

RHEOS.jl – A Julia Package for Rheology Data Analysis

Jonathan Louis Kaplan¹, Alessandra Bonfanti¹, and Alexandre J Kabla¹

¹ Engineering Department, University of Cambridge, United Kingdom

DOI: [10.21105/joss.01700](https://doi.org/10.21105/joss.01700)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 25 August 2019

Published: 24 September 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

Rheology is the science of deformation and flow, with a focus on materials that do not exhibit simple linear elastic or viscous Newtonian behaviours. Rheology plays an important role in the characterisation of soft viscoelastic materials commonly found in the food and cosmetics industries, as well as in biology and bioengineering. Empirical and theoretical approaches are commonly used to identify and quantify material behaviours based on experimental data.

RHEOS (RHEology, Open-Source) is a software package designed to make the analysis of rheological data simpler, faster, and more reproducible. RHEOS is currently limited to the broad family of linear viscoelastic models. A particular strength of the library is its ability to handle rheological models containing fractional derivatives, which have demonstrable utility for the modelling of biological materials (Aime, Cipelletti, & Ramos, 2018; Bonfanti, Fouchard, Khalilgharibi, Charras, & Kabla, 2019; Bouzid et al., 2018; Kaplan, Torode, Daher, & Braybrook, 2019), but have hitherto remained in relative obscurity – possibly due to their mathematical and computational complexity. RHEOS is written in Julia (Bezanson, Edelman, Karpinski, & Shah, 2017), which provides excellent computational efficiency and approachable syntax. RHEOS is fully documented and has extensive testing coverage.

To our knowledge, there is to this date no other software package that offers RHEOS' broad selection of rheology analysis tools and extensive library of both traditional and fractional models. It has been used to process data and validate a model in Bonfanti et al. (2019), and is currently in use for several ongoing projects.

It should be noted that RHEOS is not an optimisation package. It builds on another optimisation package, NLOpt (Johnson, n.d.), by adding a large number of abstractions and functionality specific to the exploration of viscoelastic data.

Statement of Need

Arbitrary stress-strain curves and broad relaxation spectra require advanced software

Many scientists and engineers who undertake rheological experiments would fit their data with one or several viscoelastic models in order to classify materials, quantify their behaviour, and predict their response to external perturbations.

Standard linear viscoelastic models take the form of an ordinary differential equation between stress σ and strain ϵ . Under simple perturbations (step or ramp in stress or strain, or frequency sweep), it is relatively straight-forward to extract time-scales and identify asymptotic behaviours required to identify parameter values. However, data often involves complex stress and strain signals, and materials whose behaviour involves a broad distribution of time-scales,

including power law behaviours. Fitting models and predicting their response in the time domain then requires computing viscoelastic hereditary integrals such as:

$$\sigma(t) = \int_0^t G(t - \tau) \frac{d\epsilon(\tau)}{d\tau} d\tau$$

where G is the relaxation response (stress response to a step in strain) of the material. G is defined analytically for classical models, but may only be available numerically in some cases. A similar relation exists to calculate the strain from the stress history. Fitting and predicting behaviour then becomes non-trivial and standardised processing tools are needed.

Learning about rheology is facilitated by the ability to explore a large database of models

Obtaining intuition for fractional viscoelastic theory can be difficult and learning material is sparse: of popular rheology textbooks published over several decades (Barnes, Hutton, & Walters, 1989; H. F. Brinson & Brinson, 2008; Findley, Lai, & Onaran, 1989; Lakes, 2009; Ward & Sweeney, 2013), fractional viscoelasticity is only mentioned briefly in one of them (Lakes, 2009). Tools are needed to support researchers with their exploration of standard and advanced models, and how they behave in response to idealised loading conditions, in particular when analytical expressions are difficult to obtain.

Extracting parameters, selecting models, and comparing materials require standardised tools

Because understanding of materials is often dependent on summarising their behaviour with a model, one must be able to test and compare a broad range of models to inform model selection and reliably identify material parameters. There are currently very limited options available in the public domain (Das, Read, & McLeish, 2012; Seifert, 2019), and most research groups have to invest significant effort into developing custom software. An open-source standardised library of models and fitting algorithms would support the rheology research community and make analysis more systematic, transparent, and reproducible.

Implementation

Features

RHEOS addresses the issues outlined in the Statement of Need in several ways.

- RHEOS includes an extensive library of both traditional and fractional viscoelastic models. Although this library will satisfy most users, it is also straightforward to add additional models to RHEOS should they need to.
- As well as being able to fit models to experimental data and predict the materials response to step loading of stress or strain, RHEOS can handle arbitrary loading for non-singular and singular models, and for constant or variable sample rates.
- For intuition-building and model exploration, RHEOS includes signal generation features so that common loading patterns (e.g. step, ramp, stairs) can be applied to unfamiliar models.

- As a convenience to the user, RHEOS also includes easy-to-use CSV importing and exporting functions, as well as a number of preprocessing functions for resampling and smoothing.

All of the above features are linked together in a seamless interface intended to be very approachable for less experienced programmers. The different paradigms of creep, relaxation, and oscillatory testing are all accounted for, and models fitted against one type of data can be used to predict against a different type of data. (For instance, fitting against relaxation data and predicting the frequency response spectrum.)

Workflow

The following schematic illustrates one of the common RHEOS workflows in which experimental time-domain viscoelastic data is fitted to a model. This model is then used to make a prediction of the behaviour so that its accuracy can be qualitatively assessed. This workflow is shown schematically in Figure 1, and the prediction of the fitted model is plotted against the original data in Figure 2.

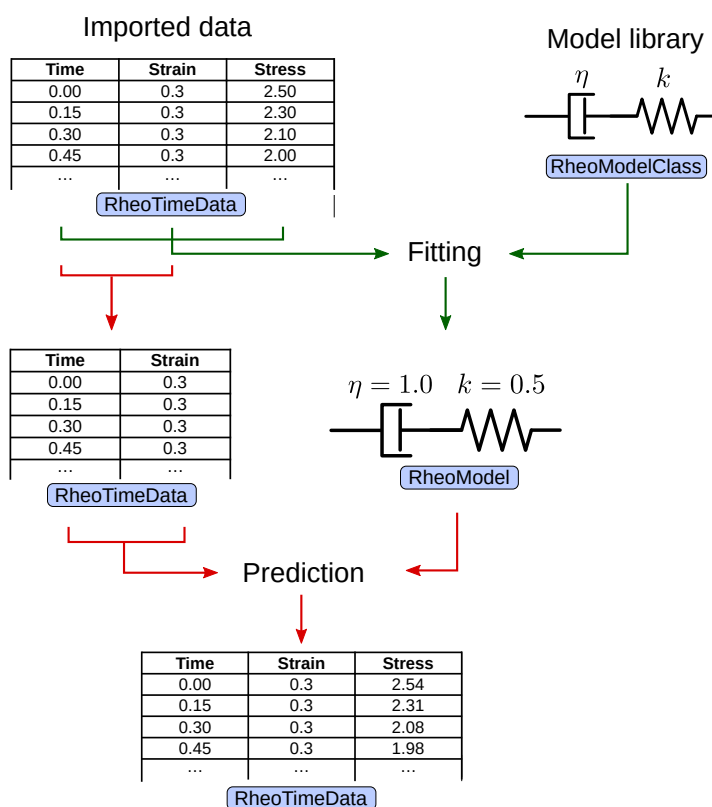


Figure 1: High level schematic of a fitting and prediction workflow from experimental data.

A brief description of this workflow is the following. A CSV is imported into a RHEOS `RheoTimeData` struct using a convenient loading function. This is then fitted to a `RheoModelClass`, which embeds expressions for key characteristics of the model (relaxation function, creep response, complex modulus) involving symbolic parameters. This results in a fitted `RheoModel` where parameters are now substituted with fixed values derived from the fitting procedure. In the prediction step, the fitted `RheoModel` is combined with partial data (here only time and strain) to simulate the stress values expected from the model. The original data and model can then be compared graphically and numerically.

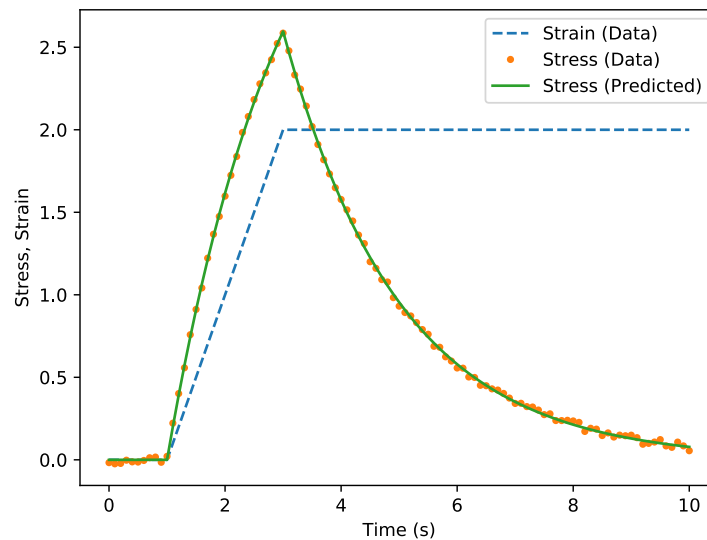


Figure 2: Qualitative assessment of the fitted model.

This example and others are available as Julia Jupyter notebooks, available in RHEOS GitHub repository and viewable on the RHEOS [documentation website](#).

Acknowledgements

JLK would like thank the George and Lillian Schiff Foundation for the PhD funding which facilitated this project. AB, JLK, and AJK acknowledge the BBSRC grants BB/M002578/1, BB/K018175/1, and BB/P003184/1. All authors would like to thank Rohit Goswami, Grant Bruer, and Adam Beall for their valuable feedback during the review process.

References

- Aime, S., Cipelletti, L., & Ramos, L. (2018). Power law viscoelasticity of a fractal colloidal gel. *Journal of Rheology*, 62(6), 1429–1441. doi:[10.1122/1.5025622](https://doi.org/10.1122/1.5025622)
- Barnes, H. A., Hutton, J. F., & Walters, K. (1989). *An introduction to rheology*. Amsterdam, The Netherlands: Elsevier.
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98. doi:[10.1137/141000671](https://doi.org/10.1137/141000671)
- Bonfanti, A., Fouchard, J., Khalilgharibi, N., Charras, G., & Kabla, A. (2019). A unified rheological model for cells and cellularised materials. *bioRxiv*. doi:[10.1101/543330](https://doi.org/10.1101/543330)
- Bouزيد, M., Keshavarz, B., Geri, M., Divoux, T., Del Gado, E., & McKinley, G. H. (2018). Computing the linear viscoelastic properties of soft gels using an optimally windowed chirp protocol. *Journal of Rheology*, 62(4), 1037–1050. doi:[10.1122/1.5018715](https://doi.org/10.1122/1.5018715)
- Brinson, H. F., & Brinson, L. C. (2008). *Polymer engineering science and viscoelasticity: An introduction*. New York: Springer.

Das, C., Read, D. J., & McLeish, T. C. (2012). Bob-rheology version 2.5. *SourceForge*. Retrieved from <https://sourceforge.net/projects/bob-rheology/>

Findley, W. N., Lai, J. S., & Onaran, K. (1989). *Creep and relaxation of nonlinear viscoelastic materials with an introduction to linear viscoelasticity*. Dover books on engineering. New York: Dover.

Johnson, S. G. (n.d.). The NLOpt nonlinear-optimization package. Retrieved from <https://github.com/stevengj/nlopt>

Kaplan, J. L., Torode, T. A., Daher, F. B., & Braybrook, S. A. (2019). On pectin methylesterification: Implications for in vitro and in vivo viscoelasticity. *bioRxiv*. doi:10.1101/565614

Lakes, R. (2009). *Viscoelastic materials*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511626722

Seifert, J. (2019). Python tools for analysis of AFM data. Retrieved from <https://github.com/jcbs/ForceMetric>

Ward, I. M., & Sweeney, J. (2013). *Mechanical properties of solid polymers* (Third edition.). Chichester, West Sussex, United Kingdom: Wiley.