Supplementary Information

pyRheo: An open-source Python package for complex rheology

Isaac Y. Miranda-Valdez,[a,*] Aaro Niinistö,[a] Tero Mäkinen,[a], Juha Lejon,[a], Juha Koivisto,[a] and Mikko J. Alava,[a]

[a]Complex Systems and Materials, Department of Applied Physics, Aalto University, P.O. Box 11000, FI-00076 Aalto, Espoo, Finland
[*] Corresponding author: isaac.mirandavaldez@aalto.fi

Abstract
This Supplementary Information accompanies the article "pyRheo: An open-source Python package for complex rheology". It provides an overview of the theoretical models implemented in pyRheo. Detailed equations and representative plots for various models are presented to elucidate their functionality within pyRheo. Here we cover the Maxwell, Springpot, Fractional Maxwell, Fractional Kelvin-Voigt, and Fractional Zener models. For each model, the documentation entails the model diagram, constitutive equation, creep compliance, relaxation modulus, and storage and loss moduli. Additionally, representative plots illustrate the practical application of these models. We also discuss how the machine learning models were trained for pyRheo, which is accompanied by confusion matrices showing the machine learning accuracy. The Supplementary Information provides several examples of fitting done with pyRheo. Furthermore, here we describe how the Mittag–Leffler function is implemented in pyRheo, highlighting the fourth-order global Padé approximation and comparing it with Garrappa's algorithm. The Supplementary Information presents an example of using Bayesian optimization to infer the initial guesses for the parameter fitting. This document also presents a tutorial on how to use pyRheo graphical user interface (GUI).

Supplementary Note 1 pyRheo workflow

The general purpose of this package is to analyze creep, stress relaxation, small amplitude oscillatory shear (SAOS), and steady shear data. pyRheo has two main features: (i) providing a machine learning decision of what model likely describes the rheological data and (ii) fitting a rheological model to the data. Next, we describe pyRheo's workflow.

Step 1: importing data

First, the user should import the data pertinent to the type of rheological dataset. Depending on the specific nature of the dataset, users should import the relevant features and their corresponding class (creep, stress relaxation, SAOS, or steady shear). pyRheo is designed to work with material functions, so one must provide at least two data vectors. For example, for creep and stress relaxation data, it is expected to import a time t vector together with its corresponding material function $J(t)$ or $G(t)$. Alternatively, for SAOS data, the user must import angular frequency ω and the materials functions storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$.

Step 2: model selection

After importing the data, the user shall select to analyze their data using the `auto` method or by manually specifying a modeL. The `auto` method uses a pre-trained Neural Network based on a Multi-Layer Perceptron (MLP). Each class has its own MLP classifier, which has been trained using 1 million computations of the corresponding material function (e.g., $J(t)$) derived from the constitutive equations of the `Maxwell`, `SpringPot`, `FractionalMaxwell`, and `FractionalKelvinVoigt`.

The accuracy (with synthetic data) of the MLP classifiers ranges from 70 to 80%. We suggest using the `auto` method as a first approximation to identify the type of rheological behavior. More detailed information about the machine learning training process and performance is disclosed in the Supplementary Note 3.

Step 3: model fitting

Parameter optimization with pyRheo follows the common practice of minimizing the weighted residual sum of squares[1,2],

$$\mathrm{RSS}_{w_i=y_i} = \sum_{i=1}^{n} \left(\frac{y_i - f(x_i)}{y_i} \right)^2 \tag{S1}$$

where the weights w_i are the measured values y_i Users may define their own initial guesses and parameter bounds (automatic bounds and random initial guesses by default). Then, to minimize RSS_{w_i}, users can choose from several minimization algorithms implemented on SciPy[3], such as Nelder–Mead, Powell, and L-BFGS-B (Powell by default).

An advantage of using pyRheo is that it addresses the challenges associated with sensitivity to initial guesses in parameter optimization. In other words, if an initial guess is close to a local minimum, the minimization algorithm may converge there instead of the global minimum. Therefore, pyRheo allows the user to restart the fitting process multiple times with random initial parameter values and then take as the final result the iteration with the lowest RSS_{w_i}. By generating a diverse set of random starting points, this brute-force approach increases the likelihood of exploring different regions of the parameter space, thus avoiding local minima.

In the Supplementary Note 2, we detail all the models implemented in pyRheo. The reader can find the constitutive equations of each model and representative plots showing how material functions change as a function of varying parameters. Furthermore, the reader can find information about how the Mittag–Leffler function is implemented in pyRheo using Zeng and Chen[4], Sarumi *et al.*[5], and Garrappa[6] concepts.

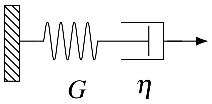
Step 4: analysis of results

After fitting the target model to the rheological data, the results are stored as an object variable that contains all the necessary components for prediction, visualization, and further data analysis. Users can learn more from pyRheo's documentation and the examples available on its GitHub repository.

Supplementary Note 2 Equations of the models implemented in pyRheo

Maxwell model (`Maxwell`)

Model diagram



Constitutive equation

$$\sigma(t) + \frac{\eta}{G}\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} = \eta\frac{\mathrm{d}\gamma(t)}{\mathrm{d}t}$$
$$\tau_c = \frac{\eta}{G} \tag{S2}$$
$$G_c = G$$

Creep compliance

$$J(t) = \frac{t}{\eta} + \frac{1}{G} \tag{S3}$$

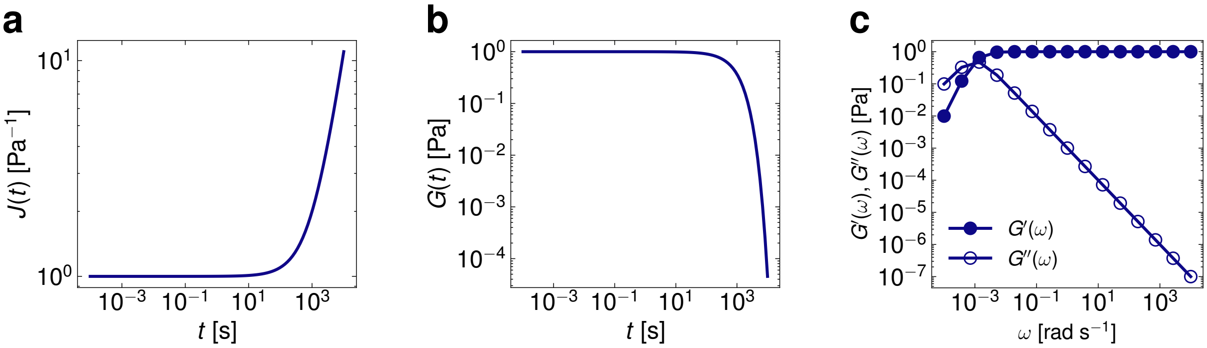
Relaxation modulus

$$G(t) = G_c \exp\left(-\frac{t}{\tau_c}\right) \tag{S4}$$

Storage modulus and Loss modulus

$$G'(\omega) = G_c\frac{(\omega\tau_c)^2}{1+(\omega\tau_c)^2}$$
$$G''(\omega) = G_c\frac{\omega\tau_c}{1+(\omega\tau_c)^2} \tag{S5}$$

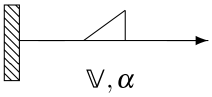
Representative plots



Supplementary Fig. 1 Representative plots of the Maxwell model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $G = 1$ Pa and $\eta = 1000$ Pa s. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Springpot model (`SpringPot`)

Model diagram



Constitutive equation

$$\sigma(t) = \mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}} \tag{S6}$$

Creep compliance

$$J(t) = \frac{1}{\mathbb{V}}\frac{t^{\alpha}}{\Gamma(1+\alpha)} \tag{S7}$$

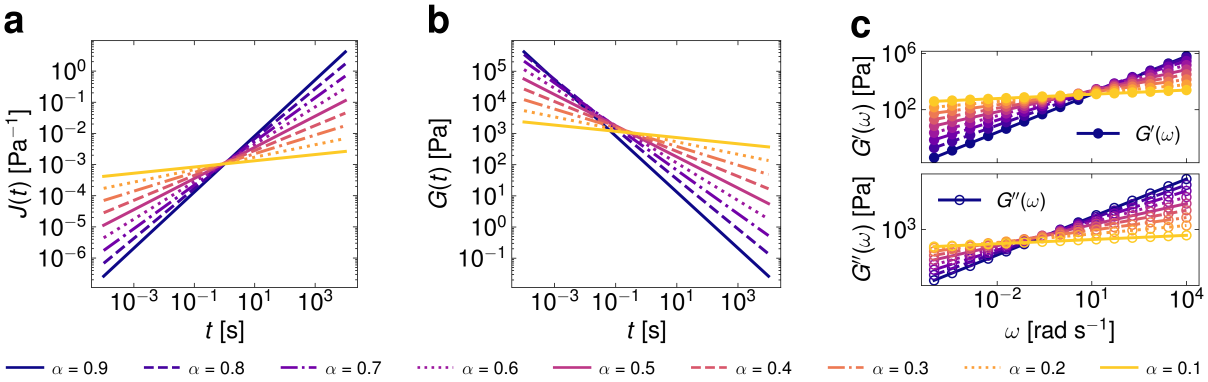
Relaxation modulus

$$G(t) = \mathbb{V}\frac{t^{-\alpha}}{\Gamma(1-\alpha)} \tag{S8}$$

Storage modulus and Loss modulus

$$\begin{aligned} G'(\omega) &= \mathbb{V}\omega^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right) \\ G''(\omega) &= \mathbb{V}\omega^{\alpha}\sin\left(\frac{\pi}{2}\alpha\right) \end{aligned} \tag{S9}$$

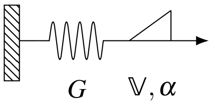
Representative plots



Supplementary Fig. 2 Representative plots of the Springpot (critical gel) model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\mathbb{V} = 1000$ Pa s$^{\alpha}$ and varying α. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Maxwell Gel model (`FractionalMaxwellGel`)

Model diagram



Constitutive equation

$$\sigma(t)+\frac{\mathbb{V}}{G}\frac{\mathrm{d}^{\alpha}\sigma(t)}{\mathrm{d}t^{\alpha}}=\mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}}$$
$$\tau_c=\left(\frac{\mathbb{V}}{G}\right)^{\frac{1}{\alpha}} \tag{S10}$$
$$G_c=\mathbb{V}\tau_c^{-\alpha}$$

Creep compliance

$$J(t)=\frac{1}{\mathbb{V}}\frac{t^{\alpha}}{\Gamma(1+\alpha)}+\frac{1}{G} \tag{S11}$$

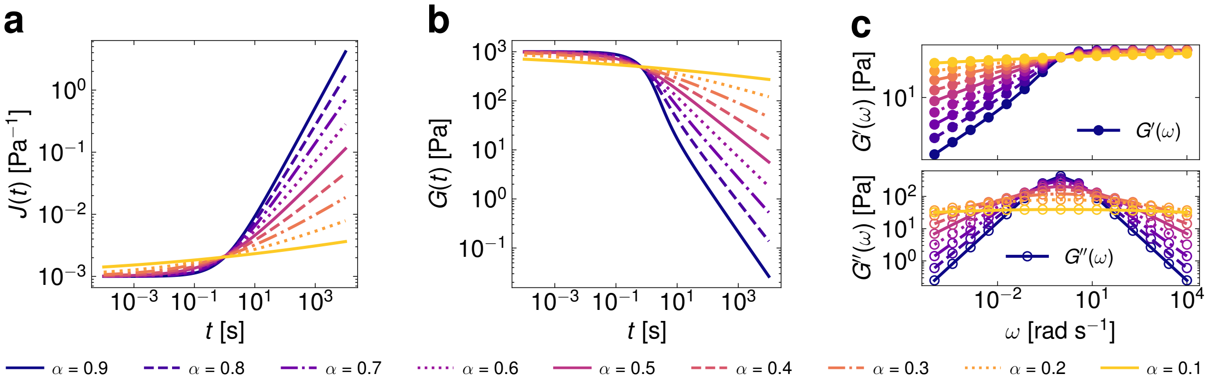
Relaxation modulus

$$G(t)=G_cE_{a,b}(z)$$
$$a=\alpha$$
$$b=1 \tag{S12}$$
$$z=-\left(\frac{t}{\tau_c}\right)^{\alpha}$$

Storage modulus and Loss modulus

$$G'(\omega)=G_c\frac{(\omega\tau_c)^{2\alpha}+(\omega\tau_c)^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right)}{1+(\omega\tau_c)^{2\alpha}+2(\omega\tau_c)^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right)}$$
$$G''(\omega)=G_c\frac{(\omega\tau_c)^{\alpha}\sin\left(\frac{\pi}{2}\alpha\right)}{1+(\omega\tau_c)^{2\alpha}+2(\omega\tau_c)^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right)} \tag{S13}$$

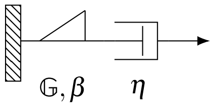
Representative plots



Supplementary Fig. 3 Representative plots of the Fractional Maxwell Gel model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\mathbb{V}=1000$ Pa s$^{\alpha}$ and $G=1000$ Pa as well as varying α. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Maxwell Liquid model (`FractionalMaxwellLiquid`)

Model diagram



Constitutive equation

$$\begin{gathered}\sigma(t) + \frac{\eta}{\mathbb{G}}\frac{\mathrm{d}^{1-\beta}\sigma(t)}{\mathrm{d}t^{1-\beta}} = \eta\frac{\mathrm{d}\gamma(t)}{\mathrm{d}t} \\ \tau_c = \left(\frac{\eta}{\mathbb{G}}\right)^{\frac{1}{1-\beta}} \\ G_c = \eta\tau_c^{-1}\end{gathered} \tag{S14}$$

Creep compliance

$$J(t) = \frac{t}{\eta} + \frac{1}{\mathbb{G}}\frac{t^{\beta}}{\Gamma(1+\beta)} \tag{S15}$$

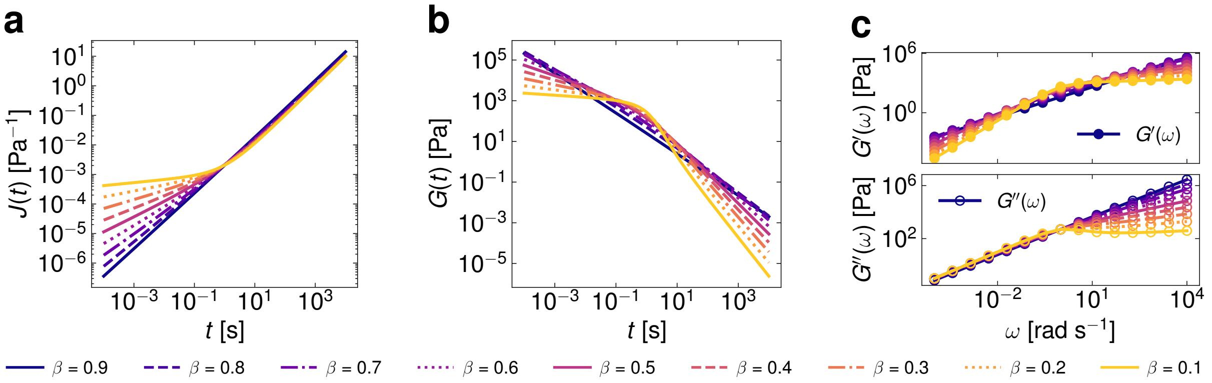
Relaxation modulus

$$\begin{aligned}G(t) &= G_c\left(\frac{t}{\tau_c}\right)^{-\beta} E_{a,b}(z) \\ a &= 1-\beta \\ b &= 1-\beta \\ z &= -\left(\frac{t}{\tau_c}\right)^{1-\beta}\end{aligned} \tag{S16}$$

Storage modulus and Loss modulus

$$\begin{aligned}G'(\omega) &= G_c\frac{(\omega\tau_c)^{2-\beta}\cos\left(\frac{\pi}{2}\beta\right)}{1+(\omega\tau_c)^{2(1-\beta)}+2(\omega\tau_c)^{1-\beta}\cos\left(\frac{\pi}{2}(1-\beta)\right)} \\ G''(\omega) &= G_c\frac{(\omega\tau_c)+(\omega\tau_c)^{2-\beta}\sin\left(\frac{\pi}{2}\beta\right)}{1+(\omega\tau_c)^{2(1-\beta)}+2(\omega\tau_c)^{1-\beta}\cos\left(\frac{\pi}{2}(1-\beta)\right)}\end{aligned} \tag{S17}$$

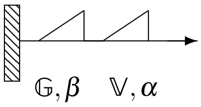
Representative plots



Supplementary Fig. 4 Representative plots of the Fractional Maxwell Liquid model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\mathbb{G} = 1000$ Pa s$^{\beta}$ and $\eta = 1000$ Pa as well as varying β. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Maxwell model (`FractionalMaxwell`)

Model diagram



Constitutive equation

$$\begin{aligned}\sigma(t)+\frac{\mathbb{V}}{\mathbb{G}}\frac{\mathrm{d}^{\alpha-\beta}\sigma(t)}{\mathrm{d}t^{\alpha-\beta}}&=\mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}}\\ \tau_c&=\left(\frac{\mathbb{V}}{\mathbb{G}}\right)^{\frac{1}{\alpha-\beta}}\\ G_c&=\mathbb{V}\tau_c^{-\alpha}\end{aligned}\tag{S18}$$

Creep compliance

$$J(t)=\frac{1}{\mathbb{V}}\frac{t^{\alpha}}{\Gamma(1+\alpha)}+\frac{1}{\mathbb{G}}\frac{t^{\beta}}{\Gamma(1+\beta)}\tag{S19}$$

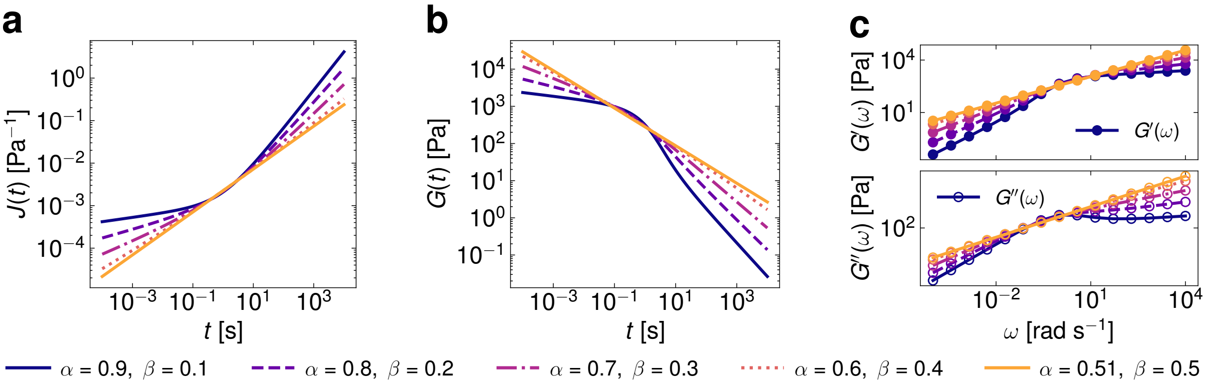
Relaxation modulus

$$\begin{aligned}G(t)&=G_c\left(\frac{t}{\tau_c}\right)^{-\beta}E_{a,b}(z)\\ a&=\alpha-\beta\\ b&=1-\beta\\ z&=-\left(\frac{t}{\tau_c}\right)^{\alpha-\beta}\end{aligned}\tag{S20}$$

Storage modulus and Loss modulus

$$\begin{aligned}G'(\omega)&=G_c\frac{(\omega\tau_c)^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right)+(\omega\tau_c)^{2\alpha-\beta}\cos\left(\frac{\pi}{2}\beta\right)}{1+(\omega\tau_c)^{2(\alpha-\beta)}+2(\omega\tau_c)^{\alpha-\beta}\cos\left(\frac{\pi}{2}(\alpha-\beta)\right)}\\ G''(\omega)&=G_c\frac{(\omega\tau_c)^{\alpha}\sin\left(\frac{\pi}{2}\alpha\right)+(\omega\tau_c)^{2\alpha-\beta}\sin\left(\frac{\pi}{2}\beta\right)}{1+(\omega\tau_c)^{2(\alpha-\beta)}+2(\omega\tau_c)^{\alpha-\beta}\cos\left(\frac{\pi}{2}(\alpha-\beta)\right)}\end{aligned}\tag{S21}$$

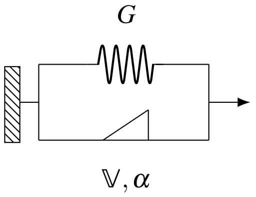
Representative plots



Supplementary Fig. 5 Representative plots of the Fractional Maxwell model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\mathbb{V}=1000$ Pa s$^{\alpha}$ and $\mathbb{G}=1000$ Pa s$^{\beta}$ as well as varying α and β. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Kelvin-Voigt-S model (`FractionalKelvinVoigtS`)

Model diagram



Constitutive equation

$$\begin{aligned} \sigma(t) &= \mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}} + G\gamma(t) \\ \tau_c &= \left(\frac{\mathbb{V}}{G}\right)^{\frac{1}{\alpha}} \\ G_c &= \mathbb{V}\tau_c^{-\alpha} \end{aligned} \tag{S22}$$

Creep compliance

$$\begin{aligned} J(t) &= \frac{t^{\alpha}}{\mathbb{V}}E_{a,b}(z) \\ a &= \alpha \\ b &= 1+\alpha \\ z &= -\left(\frac{t}{\tau_c}\right)^{\alpha} \end{aligned} \tag{S23}$$

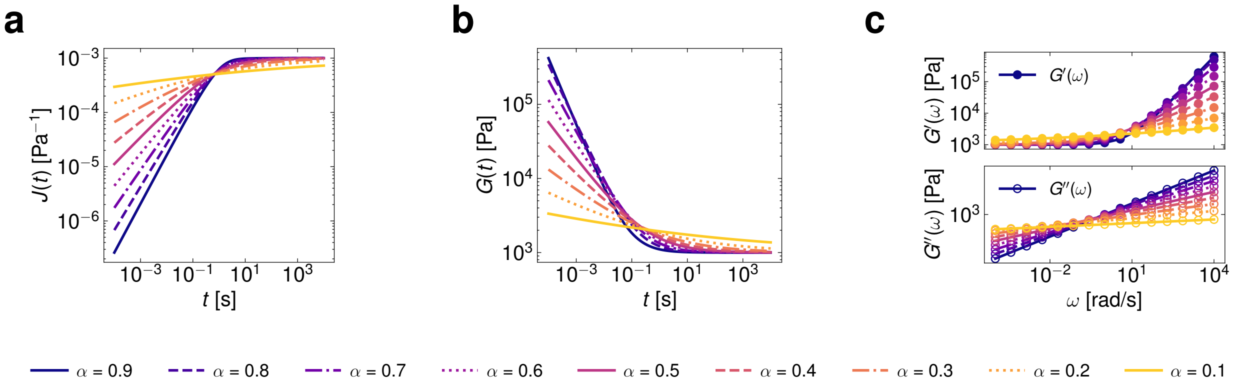
Relaxation modulus

$$G(t) = \mathbb{V}\frac{t^{-\alpha}}{\Gamma(1-\alpha)} + G \tag{S24}$$

Storage modulus and Loss modulus

$$\begin{aligned} G'(\omega) &= \mathbb{V}\omega^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right) + G \\ G''(\omega) &= \mathbb{V}\omega^{\alpha}\sin\left(\frac{\pi}{2}\alpha\right) \end{aligned} \tag{S25}$$

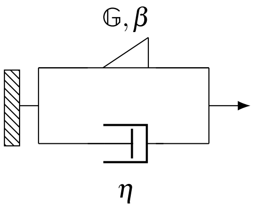
Representative plots



Supplementary Fig. 6 Representative plots of the Fractional Kelvin-Voigt-S model. The plots were computed using $\mathbb{V} = 1000$ Pa s$^{\alpha}$ and $G = 1000$ Pa, varying α. **a)** Time-dependent creep compliance $J(t)$ under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$.

Fractional Kelvin-Voigt-D model (`FractionalKelvinVoigtD`)

Model diagram



Constitutive equation

$$\begin{aligned}\sigma(t) &= \eta\frac{\mathrm{d}\gamma(t)}{\mathrm{d}t} + \mathbb{G}\frac{\mathrm{d}^{\beta}\gamma(t)}{\mathrm{d}t^{\beta}} \\ \tau_c &= \left(\frac{\eta}{\mathbb{G}}\right)^{\frac{1}{1-\beta}} \\ G_c &= \eta\tau_c^{-1}\end{aligned} \tag{S26}$$

Creep compliance

$$\begin{aligned}J(t) &= \frac{t}{\eta}E_{a,b}(z) \\ a &= 1-\beta \\ b &= 1+1 \\ z &= -\left(\frac{t}{\tau_c}\right)^{1-\beta}\end{aligned} \tag{S27}$$

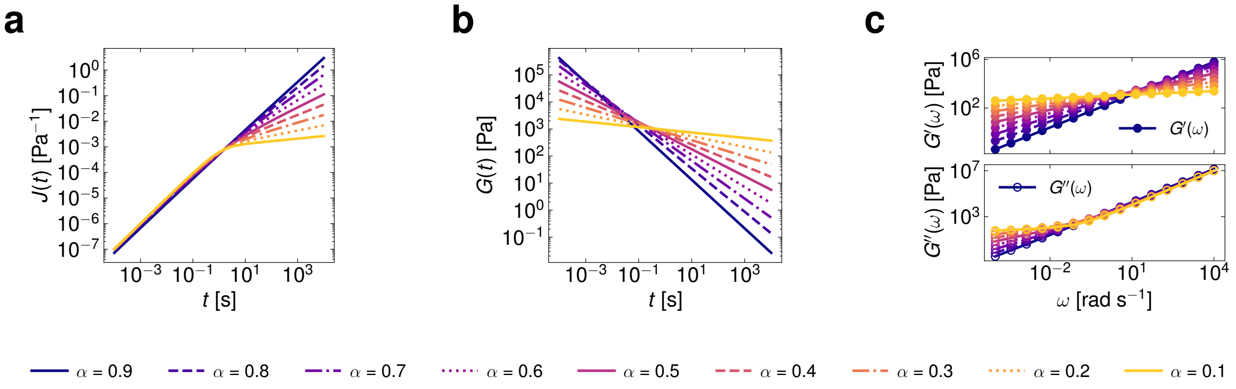
Relaxation modulus

$$G(t) = \eta\Delta(t) + \mathbb{G}\frac{t^{-\beta}}{\Gamma(1-\beta)} \tag{S28}$$

Storage modulus and Loss modulus

$$\begin{aligned}G'(\omega) &= \mathbb{G}\omega^{\beta}\cos\left(\frac{\pi}{2}\beta\right) \\ G''(\omega) &= \eta\omega + \mathbb{G}\omega^{\beta}\sin\left(\frac{\pi}{2}\beta\right)\end{aligned} \tag{S29}$$

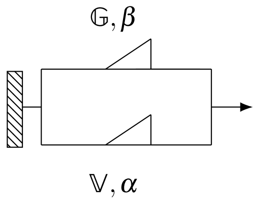
Representative plots



Supplementary Fig. 7 Representative plots of the Fractional Kelvin-Voigt-D model. The plots were computed using $\mathbb{V} = 1000$ Pa s$^{\alpha}$ and $\mathbb{G} = 1000$ Pa s$^{\beta}$, varying β. **a)** Time-dependent creep compliance $J(t)$ under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$.

Fractional Kelvin-Voigt model (`FractionalKelvinVoigt`)

Model diagram



Constitutive equation

$$\begin{aligned} \sigma(t) &= \mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}} + \mathbb{G}\frac{\mathrm{d}^{\beta}\gamma(t)}{\mathrm{d}t^{\beta}} \\ \tau_c &= \left(\frac{\mathbb{V}}{\mathbb{G}}\right)^{\frac{1}{\alpha-\beta}} \\ G_c &= \mathbb{V}\tau_c^{-\alpha} \end{aligned} \tag{S30}$$

Creep compliance

$$\begin{aligned} J(t) &= \frac{t^{\alpha}}{\mathbb{V}}E_{a,b}(z) \\ a &= \alpha - \beta \\ b &= 1 + \alpha \\ z &= -\left(\frac{t}{\tau_c}\right)^{\alpha-\beta} \end{aligned} \tag{S31}$$

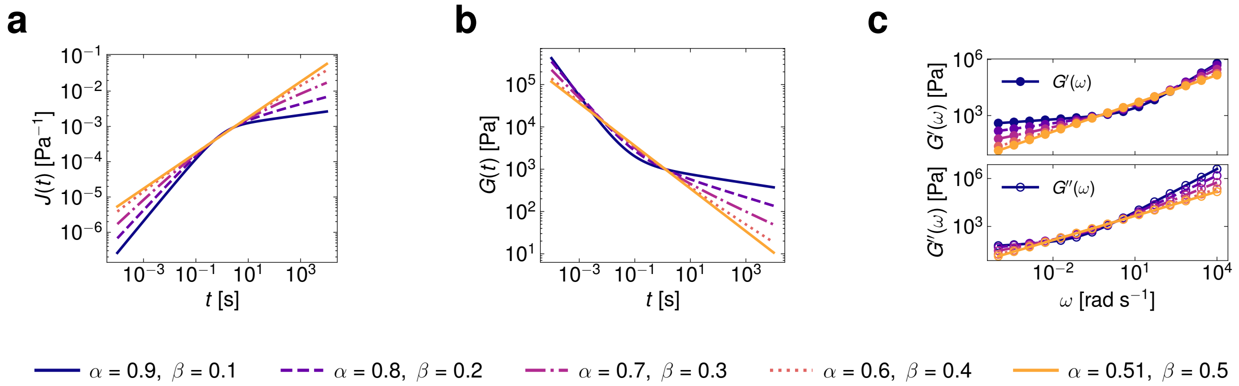
Relaxation modulus

$$G(t) = \mathbb{V}\frac{t^{-\alpha}}{\Gamma(1-\alpha)} + \mathbb{G}\frac{t^{-\beta}}{\Gamma(1-\beta)} \tag{S32}$$

Storage modulus and Loss modulus

$$\begin{aligned} G'(\omega) &= \mathbb{V}\omega^{\alpha}\cos\left(\frac{\pi}{2}\alpha\right) + \mathbb{G}\omega^{\beta}\cos\left(\frac{\pi}{2}\beta\right) \\ G''(\omega) &= \mathbb{V}\omega^{\alpha}\sin\left(\frac{\pi}{2}\alpha\right) + \mathbb{G}\omega^{\beta}\sin\left(\frac{\pi}{2}\beta\right) \end{aligned} \tag{S33}$$

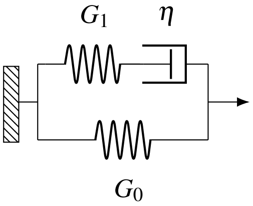
Representative plots



Supplementary Fig. 8 Representative plots of the Fractional Kelvin-Voigt model. The plots were computed using $\mathbb{V} = 1000$ Pa s$^{\alpha}$ and $\mathbb{G} = 1000$ Pa s$^{\beta}$, varying α and β. **a)** Time-dependent creep compliance $J(t)$ under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$.

Zener model (Zener)

Model diagram



Constitutive equation

$$\sigma(t)+\frac{\eta}{G_1}\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}=G_0\gamma(t)+\eta\frac{G_0+G_1}{G_1}\frac{\mathrm{d}\gamma(t)}{\mathrm{d}t}$$
$$\tau_c=\frac{\eta}{G_1} \tag{S34}$$
$$G_1=G_\infty-G_0$$

Creep compliance

$$J(t)=\frac{1}{G_0}-\frac{G_1}{G_0(G_0+G_1)}\exp\left(-\frac{G_0}{G_0+G_1}\frac{t}{\tau_c}\right) \tag{S35}$$

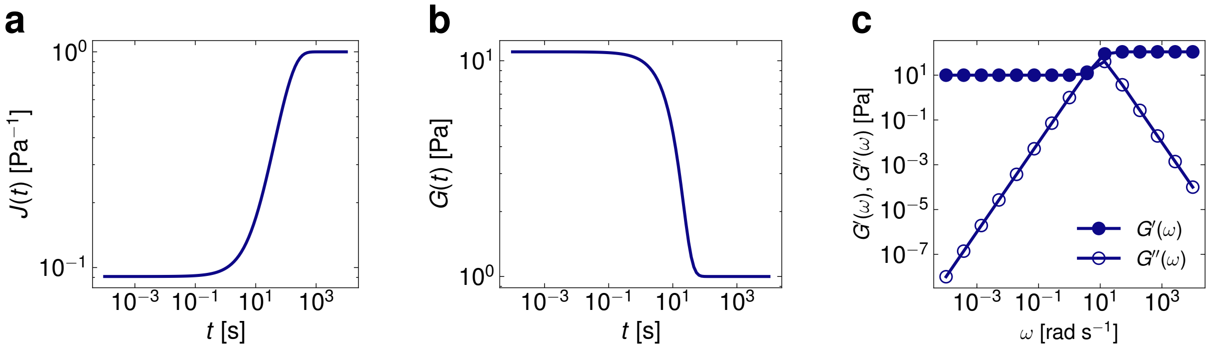
Relaxation modulus

$$G(t)=G_0+G_1\exp\left(-\frac{t}{\tau_c}\right) \tag{S36}$$

Storage modulus and Loss modulus

$$G'(\omega)=G_0+G_1\frac{(\omega\tau_c)^2}{1+(\omega\tau_c)^2}$$
$$G''(\omega)=G_1\frac{\omega\tau_c}{1+(\omega\tau_c)^2} \tag{S37}$$

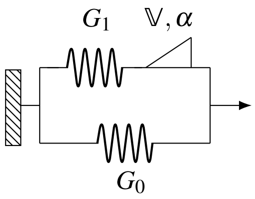
Representative plots



Supplementary Fig. 9 Representative plots of the Zener model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $G_0 = 10$ Pa, $G_1 = 10$ Pa, and $\eta = 100$ Pa s. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Zener Solid-S model (`FractionalZenerSolidS`)

Model diagram



Constitutive equation

$$\sigma(t)+\frac{\mathbb{V}}{G_1}\frac{\mathrm{d}^\alpha\sigma(t)}{\mathrm{d}t^\alpha}=G_0\gamma(t)+\mathbb{V}\frac{\mathrm{d}^\alpha\gamma(t)}{\mathrm{d}t^\alpha}+G_0\frac{\mathbb{V}}{G_1}\frac{\mathrm{d}^\alpha\gamma(t)}{\mathrm{d}t^\alpha}$$

$$\tau_c=\left(\frac{\mathbb{V}}{G_1}\right)^{\frac{1}{\alpha}} \qquad \text{(S38)}$$

$$G_c=\mathbb{V}\tau_c^{-\alpha}$$

$$G_1=G_\infty-G_0.$$

Creep compliance

$$J(t)=\frac{1}{G_0+G_1}+\frac{G_1}{G_0(G_0+G_1)}\left\{1-E_{a,b}\left[\left(\frac{G_0}{G_0+G_1}\right)z\right]\right\}$$

$$a=\alpha,\quad b=1 \qquad \text{(S39)}$$

$$z=-\left(\frac{t}{\tau_c}\right)^\alpha$$

Relaxation modulus

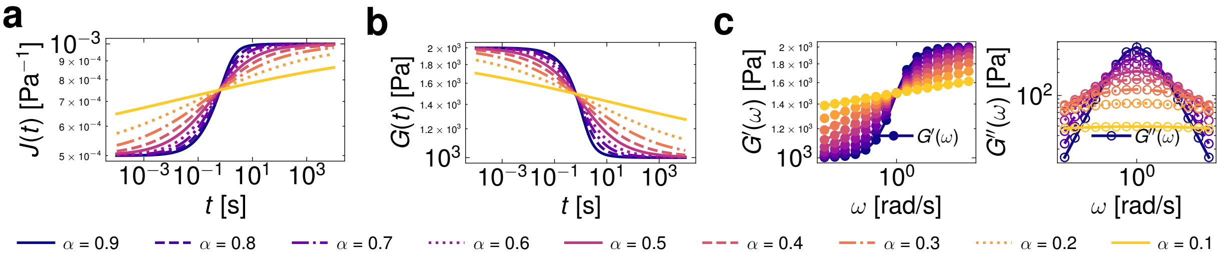
$$G(t)=G_0+G_1E_{a,b}(z) \qquad \text{(S40)}$$

Storage modulus and Loss modulus

$$G'(\omega)=G_0+\frac{(G_1)^2\mathbb{V}\omega^\alpha\cos\left(\frac{\pi}{2}\alpha\right)+(\mathbb{V}\omega^\alpha)^2G_1}{(\mathbb{V}\omega^\alpha)^2+(G_1)^2+2\mathbb{V}\omega^\alpha G_1\cos\left(\frac{\pi}{2}\alpha\right)}$$

$$G''(\omega)=\frac{(G_1)^2\mathbb{V}\omega^\alpha\sin\left(\frac{\pi}{2}\alpha\right)}{(\mathbb{V}\omega^\alpha)^2+(G_1)^2+2\mathbb{V}\omega^\alpha G_1\cos\left(\frac{\pi}{2}\alpha\right)} \qquad \text{(S41)}$$

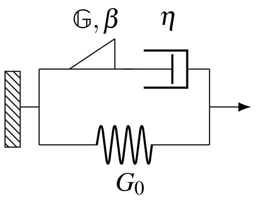
Representative plots



Supplementary Fig. 10 Representative plots of the Fractional Zener Solid-S model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $G_0 = 1000$ Pa, $\mathbb{V} = 1000$ Pa s$^\alpha$, and $G_1 = 1000$ Pa. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Zener Liquid-S model (`FractionalZenerLiquidS`)

Model diagram



Constitutive equation

$$\sigma(t)+\frac{\eta}{\mathbb{G}}\frac{\mathrm{d}^{1-\beta}\sigma(t)}{\mathrm{d}t^{1-\beta}}=G_0\gamma(t)+\eta\frac{\mathrm{d}\gamma(t)}{\mathrm{d}t}+G_0\frac{\eta}{\mathbb{G}}\frac{\mathrm{d}^{1-\beta}\gamma(t)}{\mathrm{d}t^{1-\beta}}$$
$$\tau_c=\left(\frac{\eta}{\mathbb{G}}\right)^{\frac{1}{1-\beta}} \tag{S42}$$
$$G_c=\eta\tau_c^{-1}$$

Creep compliance (Laplace form)

$$\hat{J}(s)=\frac{1}{s^2}\frac{1+\frac{\eta}{\mathbb{G}}s^{1-\beta}}{\eta+\frac{G_0}{s}+\frac{G_0\eta}{\mathbb{G}}s^{-\beta}} \tag{S43}$$

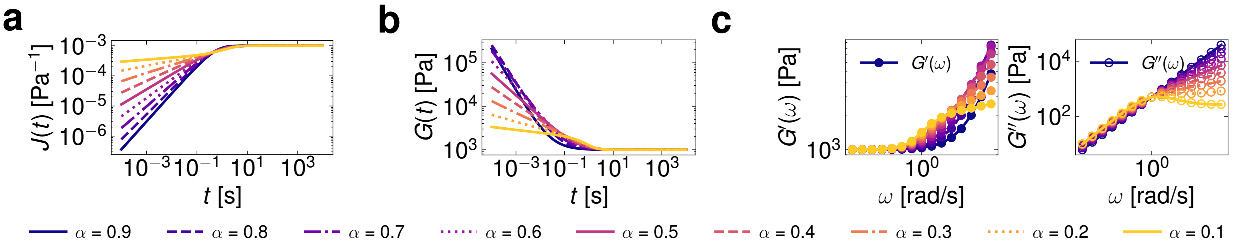
Relaxation modulus

$$G(t)=G_0+\mathbb{G}t^{-\beta}E_{a,b}(z)$$
$$a=1-\beta,\quad b=1-\beta \tag{S44}$$
$$z=-\left(\frac{t}{\tau_c}\right)^{1-\beta}$$

Storage modulus and Loss modulus

$$G'(\omega)=G_0+\frac{(\eta\omega)^2\,\mathbb{G}\omega^\beta\cos\left(\frac{\pi}{2}\beta\right)}{(\eta\omega)^2+\left(\mathbb{G}\omega^\beta\right)^2+(2\eta\omega)\,\mathbb{G}\omega^\beta\cos\left(\frac{\pi}{2}(1-\beta)\right)}$$
$$G''(\omega)=\frac{\left(\mathbb{G}\omega^\beta\right)^2(\eta\omega)+(\eta\omega)^2\,\mathbb{G}\omega^\beta\sin\left(\frac{\pi}{2}\beta\right)}{(\eta\omega)^2+\left(\mathbb{G}\omega^\beta\right)^2+(2\eta\omega)\,\mathbb{G}\omega^\beta\cos\left(\frac{\pi}{2}(1-\beta)\right)} \tag{S45}$$

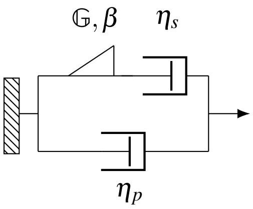
Representative plots



Supplementary Fig. 11 Representative plots of the Fractional Zener Liquid-S model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $G_0=1000$ Pa, $\mathbb{G}=1000$ Pa s$^\beta$, and $\eta=1000$ Pa s. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Zener Liquid-D model (`FractionalZenerLiquidD`)

Model diagram



Constitutive equation

$$\begin{gathered}\sigma(t) + \frac{\eta_s}{\mathbb{G}} \frac{\mathrm{d}^{1-\beta}}{\mathrm{d}t^{1-\beta}} \sigma(t) = (\eta_s + \eta_p) \frac{\mathrm{d}\gamma(t)}{\mathrm{d}t} + \frac{\eta_s \eta_p}{\mathbb{G}} \frac{\mathrm{d}^{2-\beta} \gamma(t)}{\mathrm{d}t^{2-\beta}} \\ \tau_c = \left(\frac{\eta_s}{\mathbb{G}}\right)^{\frac{1}{1-\beta}} \\ G_c = \eta_s \tau_c^{-1}\end{gathered} \tag{S46}$$

Creep compliance (Laplace form)

$$\hat{J}(s) = \frac{1}{s} \frac{\left(\eta_s s + \mathbb{G} s^\beta\right)}{\eta_s s \mathbb{G} s^\beta + \eta_p s \left(\eta_s s + \mathbb{G} s^\beta\right)} \tag{S47}$$

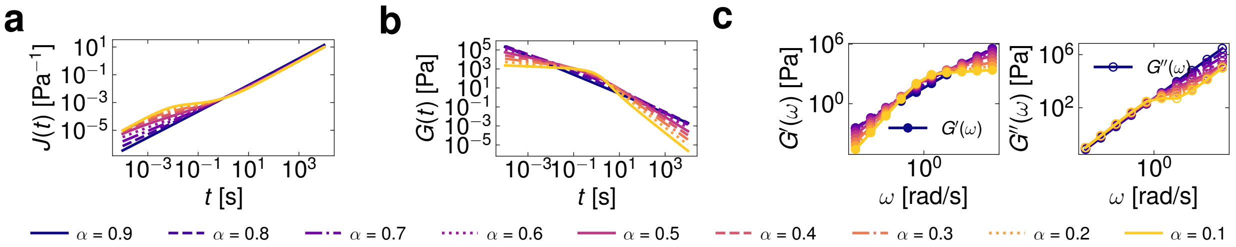
Relaxation modulus

$$\begin{gathered}G(t) = \eta_p \Delta(t) + \mathbb{G} t^{-\beta} E_{a,b}(z) \\ a = 1 - \beta, \quad b = 1 - \beta \\ z = -\left(\frac{t}{\tau_c}\right)^{1-\beta}\end{gathered} \tag{S48}$$

Storage modulus and Loss modulus

$$\begin{aligned}G'(\omega) &= \frac{\eta_s^2 \mathbb{G} \omega^{\beta+2} \cos(\frac{\pi}{2}\beta)}{(\eta_s \omega)^2 + (\mathbb{G}\omega^\beta)^2 + 2\eta_s \mathbb{G} \omega^{\beta+1} \sin(\frac{\pi}{2}\beta)} \\ G''(\omega) &= \eta_p \omega + \frac{\mathbb{G}^2 \eta_s \omega^{2\beta+1} + \eta_s^2 \mathbb{G} \omega^{\beta+2} \sin(\frac{\pi}{2}\beta)}{(\eta_s \omega)^2 + (\mathbb{G}\omega^\beta)^2 + 2\eta_s \mathbb{G} \omega^{\beta+1} \sin(\frac{\pi}{2}\beta)}\end{aligned} \tag{S49}$$

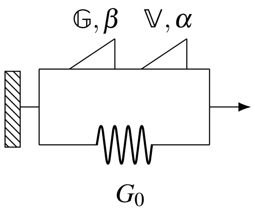
Representative plots



Supplementary Fig. 12 Representative plots of the Fractional Zener Liquid-D model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\eta_s = 1000$ Pa s, $\mathbb{G} = 1000$ Pa s$^\beta$, and $\eta_p = 10$ Pa s. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Zener-S model (`FractionalZenerS`)

Model diagram



Constitutive equation

$$\begin{gathered}\sigma(t)+\frac{\mathbb{V}}{\mathbb{G}}\frac{\mathrm{d}^{\alpha-\beta}\sigma(t)}{\mathrm{d}t^{\alpha-\beta}}=G_0\gamma(t)+\mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}}+G_0\frac{\mathbb{V}}{\mathbb{G}}\frac{\mathrm{d}^{\alpha-\beta}\gamma(t)}{\mathrm{d}t^{\alpha-\beta}}\\ \tau_c=\left(\frac{\mathbb{V}}{\mathbb{G}}\right)^{\frac{1}{\alpha-\beta}}\\ G_c=\mathbb{V}\tau_c^{-\alpha}\end{gathered}\tag{S50}$$

Creep compliance (Laplace form)

$$\hat{J}(s)=\frac{1}{s}\frac{\left(\mathbb{V}s^{\alpha}+\mathbb{G}s^{\beta}\right)}{\mathbb{V}s^{\alpha}\mathbb{G}s^{\beta}+G_0\left(\mathbb{V}s^{\alpha}+\mathbb{G}s^{\beta}\right)}\tag{S51}$$

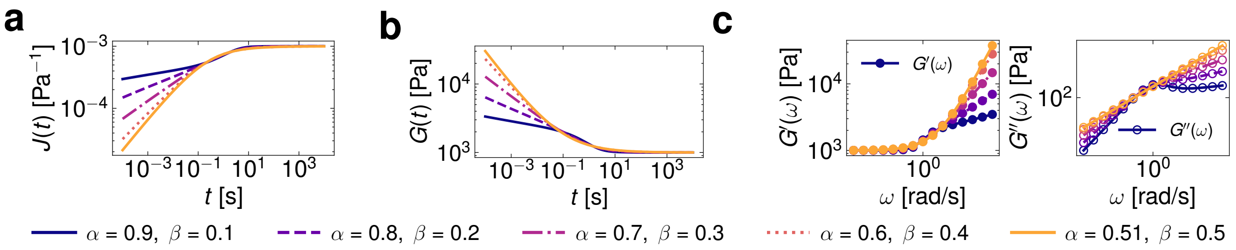
Relaxation modulus

$$\begin{gathered}G(t)=G_0+\mathbb{G}t^{-\beta}E_{a,b}(z)\\ a=\alpha-\beta,\quad b=1-\beta\\ z=-\left(\frac{t}{\tau_c}\right)^{\alpha-\beta}\end{gathered}\tag{S52}$$

Storage modulus and Loss modulus

$$\begin{gathered}G'(\omega)=G_0+\frac{(\mathbb{G}\omega^{\beta})^2(\mathbb{V}\omega^{\alpha})\cos\left(\frac{\pi}{2}\alpha\right)+(\mathbb{V}\omega^{\alpha})^2(\mathbb{G}\omega^{\beta})\cos\left(\frac{\pi}{2}\beta\right)}{(\mathbb{V}\omega^{\alpha})^2+(\mathbb{G}\omega^{\beta})^2+2(\mathbb{V}\omega^{\alpha})(\mathbb{G}\omega^{\beta})\cos\left(\frac{\pi}{2}(\alpha-\beta)\right)}\\ G''(\omega)=\frac{(\mathbb{G}\omega^{\beta})^2(\mathbb{V}\omega^{\alpha})\sin\left(\frac{\pi}{2}\alpha\right)+(\mathbb{V}\omega^{\alpha})^2(\mathbb{G}\omega^{\beta})\sin\left(\frac{\pi}{2}\beta\right)}{(\mathbb{V}\omega^{\alpha})^2+(\mathbb{G}\omega^{\beta})^2+2(\mathbb{V}\omega^{\alpha})(\mathbb{G}\omega^{\beta})\cos\left(\frac{\pi}{2}(\alpha-\beta)\right)}\end{gathered}\tag{S53}$$

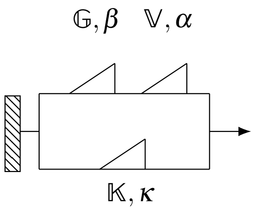
Representative plots



Supplementary Fig. 13 Representative plots of the Fractional Zener-S model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\mathbb{V} = 1000$ Pa s$^{\alpha}$, $\mathbb{G} = 1000$ Pa s$^{\beta}$, and $G_0 = 1000$ Pa. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Fractional Zener model (`FractionalZener`)

Model diagram



Constitutive equation

$$\begin{gathered}
\sigma(t)+\frac{\mathbb{V}}{\mathbb{G}}\frac{\mathrm{d}^{\alpha-\beta}\sigma(t)}{\mathrm{d}t^{\alpha-\beta}}=\mathbb{K}\frac{\mathrm{d}^{\kappa}\gamma(t)}{\mathrm{d}t^{\kappa}}+\mathbb{V}\frac{\mathrm{d}^{\alpha}\gamma(t)}{\mathrm{d}t^{\alpha}}+\mathbb{K}\frac{\mathbb{V}}{\mathbb{G}}\frac{\mathrm{d}^{\alpha+\kappa-\beta}\gamma(t)}{\mathrm{d}t^{\alpha+\kappa-\beta}}\\
\tau_c=\left(\frac{\mathbb{V}}{\mathbb{G}}\right)^{\frac{1}{\alpha-\beta}}\\
G_c=\mathbb{V}\tau_c^{-\alpha}
\end{gathered}\tag{S54}$$

Creep compliance (Laplace form)

$$\hat{J}(s)=\frac{1}{s}\frac{\left(\mathbb{V}s^{\alpha}+\mathbb{G}s^{\beta}\right)}{\mathbb{V}s^{\alpha}\mathbb{G}s^{\beta}+\mathbb{K}s^{\kappa}\left(\mathbb{V}s^{\alpha}+\mathbb{G}s^{\beta}\right)}\tag{S55}$$

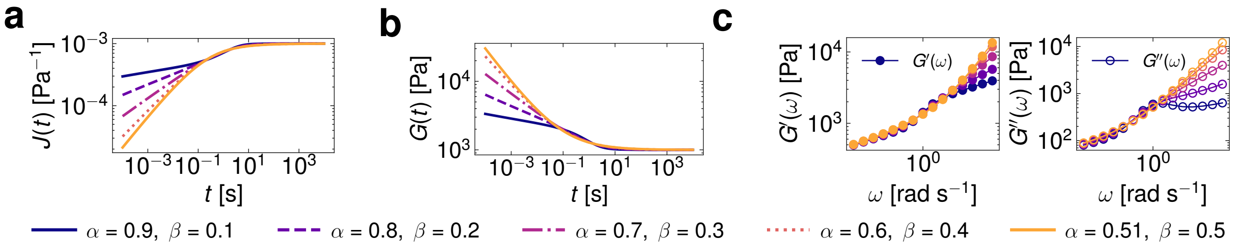
Relaxation modulus

$$\begin{gathered}
G(t)=\frac{\mathbb{K}}{\Gamma(1-\kappa)}t^{-\kappa}+\mathbb{G}t^{-\beta}E_{a,b}(z)\\
a=\alpha-\beta,\quad b=1-\beta\\
z=-\left(\frac{t}{\tau_c}\right)^{\alpha-\beta}
\end{gathered}\tag{S56}$$

Storage modulus and Loss modulus

$$\begin{gathered}
G'(\omega)=(\mathbb{K}\omega^{\kappa})\cos\left(\frac{\pi}{2}\kappa\right)+\frac{(\mathbb{G}\omega^{\beta})^2(\mathbb{V}\omega^{\alpha})\cos\left(\frac{\pi}{2}\alpha\right)+(\mathbb{V}\omega^{\alpha})^2(\mathbb{G}\omega^{\beta})\cos\left(\frac{\pi}{2}\beta\right)}{(\mathbb{V}\omega^{\alpha})^2+(\mathbb{G}\omega^{\beta})^2+2(\mathbb{V}\omega^{\alpha})(\mathbb{G}\omega^{\beta})\cos\left(\frac{\pi}{2}(\alpha-\beta)\right)}\\
G''(\omega)=(\mathbb{K}\omega^{\kappa})\sin\left(\frac{\pi}{2}\kappa\right)+\frac{(\mathbb{G}\omega^{\beta})^2(\mathbb{V}\omega^{\alpha})\sin\left(\frac{\pi}{2}\alpha\right)+(\mathbb{V}\omega^{\alpha})^2(\mathbb{G}\omega^{\beta})\sin\left(\frac{\pi}{2}\beta\right)}{(\mathbb{V}\omega^{\alpha})^2+(\mathbb{G}\omega^{\beta})^2+2(\mathbb{V}\omega^{\alpha})(\mathbb{G}\omega^{\beta})\cos\left(\frac{\pi}{2}(\alpha-\beta)\right)}
\end{gathered}\tag{S57}$$

Representative plots



Supplementary Fig. 14 Representative plots of the Fractional Zener model illustrating different viscoelastic behaviors under various types of stimulus. The plots were computed using fixed parameter values of $\mathbb{V}=1000$ Pa s$^{\alpha}$, $\mathbb{G}=1000$ Pa s$^{\beta}$, $\mathbb{K}=1000$ Pa s$^{\kappa}$, and $\kappa=0.0$. **a)** Time-dependent creep compliance $J(t)$ showing the variation of strain over time under constant stress. **b)** Stress relaxation behavior depicting a decrease in the relaxation modulus $G(t)$ under constant strain over time. **c)** Oscillatory response demonstrating the storage modulus $G'(\omega)$ and loss modulus $G''(\omega)$ under frequency-changing load.

Viscosity models

Herschel–Bulkley (`HerschelBulkley`)

$$\eta(\dot{\gamma}) = \frac{\sigma_0}{\dot{\gamma}} + k\dot{\gamma}^{n-1} \tag{S58}$$

The Herschel–Bulkley model[7] describes a yield stress fluid where $\eta(\dot{\gamma})$ is the apparent viscosity, σ_0 is the yield stress, k is the consistency index, $\dot{\gamma}$ is the shear rate, and n is the flow behavior index.

Bingham (`Bingham`)

$$\eta(\dot{\gamma}) = \frac{\sigma_0}{\dot{\gamma}} + \eta_{pl} \tag{S59}$$

The Bingham model[8] describes a fluid with a linear relationship between shear stress and shear rate beyond a yield stress σ_0, where η_{pl} is the plastic viscosity.

Power-Law (`PowerLaw`)

$$\eta(\dot{\gamma}) = k\dot{\gamma}^{n-1} \tag{S60}$$

The Power-Law model is used for non-Newtonian fluids, where $\eta(\dot{\gamma})$ is the apparent viscosity, k is the consistency index, $\dot{\gamma}$ is the shear rate, and n is the flow behavior index.

Carreau–Yasuda (`CarreauYasuda`)

$$\eta(\dot{\gamma}) = \eta_\infty + (\eta_0 - \eta_\infty)\left[1 + (k\dot{\gamma})^a\right]^{\frac{n-1}{a}} \tag{S61}$$

The Carreau–Yasuda[9] model describes shear-thinning behavior, where $\eta(\dot{\gamma})$ is the apparent viscosity, η_0 is the zero-shear viscosity, η_∞ is the infinite-shear viscosity, k is a time constant, a is a fitting parameter, and n is the power-law index.

Cross (`Cross`)

$$\eta(\dot{\gamma}) = \eta_\infty + \frac{\eta_0 - \eta_\infty}{1 + (k\dot{\gamma})^n} \tag{S62}$$

The Cross model[10] also describes shear-thinning behavior, where $\eta(\dot{\gamma})$ is the apparent viscosity, η_0 is the zero-shear viscosity, η_∞ is the infinite-shear viscosity, k is a time constant, and n is the power-law index.

Casson (`Casson`)

$$\eta(\dot{\gamma}) = \left(\frac{\sqrt{\sigma_0}}{\sqrt{\dot{\gamma}}} + \sqrt{\eta_{pl}}\right)^2 \tag{S63}$$

The Casson model[11] describes the behavior of certain yield stress fluids, where $\eta(\dot{\gamma})$ is the apparent viscosity, σ_0 is the yield stress, and η_{pl} is the plastic viscosity.

Supplementary Note 3 Multi-Layer Perceptron classifier

This section describes the training process of the Multi-Layer Perceptron (MLP) classifier for analyzing data related to creep, stress relaxation, SAOS, and steady shear. We chose the MLP classifier because of its computational efficiency. Additionally, once trained, the MLP model file is considerably smaller than files generated by Random Forest classifiers; MLP model sizes typically range from kilobytes, while Random Forest models can reach gigabytes. The latter is important to facilitate the distribution of pyRheo.

The source code for training the MLP is available on the pyRheo GitHub page, allowing users to modify it to include additional models or adjust the number of datasets used for training. Each data type has its corresponding script, with the scripts for creep, relaxation, SAOS, and steady shear being similar but tailored to fit the specific models relevant to each data type. These scripts utilize various libraries for numerical computations, data manipulation, machine learning, and data visualization, which provide the essential tools and functions needed for the tasks at hand.

Currently, the MLP is trained with the following models: Maxwell, SpringPot, FractionalMaxwellGel, FractionalMaxwellLiquid, FractionalMaxwell, and FractionalKelvinVoigt. Zener models are not included, as they overlap with the aforementioned rheological behaviors. In the future, the MLP model can be expanded to function as a multi-label classifier to incorporate Zener models.

Step 1: Generating synthetic data

To train the MLP models, we started by generating synthetic data using several predefined models, such as the Maxwell Model and the Fractional Maxwell Model. Each of these models simulates a specific type of material behavior. The task of the MLP is to classify the input data the user provides into one of these predefined models. We generated 1 million synthetic datasets for each classifier, assigning random parameter values to the model functions and time (angular frequency or shear rate) intervals.

Step 2: Scaling synthetic data

Before training the MLP, we scale each dataset to ensure optimal performance. First, we convert the dataset to a logarithmic scale. Then, we standardize it by subtracting the mean and scaling it to have a unit variance. This approach ensures that the user-provided data falls within the numerical limits of the MLP, enhancing the overall performance of the machine learning model.

Step 3: Principal Component Analysis (PCA) of the scaled synthetic data

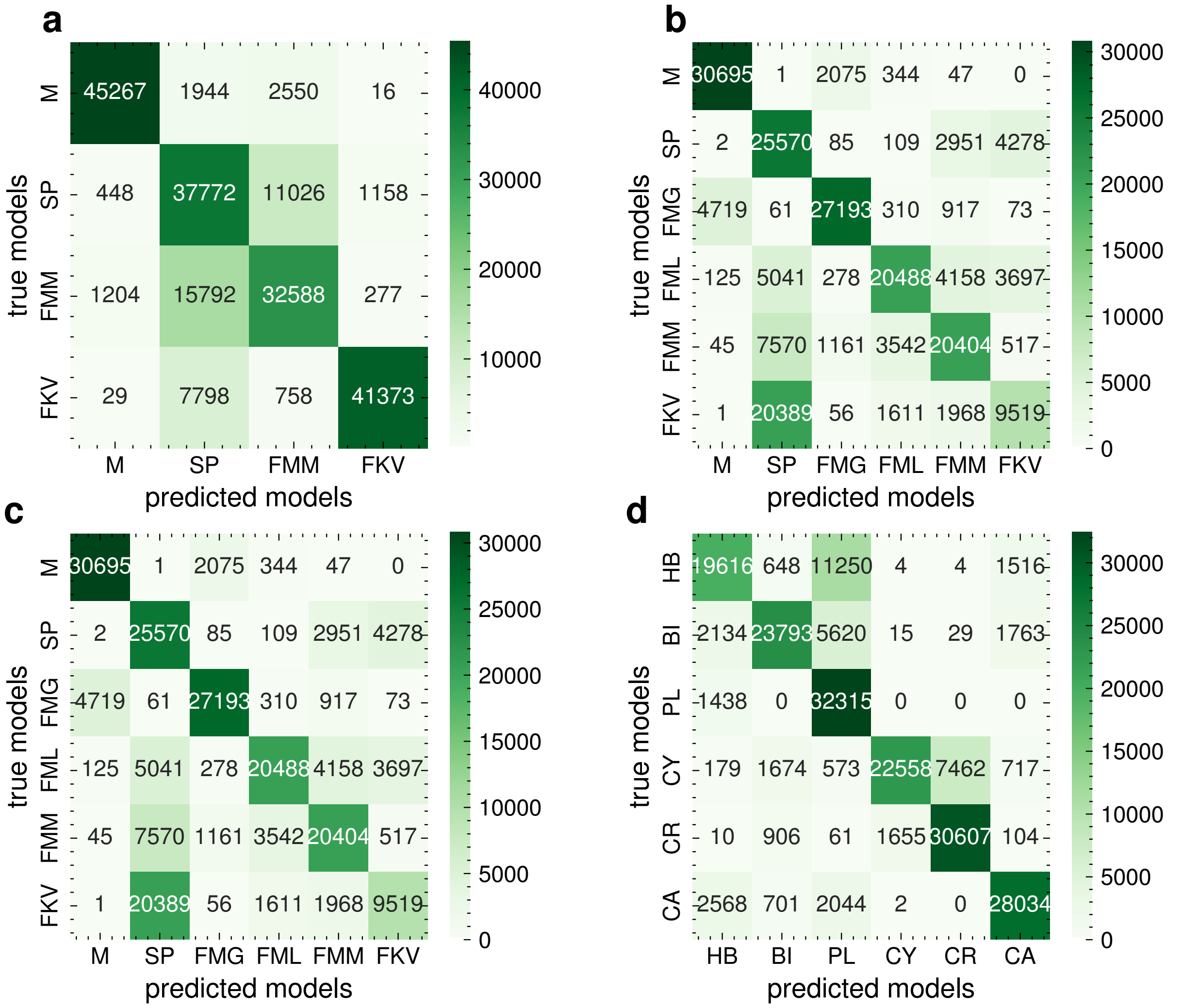
PCA is used to reduce the dimensionality of the data while retaining the most important information[12,13]. This makes it easier for machine learning models to analyze the data without the burden of using too many variables. The scaled data is transformed into a new coordinate system where the most significant features (10 principal components) are identified. This reduces the number of variables needed to describe the data while preserving important patterns.

Step 4: Training machine learning models

Machine learning models are trained to classify the different types of rheological behavior based on synthetic data. The transformed data (principal components) is split into training (80%) and testing sets (20%). With the training set, we train the MLP classifier. The trained models predict the type of rheological behavior (model type) for the testing data.

Step 5: Evaluating Model Performance

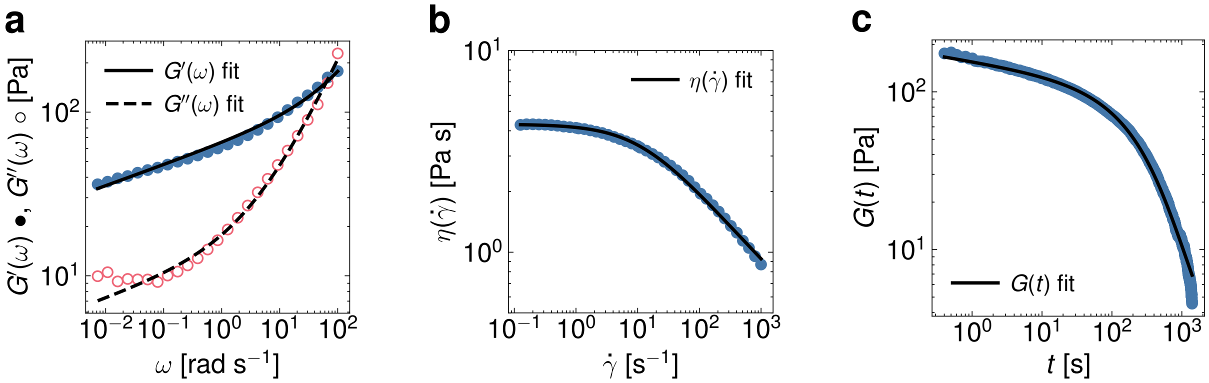
Finally, we evaluate how well the trained machine learning model can predict the type of rheological behavior on new data. The predictions made by the model using the test set are compared to the actual model types (true labels). In Supplementary Fig.15, we show the confusion matrices generated to visualize where the model performs well and makes errors. The MLP models have an accuracy of 70 to 80%. In particular, we can observed from Supplementary Fig.15 that the MLP confuses the `SpringPot` and `FractionalKelvinVoigt`..



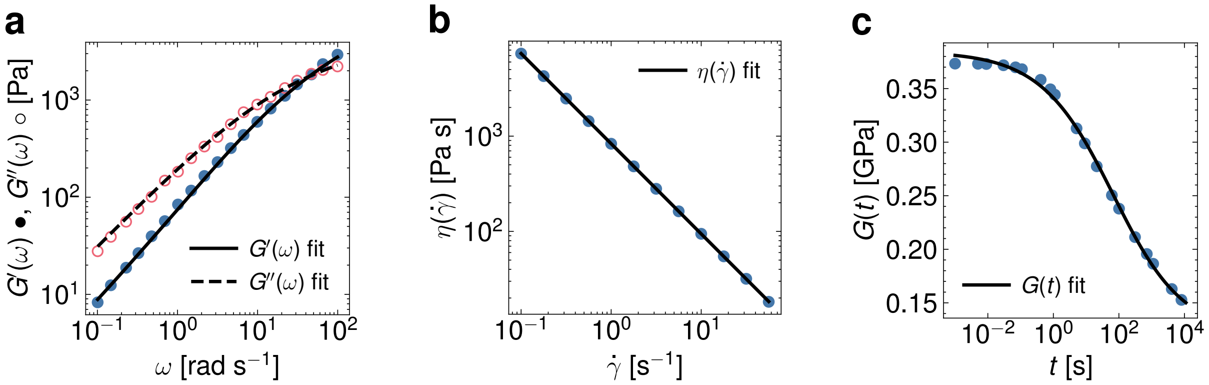
Supplementary Fig. 15 Confusion matrices illustrating the performance of the trained MLP models in predicting rheological behavior types. Each subfigure corresponds to a different rheological data class: **a)** creep data, **b)** relaxation data, **c)** SAOS data, and **d)** steady shear data. The rows represent the true labels, whereas the columns denote the predicted labels. The models achieve an accuracy between 70% and 80%.

Supplementary Note 4 Complementary examples of predicting and fitting tasks with pyRheo

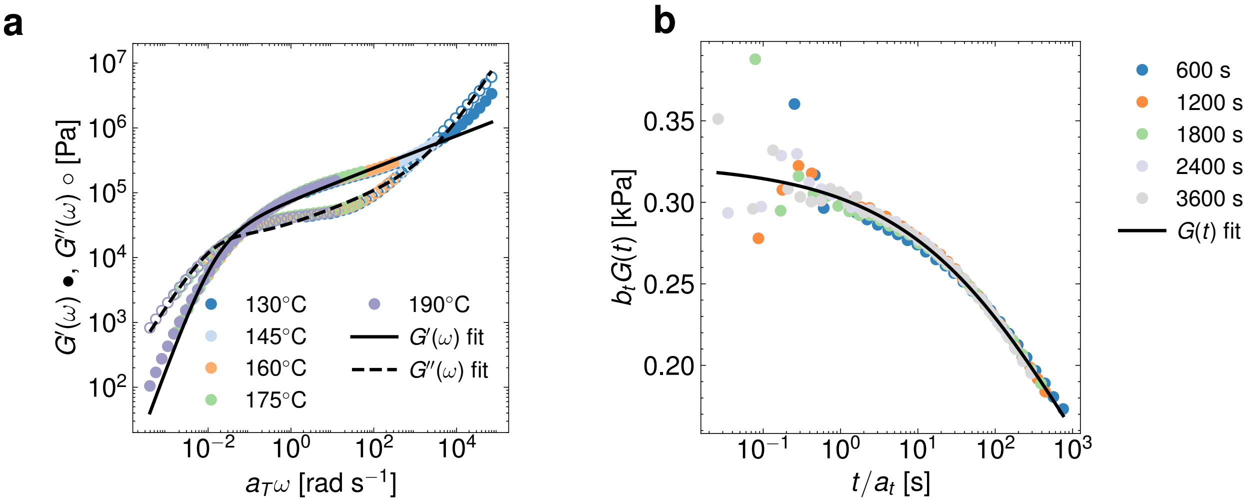
We show in Supplementary Fig.16, Supplementary Fig.17, Supplementary Fig.18, Supplementary Fig.19, Supplementary Fig.20, and Supplementary Fig.21 examples of fitting with pyRheo. The parameter values of each model can be found in their corresponding Jupyter Notebook.



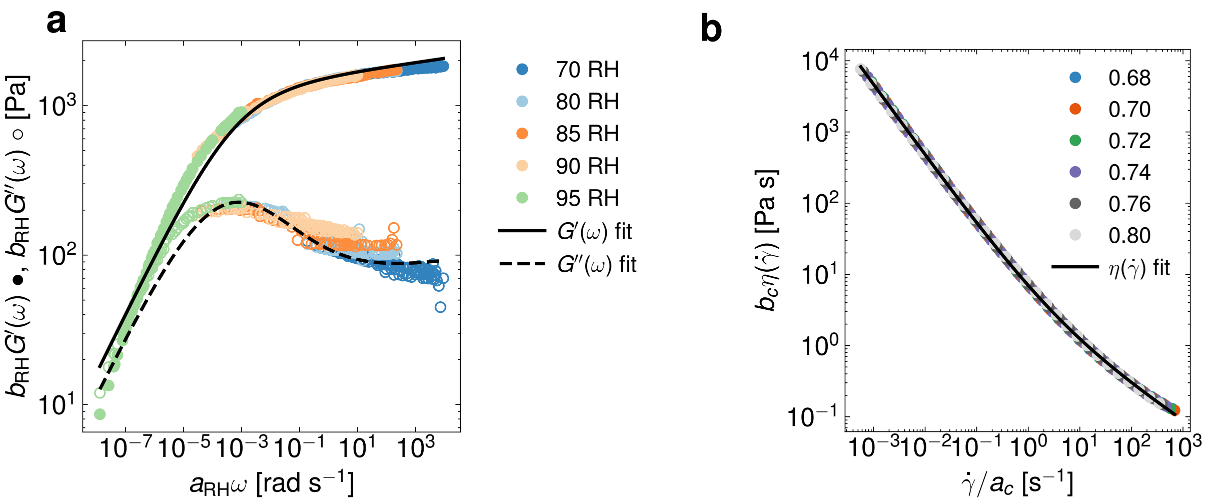
Supplementary Fig. 16 Fitting results obtained using pyRheo. **a)** Storage and Loss modulus as a function of angular frequency $G'(\omega)$, $G''(\omega)$ measured for a chia pudding fitted with `FractionalKelvinVoigt`. **b)** Shear viscosity as function of shear rate $\eta(\dot{\gamma})$ measured for an ethylcellulose solution in toluene fitted with `CarreauYasuda`. **c)** Relaxation modulus as function of time $G(t)$ measured for a shaving foam fitted with `FractionalMaxwell` (data from Lavergne *et al.*[14]).



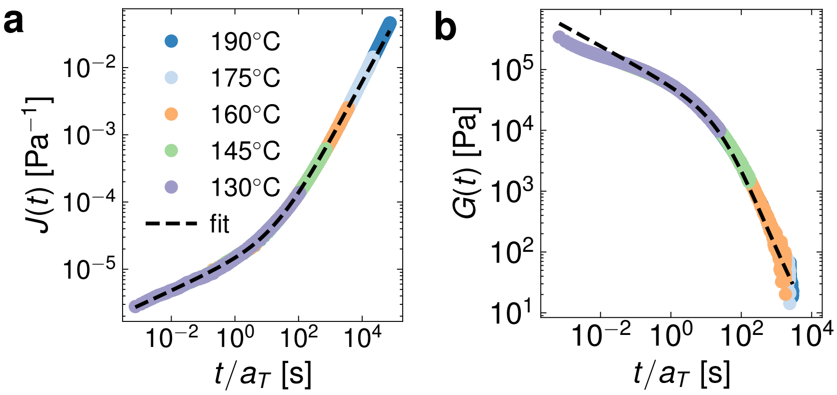
Supplementary Fig. 17 Fitting results obtained using pyRheo. **a)** Storage and Loss modulus as a function of angular frequency $G'(\omega)$, $G''(\omega)$ measured for a metal-coordinating polymer network fitted with `FractionalMaxwell` (data from Epstein *et al.*[15]). **b)** Shear viscosity as function of shear rate $\eta(\dot{\gamma})$ measured for a cellulose nanofiber hydrogel fitted with `PowerLaw` (data from Miranda-Valdez *et al.*[16]). **c)** Relaxation modulus as function of time $G(t)$ measured for a polypropylene sample fitted with `FractionalZenerSolidS`.



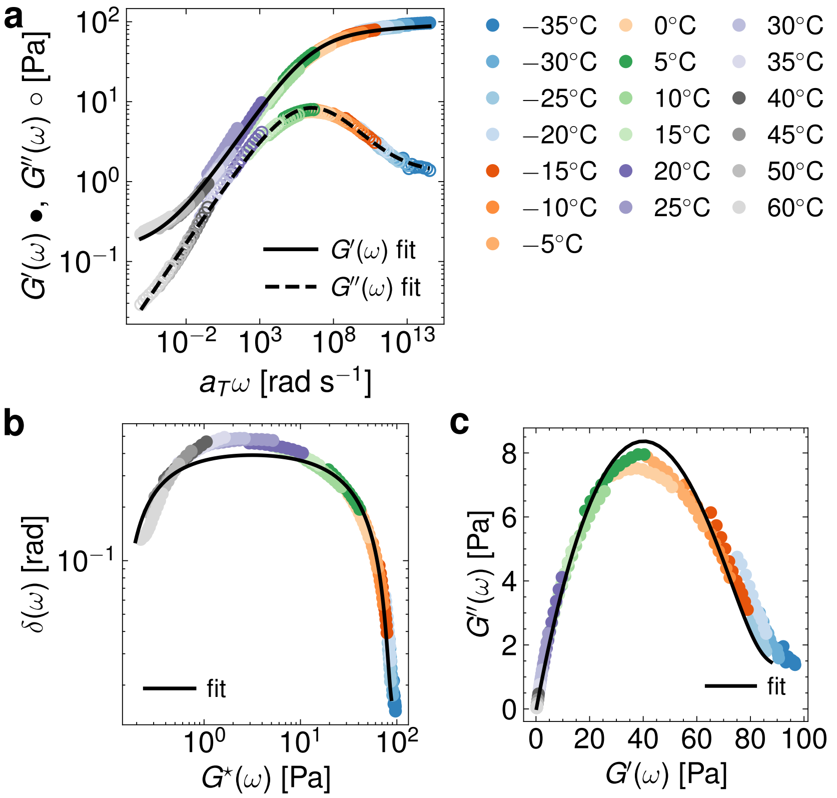
Supplementary Fig. 18 Fitting results obtained using pyRheo. **a)** Storage and Loss modulus master curve as a function of angular frequency $G'(\omega)$, $G''(\omega)$ measured for polystyrene ($T_{\mathrm{ref}} = 160°\mathrm{C}$) fitted with `FractionalZenerLiquidD` (data from Ricarte and Shanbhag[17]). **b)** Relaxation modulus master curve as function of time $G(t)$ measured for a laponite with different aging times ($t_{\mathrm{ref}} = 600$ s) fitted with `FractionalMaxwellGel` (data from Lennon *et al.*[18]).



Supplementary Fig. 19 Fitting results obtained using pyRheo. **a)** Storage and Loss modulus master curve as a function of angular frequency $G'(\omega)$, $G''(\omega)$ measured for a viscoelastic material at different relative humidity (RH) ($\mathrm{RH}_{\mathrm{ref}} = 90$) fitted with `FractionalMaxwell` (data from Lennon *et al.*[18]). **b)** Viscosity master curve as function of shear rate $\eta(\dot{\gamma})$ measured for a castor oil emulsion with different oil contents ($c_{\mathrm{ref}} = 0.68$) fitted with `HerschelBulkley` (the dataset was taken from Lennon *et al.*[18] who adapted it from Dekker *et al.*[19]).



Supplementary Fig. 20 Master curves for the linear viscoelastic behavior of polystyrene at various temperatures. **a)** Creep compliance $J(t)$ master curve ($T_{\mathrm{ref}} = 160°\mathrm{C}$) constructed using the time-temperature superposition (TTS). The curve is fitted using the `auto` function in pyRheo, which classifies the data as a `FractionalMaxwell`. **b)** Relaxation modulus $G(t)$ master curve classified and fitted with `FractionalMaxwell`. The raw data was reproduced from Ricarte and Shanbhag[17].



Supplementary Fig. 21 Master curves for the linear viscoelastic behavior of a vinyl nitrile solid foam at various temperatures. **a)** Storage modulus $G'(\omega)$ and Loss modulus $G''(\omega)$ master curves ($T_{\mathrm{ref}} = 25°\mathrm{C}$) constructed using the time-temperature superposition (TTS) and fitted with `FractionalZenerS`, constituted by a `FractionalMaxwell` connected in parallel to a spring. **b)** Van Gurp–Palmen plot and **c)** Cole–Cole representation of the master curve. The raw data was reproduced from Landauer *et al.*[20].

Supplementary Note 5 Mittag–Leffler function in pyRheo

Global Padé approximations

In pyRheo, we have implemented the second-order global Padé approximation by Zeng and Chen[4] and fourth-order global Padé approximations published by Sarumi *et al.*[5].

Second-order approximation

The second-order approximation is of the type $R^{3,2}_{a,b}(x)$. The approximation is given by

$$R^{3,2}_{a,b}(-x) = \frac{1}{\Gamma(b-a)}\frac{p_1 + x}{q_0 + q_1 x + x^2}, \qquad b > a, \tag{S64}$$

with

$$\begin{aligned} p_1 &= c_{a,b}\left[\Gamma(b)\Gamma(b+a) - \frac{\Gamma(b+a)\Gamma^2(b-a)}{\Gamma(b-2a)}\right], \\ q_0 &= c_{a,b}\left[\frac{\Gamma^2(b)\Gamma(b+a)}{\Gamma(b-a)} - \frac{\Gamma(b)\Gamma(b+a)\Gamma(b-a)}{\Gamma(b-2a)}\right], \\ q_1 &= c_{a,b}\left[\Gamma(b)\Gamma(b+a) - \frac{\Gamma^2(b)\Gamma(b-a)}{\Gamma(b-2a)}\right], \\ c_{a,b} &= \frac{1}{\Gamma(b+a)\Gamma(b-a) - \Gamma^2(b)}, \end{aligned} \tag{S65}$$

and

$$R^{3,2}_{a,a}(-x) = \frac{a}{\Gamma(1+a) + \frac{2\Gamma(1-a)^2}{\Gamma(1-2a)}x + \Gamma(1-a)x^2}, \qquad 0 < a < 1. \tag{S66}$$

Fourth-order approximation

The fourth-order global Padé approximation ($\nu = 4$) correspond to the types (m,n) with $m+n=9$. They include the types $(5,4)$, $(6,3)$, and $(7,2)$. pyRheo uses the approximation $R^{m,n}_{a,b}$ for $\nu = 4$, which takes the form

$$R^{m,n}_{a,b}(-x) = \begin{cases} \dfrac{1}{\Gamma(b-a)}\dfrac{p_1 + p_2 x + p_3 x^2 + x^3}{q_0 + q_1 x + q_2 x^2 + q_3 x^3 + x^4}, & b > a, \\ \\ \dfrac{-1}{\Gamma(-a)}\dfrac{\hat{p}_2 + \hat{p}_3 x + x^2}{\hat{q}_0 + \hat{q}_1 x + \hat{q}_2 x^2 + \hat{q}_3 x^3 + x^4}, & b = a. \end{cases} \tag{S67}$$

The unknown coefficients are obtained by solving the following systems. For $b > a$, the coefficients satisfy the system

Coefficients of $R^{5,4}_{a,b}$

$$\begin{bmatrix} 1 & 0 & 0 & -\dfrac{\Gamma(b-a)}{\Gamma(b)} & 0 & 0 & 0 \\ 0 & 1 & 0 & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b)} & 0 & 0 \\ 0 & 0 & 1 & -\dfrac{\Gamma(b-a)}{\Gamma(b+2a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b)} & 0 \\ 0 & 0 & 0 & \dfrac{\Gamma(b-a)}{\Gamma(b+3a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b+2a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b)} \\ 1 & 0 & 0 & 0 & -1 & \dfrac{\Gamma(b-a)}{\Gamma(b-2a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b-3a)} \\ 0 & 1 & 0 & 0 & 0 & -1 & \dfrac{\Gamma(b-a)}{\Gamma(b-2a)} \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -\dfrac{\Gamma(b-a)}{\Gamma(b-4a)} \\ \dfrac{\Gamma(b-a)}{\Gamma(b-3a)} \\ -\dfrac{\Gamma(b-a)}{\Gamma(b-2a)} \end{pmatrix}. \tag{S68}$$

For $b = a$ the coefficients satisfy the system

$$\begin{bmatrix} 1 & 0 & \frac{\Gamma(-a)}{\Gamma(a)} & 0 & 0 & 0 \\ 0 & 1 & -\frac{\Gamma(-a)}{\Gamma(2a)} & \frac{\Gamma(-a)}{\Gamma(a)} & 0 & 0 \\ 0 & 0 & \frac{\Gamma(-a)}{\Gamma(3a)} & -\frac{\Gamma(-a)}{\Gamma(2a)} & \frac{\Gamma(-a)}{\Gamma(a)} & 0 \\ 0 & 0 & 0 & -1 & -\frac{\Gamma(-a)}{\Gamma(-2a)} & 0 \\ 1 & 0 & 0 & 0 & -1 & \frac{\Gamma(-a)}{\Gamma(-2a)} \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \hat{p}_2 \\ \hat{p}_3 \\ \hat{q}_0 \\ \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ -\frac{\Gamma(-a)}{\Gamma(-2a)} \end{pmatrix}. \qquad \text{(S69)}$$

Coefficients of $R^{6,3}_{a,b}$

For $b > a$, the coefficients satisfy the system

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{\Gamma(b-a)}{\Gamma(b)} & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{\Gamma(b-a)}{\Gamma(b+a)} & -\frac{\Gamma(b-a)}{\Gamma(b)} & 0 & 0 \\ 0 & 0 & 1 & -\frac{\Gamma(b-a)}{\Gamma(b+2a)} & \frac{\Gamma(b-a)}{\Gamma(b+a)} & -\frac{\Gamma(b-a)}{\Gamma(b)} & 0 \\ 0 & 0 & 0 & \frac{\Gamma(b-a)}{\Gamma(b+3a)} & -\frac{\Gamma(b-a)}{\Gamma(b+2a)} & \frac{\Gamma(b-a)}{\Gamma(b+a)} & -\frac{\Gamma(b-a)}{\Gamma(b)} \\ 0 & 0 & 0 & -\frac{\Gamma(b-a)}{\Gamma(b+4a)} & \frac{\Gamma(b-a)}{\Gamma(b+3a)} & -\frac{\Gamma(b-a)}{\Gamma(b+2a)} & \frac{\Gamma(b-a)}{\Gamma(b+a)} \\ 0 & 1 & 0 & 0 & 0 & -1 & \frac{\Gamma(b-a)}{\Gamma(b-2a)} \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ \frac{\Gamma(b-a)}{\Gamma(b)} \\ \frac{\Gamma(b-a)}{\Gamma(b-3a)} \\ -\frac{\Gamma(b-a)}{\Gamma(b-2a)} \end{pmatrix}, \qquad \text{(S70)}$$

For $b = a$, the coefficients satisfy the system

$$\begin{bmatrix} 1 & 0 & \frac{\Gamma(-a)}{\Gamma(a)} & 0 & 0 & 0 \\ 0 & 1 & -\frac{\Gamma(-a)}{\Gamma(2a)} & \frac{\Gamma(-a)}{\Gamma(a)} & 0 & 0 \\ 0 & 0 & \frac{\Gamma(-a)}{\Gamma(3a)} & -\frac{\Gamma(-a)}{\Gamma(2a)} & \frac{\Gamma(-a)}{\Gamma(a)} & 0 \\ 0 & 0 & -\frac{\Gamma(-a)}{\Gamma(4a)} & \frac{\Gamma(-a)}{\Gamma(3a)} & -\frac{\Gamma(-a)}{\Gamma(2a)} & \frac{\Gamma(-a)}{\Gamma(a)} \\ 1 & 0 & 0 & 0 & -1 & \frac{\Gamma(-a)}{\Gamma(-2a)} \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \hat{p}_2 \\ \hat{p}_3 \\ \hat{q}_0 \\ \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ \frac{\Gamma(-a)}{\Gamma(-3a)} \\ -\frac{\Gamma(-a)}{\Gamma(-2a)} \end{pmatrix}. \qquad \text{(S71)}$$

Coefficients of $R^{7,2}_{a,b}$

For $b > a$, the coefficients satisfy the system

$$\begin{bmatrix} 1 & 0 & 0 & -\dfrac{\Gamma(b-a)}{\Gamma(b)} & 0 & 0 & 0 \\ 0 & 1 & 0 & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b)} & 0 & 0 \\ 0 & 0 & 1 & -\dfrac{\Gamma(b-a)}{\Gamma(b+2a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b)} & 0 \\ 0 & 0 & 0 & \dfrac{\Gamma(b-a)}{\Gamma(b+3a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b+2a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b)} \\ 0 & 0 & 0 & -\dfrac{\Gamma(b-a)}{\Gamma(b+4a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+3a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b+2a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+a)} \\ 0 & 0 & 0 & \dfrac{\Gamma(b-a)}{\Gamma(b+5a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b+4a)} & \dfrac{\Gamma(b-a)}{\Gamma(b+3a)} & -\dfrac{\Gamma(b-a)}{\Gamma(b+2a)} \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ \dfrac{\Gamma(b-a)}{\Gamma(b)} \\ -\dfrac{\Gamma(b-a)}{\Gamma(b+a)} \\ -\dfrac{\Gamma(b-a)}{\Gamma(b-2a)} \end{pmatrix}. \tag{S72}$$

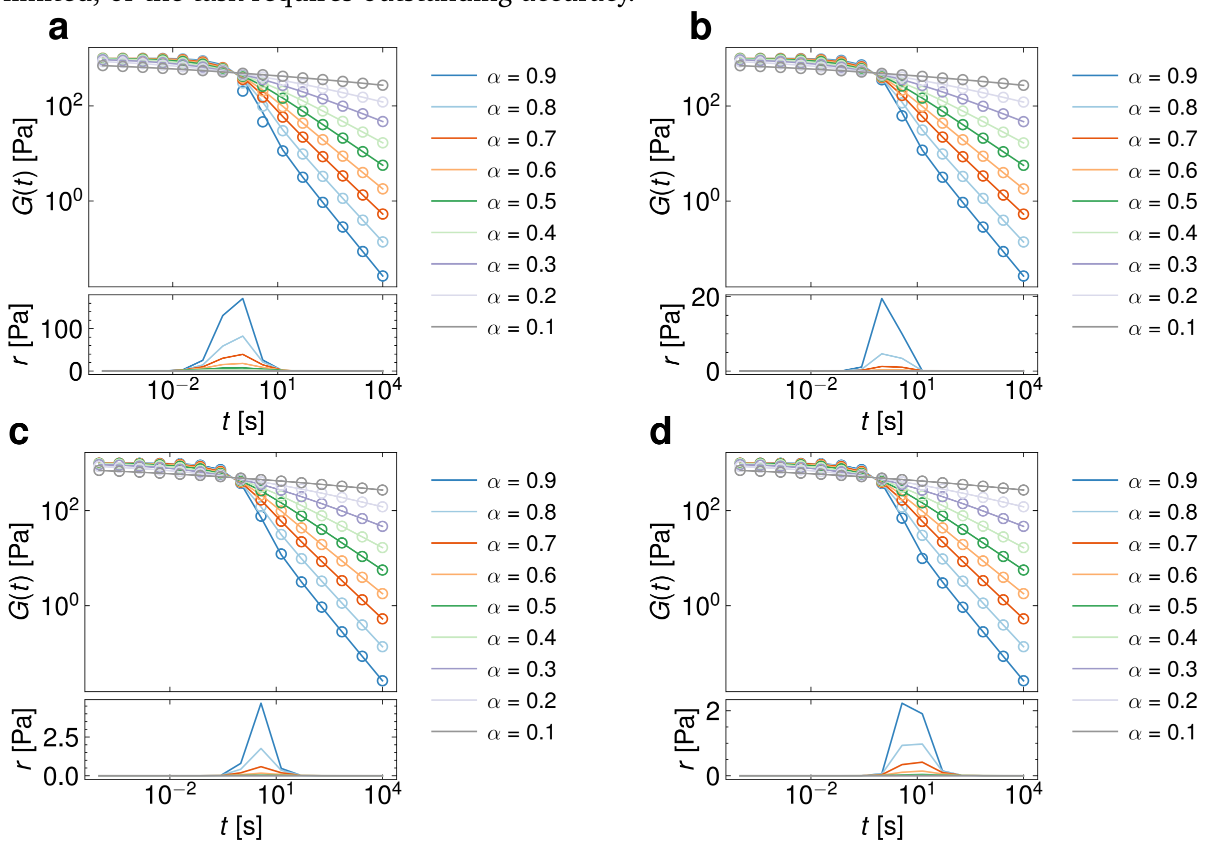
For $b = a$, the coefficients satisfy the system

$$\begin{bmatrix} 1 & 0 & \dfrac{\Gamma(-a)}{\Gamma(a)} & 0 & 0 & 0 \\ 0 & 1 & -\dfrac{\Gamma(-a)}{\Gamma(2a)} & \dfrac{\Gamma(-a)}{\Gamma(a)} & 0 & 0 \\ 0 & 0 & \dfrac{\Gamma(-a)}{\Gamma(3a)} & -\dfrac{\Gamma(-a)}{\Gamma(2a)} & \dfrac{\Gamma(-a)}{\Gamma(a)} & 0 \\ 0 & 0 & -\dfrac{\Gamma(-a)}{\Gamma(4a)} & \dfrac{\Gamma(-a)}{\Gamma(3a)} & -\dfrac{\Gamma(-a)}{\Gamma(2a)} & -\dfrac{\Gamma(-a)}{\Gamma(a)} \\ 0 & 0 & \dfrac{\Gamma(-a)}{\Gamma(5a)} & -\dfrac{\Gamma(-a)}{\Gamma(4a)} & \dfrac{\Gamma(-a)}{\Gamma(3a)} & -\dfrac{\Gamma(-a)}{\Gamma(2a)} \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \hat{p}_2 \\ \hat{p}_3 \\ \hat{q}_0 \\ \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ \dfrac{\Gamma(-a)}{\Gamma(-3a)} \\ -\dfrac{\Gamma(-a)}{\Gamma(-2a)} \end{pmatrix}. \tag{S73}$$

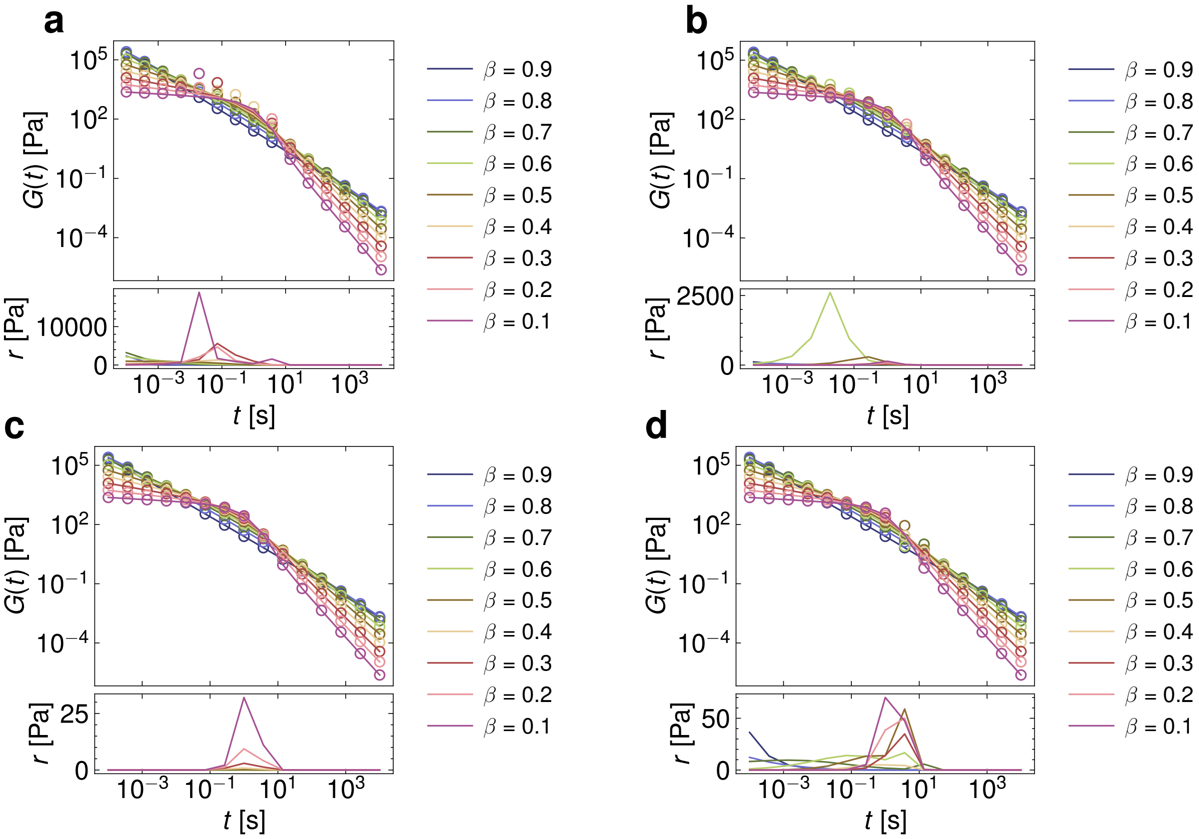
Supplementary Note 6 Comparison between global Padé approximations and Garrappa's algorithm

pyRheo also provides users with the option to use Garrappa's algorithm[6] for computing the Mittag–Leffler function. The algorithm calculates the Mittag–Leffler function based on the numerical inversion of its Laplace transform (LT). An optimal parabolic contour is chosen based on the distance to and strength of the singularities of the Laplace Transform, which aims to minimize computational effort and reduce error propagation. Next, we show how the Padé approximations perform compared to Garrappa's algorithm. In Supplementary Fig. 22, Supplementary Fig. 23, and Supplementary Fig. 24, we simply compute Eq. S12, Eq. S16, and Eq. S20 respectively and show the difference in the stress relaxation $G(t)$ function when using Padé approximations and Garrappa's algorithm.

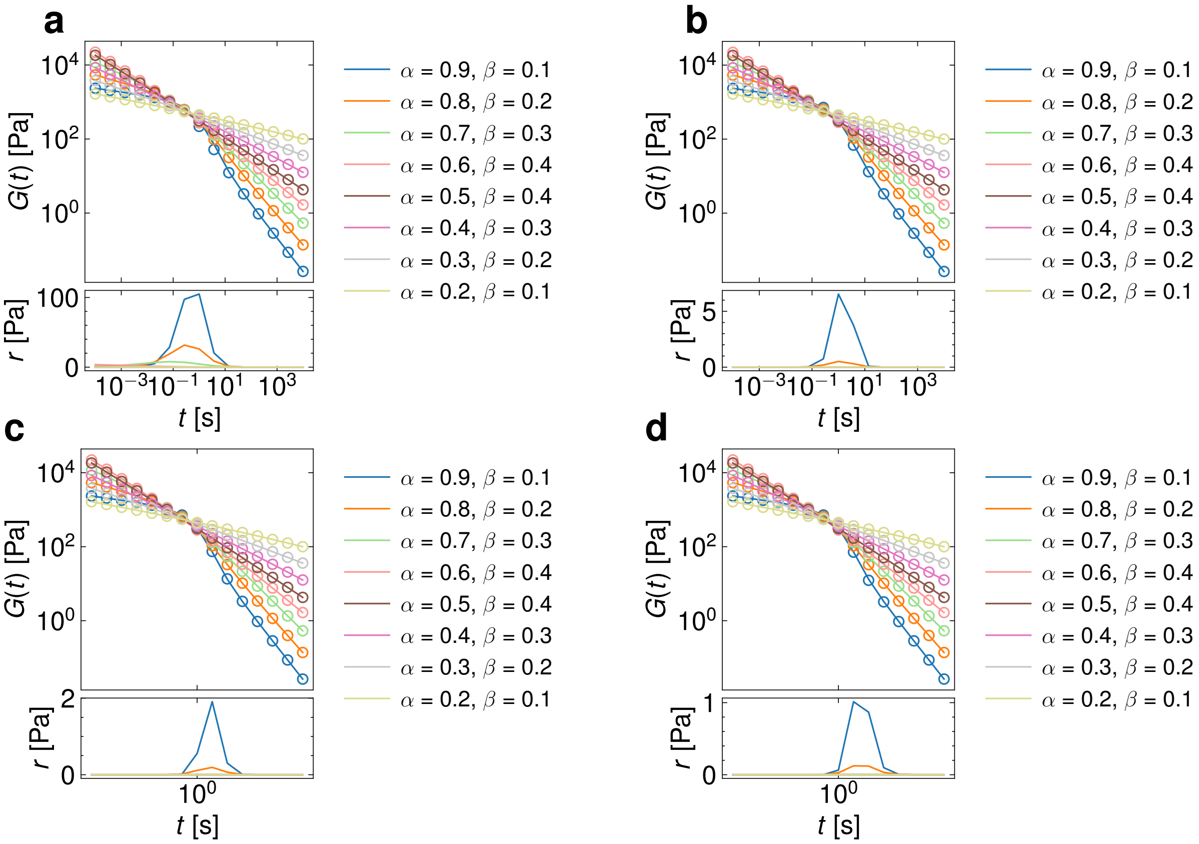
As seen in Supplementary Fig.22, Supplementary Fig. 23, and Supplementary Fig. 24, the Padé approximation shows a residual r peak when the Fractional Maxwell models approach traditional Maxwellian behaviors. The residual, at the same time, reduces its intensity when using more comprehensive Padé approximations. The user might consider using Garrappa's implementation of the Mittag–Leffler function when access to computational resources is not limited, or the task requires outstanding accuracy.



Supplementary Fig. 22 Stress relaxation functions $G(t)$ computed for different fractional orders α using the `FractionalMaxwellGel`. The solid lines represent the functions computed using Garrappa's algorithm[6], while the empty markers represent those computed using the global Padé approximation[4,5] **a)** $R^{3,2}_{a,b}$, **b)** $R^{5,4}_{a,b}$, **c)** $R^{6,3}_{a,b}$, and **d)** $R^{7,2}_{a,b}$. Each figure also includes a residual r plot, highlighting the main differences between both approaches. All the functions were computed by assigning a value of 1000 to $\mathbb{V}$ and G.



Supplementary Fig. 23 Stress relaxation functions $G(t)$ computed for different fractional orders β using the `FractionalMaxwellLiquid`. The solid lines represent the functions computed using Garrappa's algorithm[6], while the empty markers represent those computed using the global Padé approximation[4,5] **a)** $R^{3,2}_{a,b}$, **b)** $R^{5,4}_{a,b}$, **c)** $R^{6,3}_{a,b}$, and **d)** $R^{7,2}_{a,b}$. Each figure also includes a residual r plot, highlighting the main differences between both approaches. All the functions were computed by assigning a value of 1000 to $\mathbb{G}$ and η.



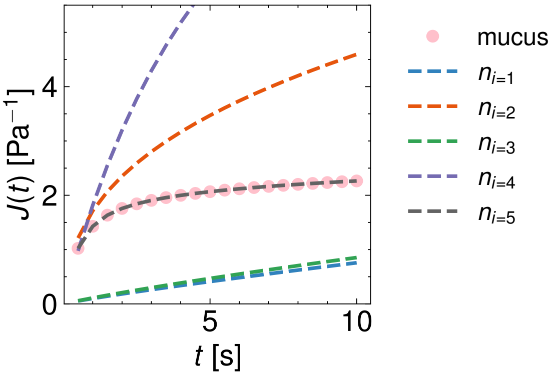
Supplementary Fig. 24 Stress relaxation functions $G(t)$ computed for different fractional orders α and β using the `FractionalMaxwell`. The solid lines represent the functions computed using Garrappa's algorithm[6], while the empty markers represent those computed using the global Padé approximation[4,5] **a)** $R^{3,2}_{a,b}$, **b)** $R^{5,4}_{a,b}$, **c)** $R^{6,3}_{a,b}$, and **d)** $R^{7,2}_{a,b}$. Each figure also includes a residual r plot, highlighting the main differences between both approaches. All the functions were computed by assigning a value of 1000 to $\mathbb{V}$ and $\mathbb{G}$.

Supplementary Note 7 Initial guesses with Bayesian optimization

Due to the sensitivity of fitting routines to initial parameter guesses, pyRheo employs three methodologies to help prevent convergence to local minima during parameter optimization. The first method involves using `manual` initial guesses, which require user expertise. The second method is to restart the fitting process using `random` initial guesses for a fixed number of iterations. In each iteration, pyRheo attempts to minimize the weighted residual sum of squares (RSS_{w_i}) and ultimately selects the best iteration, based on the lowest RSS_{w_i}. The `random` method is robust. For example, the fittings in the main manuscript were generated with a maximum of 10 iterations.

A third method that pyRheo offers for defining initial guesses is Bayesian Optimization (BO)[21–23]. In this approach, pyRheo creates a mapping from the parameter space $\mathscr{P}$ to the error space $\mathscr{E}$ using Gaussian Process Regression (GPR), represented as $g : \mathscr{P} \to \mathscr{E}$, where g is the Gaussian Process. The surrogate model $\varepsilon = g(p)$ (with $\varepsilon \in \mathscr{E}$ and $p \in \mathscr{P}$) is developed by computing the constitutive equation of the target model with fixed parameter values and then recording the difference (residuals) between this computation and the data being analyzed. The goal of BO is to minimize ε by exploring various combinations of parameter values, guided by an acquisition function known as Expected Improvement, which balances exploration and exploitation of the parameter space[24]. Afterward, pyRheo uses the BO solution as the initial guess for the minimization algorithm.

In Supplementary Fig. 25, we show how the BO works when used as a method to define the initial guesses. In the case of Supplementary Fig. 25, we fit the creep data of a mucus gel[25] using a `FractionalKelvinVoigtD` model. The Bayesian optimization learns from every iteration n_i and, in the end, finds the best combination of parameter values, which pyRheo uses to initialize the final minimization algorithm.



Supplementary Fig. 25 Iterative fitting with Bayesian Optimization (BO) as a method to define initial guesses in the `FractionalKelvinVoigtD` model when fitting the creep data of a mucus gel. Every iteration n_i in the BO process refines the understanding of the parameter space, effectively guiding the selection of the optimal parameter combination. The approach uses the Expected Improvement acquisition function to explore and minimize the error space ε. The final best combination of parameters is then used as the initial guess for subsequent minimization algorithms. The BO method shows significant effectiveness in preventing convergence to local minima, improving the robustness of parameter optimization.

Supplementary Note 8 Graphical user interface (GUI) tutorial

This tutorial provides step-by-step instructions on how to run a Python file using Anaconda on both Windows and Linux. The GUI for pyRheo is launched by executing either `pyRheo_creep_GUI.py`, `pyRheo_relaxation_GUI.py`, `pyRheo_oscillation_GUI.py`, or `pyRheo_rotation_GUI.py`. We note that for simplicity, the GUI works only using `auto` bounds and `random` initial guesses.

Prerequisites

Before getting started, ensure that you meet all the necessary requirements for using pyRheo. A recommended approach is to use Anaconda for managing dependencies and environments. Download Anaconda and follow the installation instructions suitable for your operating system.

Step 1: Clone the GitHub repository

Download pyRheo package from https://github.com/mirandi1/pyRheo.

Windows and Linux

1. Open a terminal (Windows: Anaconda Prompt or Command Prompt; Linux: Terminal).
2. Change the directory to where you want to clone the repository:

```
cd /path/to/your/directory
```

3. Clone the GitHub repository using `git` (assuming Git is installed):

```
git clone https://github.com/mirandi1/pyRheo
```

4. Navigate into the cloned repository and to the GUI folder:

```
cd pyRheo/gui
```

Step 2: Run the Python file

Execute the specific Python file. For example, in this tutorial, we execute `pyRheo_creep_GUI.py`.

Windows and Linux

1. Ensure you are in the correct directory:

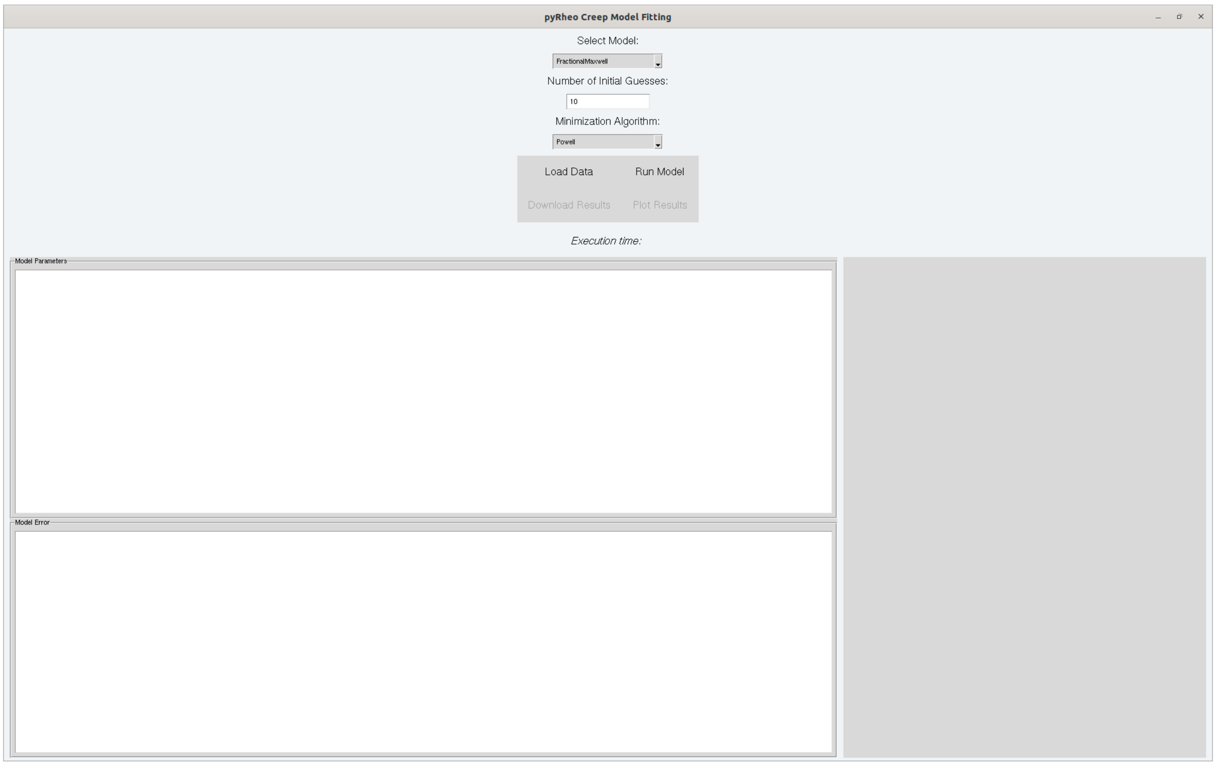
```
cd pyRheo/gui
```

2. Run the Python file:

```
python pyRheo_creep_GUI.py
```

3. The GUI should prompt as a new window, see Supplementary Fig.26

Upon running the command, the GUI window titled "pyRheo Creep Model Fitting" will appear.



Select Model: FractionalMaxwell

Number of Initial Guesses: 10

Minimization Algorithm: Powell

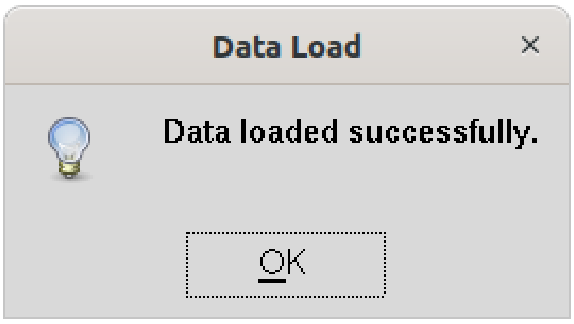
Supplementary Fig. 26 Screenshot of the pyRheo GUI - Initial screen.

Step 3: Using the GUI

Loading data

1. Click on the "**Load Data**" button to select the data file to use. The data file should be a CSV file with columns with header "Time" and "Creep Compliance", followed by their corresponding data. The user can look at the file `creep_ps190_data.csv` to understand the requirements in the file structure before loading. The delimiter between columns must be "tab", and the decimal must be ".". Column headers in case of relaxation: "Time" and "Relaxation Modulus". Column headers in case of SAOS: "Angular Frequency" and "Storage Modulus", and "Loss Modulus". Column headers in case of steady shear: "Shear Rate" and "Viscosity".

2. After successfully loading the data, a message box will notify you (Supplementary Fig. 27).

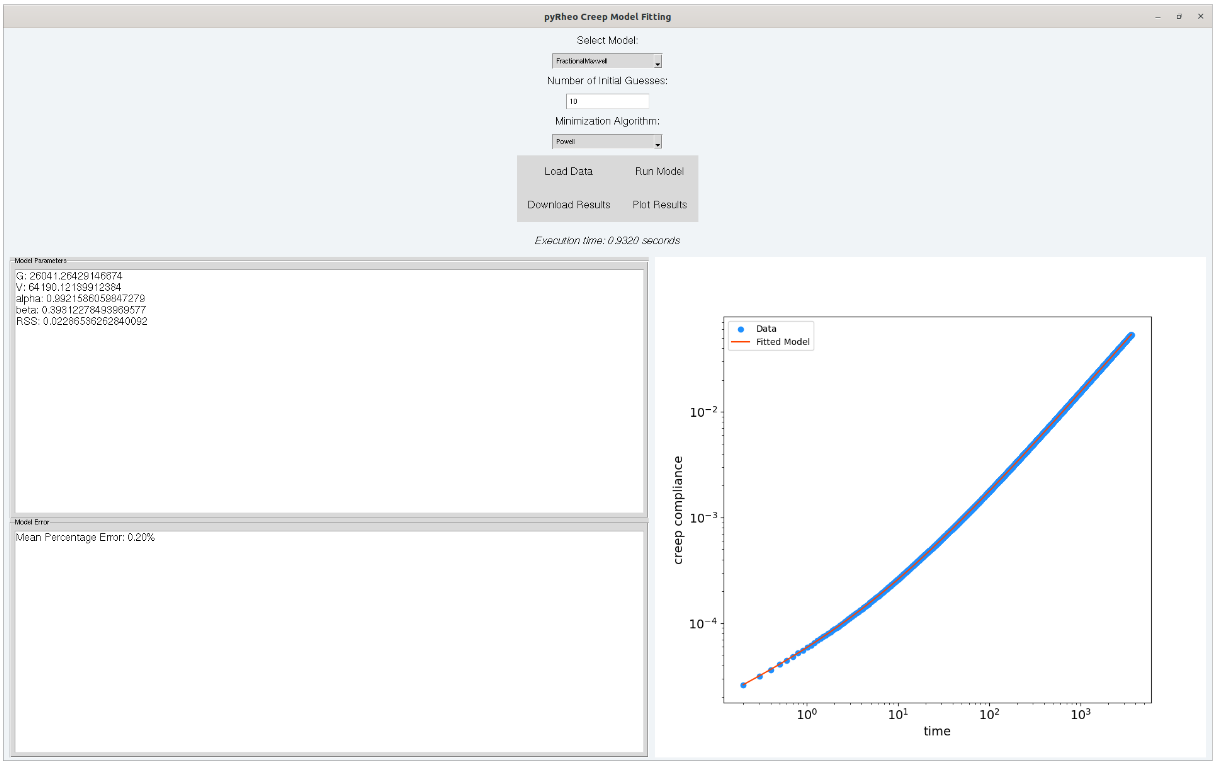


Data loaded successfully.

Supplementary Fig. 27 Successful data load prompt.

Selecting the model and algorithm

1. Use the drop-down menu labeled "**Select Model:**" to choose the desired model.



Select Model: FractionalMaxwell

Number of Initial Guesses: 10

Minimization Algorithm: Powell

Execution time: 0.9320 seconds

Model Parameters

G: 26041.26429146674
V: 64190.12139912384
alpha: 0.9921586059847279
beta: 0.39312278493969577
RSS: 0.02286536262840092

Model Error

Mean Percentage Error: 0.20%

Supplementary Fig. 28 Screenshot of pyRheo GUI - After loading and modeling data.

2. Enter the number of initial guesses in the field labeled "**Number of Initial Guesses:**". The default value is set to 10.

3. Use the drop-down menu labeled "**Minimization Algorithm:**" to select the optimization algorithm.

Running the model

1. Click the "**Run Model**" button to start the model fitting process.

2. The execution time will be displayed in the bottom part of the GUI.

3. After the model has run, the model parameters and fitting error will be displayed in the respective sections.

Plotting results

1. After running the model, click the "**Plot Results**" button to visualize the fitting results. The plot will display the original data points and the fitted model as in Supplementary Fig. 28.

Downloading Results

1. Click the "**Download Results**" button to save the fitting results.

2. You will be prompted to choose a location and filename for the results. The results will be saved as a CSV file.

Notes and references

[1] P. K. Singh, J. M. Soulages and R. H. Ewoldt, *Rheologica Acta*, 2019, **58**, 341–359.

[2] J. Song, N. Holten-Andersen and G. H. McKinley, *Soft Matter*, 2023, **19**, 7885–7906.

[3] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, *Nature Methods*, 2020, **17**, 261–272.

[4] C. Zeng and Y. Q. Chen, *Fractional Calculus and Applied Analysis*, 2015, **18**, 1492–1506.

[5] I. O. Sarumi, K. M. Furati and A. Q. M. Khaliq, *Journal of Scientific Computing*, 2020, **82**, 1–27.

[6] R. Garrappa, *SIAM Journal on Numerical Analysis*, 2015, **53**, 1350–1369.

[7] W. H. Herschel and R. Bulkley, *Colloid and Polymer Science*, 1926, **39**, 291–300.

[8] E. C. Bingham, *Bulletin of the Bureau of Standards*, 1916, **13**, 309–353.

[9] K. Yasuda, R. C. Armstrong and R. E. Cohen, *Rheologica Acta*, 1981, **20**, 163–178.

[10] M. M. Cross, *Journal of Colloid Science*, 1965, **20**, 417–437.

[11] N. Casson, *Rheology of Disperse Systems*, Pergamon Press, 1959, pp. 84–104.

[12] I. Y. Miranda-Valdez, T. Mäkinen, S. Coffeng, A. Päivänsalo, L. Jannuzzi, L. Viitanen, J. Koivisto and M. J. Alava, *Materials Horizons*, 2025, Advance Article.

[13] J. Rickman, T. Lookman and S. Kalinin, *Acta Materialia*, 2019, **168**, 473–510.

[14] F. A. Lavergne, P. Sollich and V. Trappe, *The Journal of Chemical Physics*, 2022, **156**, 154901.

[15] E. S. Epstein, L. Martinetti, R. H. Kollarigowda, O. Carey-De La Torre, J. S. Moore, R. H. Ewoldt and P. V. Braun, *Journal of the American Chemical Society.*, 2019-02-27, **141**, 3597–3604.

[16] I. Y. Miranda-Valdez, M. Sourroubille, T. Mäkinen, J. G. Puente-Córdova, A. Puisto, J. Koivisto and M. J. Alava, *Cellulose*, 2024, **31**, 1545–1558.

[17] R. G. Ricarte and S. Shanbhag, *Polymer Chemistry*, 2024, **15**, 815–846.

[18] K. R. Lennon, G. H. McKinley and J. W. Swan, *Data-Centric Engineering*, 2023, **4**, e13.

[19] R. I. Dekker, M. Dinkgreve, H. D. Cagny, D. J. Koeze, B. P. Tighe and D. Bonn, *Journal of Non-Newtonian Fluid Mechanics*, 2018, **261**, 33–37.

[20] A. K. Landauer, O. L. Kafka, N. H. Moser, I. Foster, B. Blaiszik and A. M. Forster, *Scientific Data*, 2023, **10**, 356.

[21] T. Head, MechCoder, G. Louppe, I. Shcherbatyi, fcharras, Z. Vinícius, cmmalone, C. Schröder, nel215, N. Campos, T. Young, S. Cereda, T. Fan, rene rex, K. K. Shi, J. Schwabedal, carlosdanielcsantos, Hvass-Labs, M. Pak and A. Fabisch, *scikit-optimize/scikit-optimize: v0.5.2*, 2018.

[22] I. Y. Miranda-Valdez, L. Viitanen, J. Mac Intyre, A. Puisto, J. Koivisto and M. Alava, *Carbohydrate Polymers*, 2022, **298**, 119921.

[23] V. Torsti, T. Mäkinen, S. Bonfanti, J. Koivisto and M. J. Alava, *APL Machine Learning*, 2024, **2**, 016119.

[24] P. I. Frazier, *A Tutorial on Bayesian Optimization*, 2018, `https://arxiv.org/abs/1807.02811`.

[25] M. A. Liegeois, M. Braunreuther, A. R. Charbit, W. W. Raymond, M. Tang, P. G. Woodruff, S. A. Christenson, M. Castro, S. C. Erzurum, E. Israel and et al., *JCI Insight*, 2024, **9**, e181024.