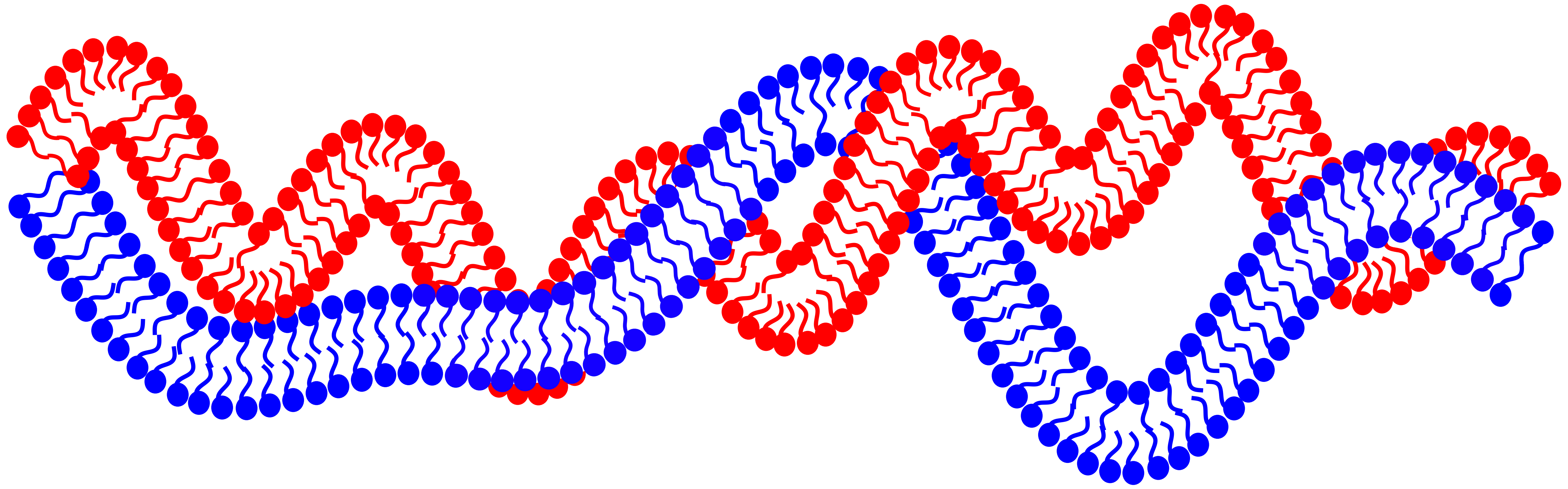
Scientific Machine Learning for Modeling and Simulating Complex Fluids



Kyle R. Lennon

HATSOPOULOS MICROFLUIDS LABORATORY

MIT ChemE

DOE CSGF

Complex fluids balance liquid-like and solid-like behavior

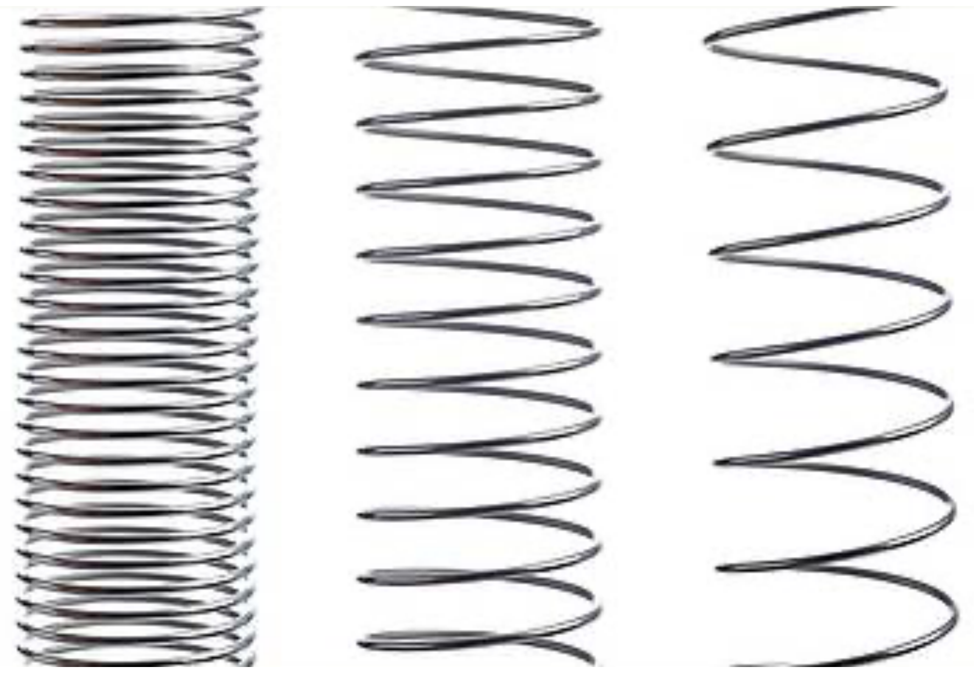
Liquid-like (viscous)

Viscoelastic

Solid-like (elastic)

Rheologists seek to understand the relationship between **stress** and **deformation** in soft materials

This relationship is **nonlinear**

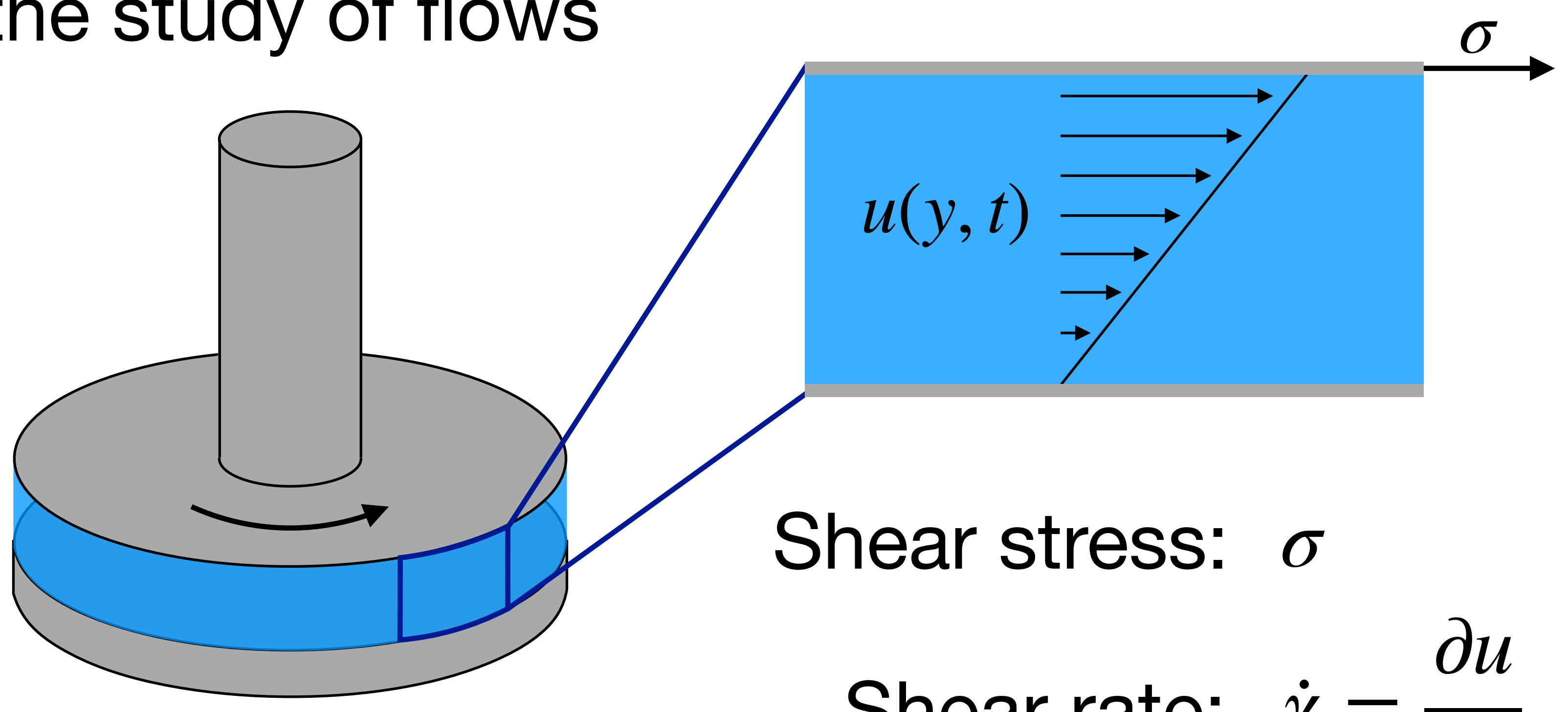


and depends on the **history** of the deformation

Rheologists measure deformations in many ways

Rheology:
the study of flows

Simple shear

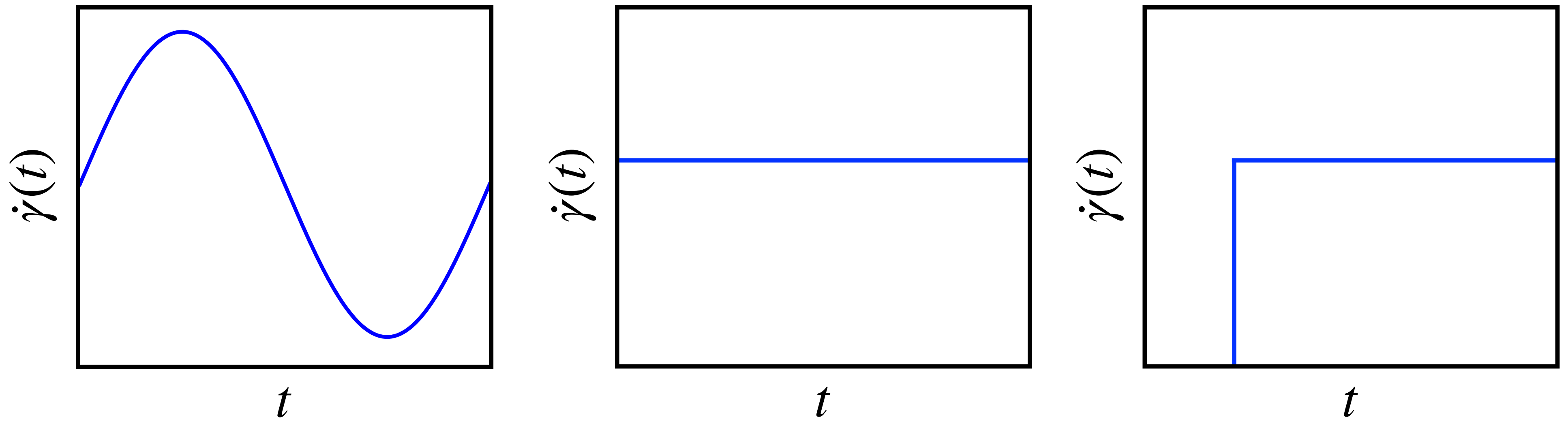


Shear stress: σ

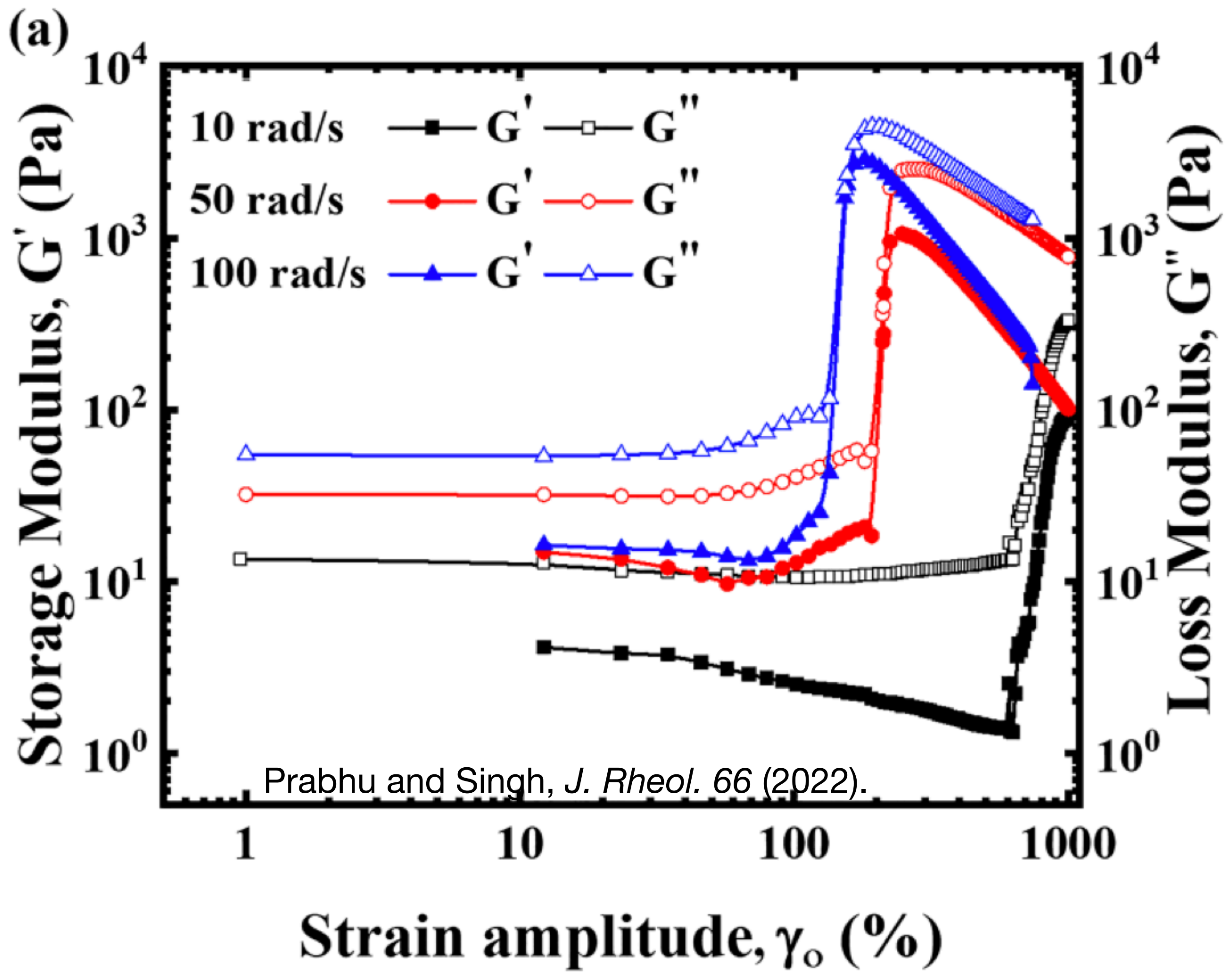
Shear rate: $\dot{\gamma} = \dfrac{\partial u}{\partial y}$

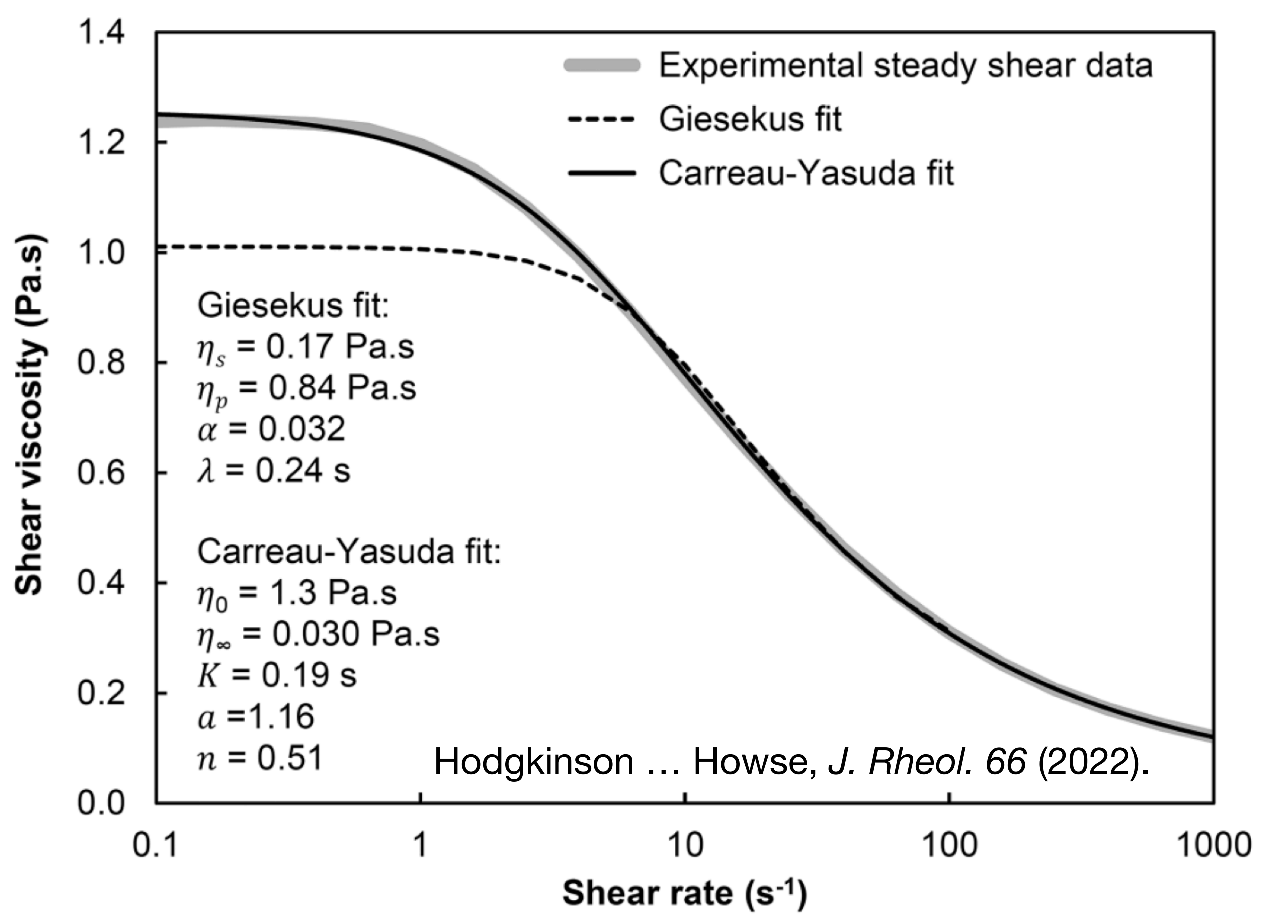
Purely viscous fluids: $\sigma = \eta(\dot{\gamma})\dot{\gamma}$

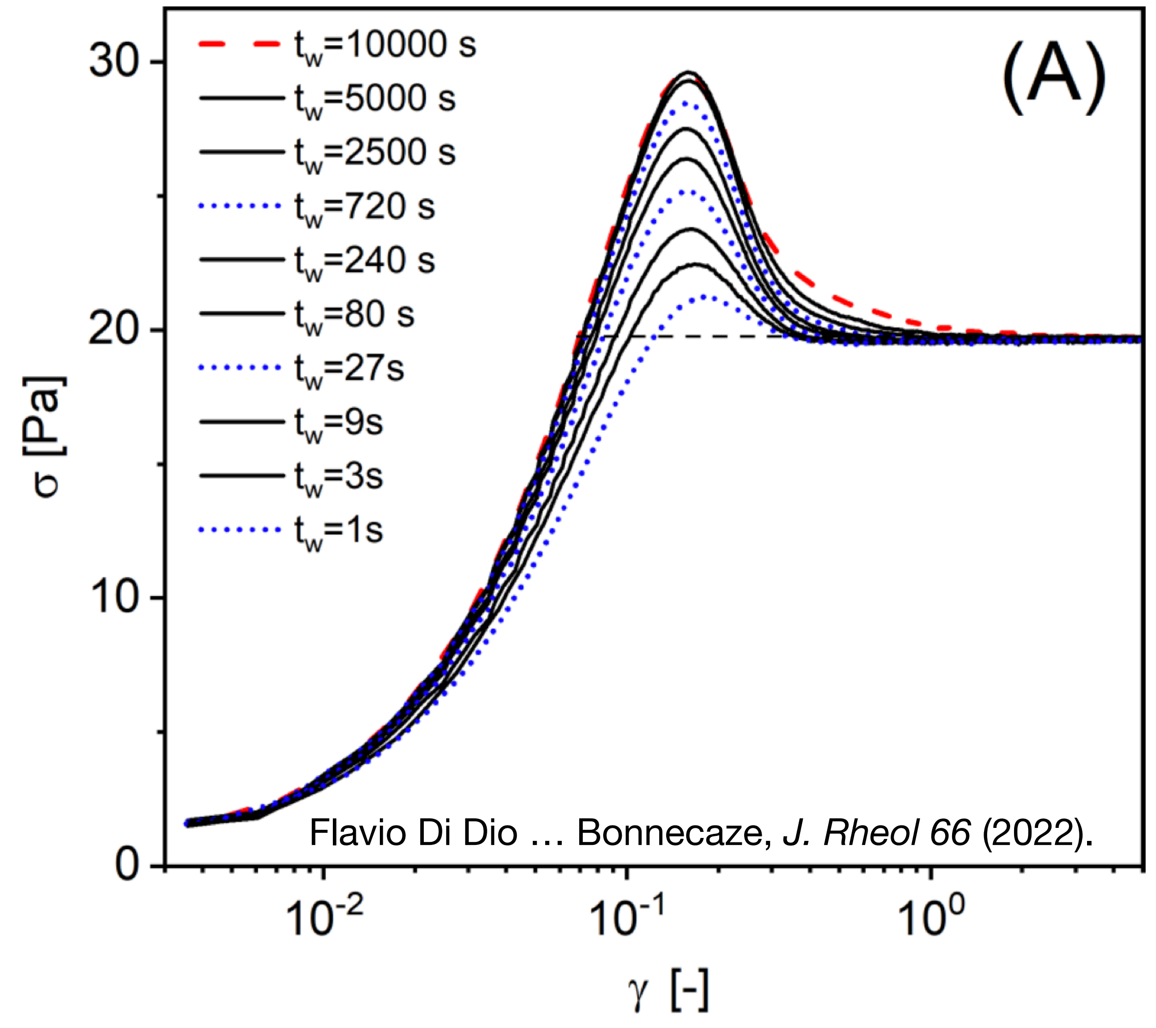
But viscoelastic fluids have **memory**, and exhibit distinct behavior for different **deformation histories**



Data coming from different experiments often have very different structures



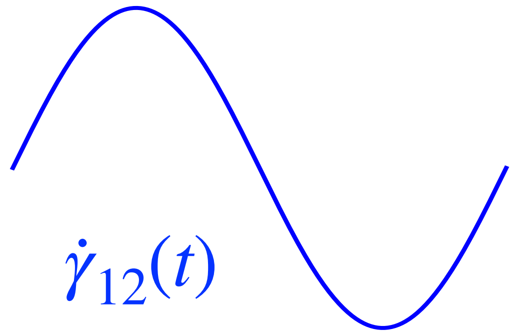




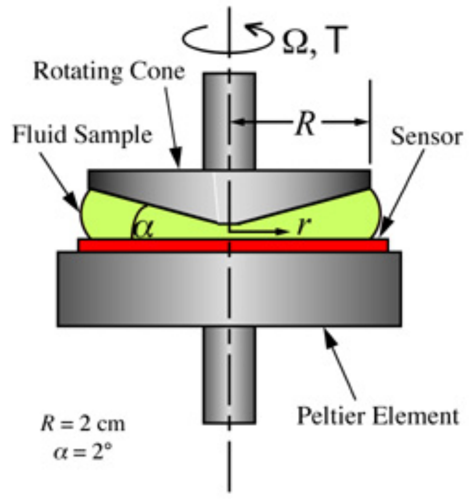
Rheological constitutive equations help us make sense of diverse data

Constitutive equations for viscoelastic fluids should mimic the material and be independent of details regarding experimental instrumentation and measurement

Constitutive equations for complex fluids relate deformations to stresses



→

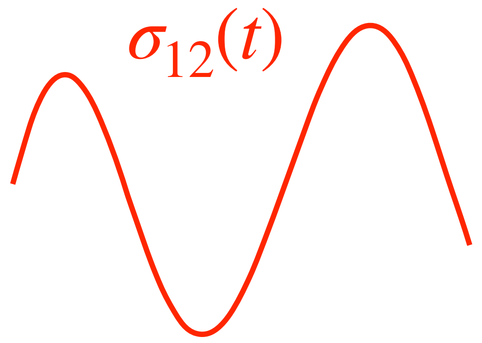


→

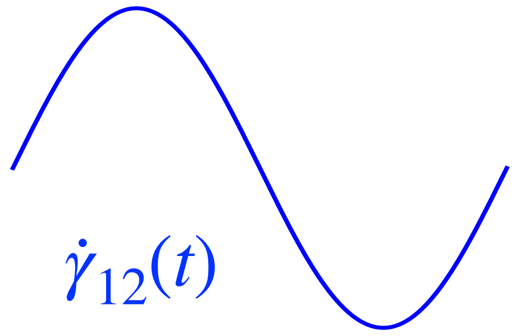
→

Transducer

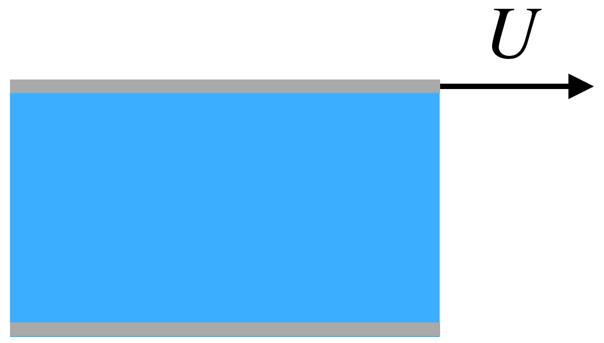
→



In a simulation of a complex fluid, the constitutive equation imitates the fluid, not the instruments



→



Boundary conditions or forcing functions

→

$\boldsymbol{\sigma}(t) = \boldsymbol{F}[\dot{\boldsymbol{\gamma}}(t)]$

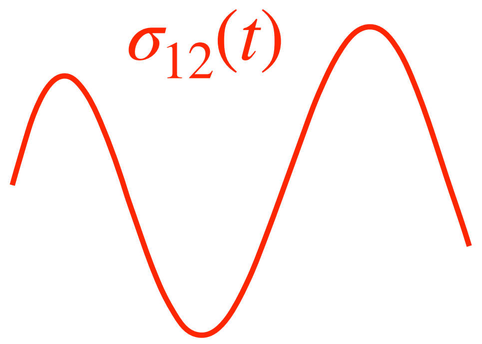
Constitutive equation

→

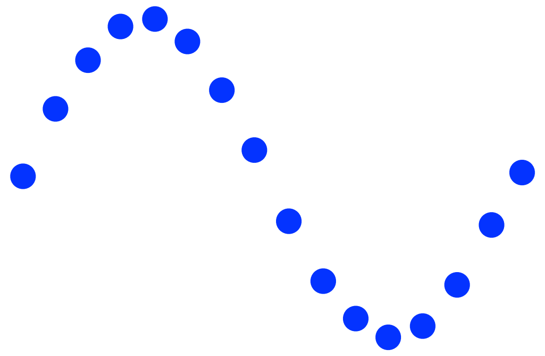
$\sigma_{12}(t) = \boldsymbol{e}_1 \cdot \boldsymbol{\sigma} \cdot \boldsymbol{e}_2$

Post-processing

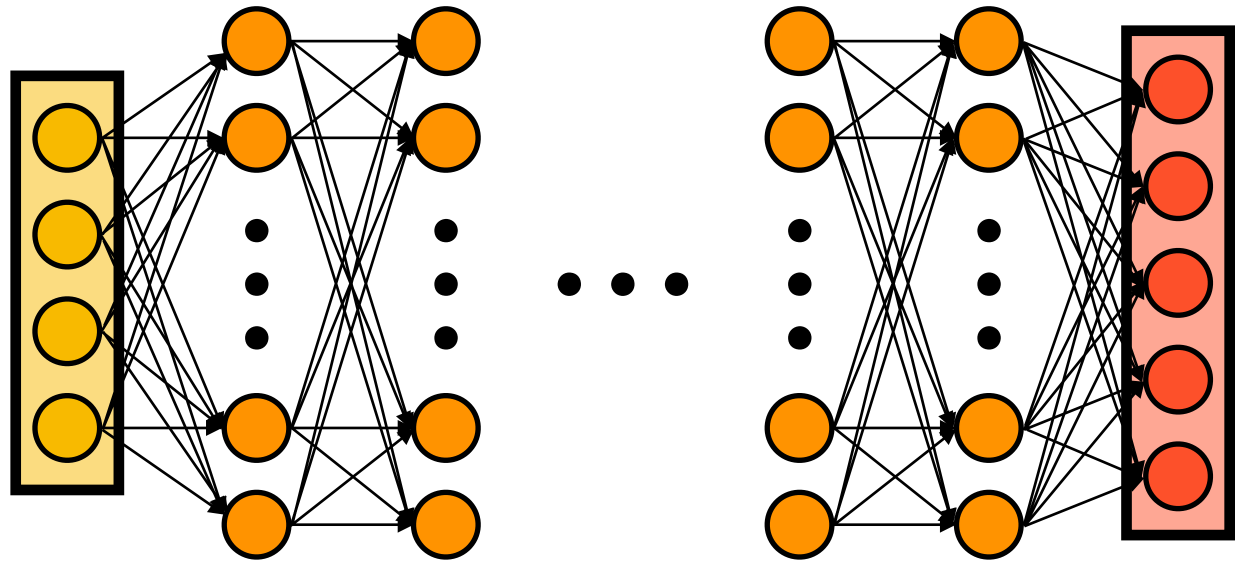
→



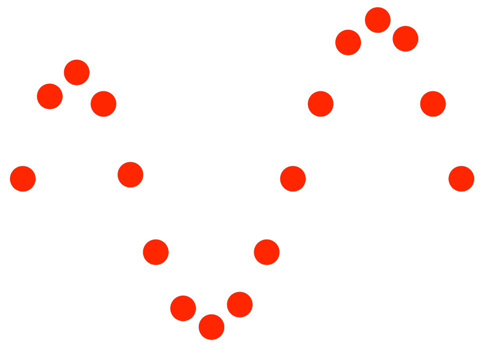
Current machine learning methods are “end-to-end”, mimicking both the instrumentation and fluid



→



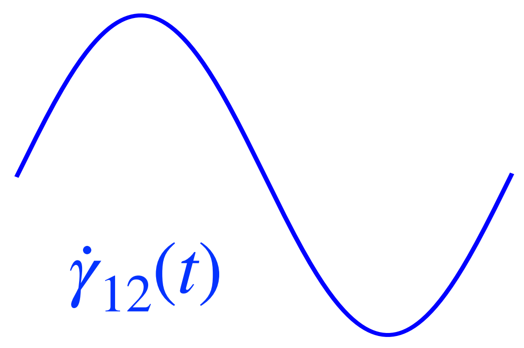
→



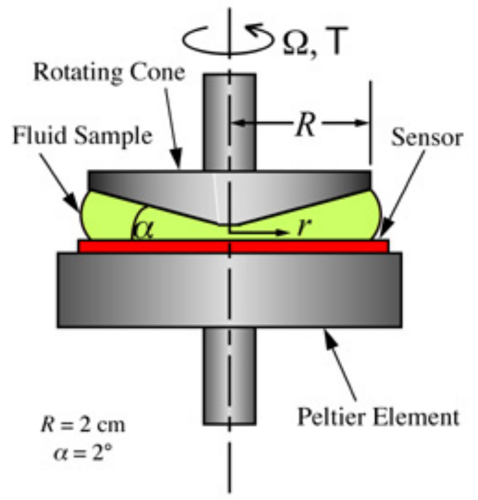
Rheological constitutive equations help us make sense of diverse data

Constitutive equations for viscoelastic fluids should mimic the material and be independent of details regarding experimental instrumentation and measurement

Constitutive equations for complex fluids relate deformations to stresses



→

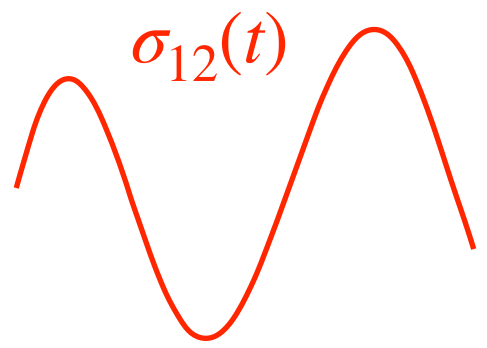


→

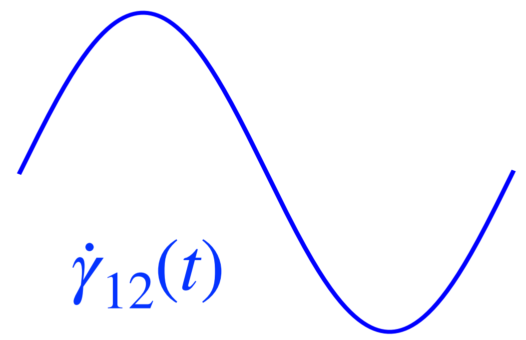
→

Transducer

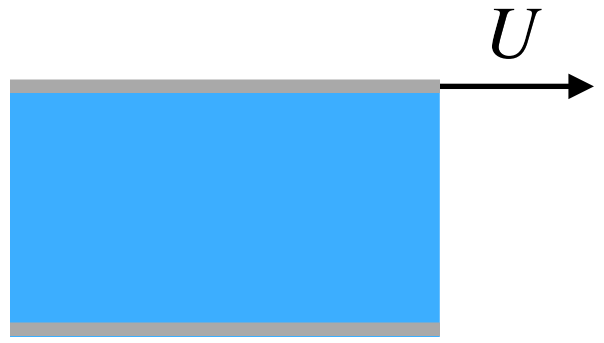
→



In a simulation of a complex fluid, the constitutive equation imitates the fluid, not the instruments



→



Boundary conditions or forcing functions

→

$\boldsymbol{\sigma}(t) = \boldsymbol{F}[\dot{\boldsymbol{\gamma}}(t)]$

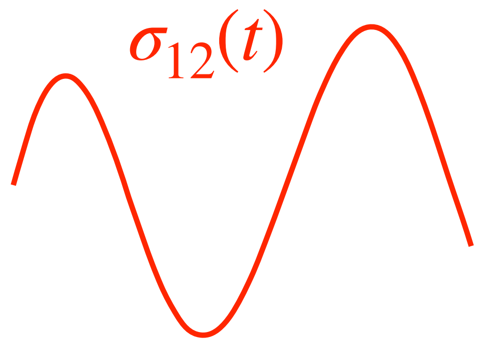
Constitutive equation

→

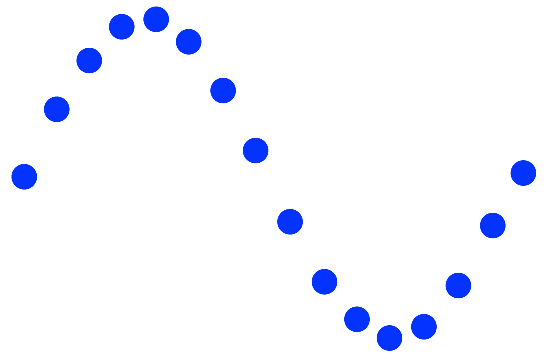
$\sigma_{12}(t) = \boldsymbol{e}_1 \cdot \boldsymbol{\sigma} \cdot \boldsymbol{e}_2$

Post-processing

→



Current machine learning methods are “end-to-end”, mimicking both the instrumentation and fluid

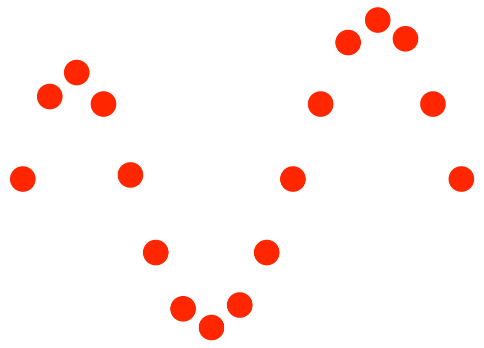


These “end-to-end” approaches …

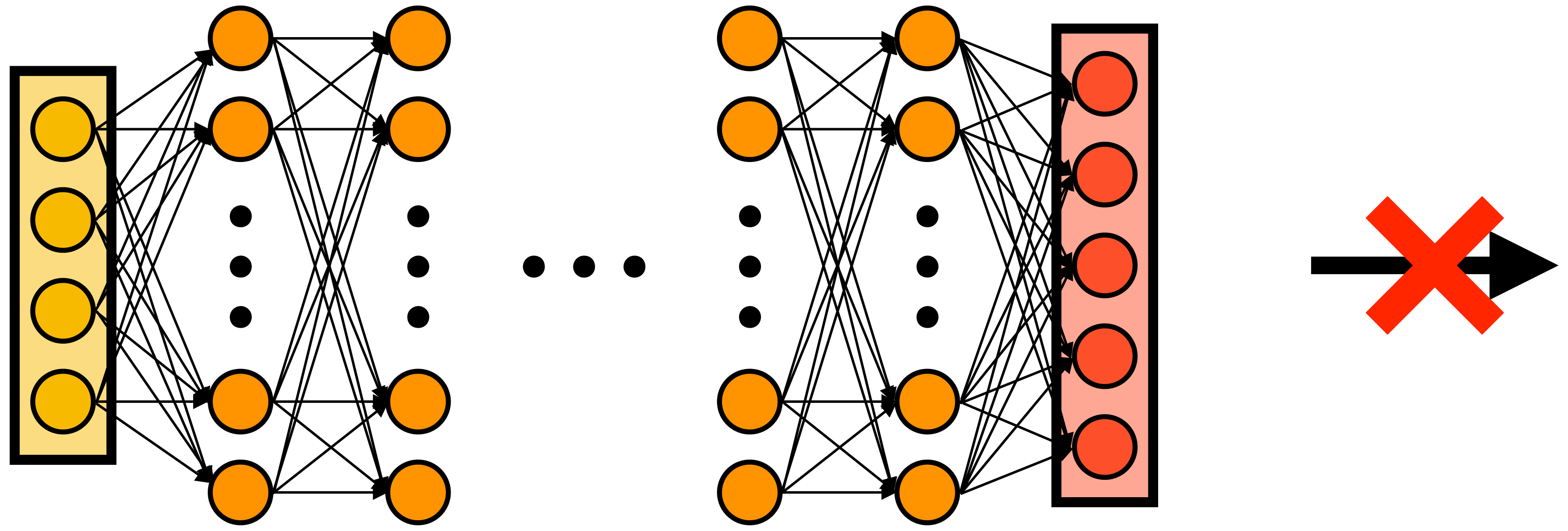
… are fixed to a specific discretization

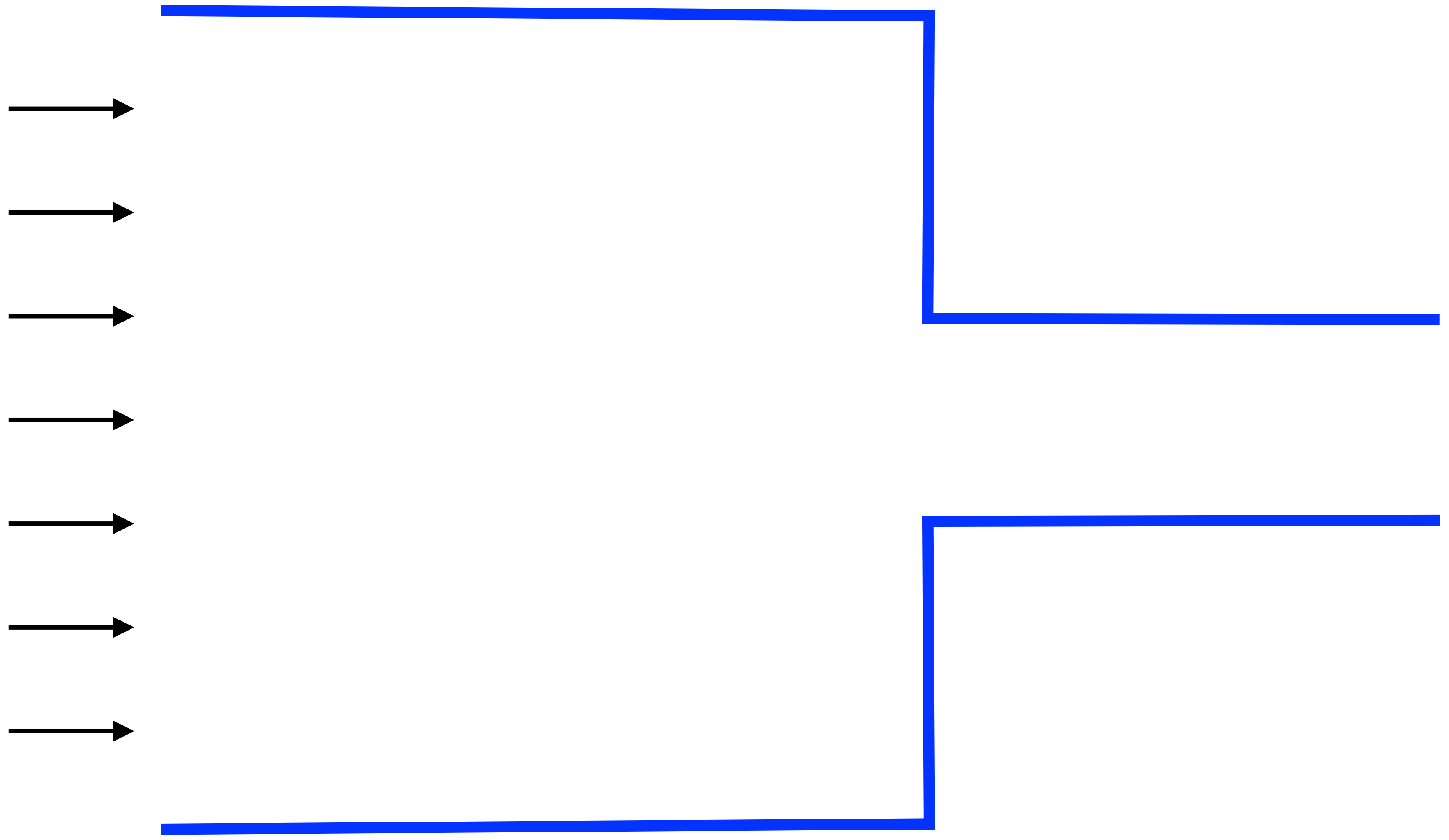
… are one-dimensional (fixed to a specific input/observable)

… can’t be used to simulate different flows



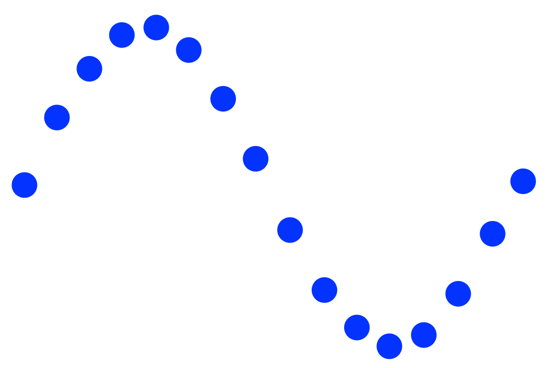
Rheological constitutive equations help us make sense of diverse data





$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f}$$

Current machine learning methods are "end-to-end", mimicking both the instrumentation and fluid

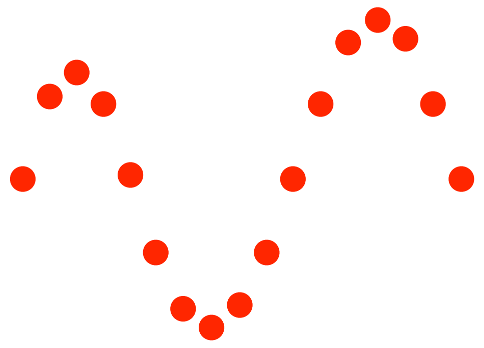


These "end-to-end" approaches …

… are fixed to a specific discretization

… are one-dimensional (fixed to a specific input/observable)

… can't be used to simulate different flows



Rheological constitutive equations help us make sense of diverse data

Goal of this talk: Design machine leaning constitutive equations that are:

- continuous-time,
- three-dimensional,
- admissible (i.e. frame-invariant),
- and trainable via laboratory-accessible data

Current machine learning methods are “end-to-end”, mimicking both the instrumentation and fluid

These “end-to-end” approaches …

… are fixed to a specific discretization

… are one-dimensional (fixed to a specific input/observable)

… can’t be used to simulate different flows

Constructing a machine learning constitutive equation from micromechanics

$$\sigma = \eta_0 \dot{\gamma}_v$$

$$\sigma = G_0 \gamma_e \longrightarrow \frac{d\sigma}{dt} = G_0 \dot{\gamma}_e$$

$$\dot{\gamma} = \dot{\gamma}_v + \dot{\gamma}_e \longrightarrow \tau \frac{d\sigma}{dt} + \sigma = \eta_0 \dot{\gamma}$$

A “learnable” Maxwell model: $\tau \frac{d\sigma}{dt} + \sigma + F(\sigma, \dot{\gamma}) = \eta_0 \dot{\gamma}$ $F =$ neural network

This is a **continuous-time** nonlinear viscoelastic model, but it is **one-dimensional**

$$\tau \frac{d\boldsymbol{\sigma}}{dt} + \boldsymbol{\sigma} + \boldsymbol{F}(\boldsymbol{\sigma}, \dot{\boldsymbol{\gamma}}) = \eta_0 \dot{\boldsymbol{\gamma}} \qquad \boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{pmatrix} \qquad \dot{\boldsymbol{\gamma}} = \begin{pmatrix} \dot{\gamma}_{11} & \dot{\gamma}_{12} & \dot{\gamma}_{13} \\ \dot{\gamma}_{12} & \dot{\gamma}_{22} & \dot{\gamma}_{23} \\ \dot{\gamma}_{13} & \dot{\gamma}_{23} & \dot{\gamma}_{33} \end{pmatrix} = \nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T$$

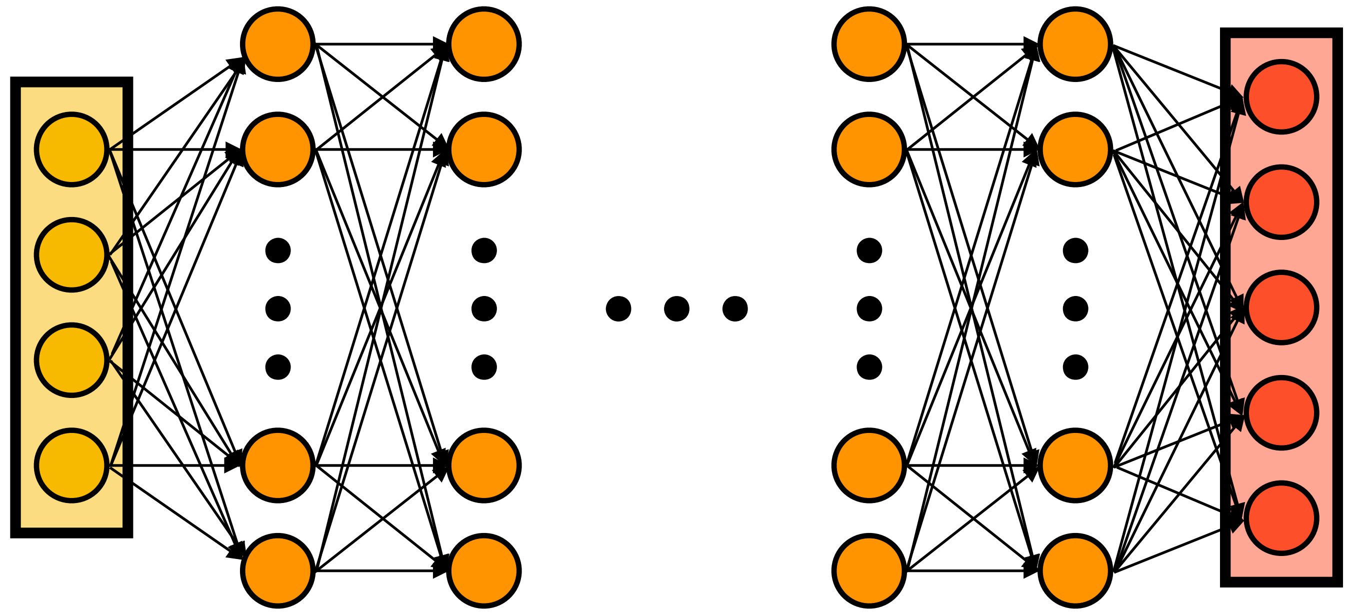
This is a **three-dimensional (tensorial)** model, but it is not **frame-invariant**

1. The stress derivative must be frame-invariant

$$\boldsymbol{Q} \cdot \frac{d\boldsymbol{\sigma}}{dt} \cdot \boldsymbol{Q}^T \neq \frac{d}{dt}\left(\boldsymbol{Q} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{Q}^T\right) \quad \text{if} \quad \boldsymbol{Q} = \boldsymbol{Q}(t)$$

$$\frac{d\boldsymbol{\sigma}}{dt} \rightarrow \overset{\nabla}{\boldsymbol{\sigma}} = \frac{D\boldsymbol{\sigma}}{Dt} + \mathbf{v} \cdot \nabla \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \nabla \mathbf{v} - (\nabla \mathbf{v})^T \cdot \boldsymbol{\sigma}$$

2. The neural network must be frame-invariant



Oldroyd, *Proc. Royal Soc. A, 200* (1950)

Embedding frame invariance within a neural network

The Theory of Matrix Polynomials and its Application to the Mechanics of Isotropic Continua

A. J. M. Spencer & R. S. Rivlin (1958)

$\boldsymbol{F}(\boldsymbol{\sigma}, \dot{\boldsymbol{\gamma}})$ has an expansion in tensor products $\boldsymbol{T}_n$ of $\dot{\boldsymbol{\gamma}}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\delta}$, whose coefficients are arbitrary functions of the invariants of $\boldsymbol{T}_n$ (denoted as λ_n)

$$\boldsymbol{F} = g_1\boldsymbol{\delta} + g_2\dot{\boldsymbol{\gamma}} + g_3\boldsymbol{\sigma} + g_4(\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}) + g_5(\dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}) + g_6(\boldsymbol{\sigma}\cdot\dot{\boldsymbol{\gamma}}) + \ldots$$

$$\boldsymbol{T}_1 = \boldsymbol{\delta} \qquad \boldsymbol{T}_2 = \dot{\boldsymbol{\gamma}} \qquad \boldsymbol{T}_3 = \boldsymbol{\sigma} \qquad \boldsymbol{T}_4 = \boldsymbol{\sigma}\cdot\boldsymbol{\sigma} \qquad \boldsymbol{T}_5 = \dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}} \qquad \boldsymbol{T}_6 = \boldsymbol{\sigma}\cdot\dot{\boldsymbol{\gamma}} \qquad g_n = f_n(\lambda_1, \lambda_2, \ldots) \qquad \lambda_n = \mathrm{tr}(\boldsymbol{T}_n)$$

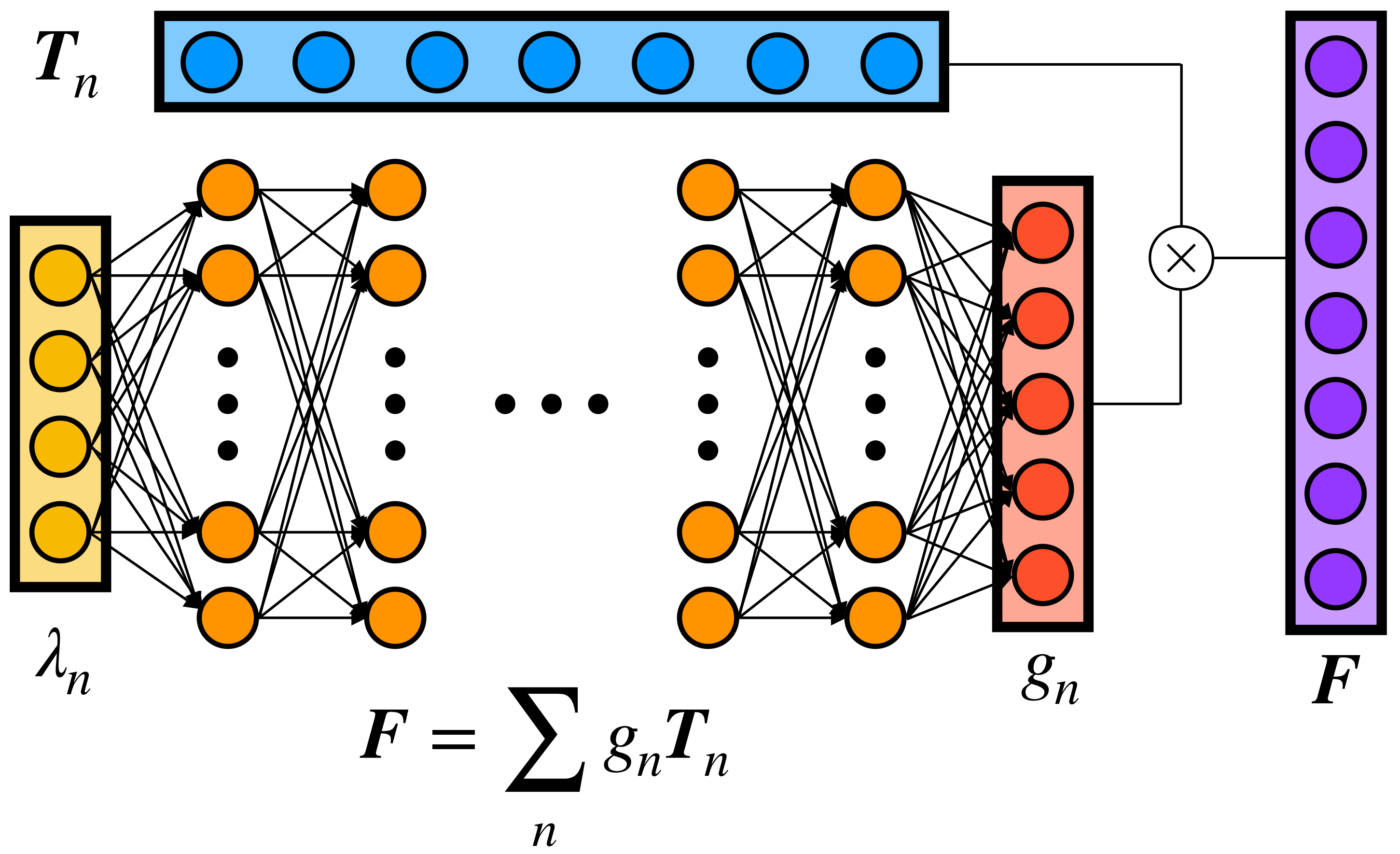
There are **only nine independent** T_n (Cayley-Hamilton)

$$\boldsymbol{T}_n = \begin{cases} \boldsymbol{\delta}\ ,\ \dot{\boldsymbol{\gamma}}\ ,\ \boldsymbol{\sigma}\ ,\ \boldsymbol{\sigma}\cdot\boldsymbol{\sigma}\ ,\ \dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}\ ,\ \dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma} \\ \dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}\ ,\ \dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}\ ,\ \dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}\cdot\boldsymbol{\sigma} \end{cases}$$

There are **only nine independent** λ_n

$$\lambda = \begin{cases} \mathrm{tr}(\boldsymbol{\sigma})\ ,\ \mathrm{tr}(\boldsymbol{\sigma}\cdot\boldsymbol{\sigma})\ ,\ \mathrm{tr}(\dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}})\ ,\ \mathrm{tr}(\dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}) \\ \mathrm{tr}(\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}\cdot\boldsymbol{\sigma})\ ,\ \mathrm{tr}(\dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}})\ ,\ \mathrm{tr}(\dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}) \\ \mathrm{tr}(\dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}\cdot\boldsymbol{\sigma})\ ,\ \mathrm{tr}(\dot{\boldsymbol{\gamma}}\cdot\dot{\boldsymbol{\gamma}}\cdot\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}) \end{cases}$$

"Tensor Basis Neural Network" (TBNN)



The "Rheological Universal Differential Equation" (RUDE): a learnable, **frame-invariant** constitutive model

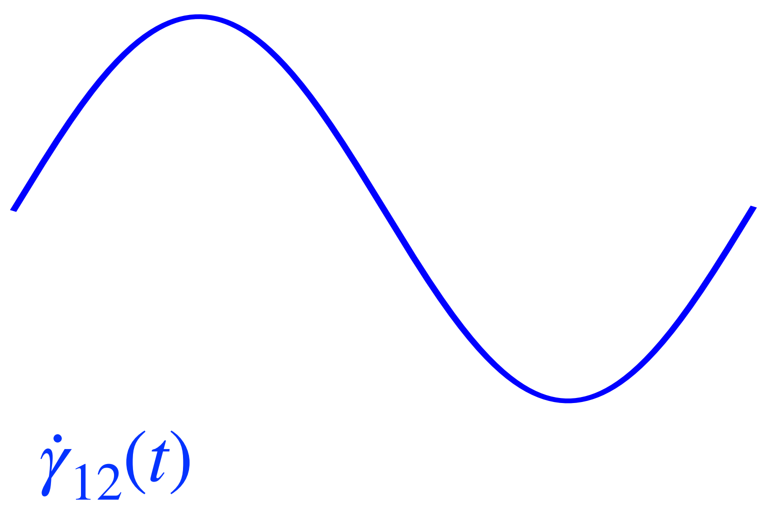
$$\tau\overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \sum_{n=1}^{9} g_n(\lambda_n;\theta)\boldsymbol{T}_n = \eta_0\dot{\boldsymbol{\gamma}}$$

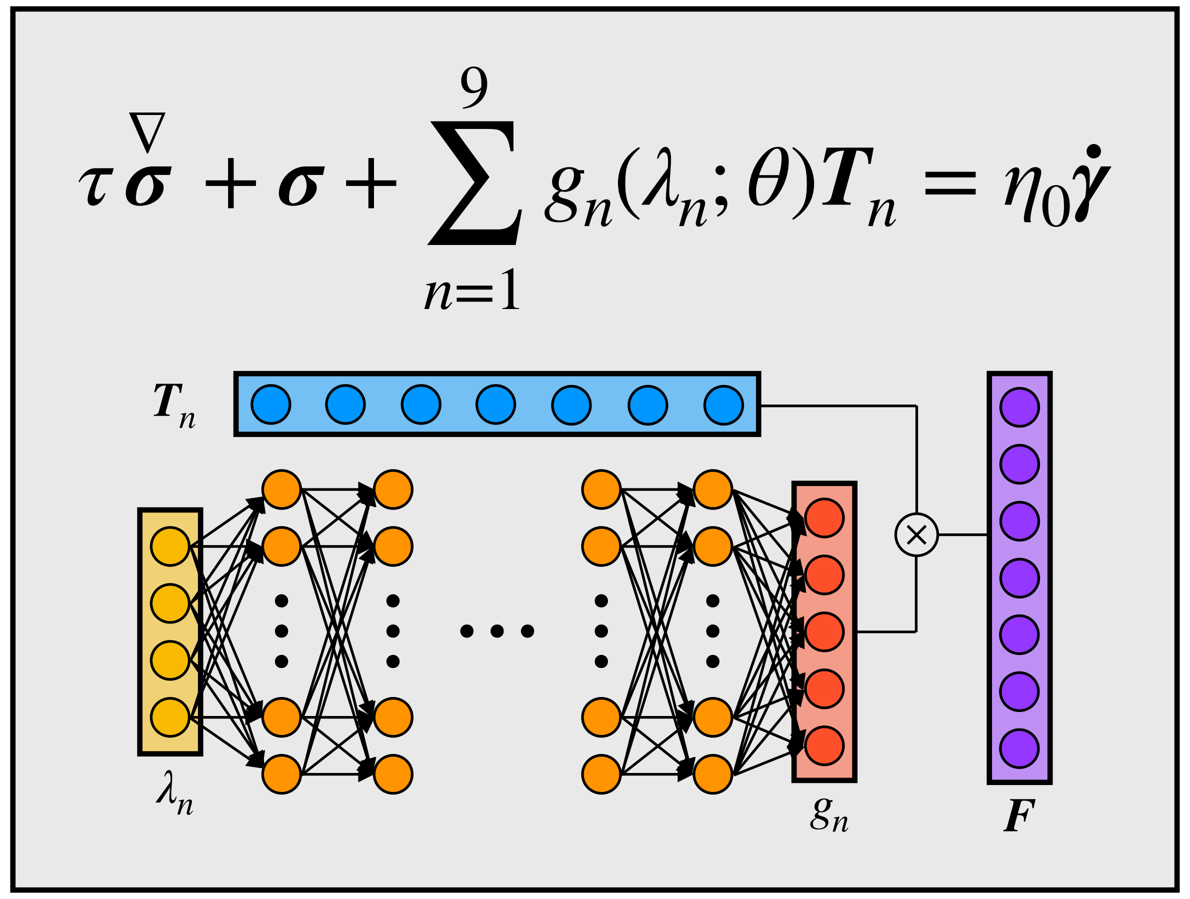
Ling … Templeton, *J. Fluid Mech.*, 807 (2016)

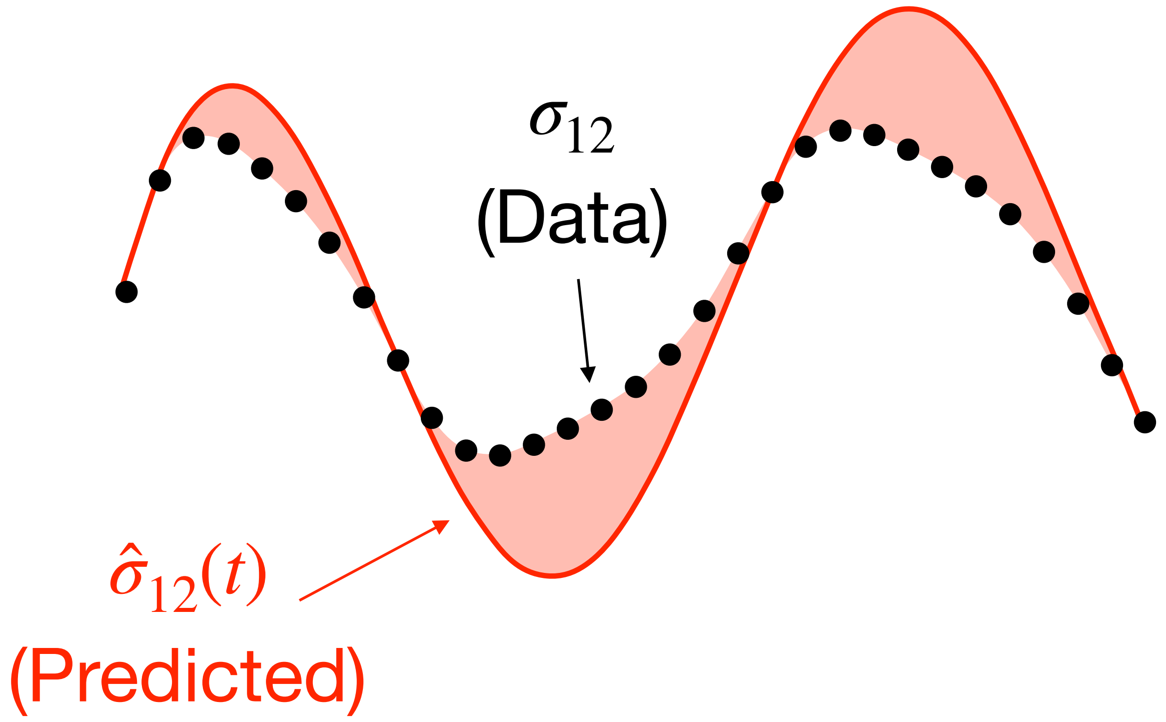
The loss function specializes on the data

$\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\gamma}}$ are symmetric tensors, so there are 12 independent quantities and 6 coupled differential equations

Homogeneous simple shear: $\dot{\gamma}_{ij} = 0$ for $(i, j) \neq (1,2)$, and specify either $\dot{\gamma}_{12}(t)$ or $\sigma_{12}(t)$







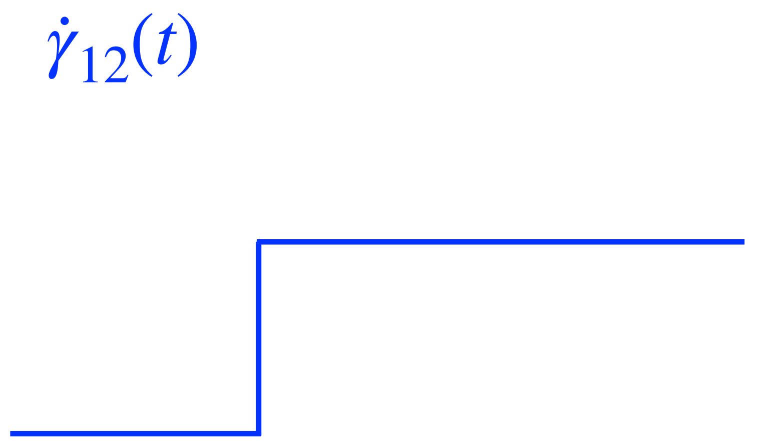
$$\mathscr{L}_{\sigma,LAOS} = \sum_{i=1}^{N} (\hat{\sigma}_{12}(t_i) - \sigma_{12}^{(i)})^2$$

K. Lennon, G.H. McKinley, J. Swan. *PNAS* (2023).

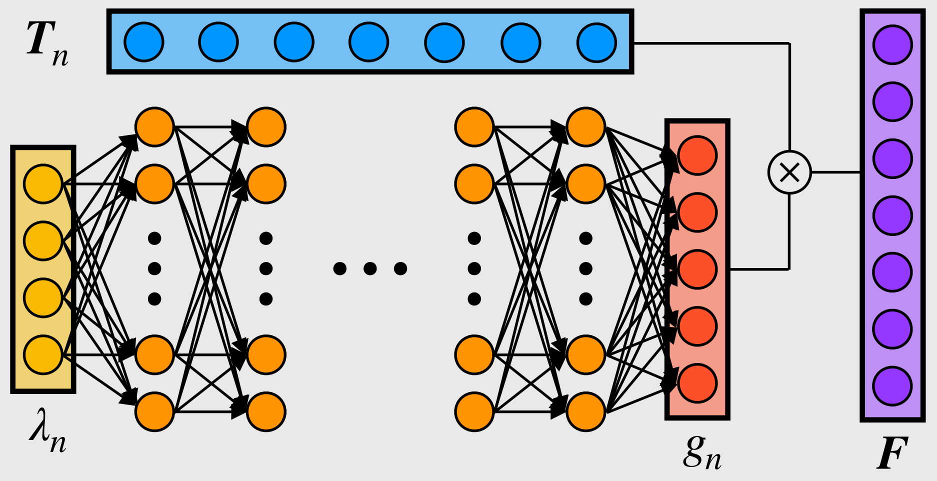
The loss function specializes on the data

$\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\gamma}}$ are symmetric tensors, so there are 12 independent quantities and 6 coupled differential equations

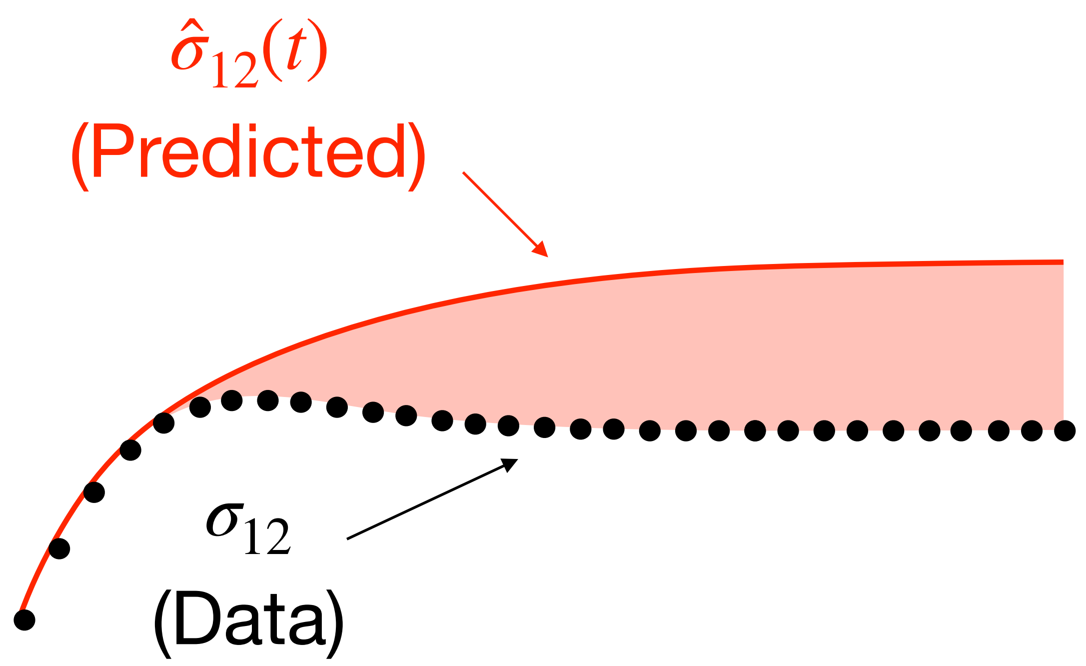
Homogeneous simple shear: $\dot{\gamma}_{ij} = 0$ for $(i, j) \neq (1,2)$, and specify either $\dot{\gamma}_{12}(t)$ or $\sigma_{12}(t)$



$$\tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \sum_{n=1}^{9} g_n(\lambda_n; \theta)\boldsymbol{T}_n = \eta_0 \dot{\boldsymbol{\gamma}}$$



$$\mathscr{L}_{\sigma,jump} = \sum_{i=1}^{N} (\hat{\sigma}_{12}(t_i) - \sigma_{12}^{(i)})^2$$

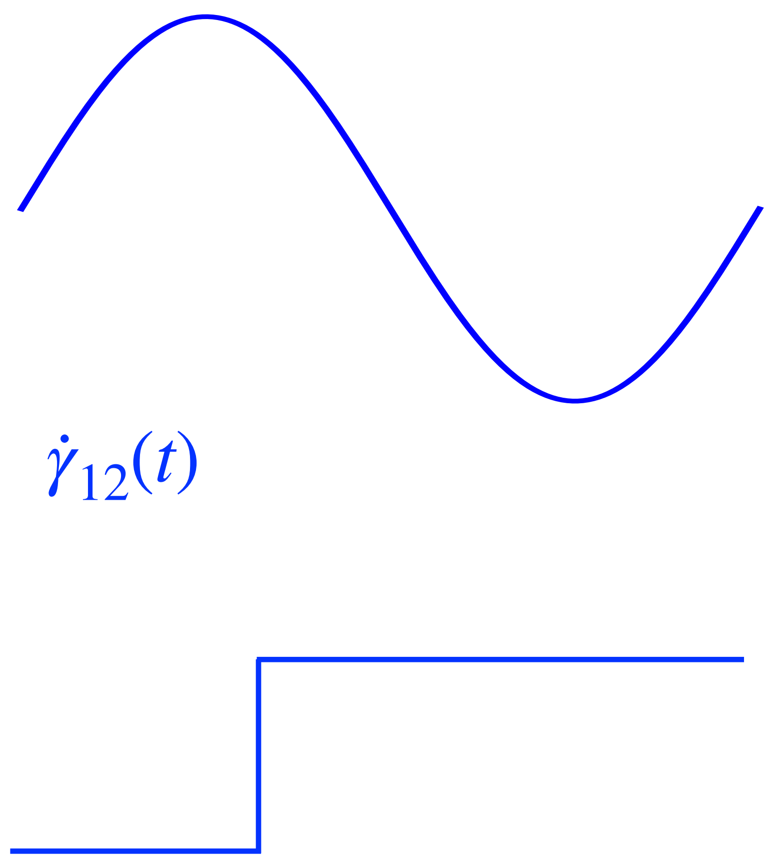


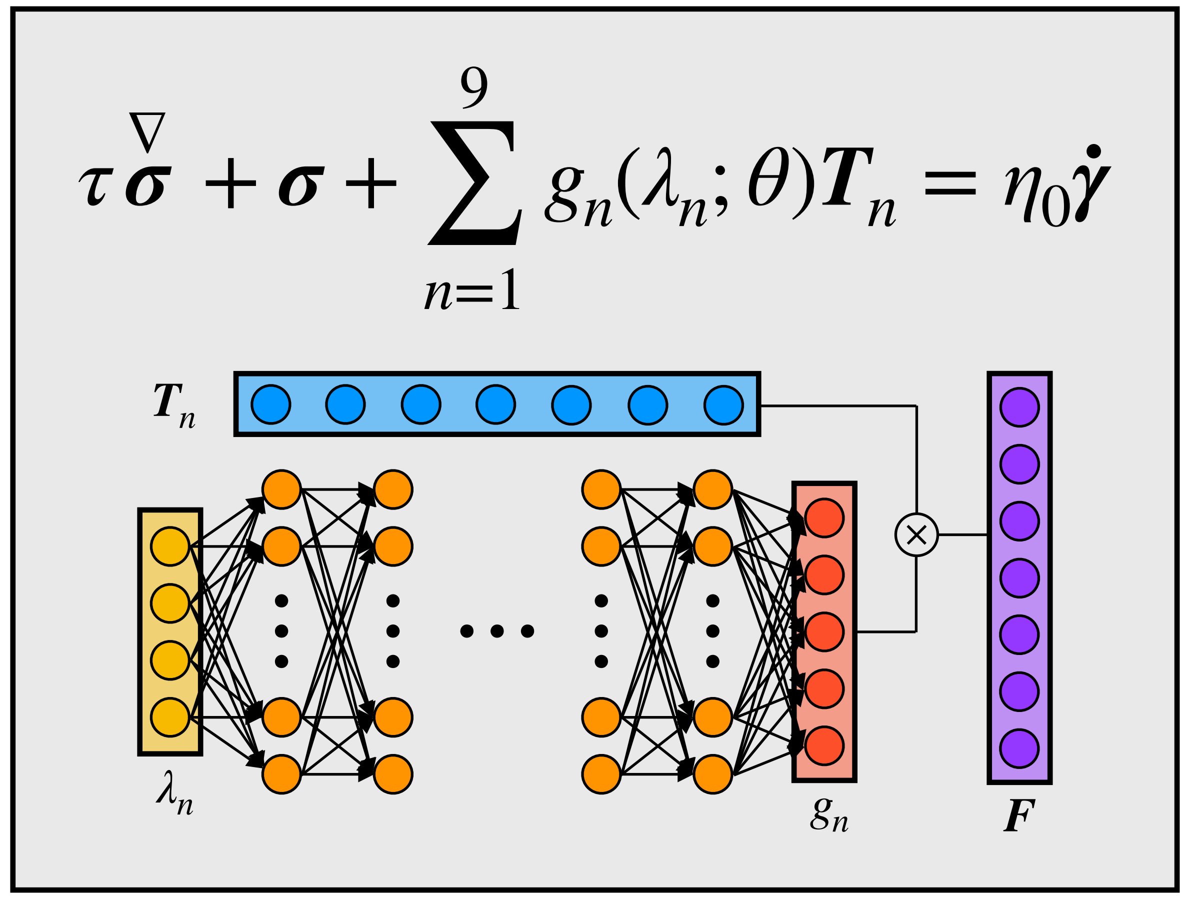
K. Lennon, G.H. McKinley, J. Swan. *PNAS* (2023).

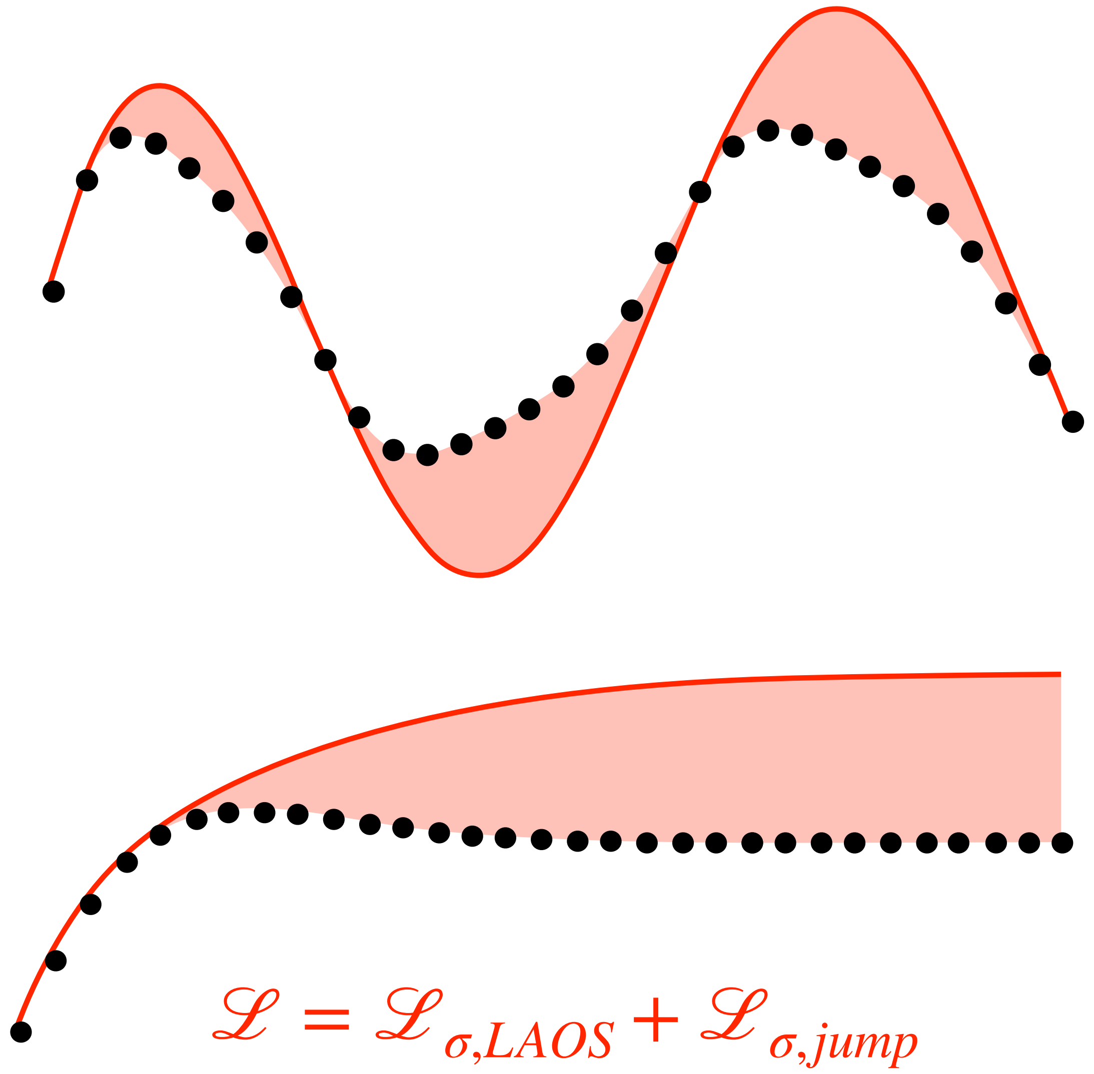
The loss function specializes on the data

$\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\gamma}}$ are symmetric tensors, so there are 12 independent quantities and 6 coupled differential equations

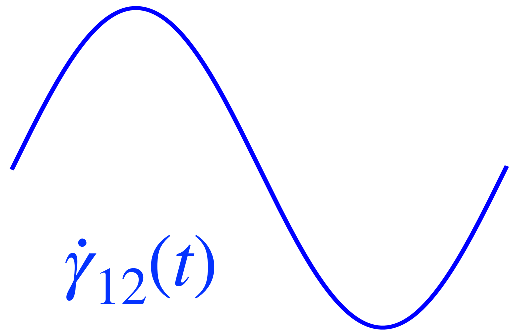
Homogeneous simple shear: $\dot{\gamma}_{ij} = 0$ for $(i,j) \neq (1,2)$, and specify either $\dot{\gamma}_{12}(t)$ or $\sigma_{12}(t)$

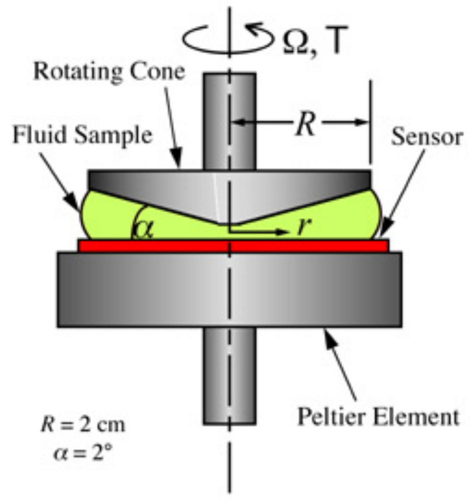




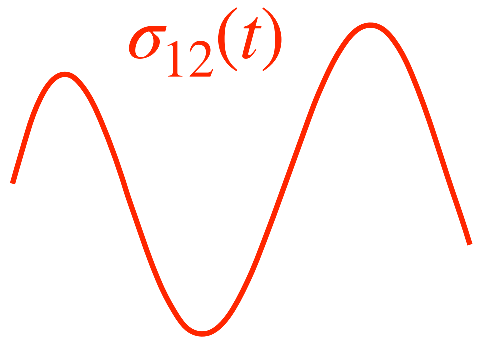


The loss function "measures" the simulated RUDE for comparison against the training data





Transducer



Boundary conditions or forcing function → RUDE → Postprocessing & loss function

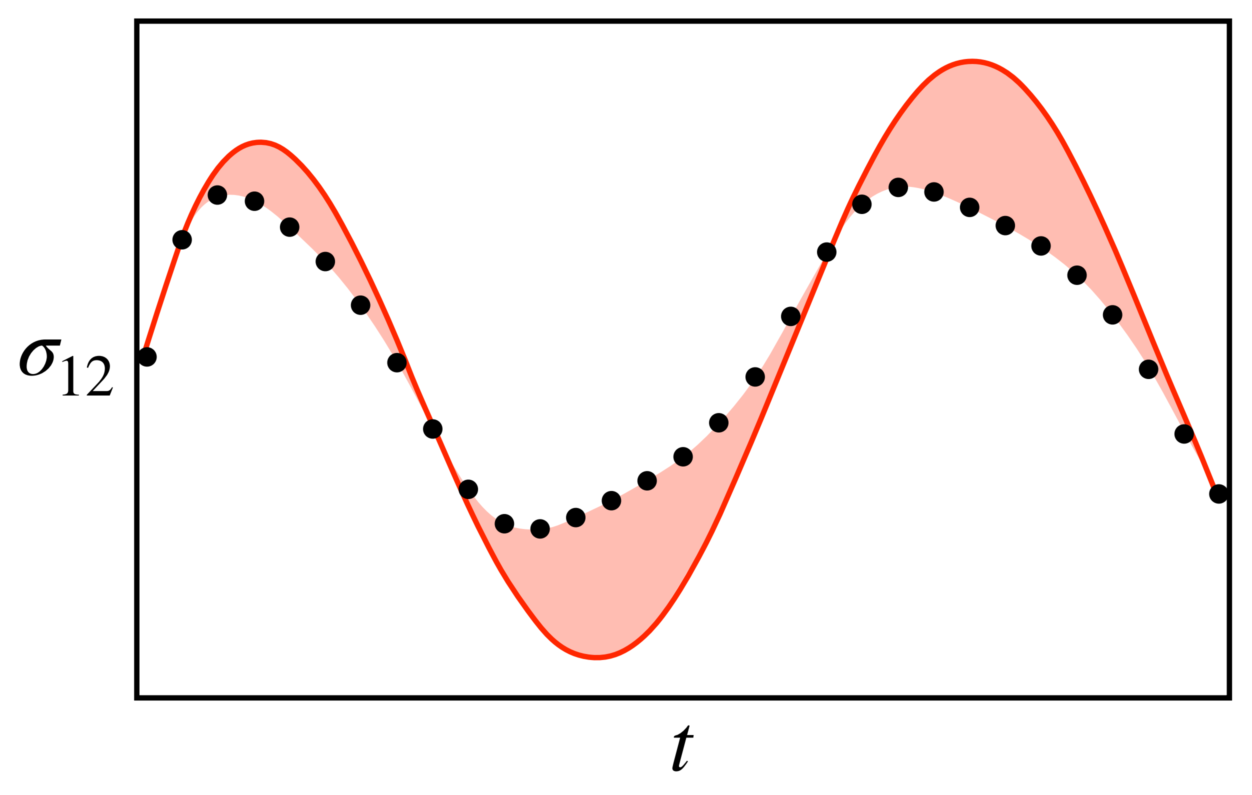
K. Lennon, G.H. McKinley, J. Swan. *PNAS* (2023).

Training a RUDE on synthetic data using Julia

Example: train a RUDE using synthetic data (Giesekus model)

$$\tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \frac{\tau\alpha}{\eta_0}\boldsymbol{\sigma}\cdot\boldsymbol{\sigma} = \eta_0 \dot{\boldsymbol{\gamma}} \qquad \alpha = 0.8$$

Training data: the shear stress (σ_{12}) in eight oscillatory tests



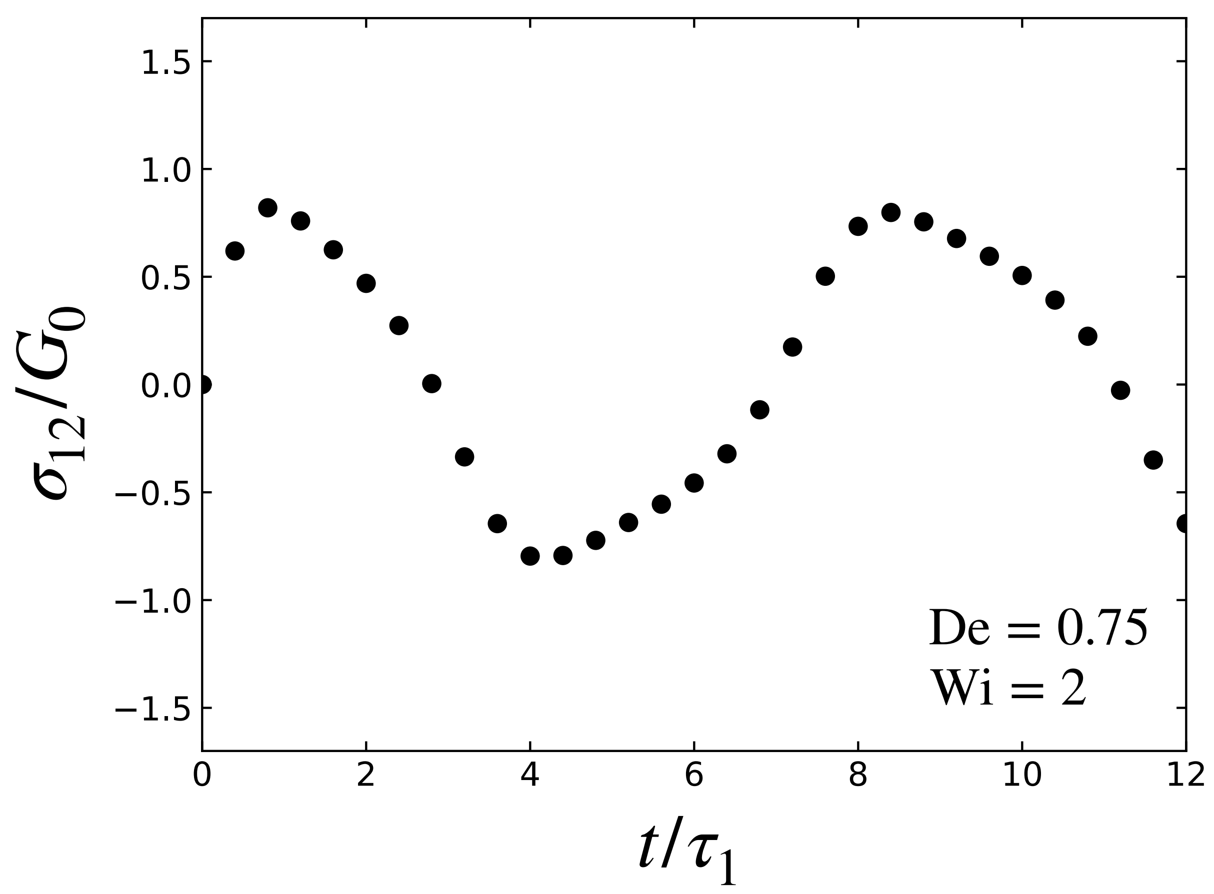
$$\dot{\gamma}(t) = \mathrm{Wi}\sin(\mathrm{De}\cdot t)$$

$$\mathrm{De} \in \{0.33, 0.5, 1, 2\} \qquad \mathrm{Wi} \in \{1, 2\}$$

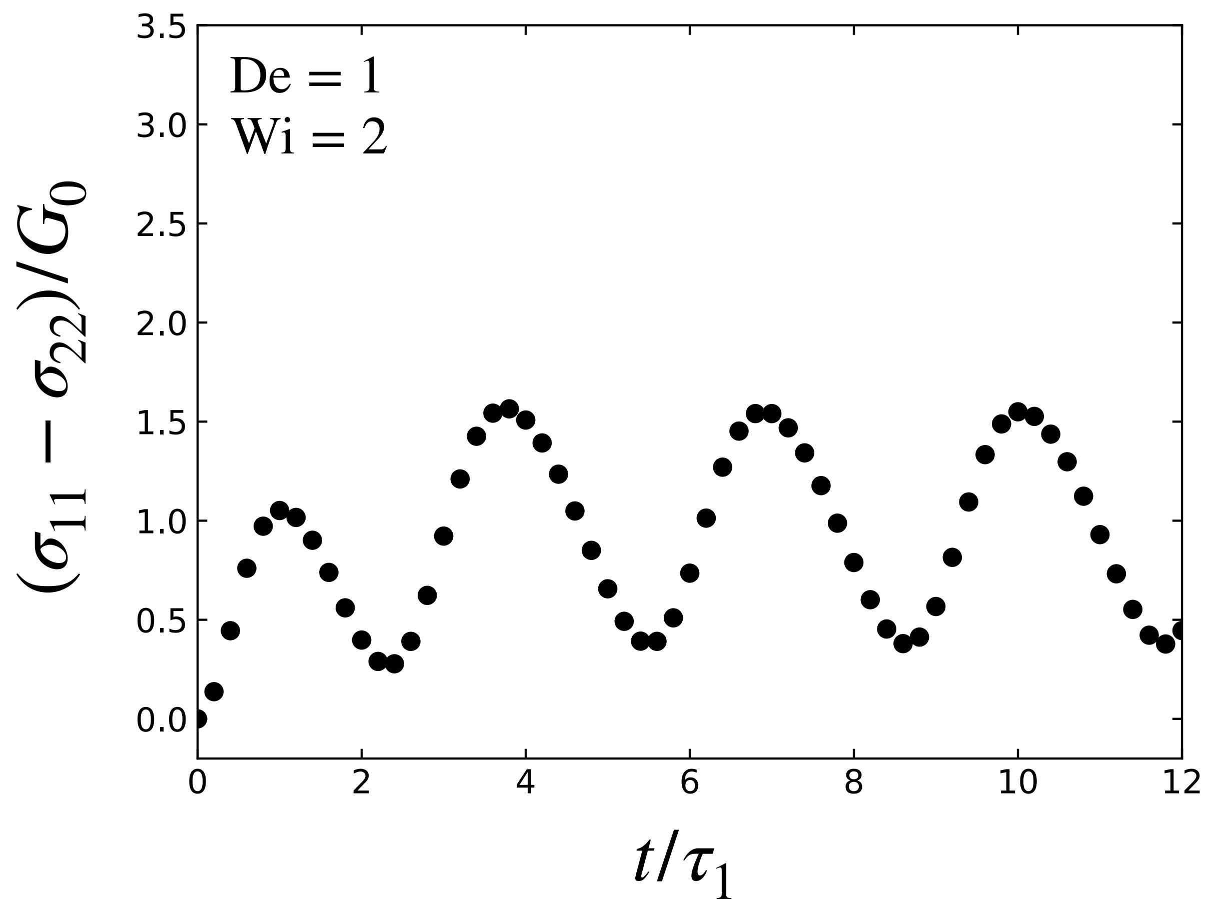
(DifferentialEquations.jl) (Flux.jl)

Test tasks:

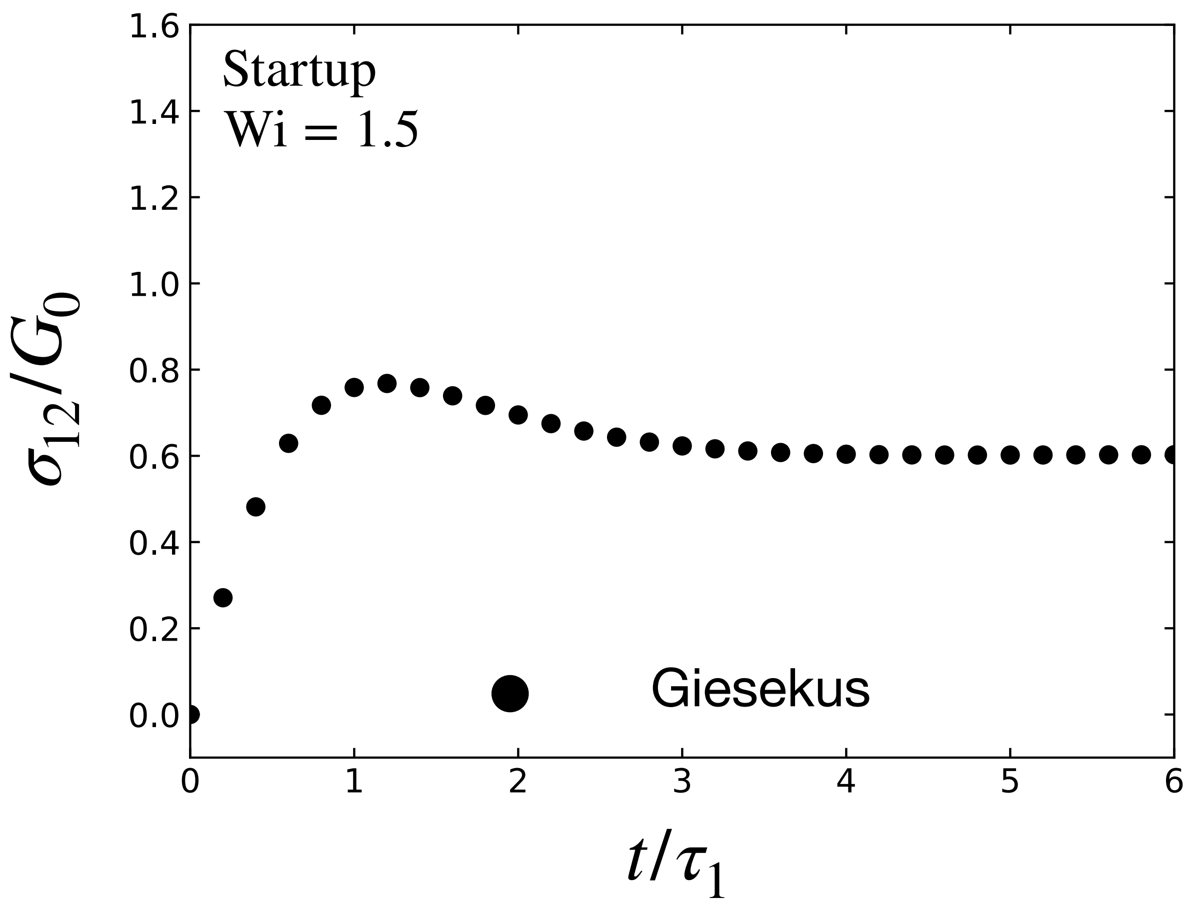
Predict σ_{12} at a new De



Predict normal stresses



Predict startup transient



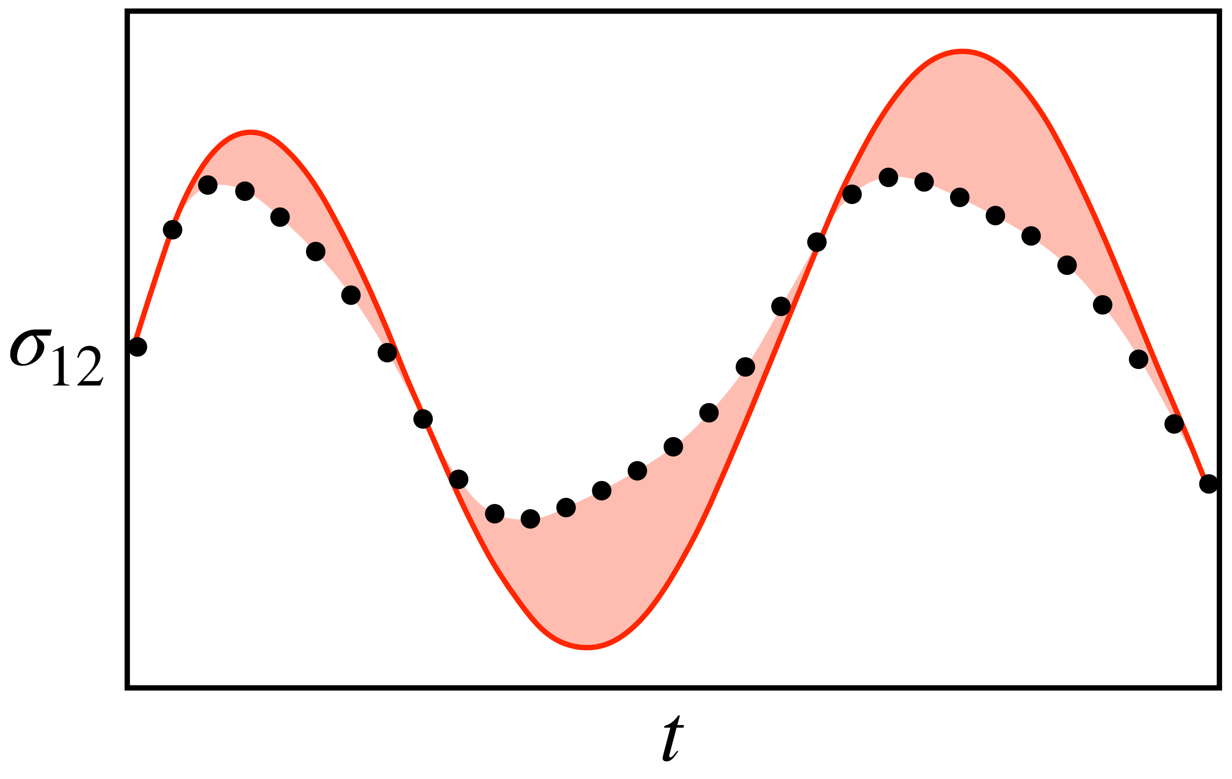
Rackauckas … Edelman, *arXiv:2001.04385* (2020)

Training a RUDE on synthetic data using Julia

Example: train a RUDE using synthetic data (Giesekus model)

$$\tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \frac{\tau\alpha}{\eta_0}\boldsymbol{\sigma}\cdot\boldsymbol{\sigma} = \eta_0 \dot{\boldsymbol{\gamma}} \qquad \alpha = 0.8$$

Training data: the shear stress (σ_{12}) in eight oscillatory tests



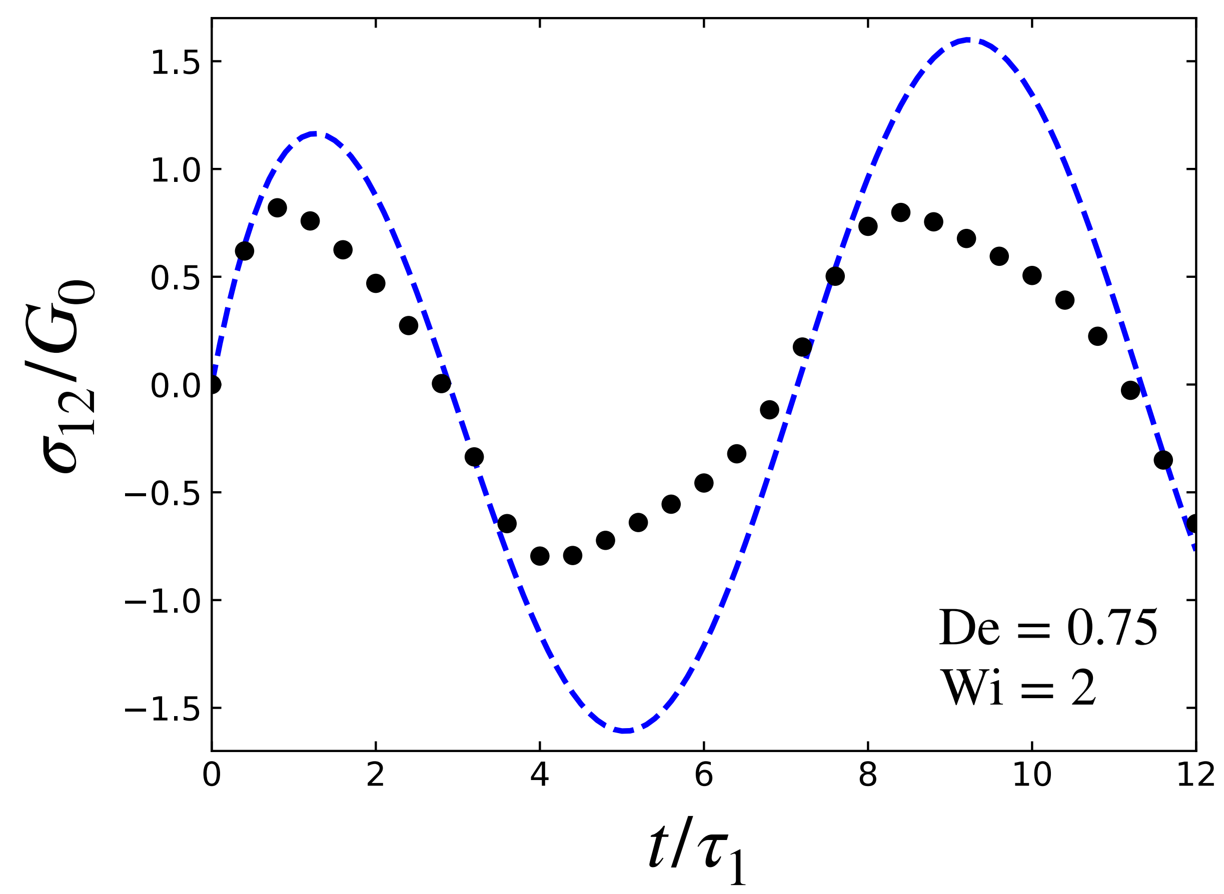
$$\dot{\gamma}(t) = \mathrm{Wi}\sin(\mathrm{De}\cdot t)$$

$$\mathrm{De} \in \{0.33, 0.5, 1, 2\} \qquad \mathrm{Wi} \in \{1, 2\}$$

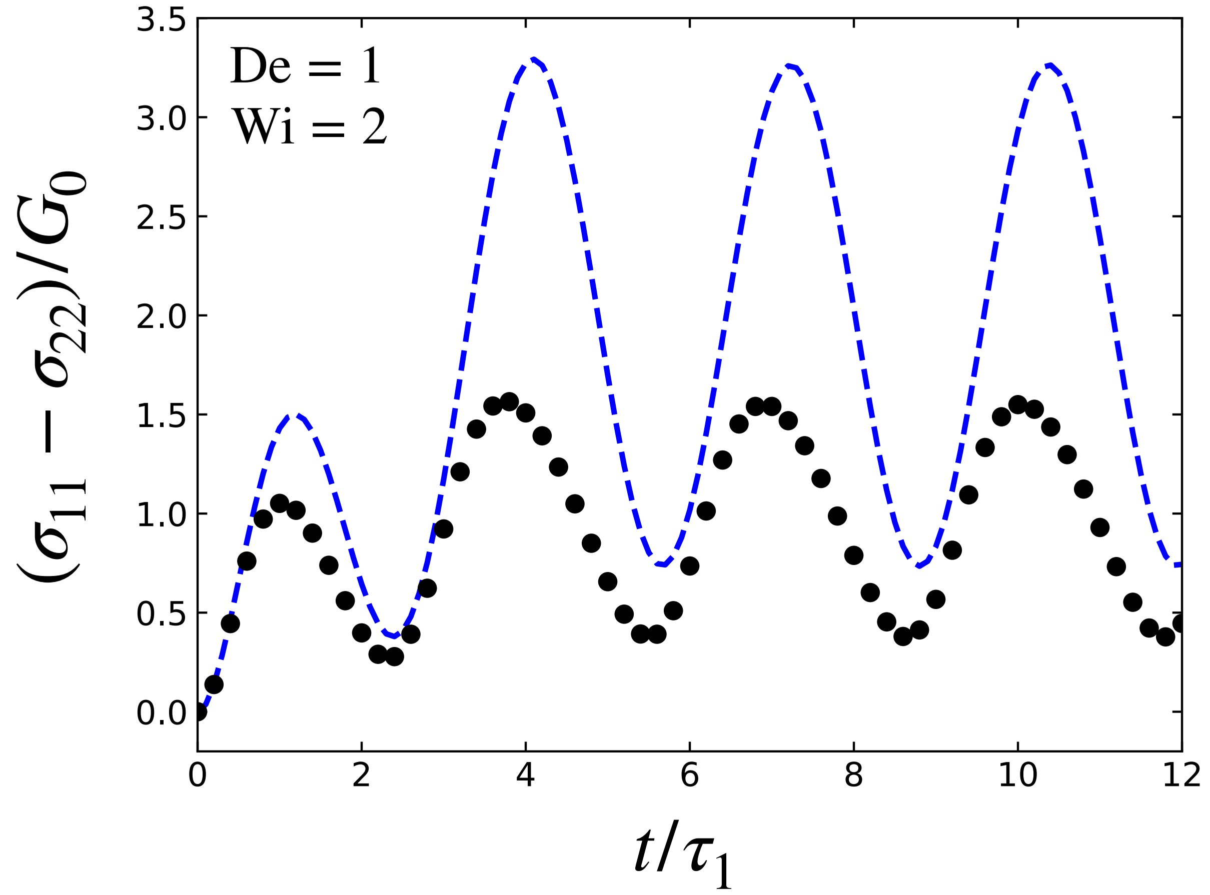
(DifferentialEquations.jl) (Flux.jl)

Test tasks:

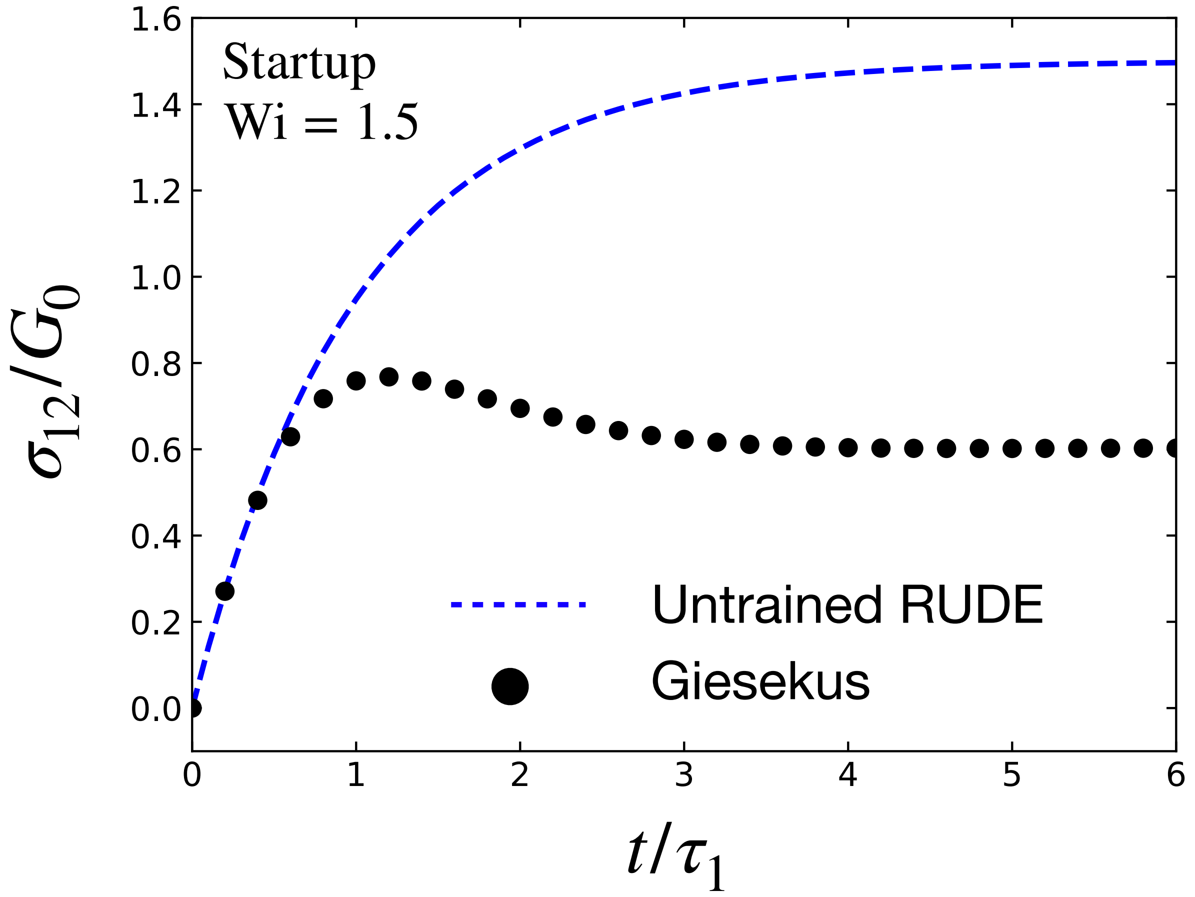
Predict σ_{12} at a new De



Predict normal stresses



Predict startup transient



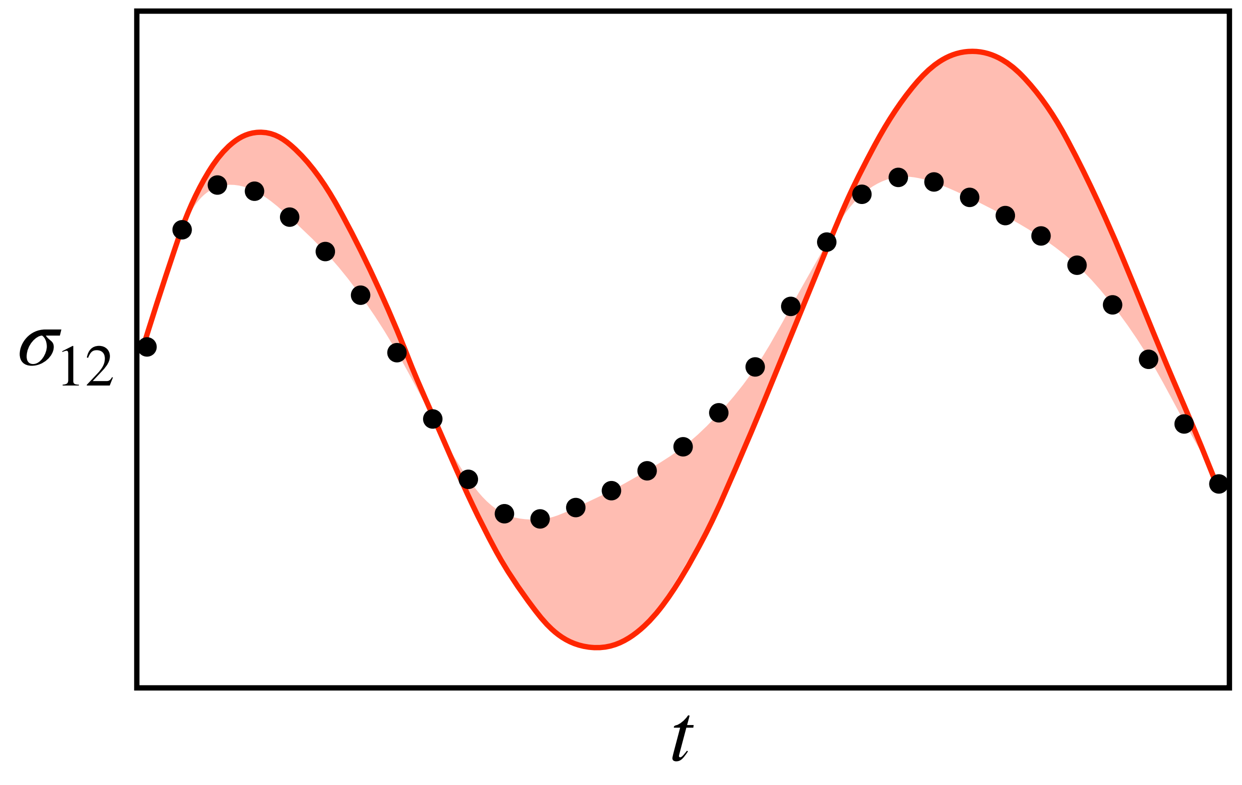
Rackauckas … Edelman, *arXiv:2001.04385* (2020)

Training a RUDE on synthetic data using Julia

Example: train a RUDE using synthetic data (Giesekus model)

$$\tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \frac{\tau\alpha}{\eta_0}\boldsymbol{\sigma}\cdot\boldsymbol{\sigma} = \eta_0 \dot{\boldsymbol{\gamma}} \qquad \alpha = 0.8$$

Training data: the shear stress (σ_{12}) in eight oscillatory tests



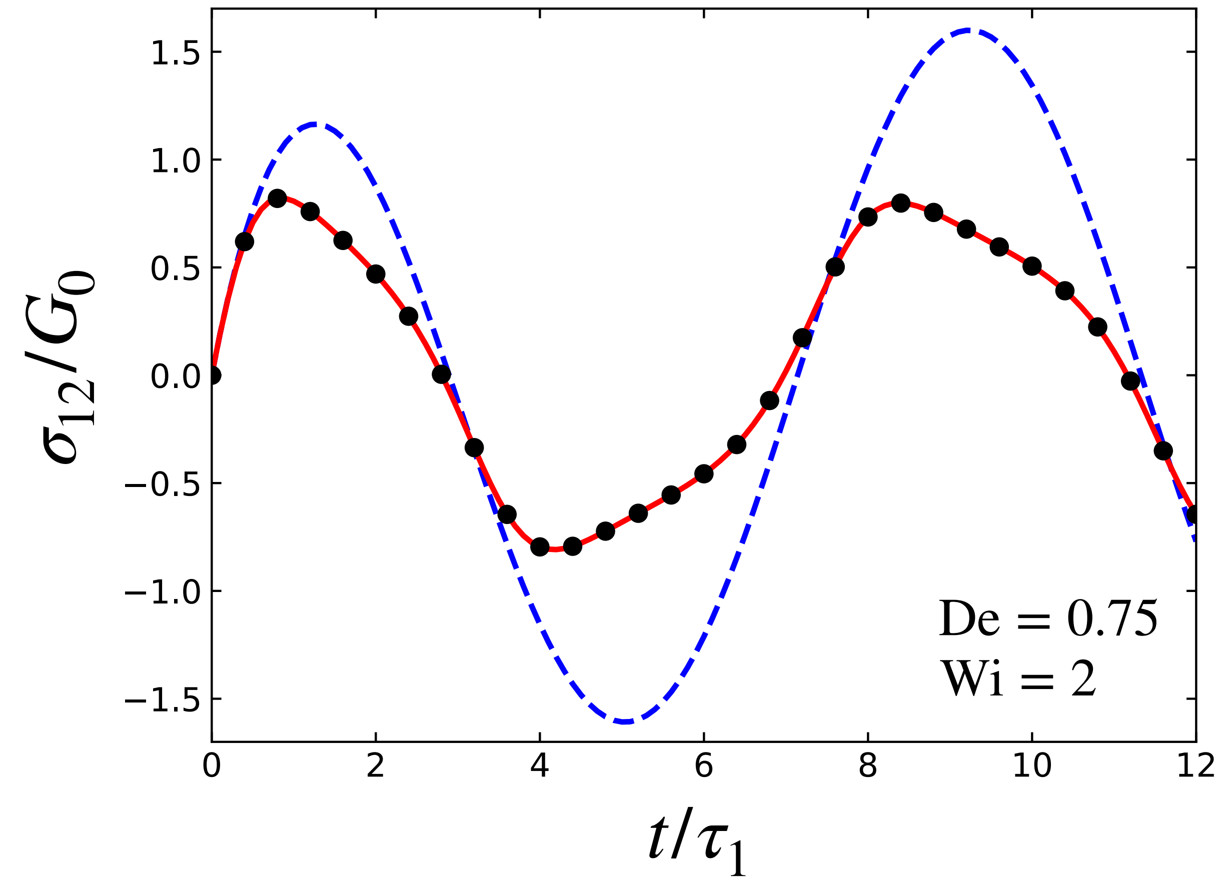
$$\dot{\gamma}(t) = \mathrm{Wi}\sin(\mathrm{De}\cdot t)$$

$$\mathrm{De} \in \{0.33, 0.5, 1, 2\} \qquad \mathrm{Wi} \in \{1, 2\}$$

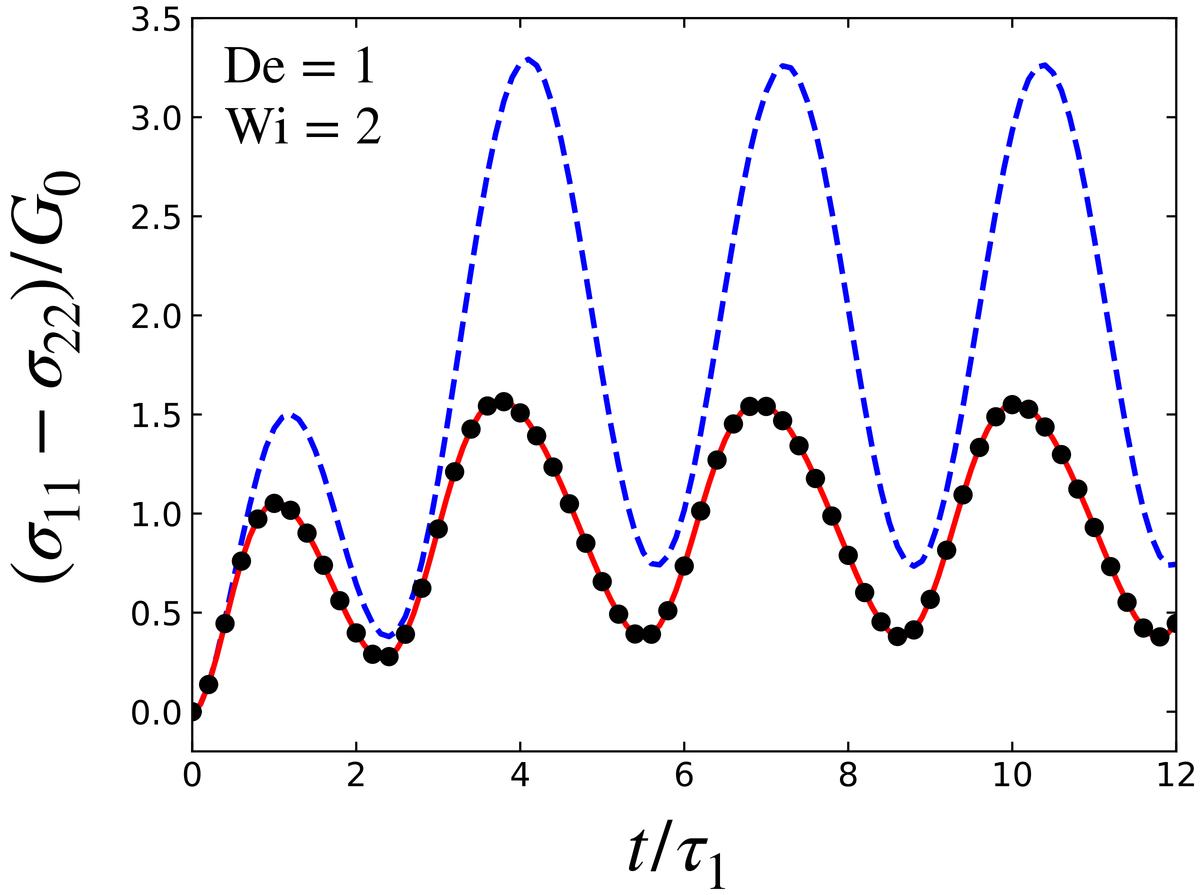
(DifferentialEquations.jl) (Flux.jl)

Test tasks:

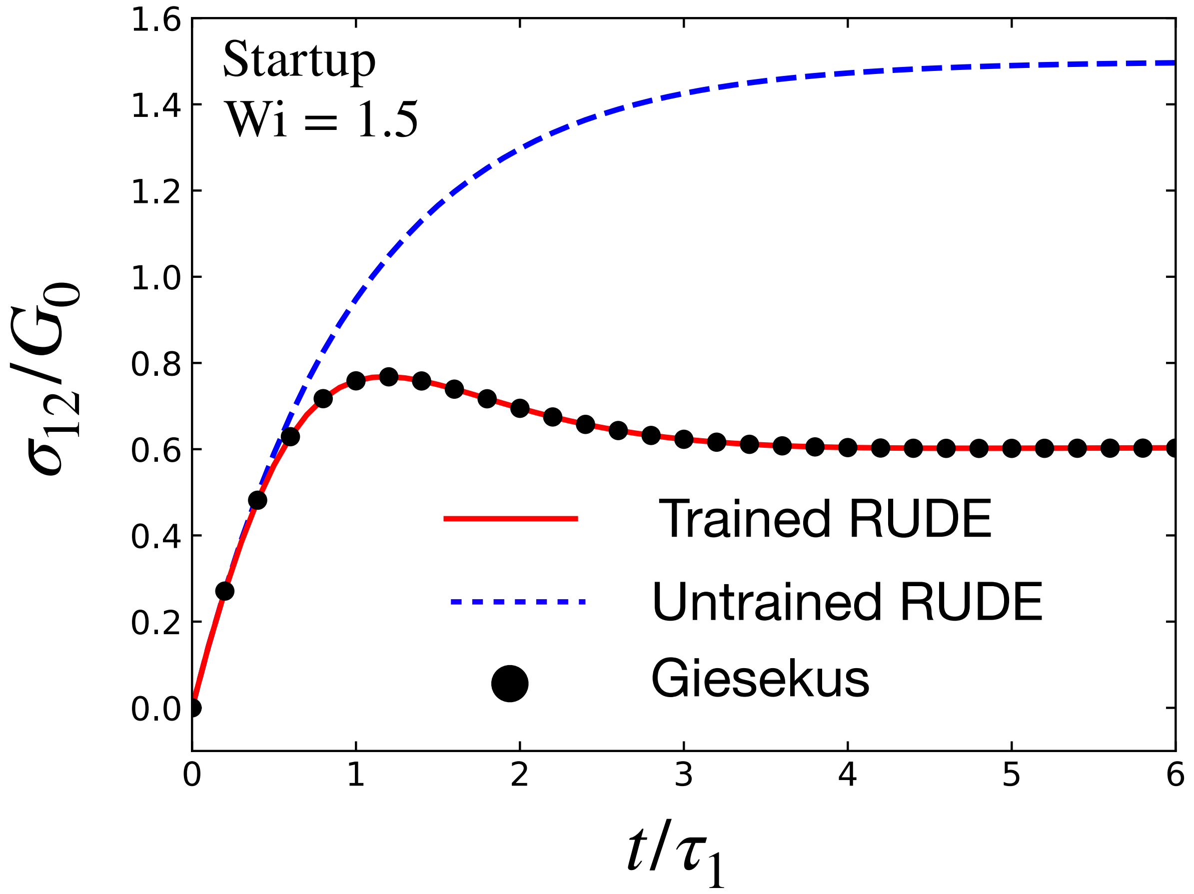
Predict σ_{12} at a new De



Predict normal stresses



Predict startup transient



The trained model reproduces the ground truth for protocols + observables not in the training set

Rackauckas … Edelman, *arXiv:2001.04385* (2020)

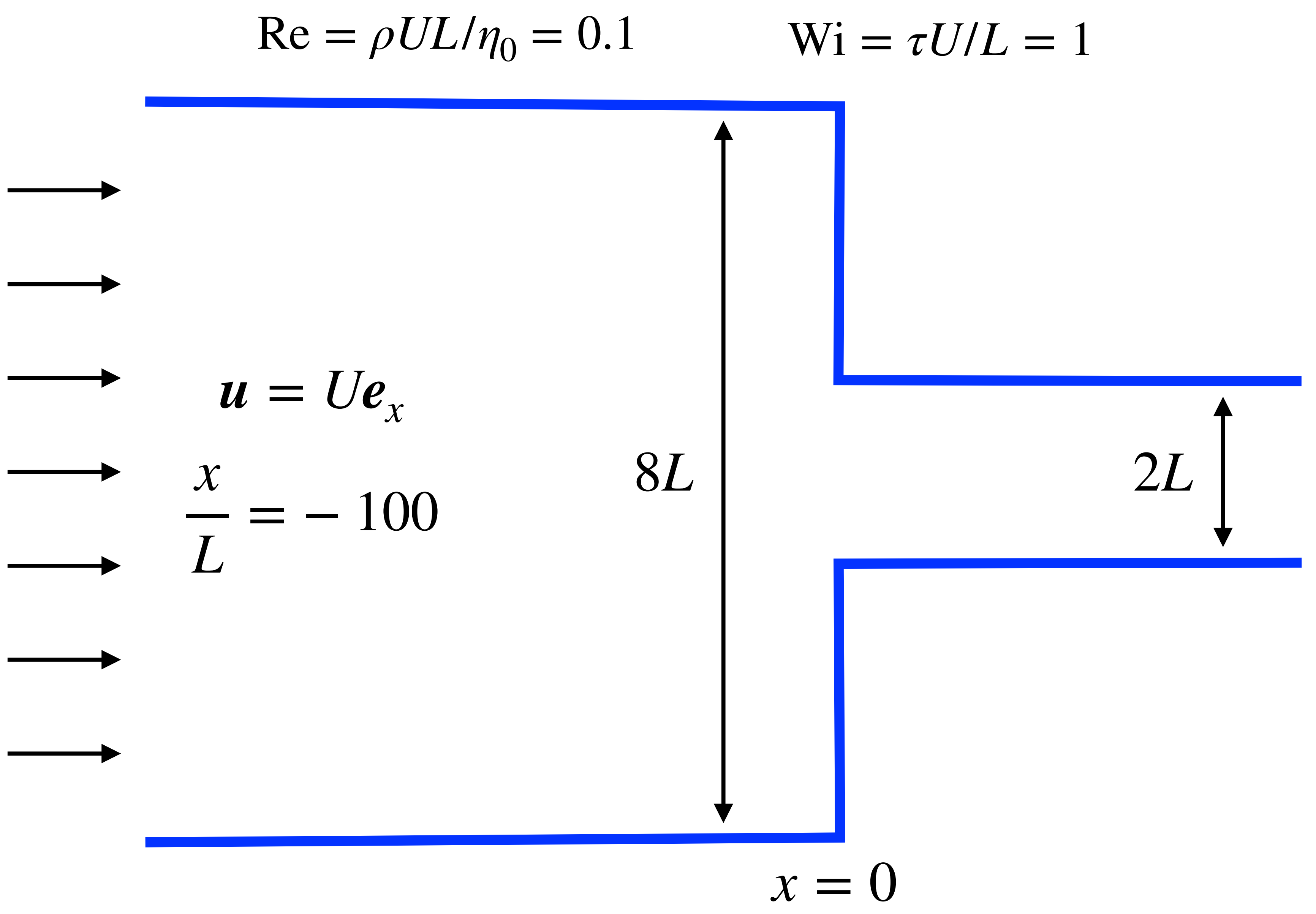
Trained RUDEs can make predictions in complicated flows

RUDEs are continuous-time tensorial models, so they are compatible with existing computational fluid dynamics tools that numerically solve the Cauchy momentum equation

OpenFOAM®   RheoTool

OpenFOAM with the RheoTool extension is a high-performance tool for simulating complex fluids with differential constitutive equations for the stress tensor

$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} \qquad \tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \sum_{n=1}^{9} g_n(\lambda_n; \theta)\boldsymbol{T}_n = \eta_0 \dot{\boldsymbol{\gamma}}$$



Weller … Fureby, *Comput. Phys., 12* (1998)   Pimenta and Alves, *JNNFM, 239* (2017)

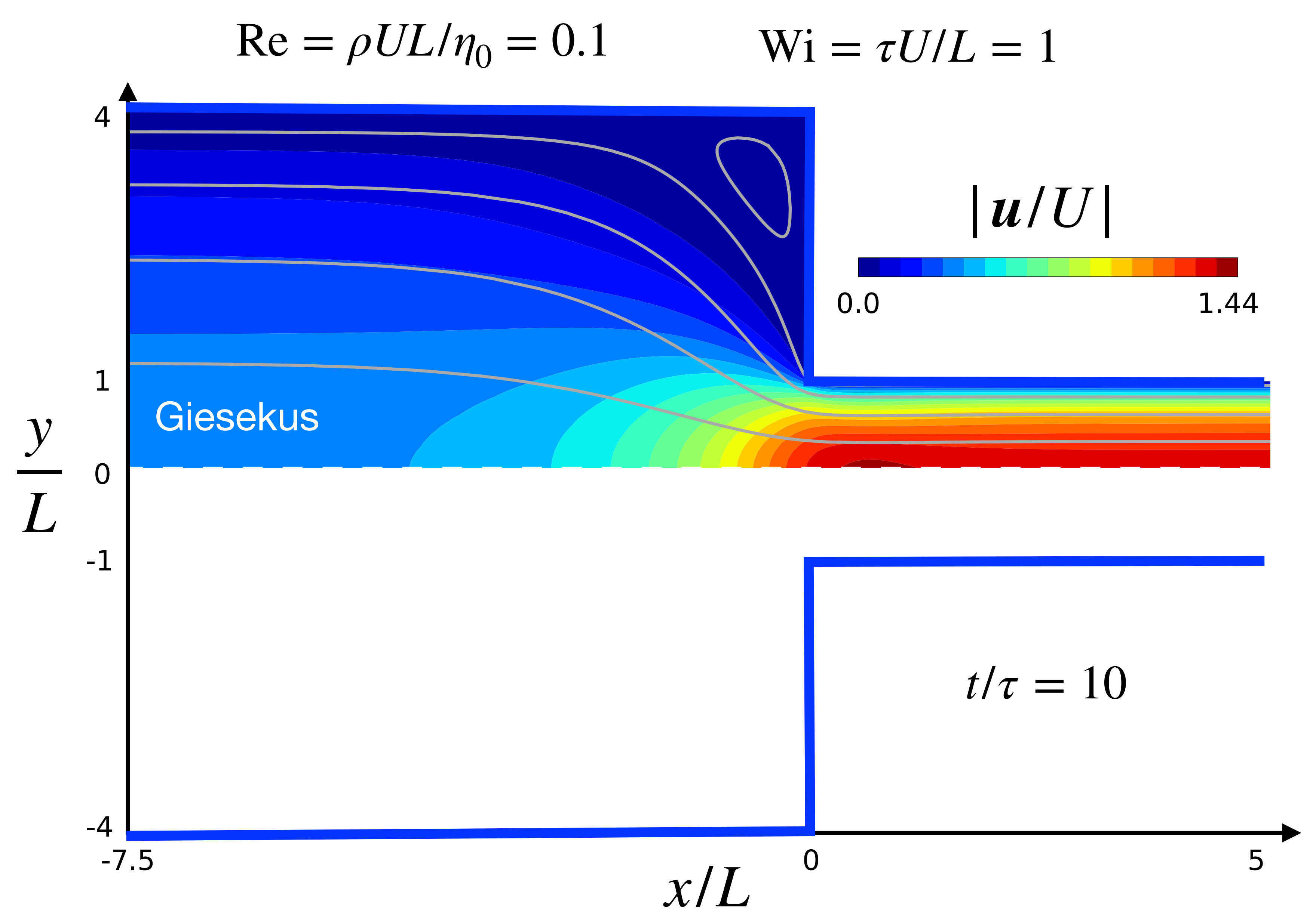
Trained RUDEs can make predictions in complicated flows

RUDEs are continuous-time tensorial models, so they are compatible with existing computational fluid dynamics tools that numerically solve the Cauchy momentum equation

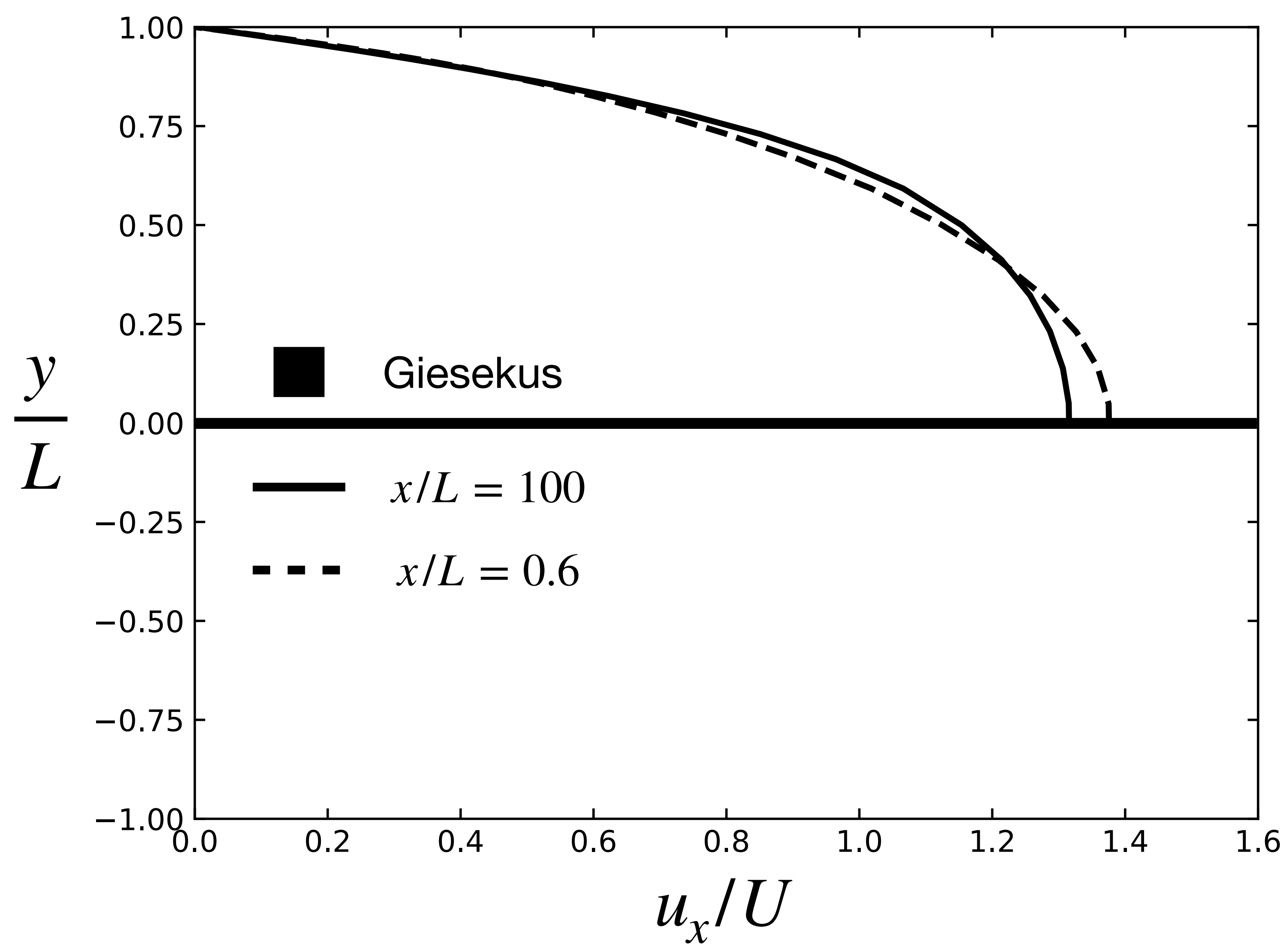
OpenFOAM® RheoTool

OpenFOAM with the RheoTool extension is a high-performance tool for simulating complex fluids with differential constitutive equations for the stress tensor

$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} \qquad \tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \sum_{n=1}^{9} g_n(\lambda_n; \theta)\boldsymbol{T}_n = \eta_0 \dot{\boldsymbol{\gamma}}$$



Velocity profiles



Weller … Fureby, *Comput. Phys., 12* (1998)   Pimenta and Alves, *JNNFM, 239* (2017)

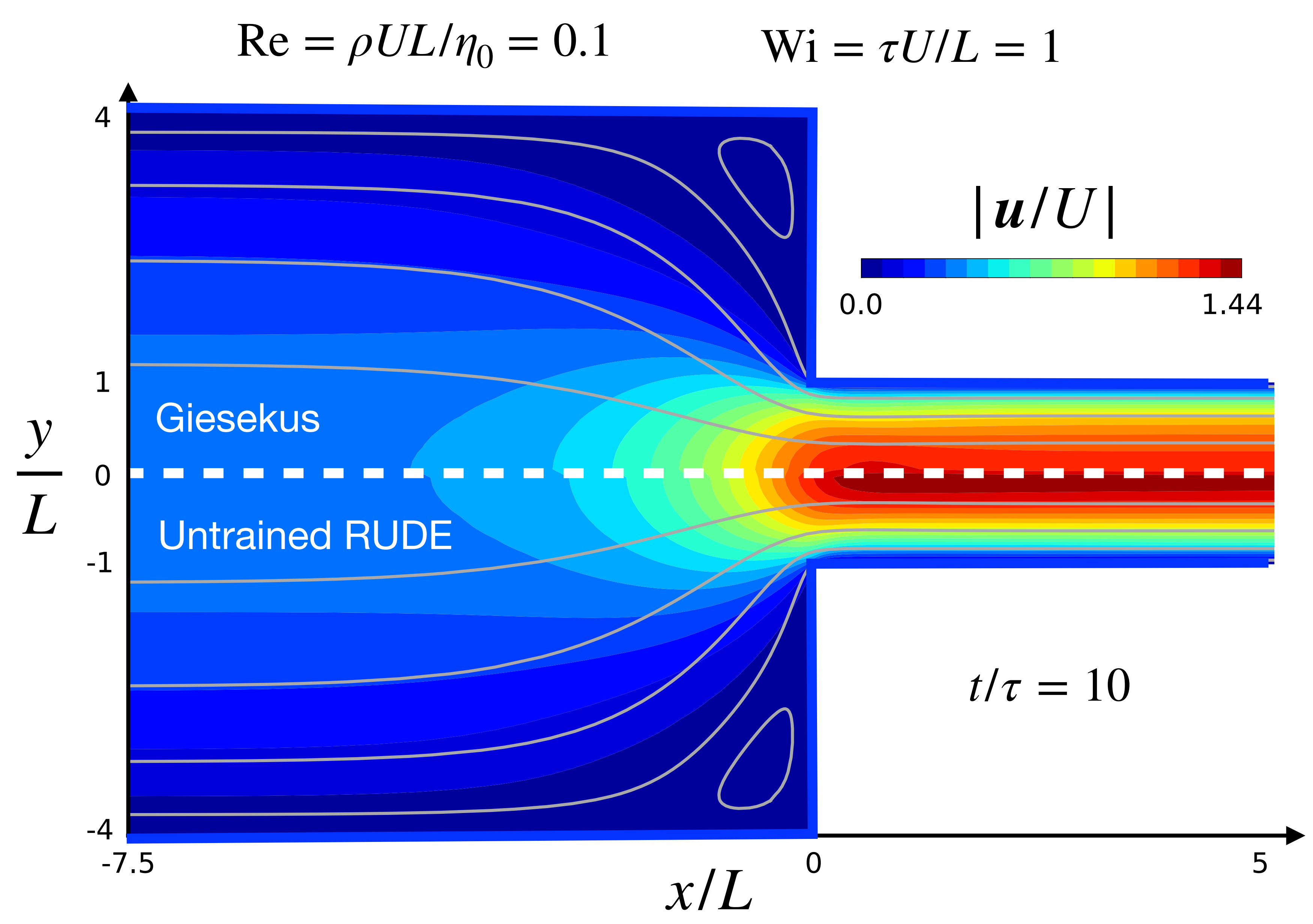
Trained RUDEs can make predictions in complicated flows

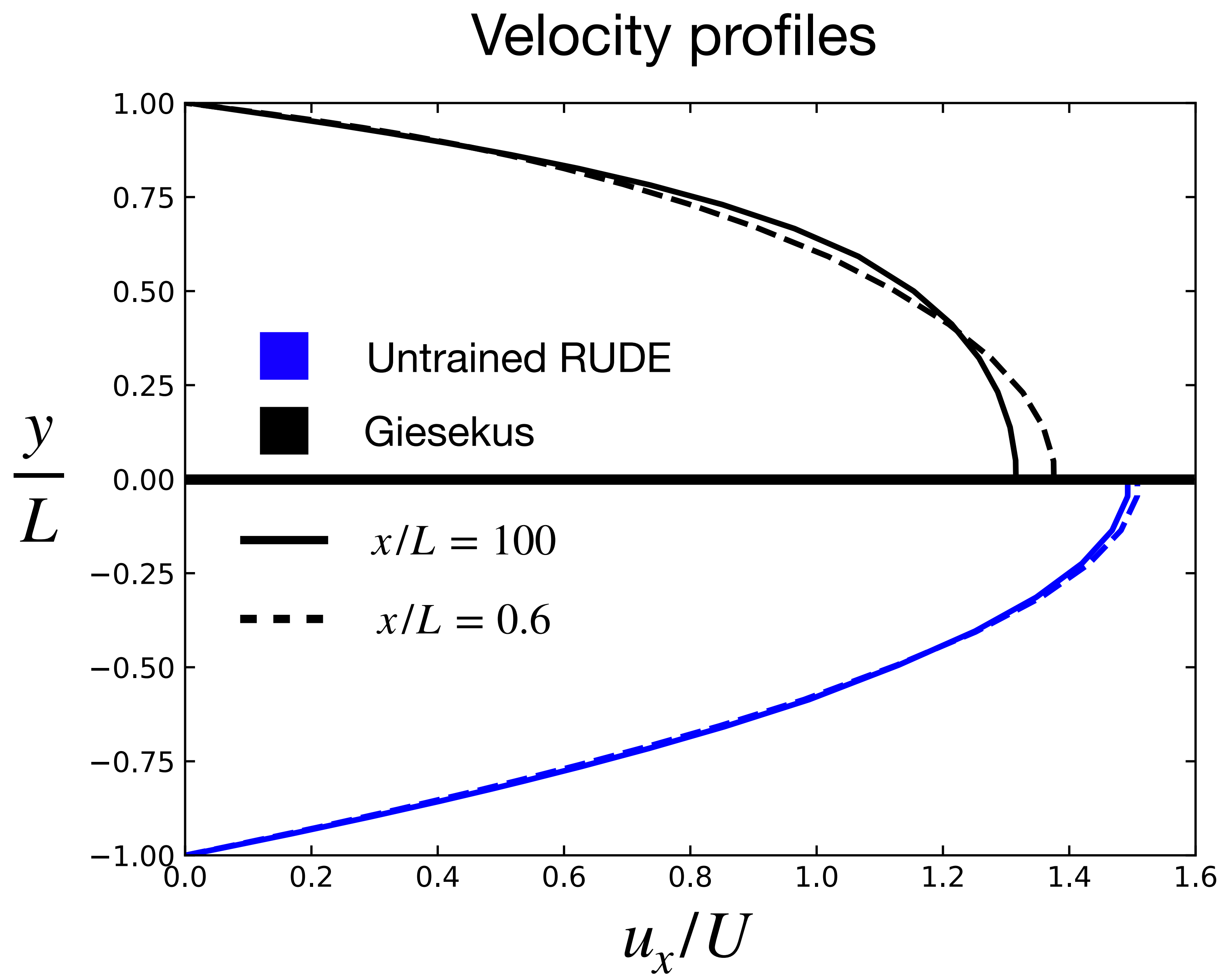
RUDEs are continuous-time tensorial models, so they are compatible with existing computational fluid dynamics tools that numerically solve the Cauchy momentum equation

OpenFOAM® RheoTool

OpenFOAM with the RheoTool extension is a high-performance tool for simulating complex fluids with differential constitutive equations for the stress tensor

$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} \qquad \tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \sum_{n=1}^{9} g_n(\lambda_n; \theta)\boldsymbol{T}_n = \eta_0 \dot{\boldsymbol{\gamma}}$$





Weller … Fureby, *Comput. Phys., 12* (1998)   Pimenta and Alves, *JNNFM, 239* (2017)

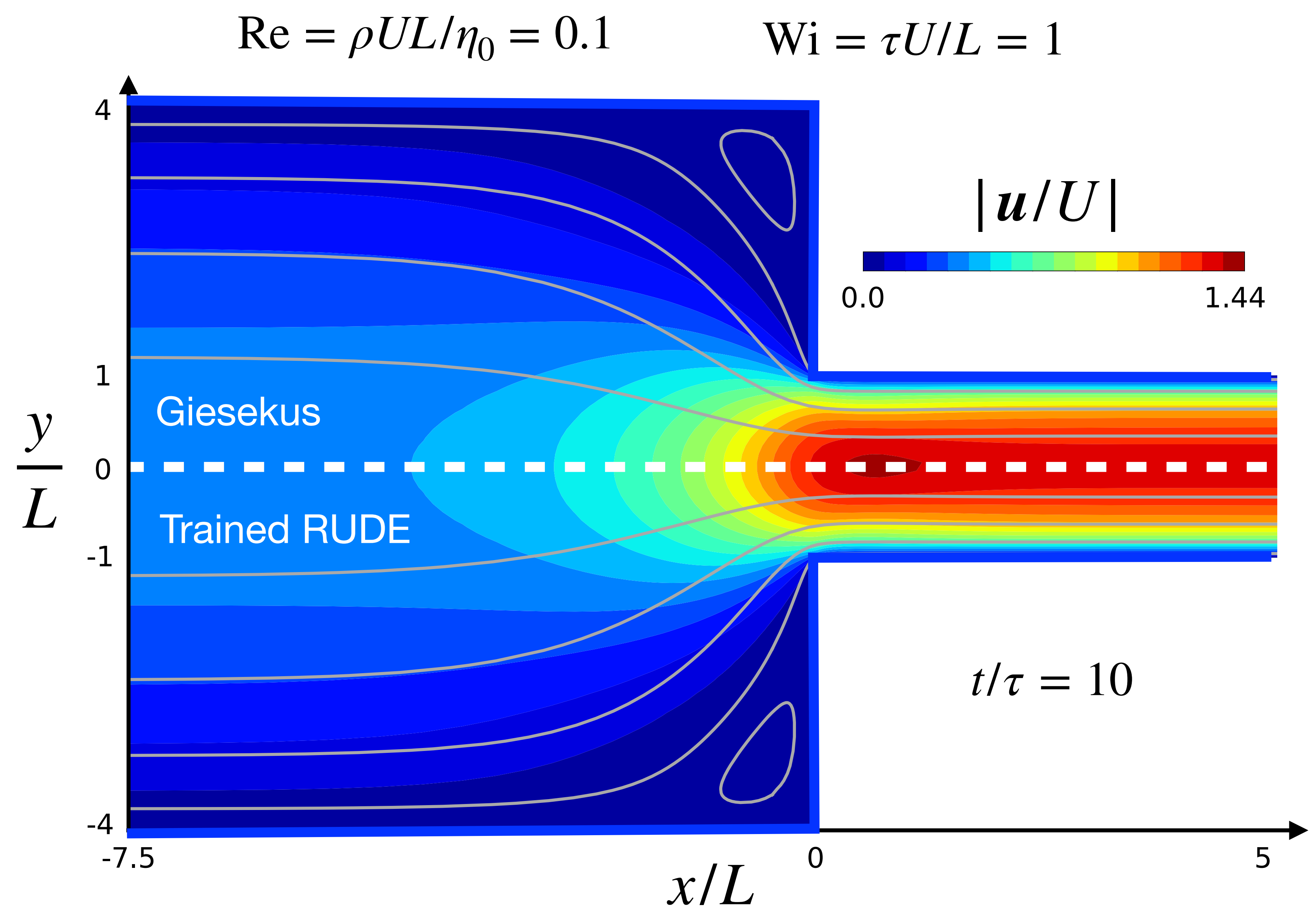
Trained RUDEs can make predictions in complicated flows

RUDEs are continuous-time tensorial models, so they are compatible with existing computational fluid dynamics tools that numerically solve the Cauchy momentum equation

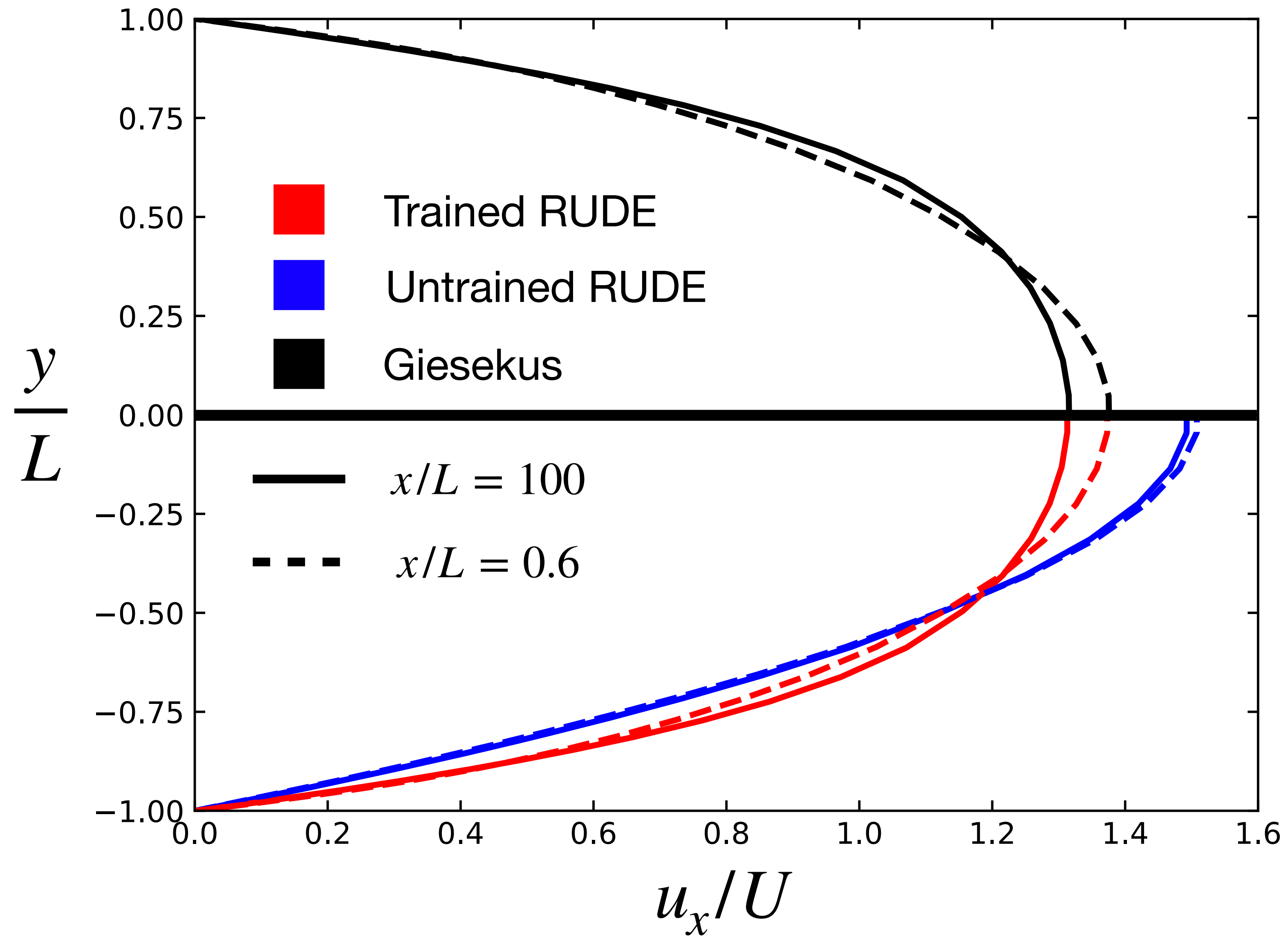
OpenFOAM® RheoTool

OpenFOAM with the RheoTool extension is a high-performance tool for simulating complex fluids with differential constitutive equations for the stress tensor

$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} \qquad \tau \overset{\nabla}{\boldsymbol{\sigma}} + \boldsymbol{\sigma} + \sum_{n=1}^{9} g_n(\lambda_n; \theta)\boldsymbol{T}_n = \eta_0 \dot{\boldsymbol{\gamma}}$$



Velocity profiles



Weller … Fureby, *Comput. Phys., 12* (1998)   Pimenta and Alves, *JNNFM, 239* (2017)

THANK YOU!!!!

DOE CSGF