of-the-art MSS, with no loss of accuracy.

Additionally, we have investigated how to develop ML solvers for Stokes flow equations, which describe the dynamics of fluids at low Reynolds numbers (e.g., swimming microorganisms or high-viscosity polymer flows), as well as how to learn optimal control strategies for agents in complex environments (e.g., swimming particles under external flows, rational individuals in a pandemic), and develop MSS for cellular tissues.

3 Significance as JHPCN Joint Research Project

In this project, we are developing ML methods to (1) accelerate large-scale simulations of soft-matter materials, particularly focusing on polymeric flows, and (2) learn optimal control strategies for agents in complex environments. Both the generation of the training data, and the learning itself, can require a considerable amount of computing resources. For example, to learn the constitutive relations of polymeric systems, we require microscopic simulations with $N_p \simeq 10^4 - 10^5$ entangled polymer chains, with $\mathcal{O}(10^2)$ different external flow conditions, in order to generate a suitable training data set. When learning optimal control strategies for swimming systems, the training typically requires $\mathcal{O}(10^3)$ epochs, with each epoch requiring a direct numerical simulation to resolve the many-body hydrodynamic interactions. In all cases, we have used custom simulation codes to generate the required training data. Unfortunately, our codes were originally written with CPU clusters in mind, so a considerable amount of effort has gone towards porting and optimizing to GPUs. For the learning, we have mainly used PyTorch and GPyTorch, as well as custom JAX/JAXOPT code, with the later providing excellent performance on the GPUs, allowing us to take full advantage of the GPU clusters (Wisteria-BDEC and FURO Type-II).

4 Outline of Research Achievements up to FY2020

Not Applicable.

5 Details of FY2021 Research Achievements

5.1 Learning for Polymer Processing Flows

5.1.1 Extension of MSS

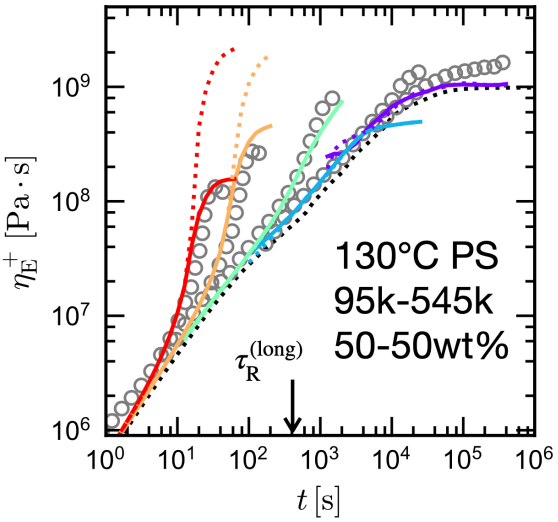

Fig. 1 Prediction of transient viscosities for a Polystyrene blend under uniaxial elongations. (symbols) Experimental data from Hengeller et al. (Rheologica Acta 55, 3030, 2016), (dotted line) simulations without SORF, (solid line) simulations with SORF, with $\tau_R^{(\mathrm{long})}$ the Rouse relaxation time. (S. Miyamoto, Master thesis, Kyoto Univ. 2022)

First, we will discuss our achievement in further developing the MSS model for polymer flows. We have performed a rheological study to extend and validate the Doi-Takimoto (DT) entanglement model for bi-disperse polymer melts. In particular, we confirmed that the extended DT model with stretch/orientation induced reduction of friction (SORF) can correctly reproduce the strain hardening under fast elongational flows that has been observed experimentally (Fig. 1).

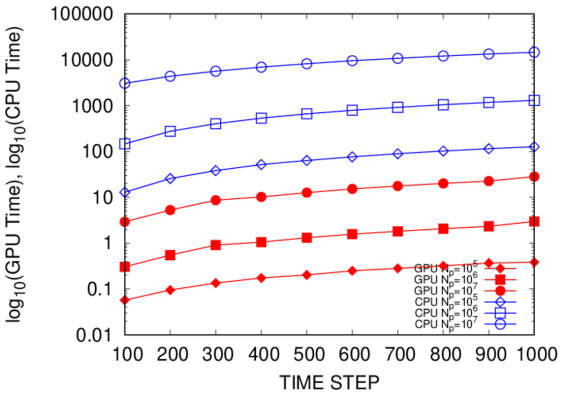

Fig. 2 Logarithm of calculation time for a non-interacting Hookean dumbbell model as a function of simulation time step, using $N_p = 10^5, 10^6, 10^7$. (filled symbols) single GPU runs, (open symbols) single CPU runs.

To accelerate the MSS, which is required to validate the results of our learned constitutive relations, one of the main goals of this project was to port our code to GPU. However, due to the complexity of the polymer entanglement inherent to the DT model, and our relative inexperience with GPU coding, we have started by porting the non-interacting polymer MSS models for both Hookean dumbbells (OpenACC and CUDA) and Rouse chains (CUDA). Given their non-interacting nature, the models were almost trivially parallelizable, requiring no specialized libraries or algorithms. Our implementation was tested by simulating planar Poiseuille flow in 1D. The single GPU calculation was $\mathcal{O}(10^3)$ times faster than a single CPU, and $\mathcal{O}(10)$ times faster than a 25 core CPU/MPI calculation (Fig.2). This dramatic speedup provided by the GPU allows us to use $N_p \gg 1$, which significantly reduces statistical errors in the calculation (whereas the CPU runs are limited to $N_p \lesssim 10^3$).

5.1.2 Learning the Constitutive Relation

To simulate large-scale polymer processing flows we have extended our learning method to learn the constitutive relation of entangled polymer melts. In particular, we have shown that it is possible to learn the constitu-

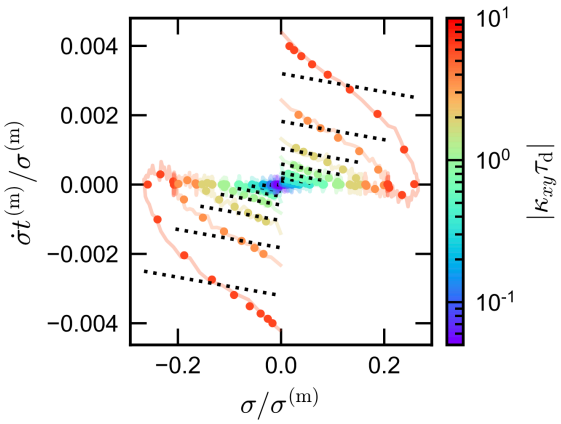

Fig. 3 Scaled training data for the constitutive relation generated from microscopic simulations of a mono-disperse Doi-Takimoto model under start-up shear flows. (lines) raw data and (symbols) smoothed data for the xy components of the shear stress σ and its time-derivative $\dot{\sigma}$.(S. Miyamoto, Master Thesis, Kyoto Univ. 2022)

tive equation for the stress of the DT model from a reduced set of microscopic training data. For this, we performed DT simulations using $N_p = 10^5$ polymer chains under 20 start-up shear flows. The training data is shown in Fig. 3, which displays the relation between the shear-stress and its time derivative. Crucially, it is seen that the relationship is two-valued for fast shear flows, which means that $\dot{\sigma}_{xy}$ cannot be a function only of σ_{xy} and κ_{xy} (with κ_{xy} the xy component of the velocity gradient tensor), i.e., additional descriptors are required to establish a constitutive relation. For this, we have introduced the number of entanglements Z. Although other descriptors are possible, Z scored the highest Pearson correlation coefficient with $\dot{\sigma}$ in our analysis. We then use a Gaussian Process (GP) regression scheme, in order to learn the constitutive relations $\dot{\sigma}_{xy}(\sigma_{xy}, Z, \kappa_{xy})$ and $\dot{Z}(\sigma_{xy}, Z, \kappa_{xy})$. For this, we have used the GPyTorch library. Unfortunately, this library does not fully support the A100 GPUs provided on Wisteria, so we were forced to run the learning using CPUs only.

Finally, we have also performed detailed microscopic (cell-level) simulations to study the effect of cell competition in proliferating tissues. This has provided us with ample training data to learn constitutive equations for cellular tissues. However, the variable density makes this a much more difficult learning task, one which we have yet to solve.

5.1.3 Flow Predictions using Learned Constitutive Relation

With the learned constitutive relations, we can perform polymer flow simulations that are $\mathcal{O}(10^2)$ times faster than full MSS, which we refer to as GP accelerated MSS (GPMSS). As a reference, MSS with $N_p = 10^5$ require ~ 0.5 s/step (8 CPU cores), whereas GPMSS require ~ 2 ms/step. Simulation results are shown in Fig.4, where good overall agreement is observed between MSS and GPMSS. The stress and entanglement profiles are in good quantitative agreement, although the velocity profiles exhibit some noticeable differences, particularly in the transient regime. These differences appear in decreasing stress regions of the constitutive relation space ($\dot{\sigma}_{xy} < 0$, $\sigma_{xy} > 0$), in which we have little to no training data, showing the limitations of Gaussian Process extrapolation. To address this issue we must improve the protocol we are using to generate the training data.

5.2 Learning for Stokesian Flows

When considering flows through complex domains or with moving boundaries, traditional flow solvers, both Eulerian (e.g., finite-element, spectral methods) and Lagrangian (e.g., smooth particle hydrodynamics), can become incredibly expensive due to the resolution requirements. In addition, for high-viscosity fluids we would like to directly solve the Stokes equation, instead of the full Navier-Stokes equation, which turns into a boundary value problem that can be cumbersome to solve, particularly for non-Newtonian fluids. In this work, leveraging the linearity of the Stokes equation, we have investigated the use of Gaussian Processes to formulate flow problems as inference problems. This allows us to consider missing

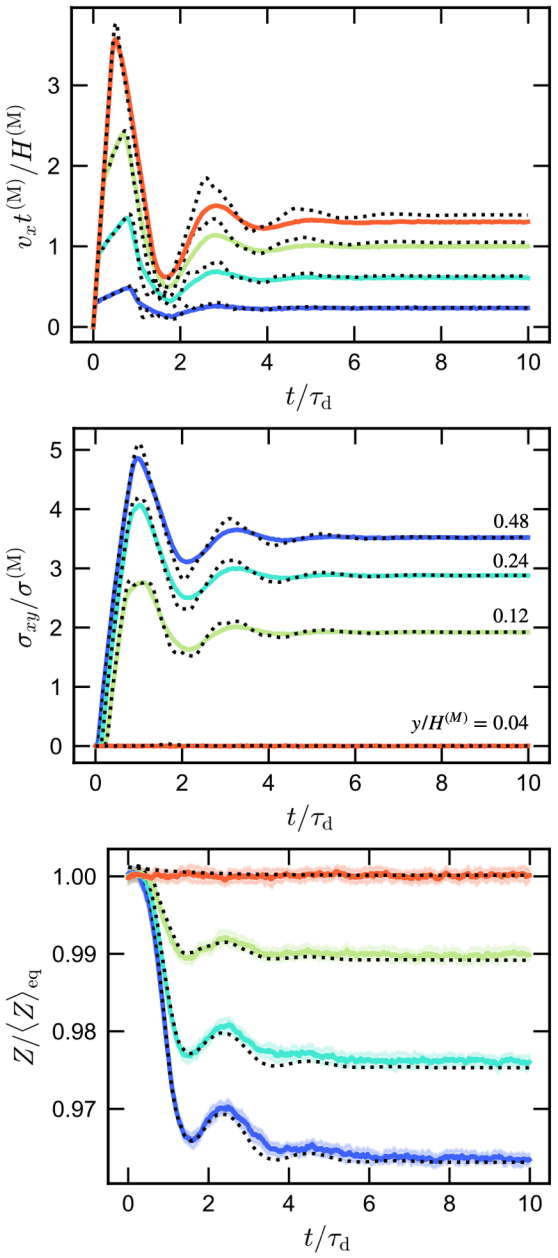

Fig. 4 Scaled simulation results for planar Poiseuille flows using full MSS (dotted black line) and GPMSS (solid colored line). (top) velocities v_x, (middle) stress σ_{xy}, and (bottom) number of entanglements Z as a function of time for different points along y the height of the channel (width $H^{(M)}$). (S. Miyamoto, Master thesis, Kyoto Univ. 2022)

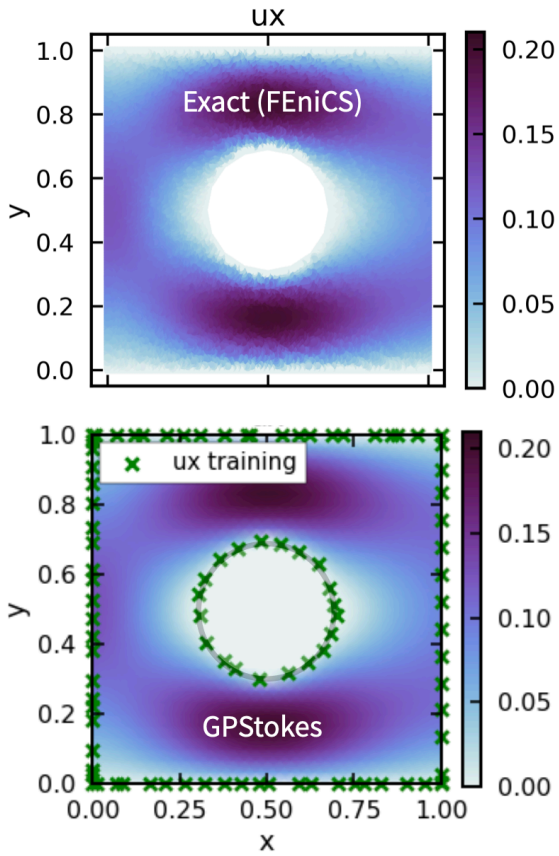

Fig. 5 2D Pressure driven flow past a cylinder. (top) reference solution obtained using the FEniCS python package with $\mathcal{O}(10^5)$ elements , (bottom) GPStokes inference with $\mathcal{O}(10^3)$ total (test + training) points. Crosses represent the training values $v = 0$ used for the velocities at the boundaries of the wall and cylinder. (K. Ogawa, Undergraduate Thesis, Kyoto Univ.)

and/or noisy data, arbitrary flow geometries with arbitrary computational grids, and it does not require the discretization of the differential operators (which plagues traditional methods).

For this, we consider the velocity u, pressure p, total force g, and velocity divergence fields $h = \nabla \cdot u$ to be given by a joint GP. We then compute the conditional GP $P(u, p \,|u_b, p_b, g, h)$, using as known training data the velocity and pressure at the boundaries, u_b and p_b, as well as the force balance $g = 0$ (Stokes equation) and the divergence

free condition $h = 0$. This allows us to encode the physics of the problem in terms of the correlations between the different random fields. For this, we rely heavily on JAX's automatic differentiation capabilities. Finally, the average for the velocity and pressure fields, which provide us with the best solution, can be computed using exact GP regression (within machine precision). Results for the case of pressure driven flow past a cylinder are shown in Fig. 5. In this case, the boundary points have been chosen randomly, whereas the training points for the force-balance and divergence-free conditions have been placed on a regular lattice (with a small random displacement to avoid singularities in the correlation matrices). This ability to use arbitrary grids allows us to easily vary the resolution in regions that require it, as well as allowing for exact treatment of sharp interfaces. We obtain excellent agreement with the reference solution (maximum absolute error is $\simeq 10^{-3}, 10^{-4}, 10^{-1}$ for v_x, v_y, and p, respectively), but require a significantly smaller number of elements. Although the training is relatively expensive, we expect this method to become attractive in situations where traditional methods fail or become too expensive, e.g., analyzing experimental data with missing information, solving particle dynamics in non-Newtonian host fluids at zero Reynolds number, or finding flow solutions for complex geometries with mixed boundary conditions.

5.3 Learning for Optimal Control

To understand how micro-organisms sense mechanical signals from their environment, we have considered how to learn optimal control strategies for swimmers navigating through non-uniform flow fields. For this, we use deep-Q learning, with prioritized experience delay and n-step learning, together with direct numerical simulations (DNS), in order to fully solve for the hydrodynamic flows. A feed-forward Neural Network (NN), with three hidden layers (100 neurons each), is used to learn the mapping between the swimmer's local information (e.g., local stress) and the preferred action (e.g., external torque). The learning is performed using the PyTorch library (using the ADAM optimizer), whereas the DNS is performed using the KAPSEL colloidal simulator (simulation grid of $32 \times 32 \times 64$ points). This required us to interface the KAPSEL C++ code with the python PyTorch libraries: every 10 simulation steps the current swimmer state (e.g., velocity, orientation, surface stresses) is fed into the NN to obtain the action to be used over the next 10 steps. Training was carried out over $\sim 10^3$ episodes, with a single episode consisting of $\sim 10^3$ DNS time-steps. Depending on the number of particles considered ($N = 1 \sim 32$), training could take between $24-72$ hours. Here, we faced significant difficulties arranging the memory transfer between C++/Python, which likely explains the excessive training times. We have successfully obtained optimal control strategies for a single force-free swimmer tasked with moving in a zig-zag shear flow, either parallel to the velocity gradient or perpendicular to the shear-plane, using only local information.

We have also considered an inverse optimal control problem, in which observations of the optimal behaviour of an agent are used to infer the underlying cost or utility function that gave rise to this motion. For this, we have considered a game theoretic formulation of social distancing in response to an epidemic. As the state of the epidemic progresses, rational individuals will adapt their strategy in such a way that maximizes their individual utility. Assuming that all individuals are the same, we arrive at a Nash equilibrium strategy as the "optimal" strategy, even though it is not the strategy that maximizes the sum total of the population's utility (the utilitarian maximum). The direct solution to such a problem will strongly depend on the choice of the individual's utility, which is impossible to measure. Therefore, we have developed a novel physics informed NN, which we refer to as a Nash Neural Network (N^3), that allows us to learn the utility in an unsupervised manner from the observed optimal behaviour (Molina et

al., arXiv:2203.13432). For this, we have encoded the underlying Euler-Lagrange (EL) equations, as well as the game-theoretic Nash equilibrium, into the structure of the Networks. Crucially, and in contrast to previous works, the N^3 self-consistently evaluates itself on the optimal control. To achieve this, we have used JAX's automatic differentiation capabilities and functional nature, which allow us to systematically derive the EL equations from black-box utilities (encoded with a simple NN), as well as JAXOPT's differentiable optimizers, which are needed to take gradients through the optimality condition at which the N^3 is evaluated. We have tested our system on synthetic data, assuming SIR dynamics with a known utility, which we were able to successfully recover. We used a feed-forward NN with three hidden layers (128 neurons each), trained with the ADAM optimizer over $\sim 10^5$ steps. We note that the complexity of the learning, caused by the highly non-linear loss function, absolutely required the use of the A100 GPUs on Wisteria: even for a relatively small number of training points($\lesssim 10^3$), training would take more than 24 hours on a single GPU.

6 Progress during FY2021 and Future Prospects

Our original plan was divided in four themes : (A) full MSS validation, (B) constitutive equation learning, (C) simulation of Polymer Processing flows, and (D) extensions to other Soft Matter systems. With regards to (A), we had intended to fully parallelize our MSS code, both the microscopic polymer simulator and macroscopic flow solver, and port it over to the GPU. While we have successfully ported the dumbbell and Rouse simulators to the GPU, we have not yet succeeded in porting the DT simulator, owing to the complexities of handling the entanglements. With regards to (B) and (C), we have successfully learned the DT constitutive relation, and used it to perform macroscopic flow simulations (resulting in $\mathcal{O}(10^2)$ speed increases). While the agreement is good overall, there are noticeable differences in the transient velocity predictions, particularly for highly elastic systems. We have traced this problem to the lack of training data in select regions of the constitutive relation space. For (D), we have considered several applications, covering a wide variety of areas in Soft Matter. In particular, using the same ML techniques used to learn the constitutive relations (i.e., Gaussian Process regression), we have developed a probabilistic flow solver for the Stokes equations. We have also developed ML methods that can be applied to direct and inverse optimal control problems, e.g., for learning efficient swimming strategies or inferring utilities from optimal behaviour.

Our research plan for next year will continue with the themes we have developed, focusing on three items: (1) Learning the constitutive relations of entangled polymer melts, (2) Learning the Stokes equation, and (3) Learning optimal control strategies. From a technical point of view, we aim to move most of our codebase to JAX, replacing PyTorch and GPyTorch, due to it's reduced development time, ability to target CPU/GPU/TPU, and high-performance. In our experience, well-written JAX code can run as fast as optimized C/C++ code. For (1), we need to improve our training protocol, in order to selectively target the regions of constitutive equation space that we have missed. In addition, we must also finish porting our DT model to the GPU, in order to fully parallelize both the MSS and the GPMSS. For the MSS, this will require us finding a suitable data-structure/algorithm to handle the insertion/deletion of entanglement points. For the GPMSS we would like to replace GPyTorch with our own custom JAX code. This will require that we implementing GPyTorch's Black-Box Matrix Matrix method. This port will also be used for item (2), in order to allow us to perform exact inference with up to millions of training points (which will likely be required for 3D flows). Regarding item (3), we will continue with the two problems we have discussed

above, optimal control of swimmers, and inverse optimal control for rational individuals. For the swimmers, we will need to optimize the data transfer between C++/Python, implement the Deep Q-Learning in JAX, incorporate torque-free control and improve the learning for many-particle systems. Finally, for the rational agents, we will work on performing the inference using only observed behaviour, without using the full state of the system (which includes Lagrange multipliers that are impossible to measure in real-life).

7 List of Publications and Presentations

Journal Papers (Refereed)

Presentations

- J. Li, S.K. Schnyder, M.S. Turner(+), R. Yamamoto, 'Cell migration and colony growth in a monolayer of model cells', 11th Liquid Matter Conference, online, 18-23 July, 2021
- T. Taniguchi, 'Multi-scale simulations of Polymer Melt Flows', 49th AIST Seminar: "Technological Prospects and Applications of Multi-scale and Coupled Analysis", online, 31 August and 2 September, 2021.
- S. Miyamoto, J.J. Molina, T. Taniguchi, 'Flow simulation of polymeric liquids using learned constitutive relations', The Society of Chemical Engineers, Japan 52nd Fall Meeting, online, 22-25 September, 2021. –Best Presentation Award–
- T. Taniguchi, Y. Hamada, T. Sato, 'Hybrid simulations of polymer melt spinning using molecular dynamics and a conventional macroscopic model', PPS-36: The 36th International Conference of the Polymer Processing Society, online, 27-29 September, 2021.
- S.Miyamoto, J.J. Molina, T. Taniguchi, 'Well-entangled polymer melt flow simulations using a Machine-Learned constitutive relation', 69th Rheology Symposium, online, 20-21 October, 2021.
- J.J. Molina and T. Taniguchi, 'A Machine-Learning Approach to Flow Problems', 69th Rheology Symposium, online, 20-21 October, 2021.
- T. Taniguchi, 'Multi-Scale Simulation for well-entangled polymer melt flows', 19th International Symposium on Applied Rheology, online, 11-12 November, 2021. –Invited Plenary–
- J.J. Molina and T. Taniguchi, 'Gaussian Processes for Machine Learning of Fluid Flows',Softbio Workshop 2021, Akita, Japan, 16-18 November, 2021. –Invited–
- S. Miyamoto, J.J. Molina, T. Taniguchi, 'Flow simulation of a polymer melt using a machine-learned constitutive relation', 29th Autumn Meeting of the Japan Society of Polymer Processing, online, 30 November- 1 December, 2021.
- J.J. Molina, S. Miyamoto, T. Taniguchi, 'Learning the constitutive relation of polymer melt flows', Pacifichem 2021, online, 16-21 December, 2021.
- J. Li, S.K. Schnyder, M.S. Turner(+), R. Yamamoto, 'Competition between two cell types under cell cycle regulation with apoptosis', American Physical Society March Meeting 2022, 14-18 March, 2022.
- S. Miyamoto, J.J. Molina, T. Taniguchi, 'Application of machine-learned constitutive relations for well-entangled polymer flows', American Physical Society March Meeting 2022, 14-18 March, 2022.
- J.J. Molina, S.K. Schnyder, M.S. Turner(+), R. Yamamoto, 'Nash Neural Networks', American Physical Society March Meeting 2022, 14-18 March, 2022.

Proceedings of International Conferences (Refereed)

Proceedings of International Conferences (Non-refereed)

Published library and relating data

Other (patents, press releases, books and so on)

- H. Shiba and T. Shimokawabe, 'Graph Neural Network Prediction of Long-Time Molecular Dynamics and its Benchmarks', IPSJ SIG Technical Report, 2022.