Physical insights from machine learning tools

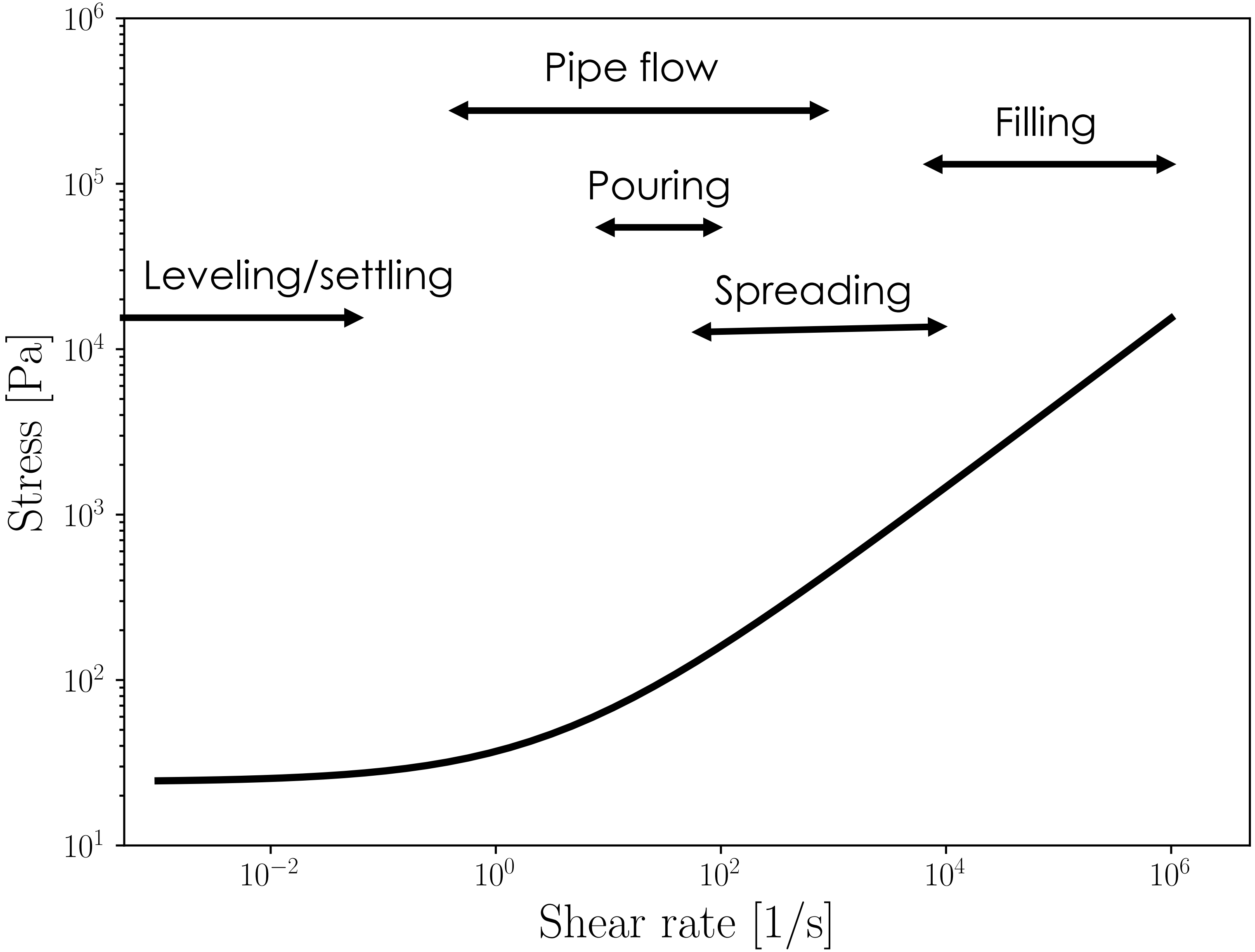
Marco Caggioni, William Hartt,
Julie Hipp, Seth Lindberg, Emilio Tozzi

93rd Annual Society of Rheology Meeting
10/10/2022

Outline

- Motivations
- Mastercurves (MC)
  - Model based MC
  - Machine Learning Based MC
- Physical insights
- Challenges for the future

Motivation



Pipe flow
Filling
Pouring
Leveling/settling
Spreading
Stress [Pa]
Shear rate [1/s]
10^6
10^5
10^4
10^3
10^2
10^1
10^{-2}
10^0
10^2
10^4
10^6

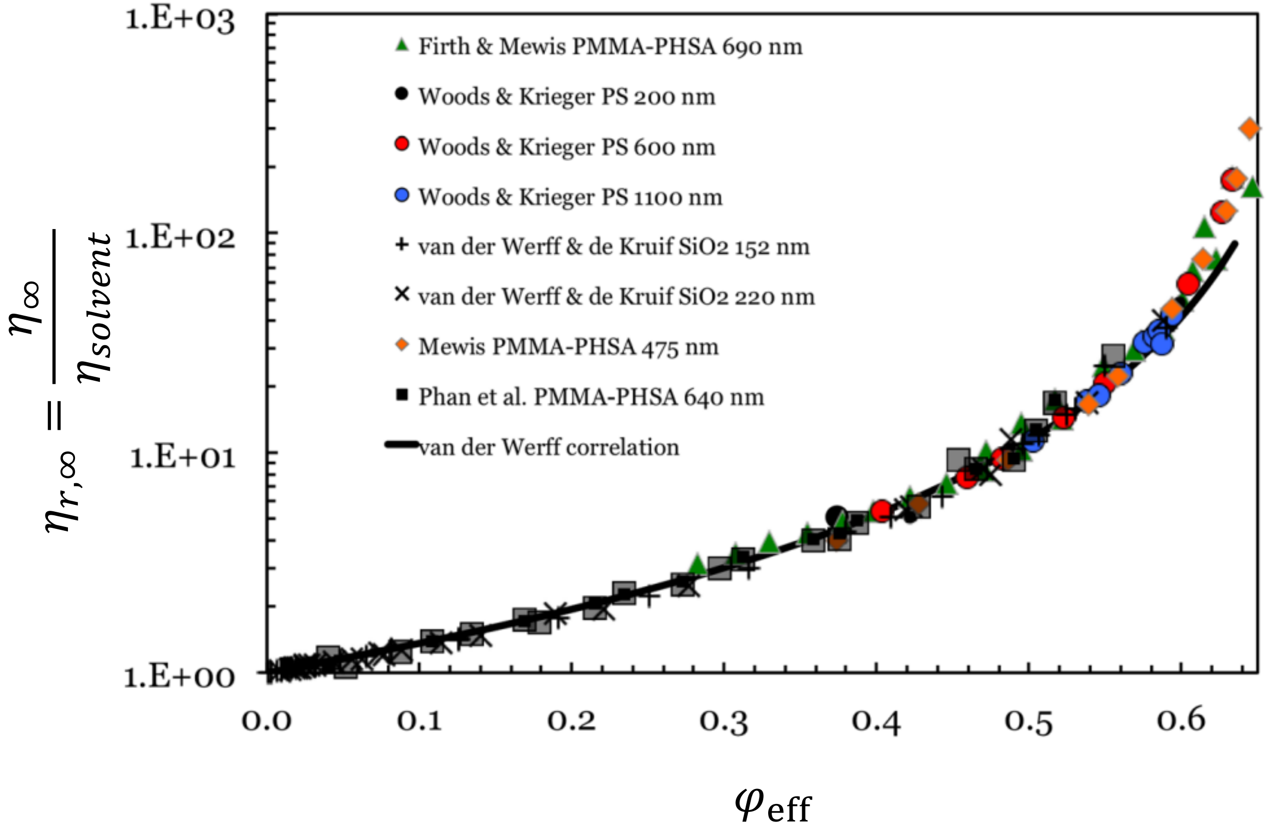
Master Curve Introduction

- Typically used to gain physical insight into behavior by plotting two meaningful dimensionless parameters against one another.

- Suspension rheology often uses Maron-Pierce type equations:

$$\eta_{r,\infty} = \frac{\eta_\infty}{\eta_{solvent}} = \left(1 - \frac{\varphi}{0.71}\right)^{-2}$$

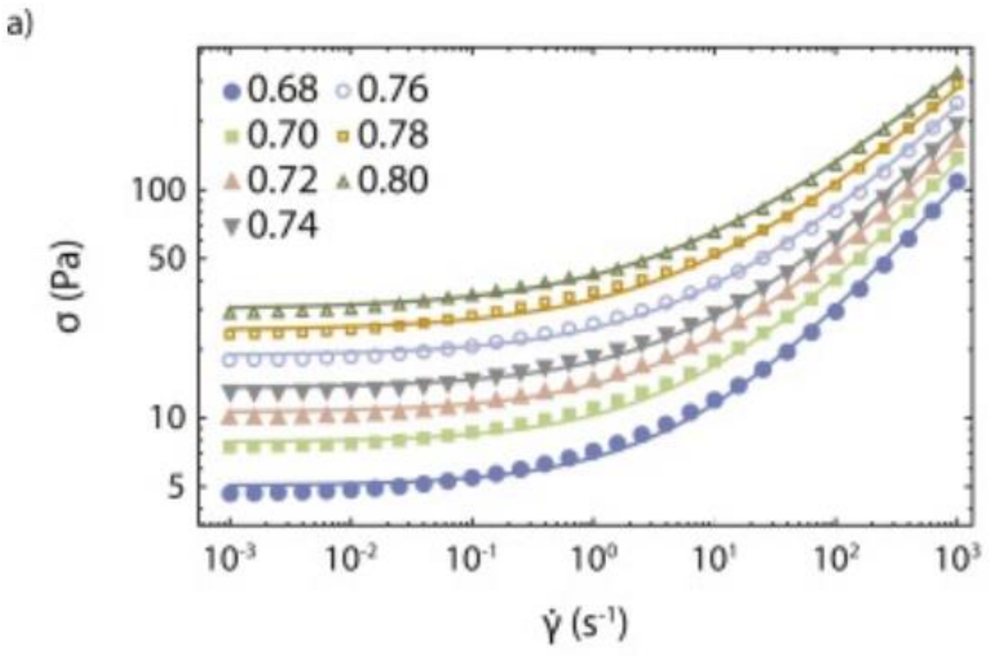
Russel, *et al. J. Rheol.* 57 (2013): 1555-1567.



Mastercurve Example

Dekker, Riande I., et al. "Scaling of flow curves: Comparison between experiments and simulations." *J. Nonnewton. Fluid Mech.* 261 (2018): 33-37.

- Oil Emulsion in SDS solution
- Variable oil volume fraction
- Equilibrium flow curve
- Herschel-Bulkley (HB) model fit → yield stress



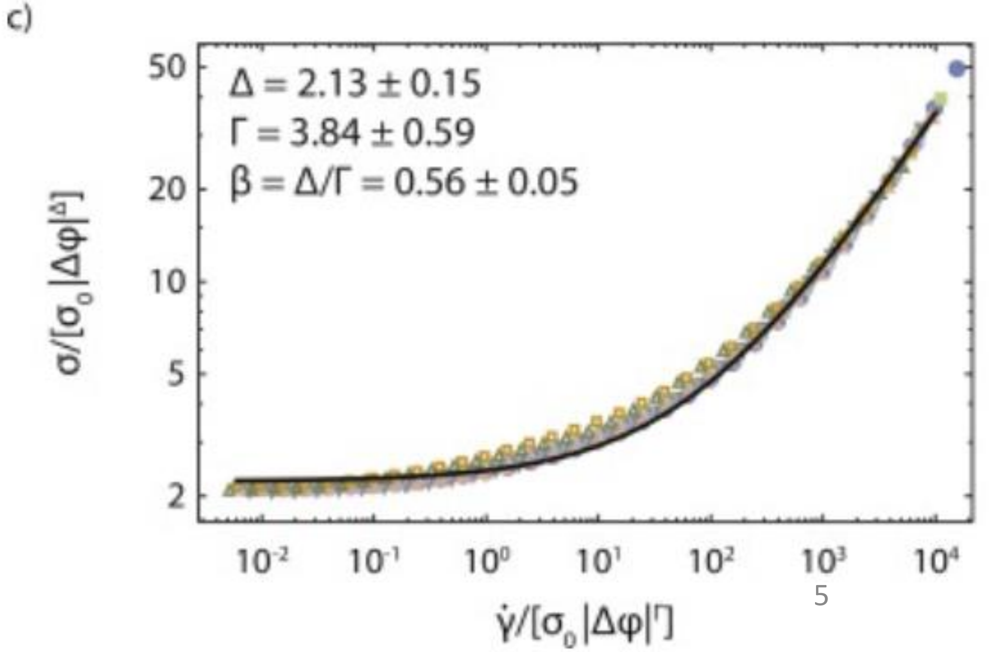
Master curve showing the collapse of the flow curves into one when plotted:
$|\Delta\phi| = \phi - \phi_c$ distance from jamming ($\phi_c = 0.64$)

Reduced stress: $\dfrac{\sigma}{\sigma_o|\Delta\phi|^{\Delta}}$ $with\ \Delta = 2.13 \pm 0.15$

Reduced shear rate: $\dfrac{\dot{\gamma}}{\sigma_o|\Delta\phi|^{\Gamma}}$ $with\ \Gamma = 3.84 \pm 0.59$

Mastercurve + HB fit β = Δ / Γ = 0.56 and K=0.19

$$\sigma = \sigma_y + K\,\dot{\gamma}^{\beta}$$

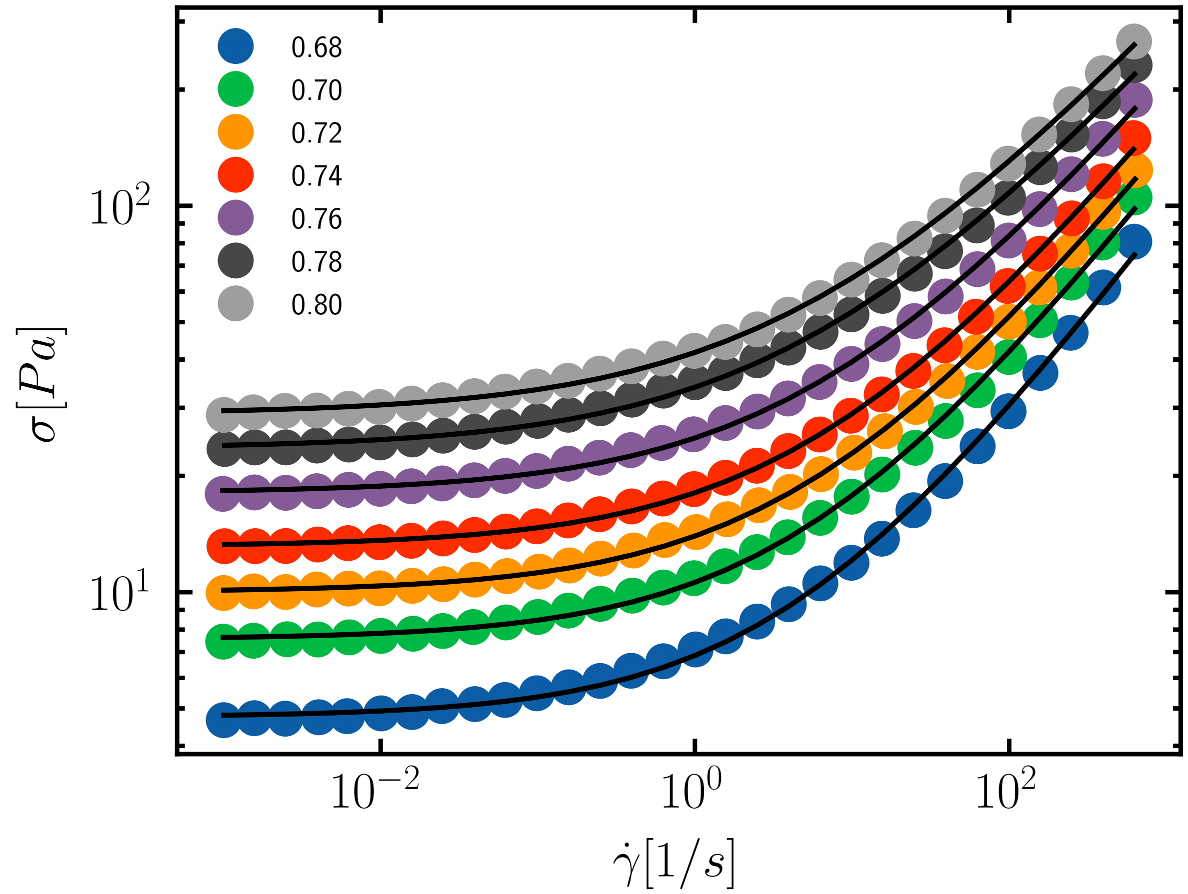


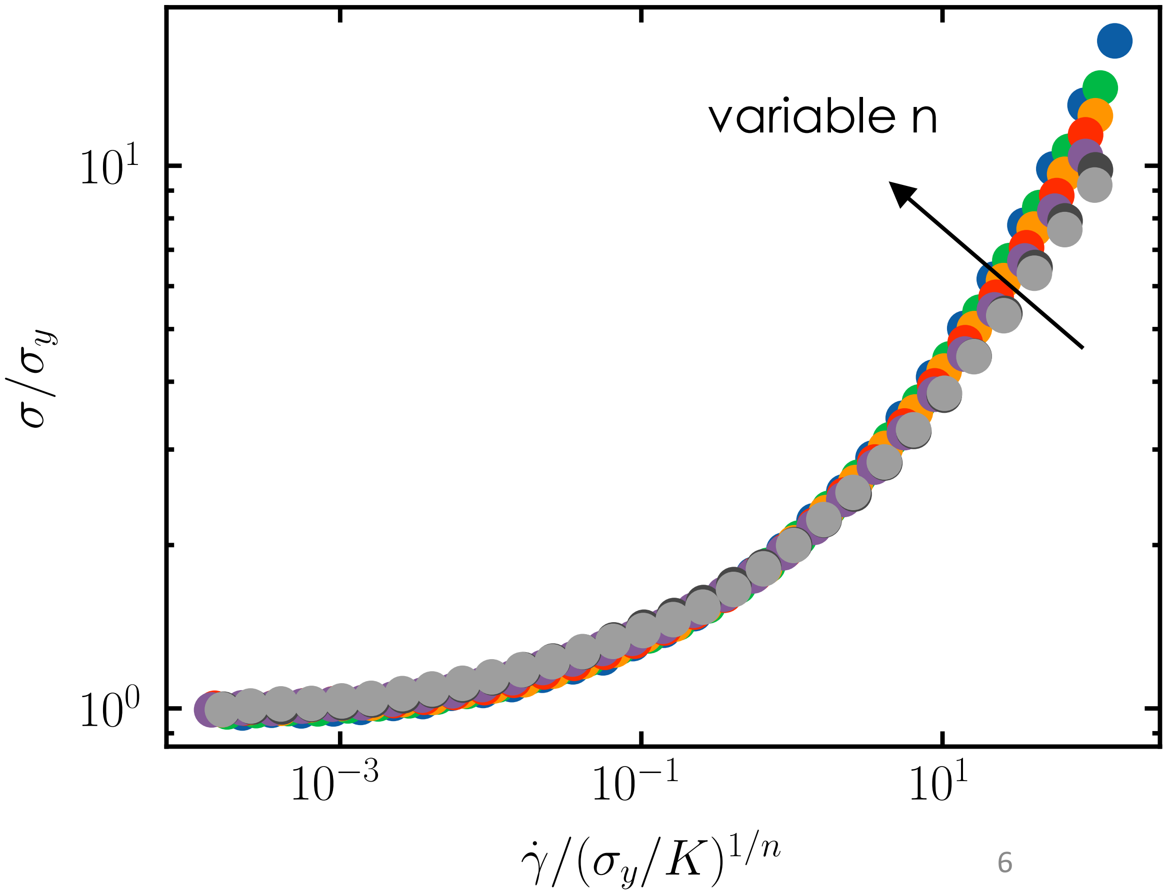
Natural scaling from rheological model: HB

Φ	K (Pa.s^n)	n	σ_y (Pa)
0.68	2.10	**0.54**	4.75
0.70	3.04	**0.53**	7.56
0.72	3.97	**0.51**	10.01
0.74	4.94	**0.51**	13.14
0.76	6.99	**0.49**	18.07
0.78	10.36	**0.46**	23.55
0.80	12.83	**0.45**	28.90

$$\sigma(\dot{\gamma}) = \sigma_y + K\dot{\gamma}^n$$

$$\frac{\sigma}{\sigma_y} = 1 + \left(\frac{\dot{\gamma}}{(\sigma_y/K)^{1/n}}\right)^n$$





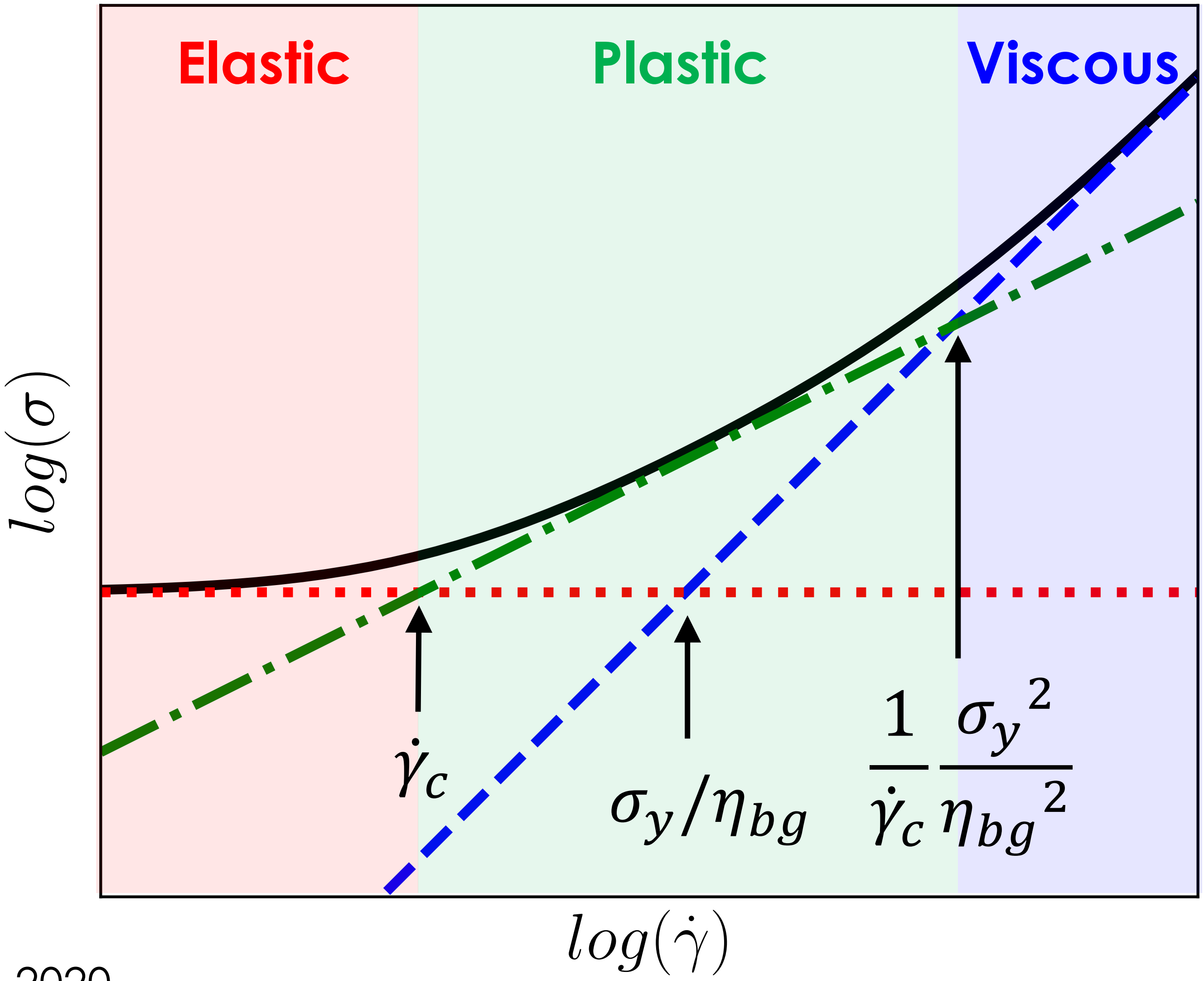
Three-component model

TC: $\sigma = \sigma_y + \sigma_y \cdot \left(\frac{\dot{\gamma}}{\dot{\gamma}_c}\right)^{\frac{1}{2}} + \eta_{bg} \cdot \dot{\gamma}$

σ_y = yield stress

$\dot{\gamma}_c$ = critical shear rate

η_{bg} = background viscosity



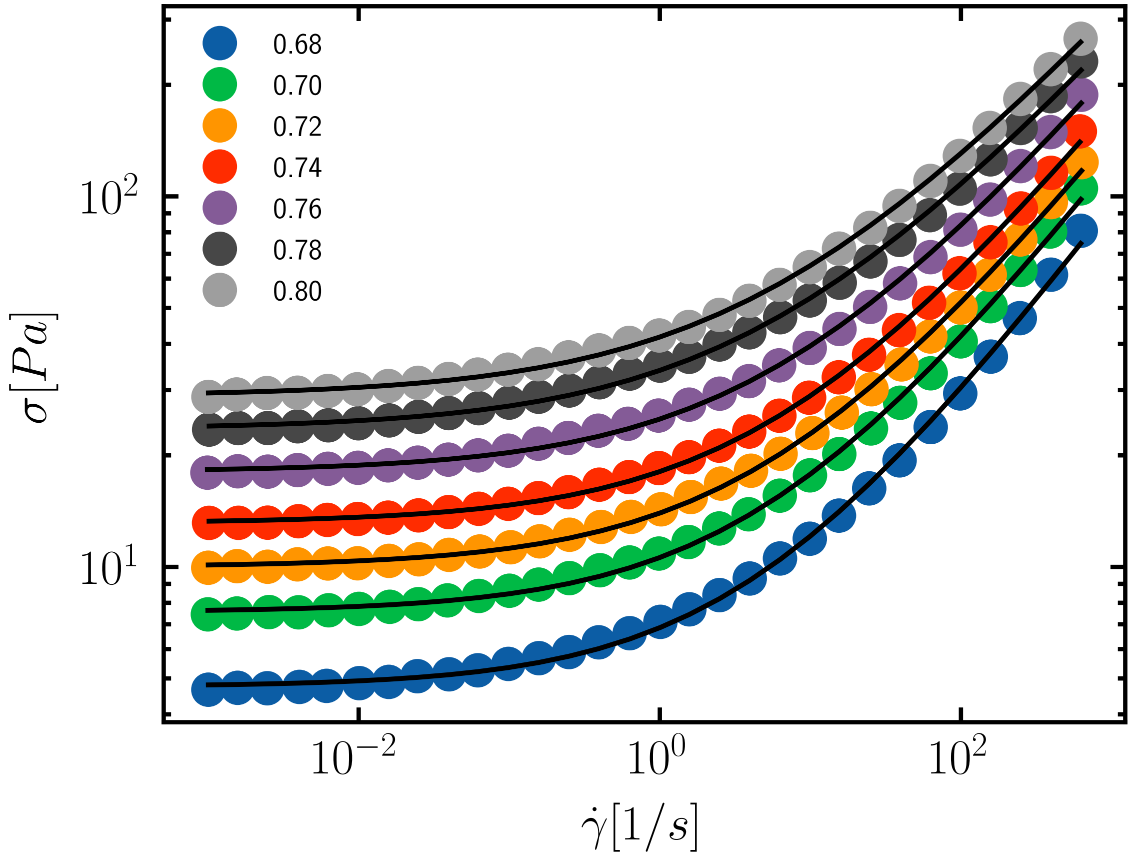
Caggioni, Trappe, Spicer *J. Rheol.* 64(2): 413-422, 2020

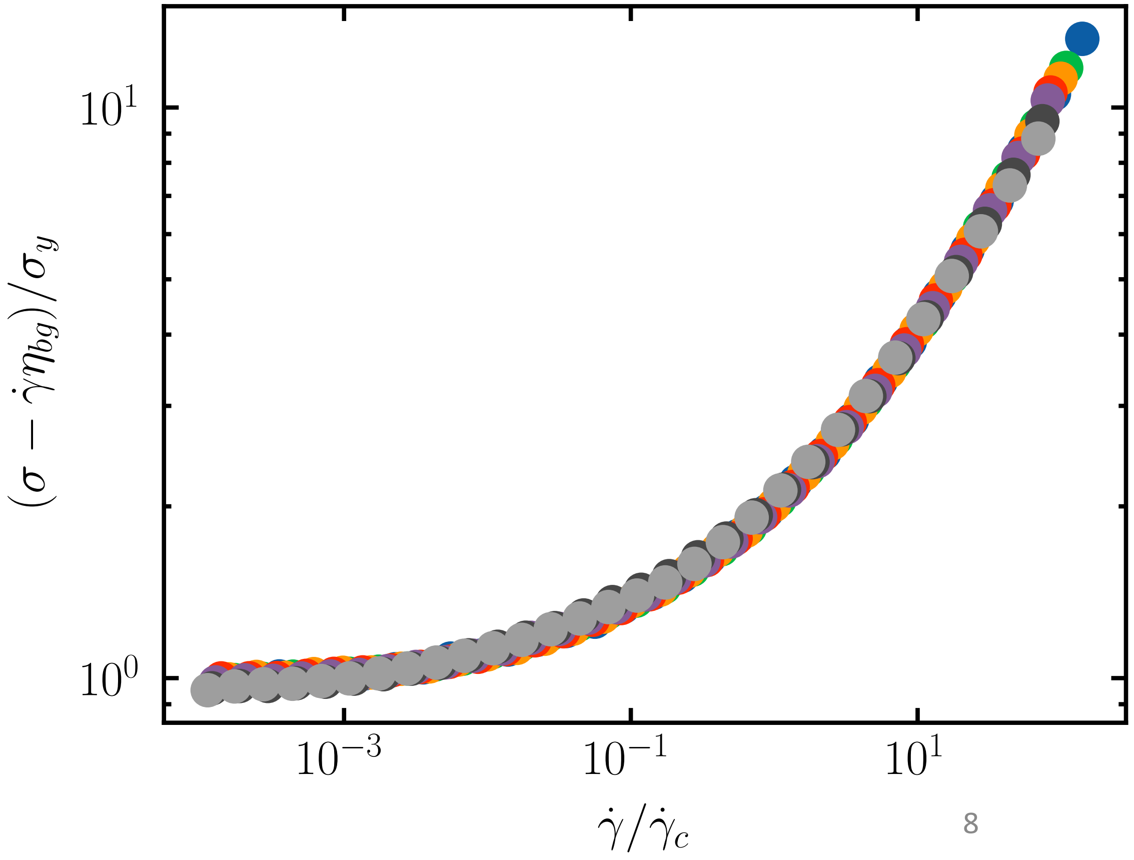
Natural scaling from rheological model: TC

$$\sigma(\dot{\gamma}) = \sigma_y + \sigma_y \left(\frac{\dot{\gamma}}{\dot{\gamma}_c}\right)^{0.5} + \eta_{bg}\dot{\gamma}$$

Φ	η_{bg} (Pa.s)	$\dot{\gamma}_c$ (s-1)	σ_y (Pa)
0.68	0.03	4.44	4.69
0.70	0.03	5.77	7.51
0.72	0.02	6.37	10.00
0.74	0.02	7.30	13.16
0.76	0.00	7.78	18.28
0.78	0.00	8.47	24.46
0.80	0.00	8.98	30.19

$$\frac{\sigma(\dot{\gamma}) - \eta_{bg}\dot{\gamma}}{\sigma_y} = 1 + \left(\frac{\dot{\gamma}}{\dot{\gamma}_c}\right)^{0.5}$$

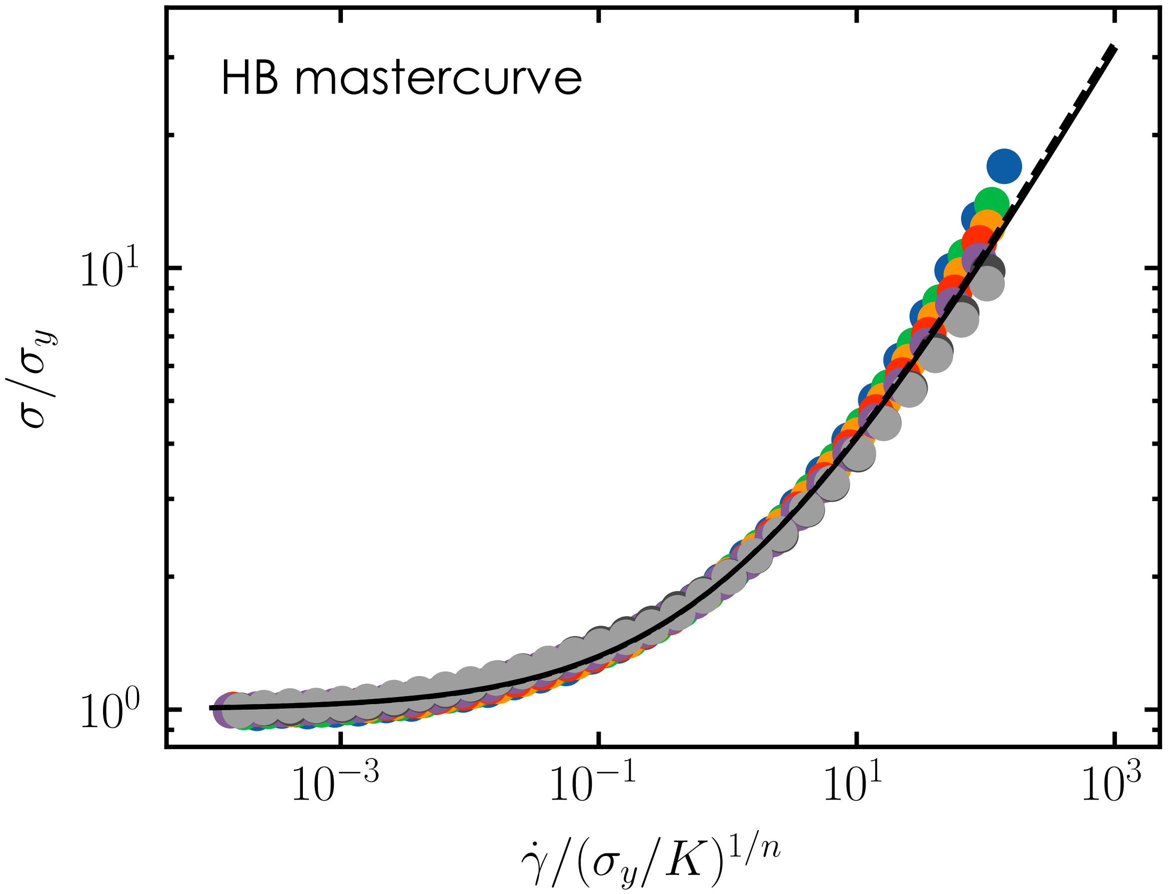




Judgement of Mastercurve Quality

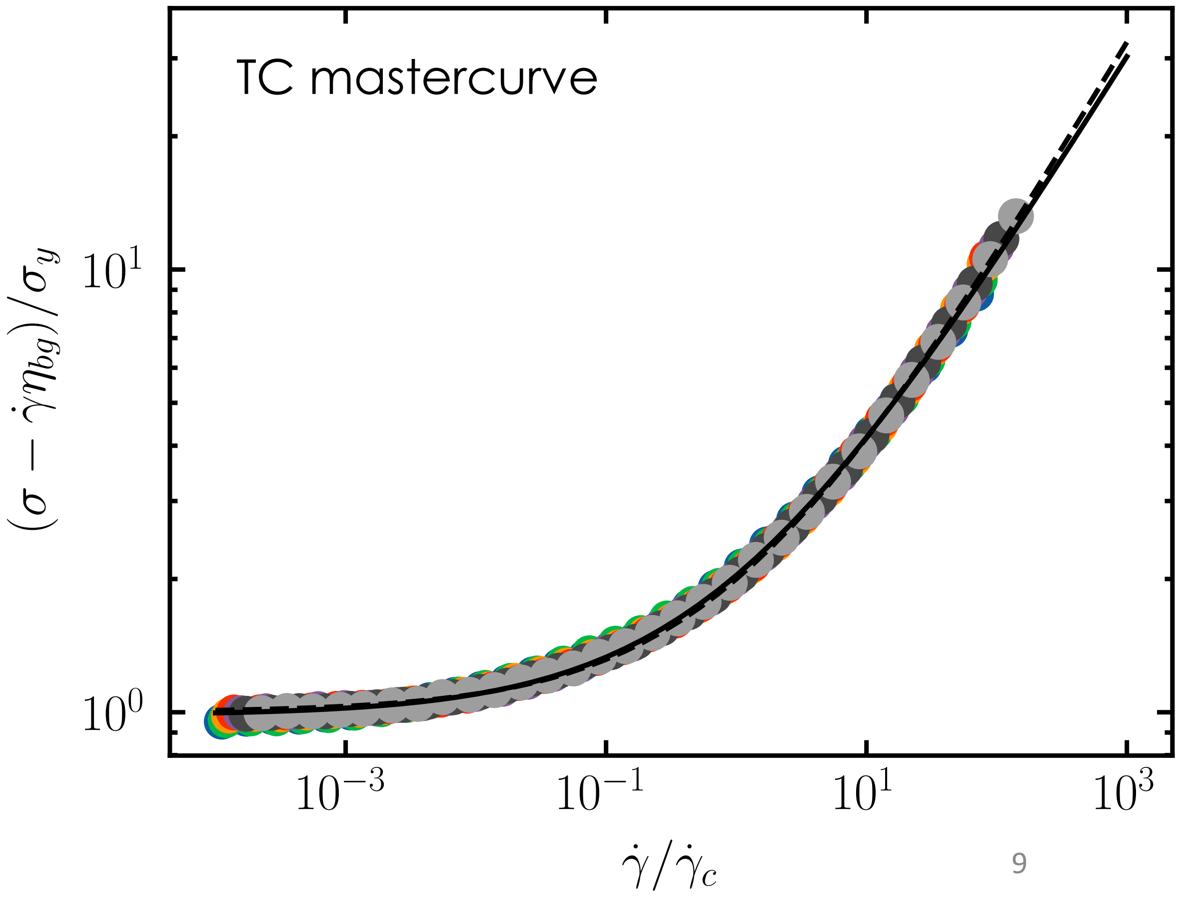
HB parameter	value	standard error
σ_y (Pa)	1.00	0.007
K (Pa.s^n)	1.00	0.017
n	0.49	0.005

$\chi^2_{HB} = 0.583$



HB parameter	value	standard error
σ_y (Pa)	0.98	0.003
K (Pa.s^n)	1.05	0.007
n	0.48	0.002

$\chi^2_{TC} = 0.095$



Mastercurve ML tool

PO39 Automatic construction of rheological master curves

Kyle R. Lennon, Gareth H. McKinley and James W. Swan

Wednesday 6:30 Ballroom 1-2-3-4 (Poster Session)

Cornell University

arXiv > physics > arXiv:2204.09521

Physics > Data Analysis, Statistics and Probability

[Submitted on 20 Apr 2022 (v1), last revised 31 May 2022 (this version, v2)]

A Data-Driven Method for Automated Data Superposition with Applications in Soft Matter Science

Kyle R. Lennon, Gareth H. McKinley, James W. Swan

The superposition of data sets with internal parametric self-similarity is a longstanding and widespread technique for the analysis of many types of experimental data across the physical sciences. Typically, this superposition is performed manually, or recently by one of a few automated algorithms. However, these methods are often heuristic in nature, are prone to user bias via manual data shifting or parameterization, and lack a native framework for handling uncertainty in both the data and the resulting model of the superposed data. In this work, we develop a data-driven, non-parametric method for superposing experimental data with arbitrary coordinate transformations, which employs Gaussian process regression to learn statistical models that describe the data, and then uses maximum a posteriori estimation to optimally superpose the data sets. This statistical framework is robust to experimental noise, and automatically produces uncertainty estimates for the learned coordinate transformations. Moreover, it is distinguished from black-box machine learning in its interpretability -- specifically, it produces a model that may itself be interrogated to gain insight into the system under study. We demonstrate these salient features of our method through its application to four representative data sets characterizing the mechanics of soft materials. In every case, our method replicates results obtained using other approaches, but with reduced bias and the addition of uncertainty estimates. This method enables a standardized, statistical treatment of self-similar data across many fields, producing interpretable data-driven models that may inform applications such as materials classification, design, and discovery.

Comments: 19 pages, 7 figures

Subjects: Data Analysis, Statistics and Probability (physics.data-an); Soft Condensed Matter (cond-mat.soft); Machine Learning (cs.LG); Computational Physics (physics.comp-ph)

Cite as: arXiv:2204.09521 [physics.data-an]
(or arXiv:2204.09521v2 [physics.data-an] for this version)
https://doi.org/10.48550/arXiv.2204.09521

Submission history

From: Kyle Lennon [view email]
[v1] Wed, 20 Apr 2022 14:58:04 UTC (1,124 KB)
[v2] Tue, 31 May 2022 15:03:44 UTC (1,124 KB)

Download:
- PDF
- Other formats

Current browse context: physics.data-an
< prev | next >
new | recent | 2204

Change to browse by: cond-mat, cond-mat.soft, cs, cs.LG, physics, physics.comp-ph

References & Citations
- INSPIRE HEP
- NASA ADS
- Google Scholar
- Semantic Scholar

Export Bibtex Citation

Bookmark

Bibliographic Tools | Code & Data | Related Papers | About arXivLabs

Code and Data Associated with this Article

arXiv Links to Code & Data (What is Links to Code & Data?)

Official Code

https://github.com/krlennon/mastercurves

Community Code

Submit your implementations of this paper on Papers With Code

Datasets Used

No dataset metadata found, you can link datasets here.

krlennon / mastercurves Public

Code | Issues | Pull requests | Actions | Projects | Security | Insights

main | 1 branch | 0 tags | Go to file | Add file | Code

Kyle Lennon add docs badge to readme · 8dfd762 on Jun 14 · 60 commits

demos	update for v0.2.0	4 months ago
docs	update setup.py description for pypi	4 months ago
mastercurves	version 0.2.0: built-in error propagation	4 months ago
.gitignore	untrack some demos	10 months ago
CITATION.cff	package software for PyPI	10 months ago
LICENSE	Initial commit	17 months ago
README.md	add docs badge to readme	4 months ago
setup.py	update setup.py description for pypi	4 months ago

README.md

mastercurves

pypi package 0.2.2 | downloads 8k | docs latest

Python package for automatically superimposing data sets to create a master curve, using Gaussian process regression and maximum a posteriori estimation.

Publication of this work is forthcoming. For now, if you use this software, please cite it using the metadata in the citation file.

About

Python package for automatically superimposing data sets to create a master curve, using Gaussian process regression and maximum a posteriori estimation.

krlennon-mastercurves.readthedocs.io/e...

python, machine-learning, automation, statistical-analysis, data-analysis, uncertainty-quantification, gaussian-processes, maximum-a-posteriori, interpretable-ai

Readme
GPL-3.0 license
Cite this repository
4 stars
1 watching
2 forks

Releases
No releases published

Packages

Search projects | Help | Sponsors | Log in | Register

mastercurves 0.2.2

pip install mastercurves

Latest version

Released: Jun 14, 2022

Python package for building master curves from data

Navigation
- Project description
- Release history
- Download files

Project links
- Homepage

Statistics

Project description

mastercurves

Python package for automatically superimposing data sets to create a master curve, using Gaussian process regression and maximum a posteriori estimation.

Publication of this work is forthcoming. For now, if you use this software, please cite it using the metadata in the citation file.

Documentation

Check out the documentation to learn more about the package and how to use its API. A tutorial and explanation of the demos are coming soon!

mastercurves 0.2.0

Search docs

INTRODUCTION
- mastercurves
- Automated Data Superposition

Quick Start
- Installation
- Creating a master curve
- Adding data to a master curve
- Defining the coordinate transformations
- Superposing the data
- Plotting the results

TUTORIALS
- Tutorial: Creating a Master Curve
- Tutorial: Adjusting the Gaussian Process Kernel

DEMOS
- Time-Temperature Superposition
- Sensitivity to Noisy Data

API
- MasterCurve
- Transforms

» Quick Start | View page source

Quick Start

Installation

To use mastercurves, first install it using pip:

```
$ pip install mastercurves
```

Creating a master curve

First import the package, then use the `MasterCurve()` constructor to initialize a master curve object:

```
from mastercurves import MasterCurve
mc = MasterCurve()
```

Adding data to a master curve

Next, collect data into three lists: `x_data`, `y_data`, and `states`. The elements of `x_data` and `y_data` should be arrays containing the x- and y-coordinates for a single state (i.e. one data set, which will be superposed with data sets comprising the other elements of `x_data` and `y_data`). The elements of `states` should be numeric values labeling the corresponding states.

When the data is ready, add it to the master curve:

```
mc.add_data(x_data, y_data, states)
```

Defining the coordinate transformations

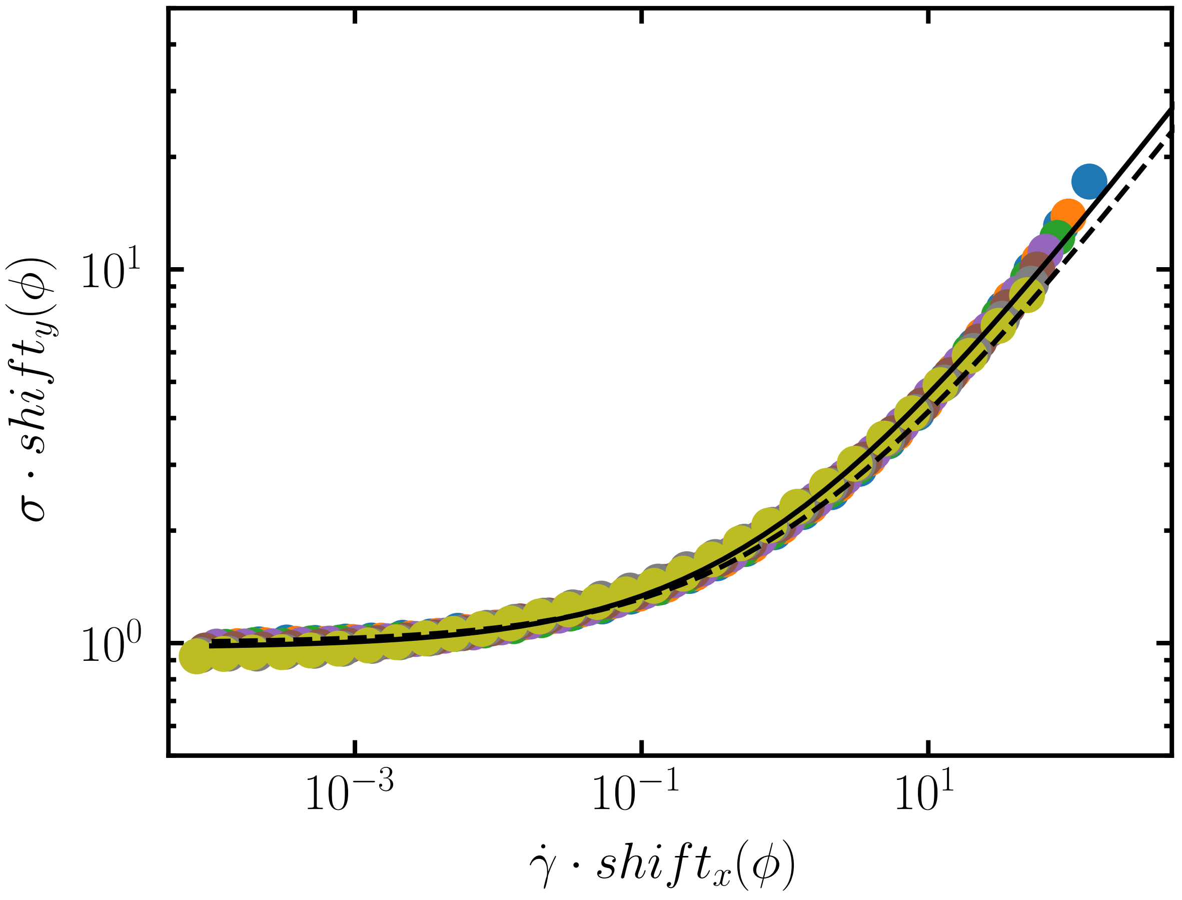
Then, add coordinate transformations to the master curve. If only horizontal shifting by a scale factor is required (the typical case for time-temperature superposition), this can be done as follows:

```
from mastercurves.transforms import Multiply
mc.add_htransform(Multiply())
```

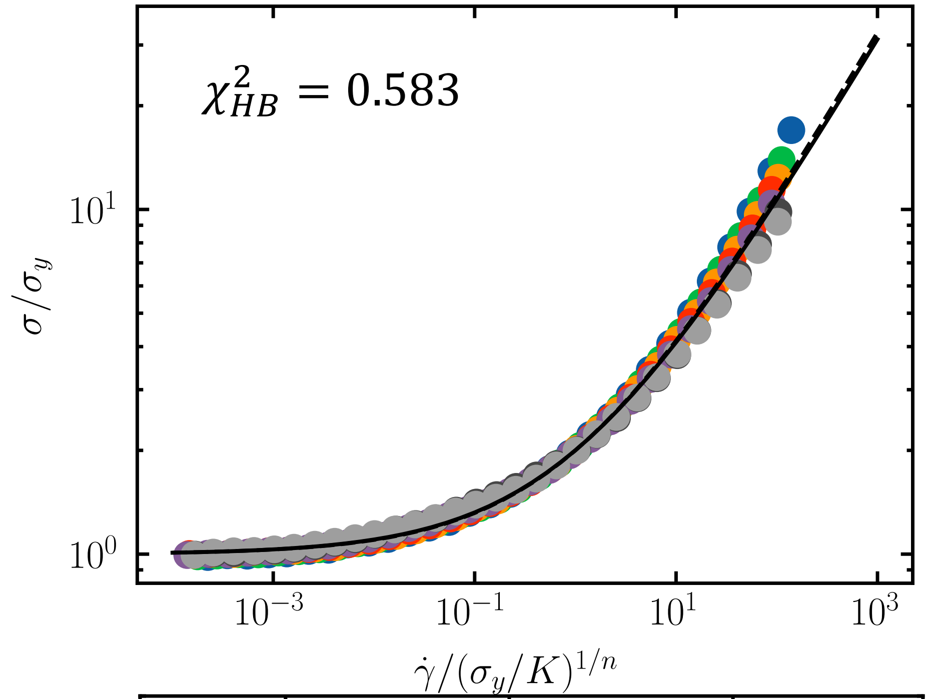
Mastercurve with just X and Y multiplication

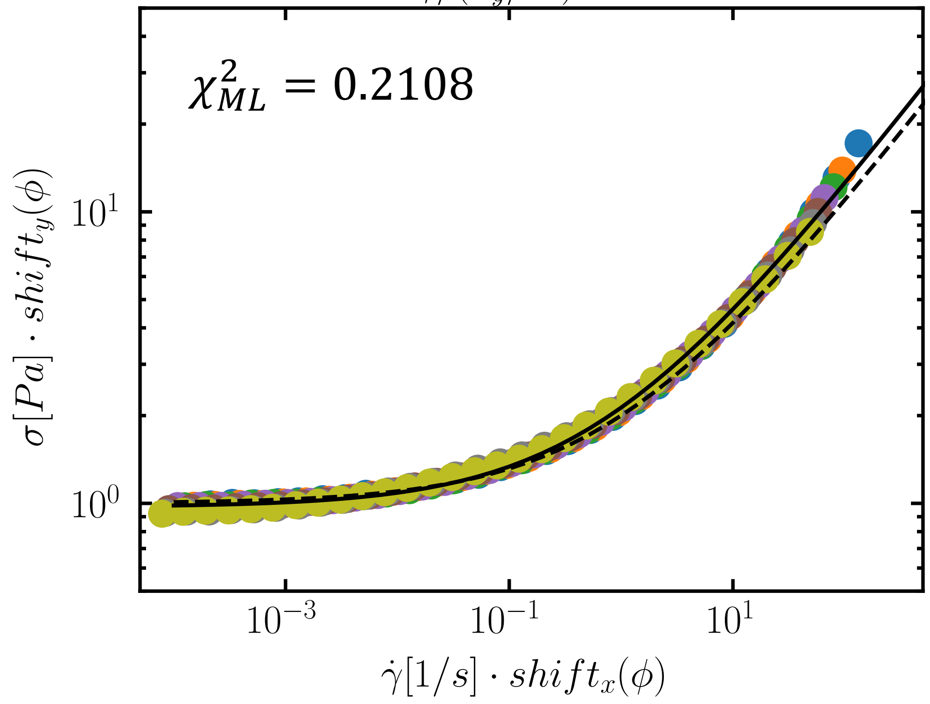
HB parameter	value	standard error
σ_y (Pa)	1.00	0.007
K (Pa.s^n)	1.00	0.017
n	0.49	0.005

$\chi^2 = 0.2108$

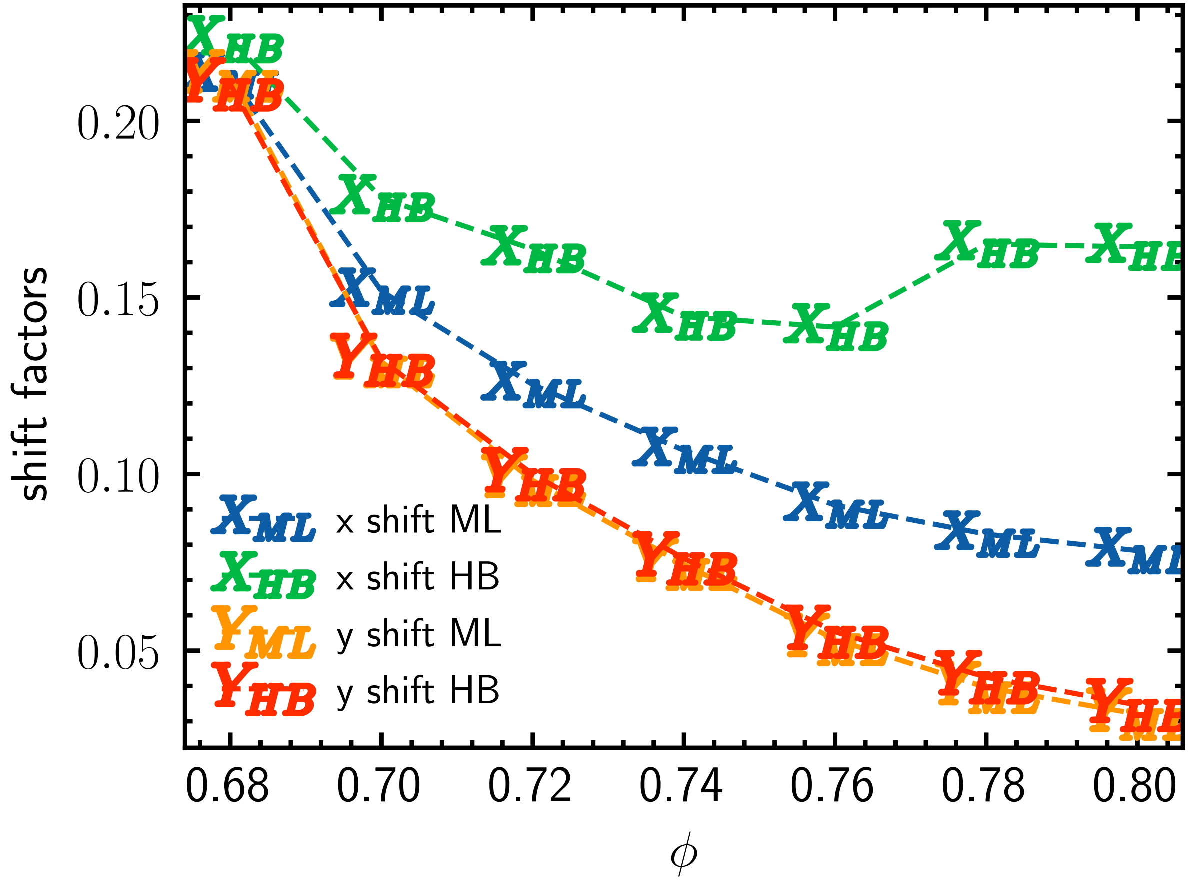



```python
mc=MasterCurve()

# Build a master curve
mc.clear()
mc.add_data(gdots, sigmas, phi)

# Add transformations
mc.add_htransform(Multiply())
mc.add_vtransform(Multiply())

# Superpose
loss = sum(mc.superpose())
mc.change_ref(0.68, 4.7, 4.7)

print(loss)

fig1, ax1, fig2, ax2, fig3, ax3 = mc.plot(log=True)
```



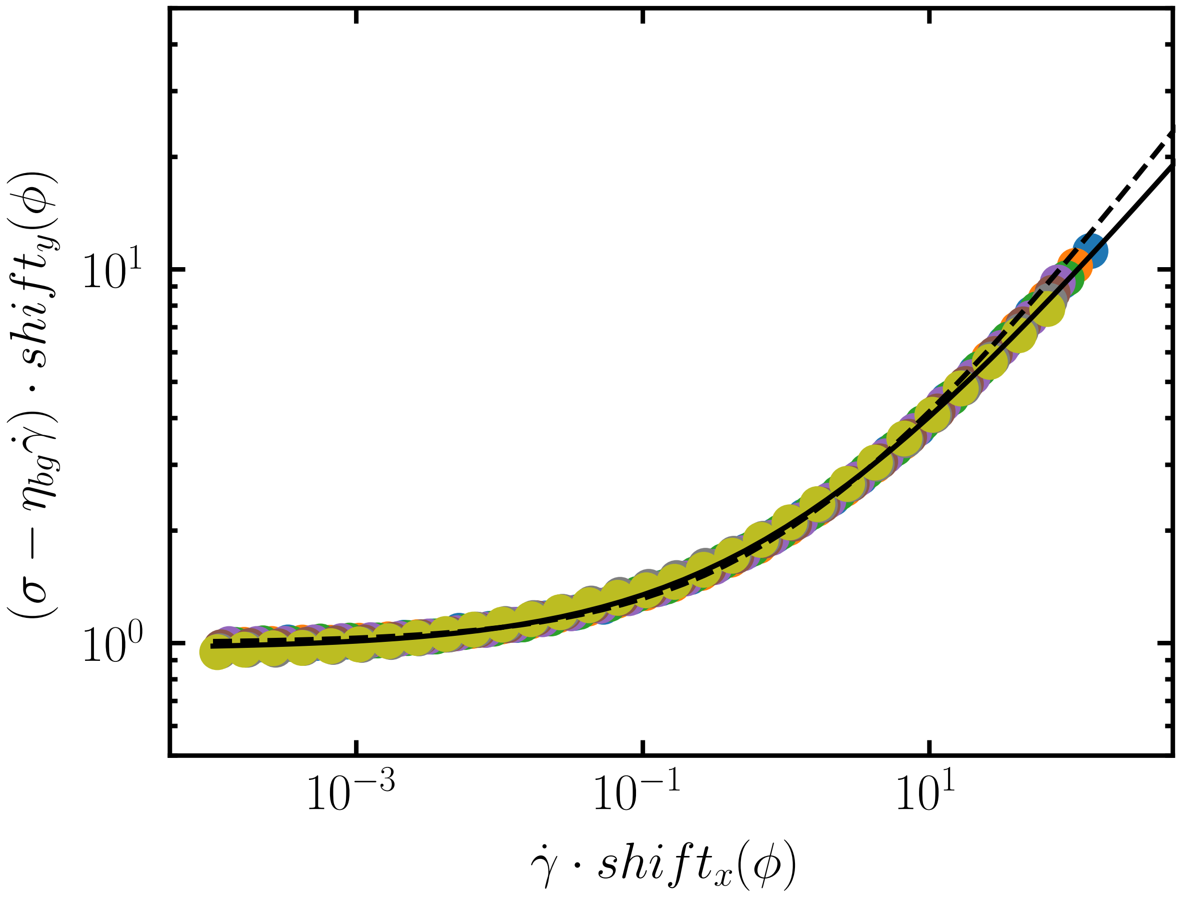


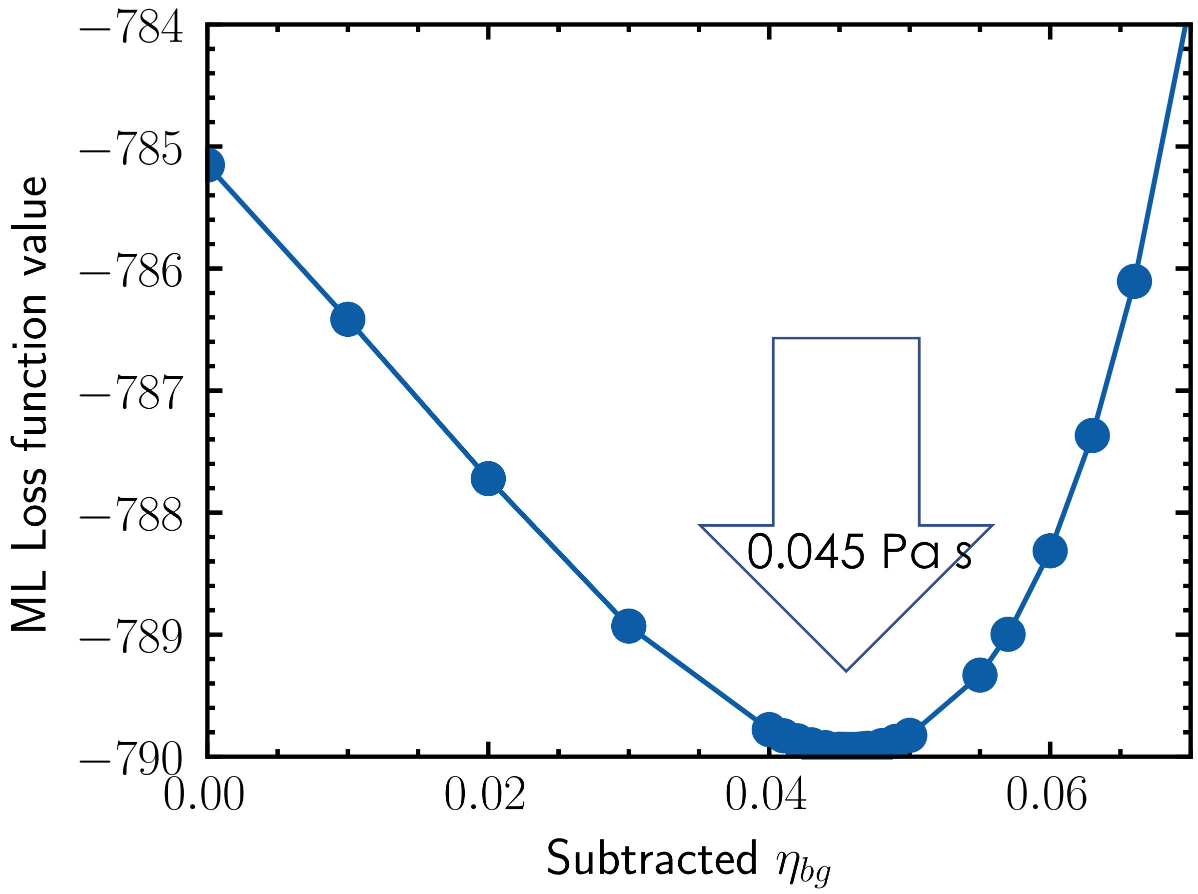
Can we gain physical insights from the shift factors?



Subtraction of a **common** background

Subtract $\eta_{bg}\,\dot{\gamma}$ from the stress data before passing to ML Mastercurve optimization tool



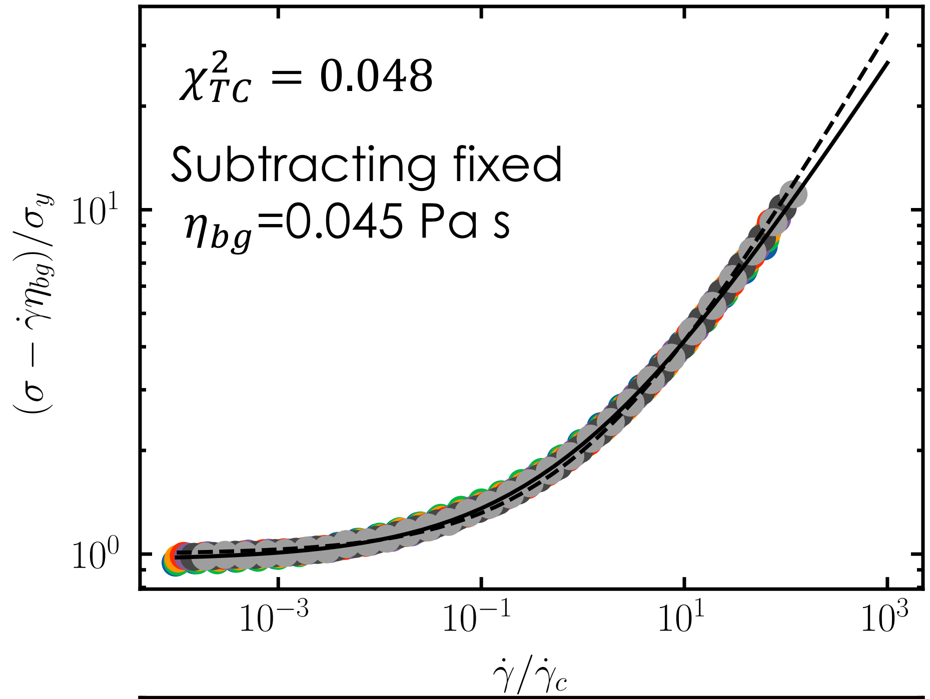


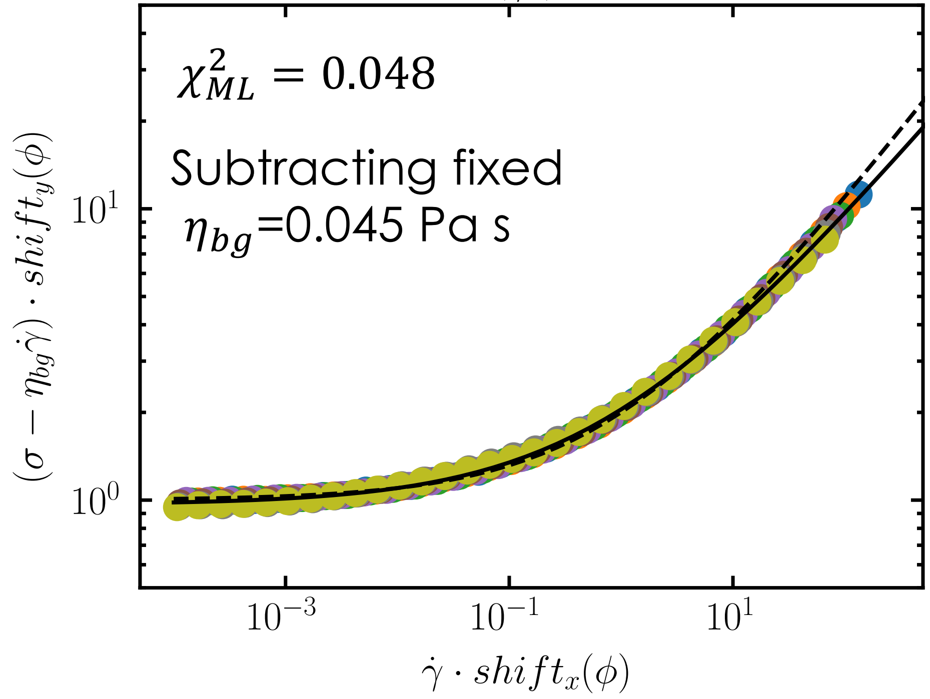
HB parameter	value	standard error
σ_y (Pa)	0.96	0.002
K (Pa.s^n)	1.08	0.005
n	0.45	0.001

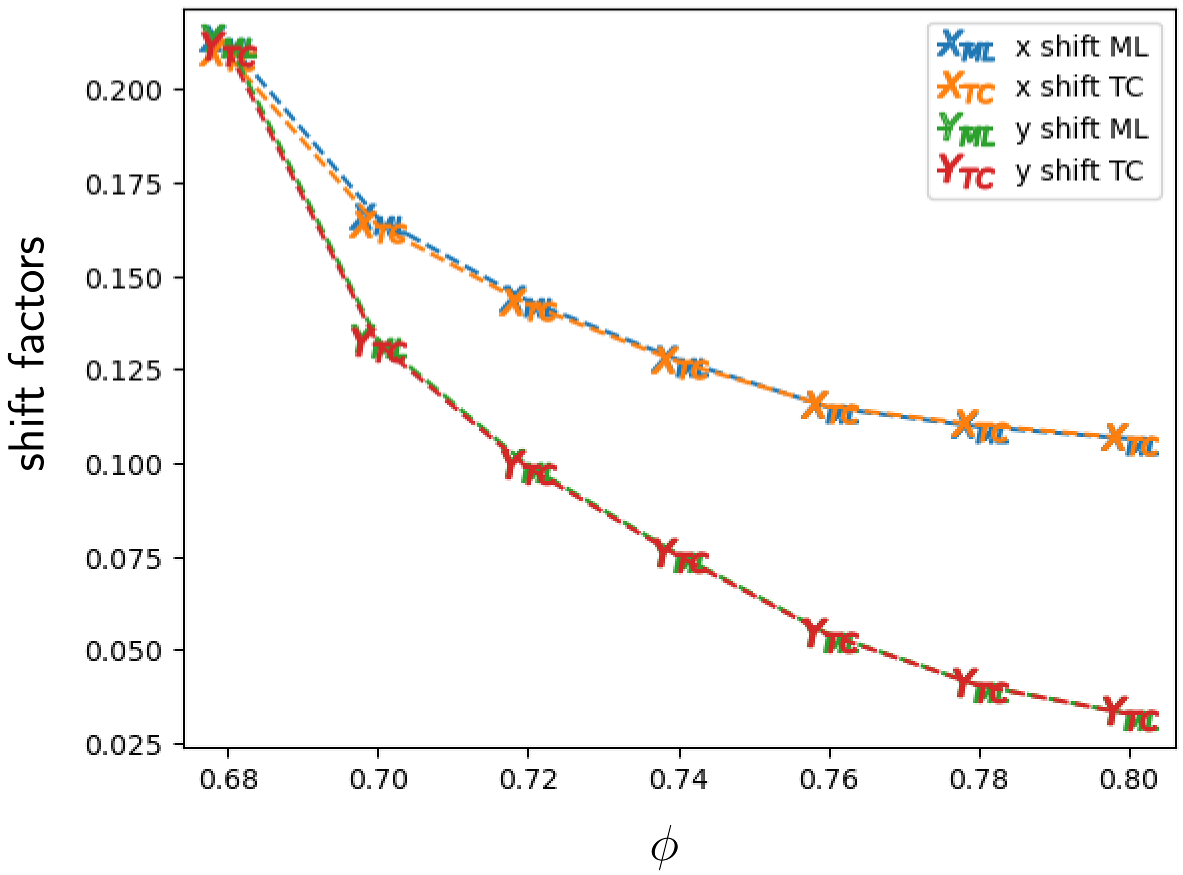
$$\chi^2_{ML} = 0.048$$

Physical insights from shift factors

Model based mastercurve (TC) and ML mastercurve support model validity

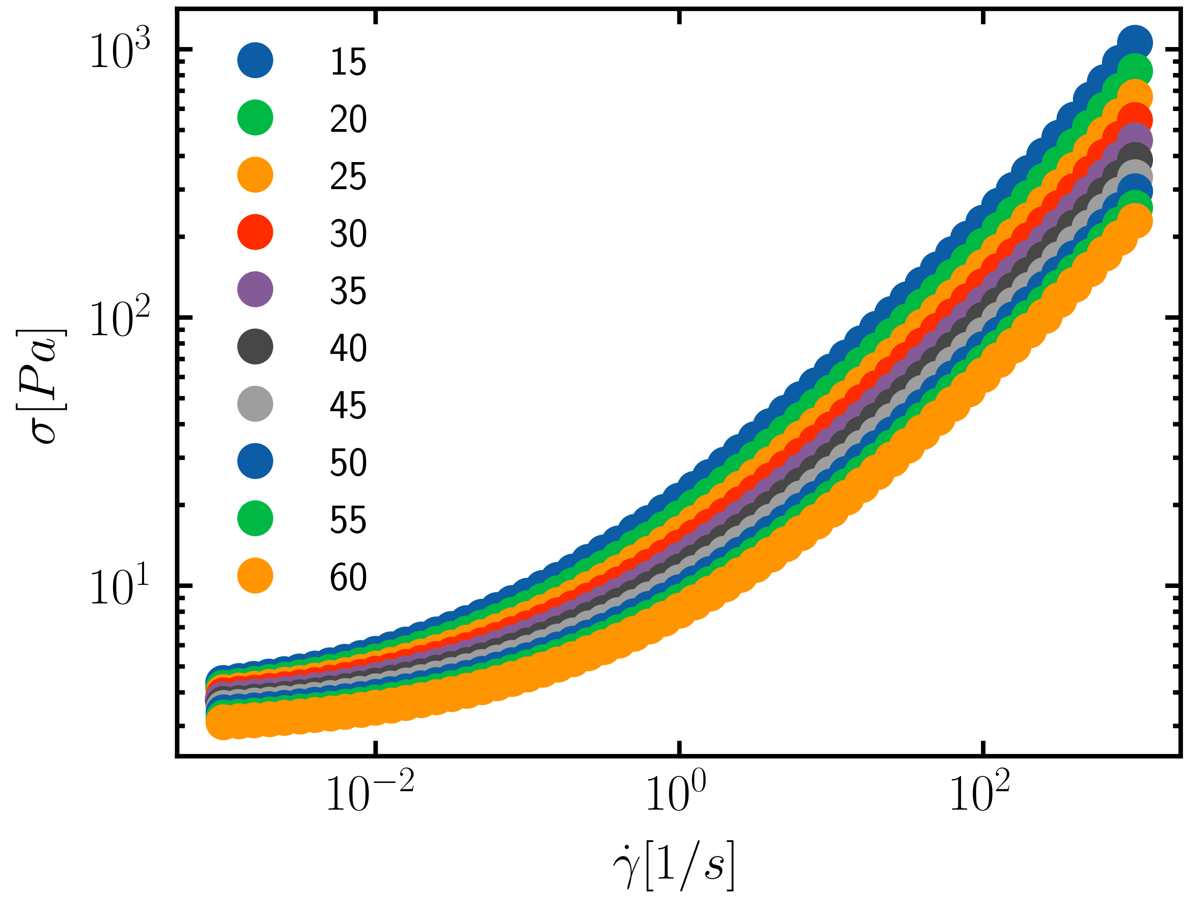


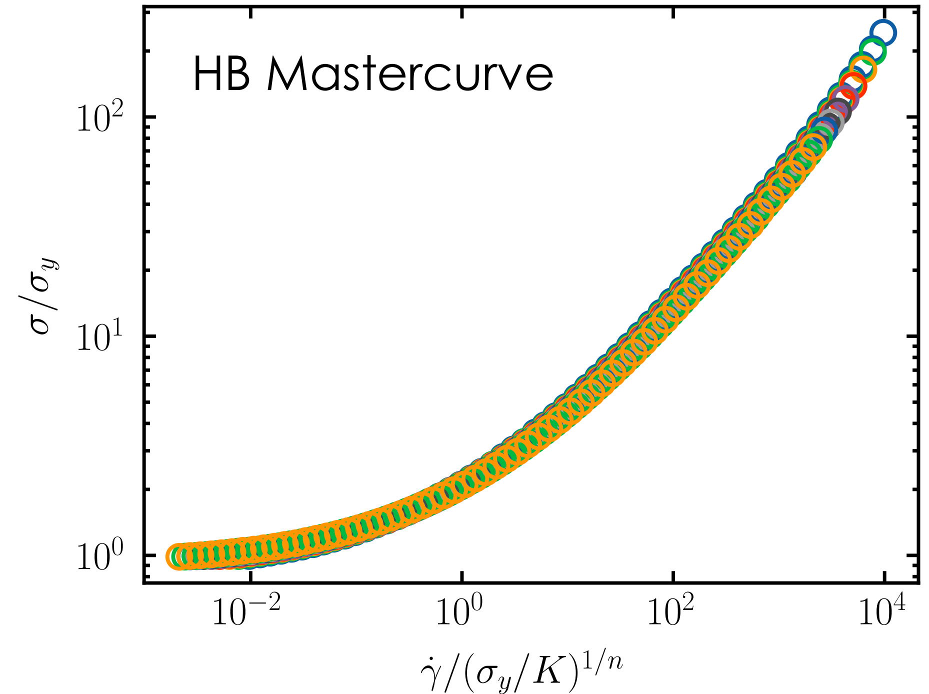


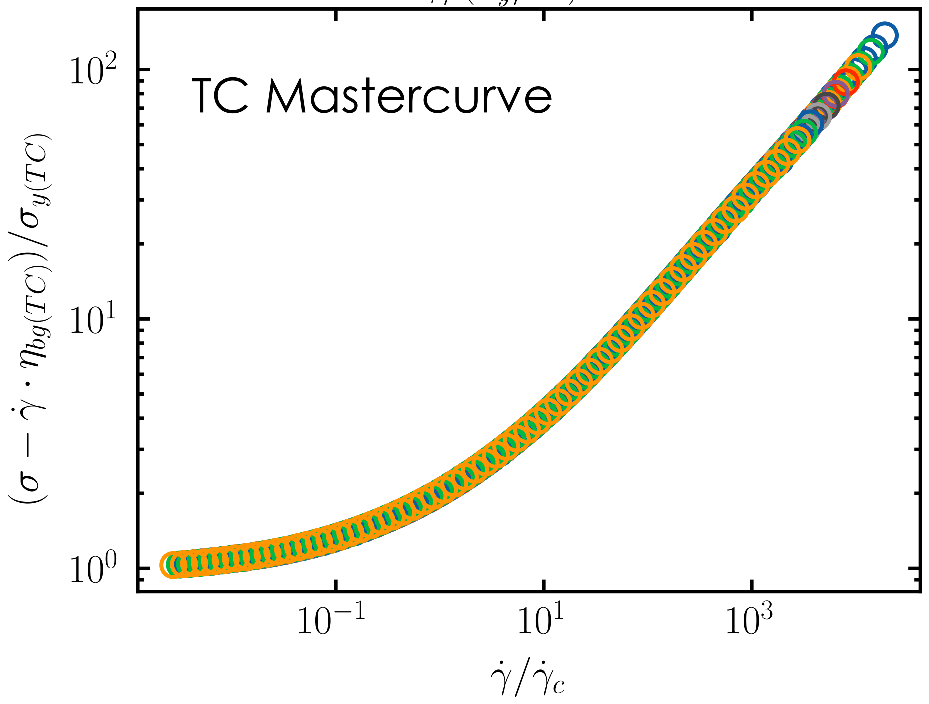


More, challenging, cases

Carbopol microgel in Propylene Glycol
Multiple datasets as a function of temperature



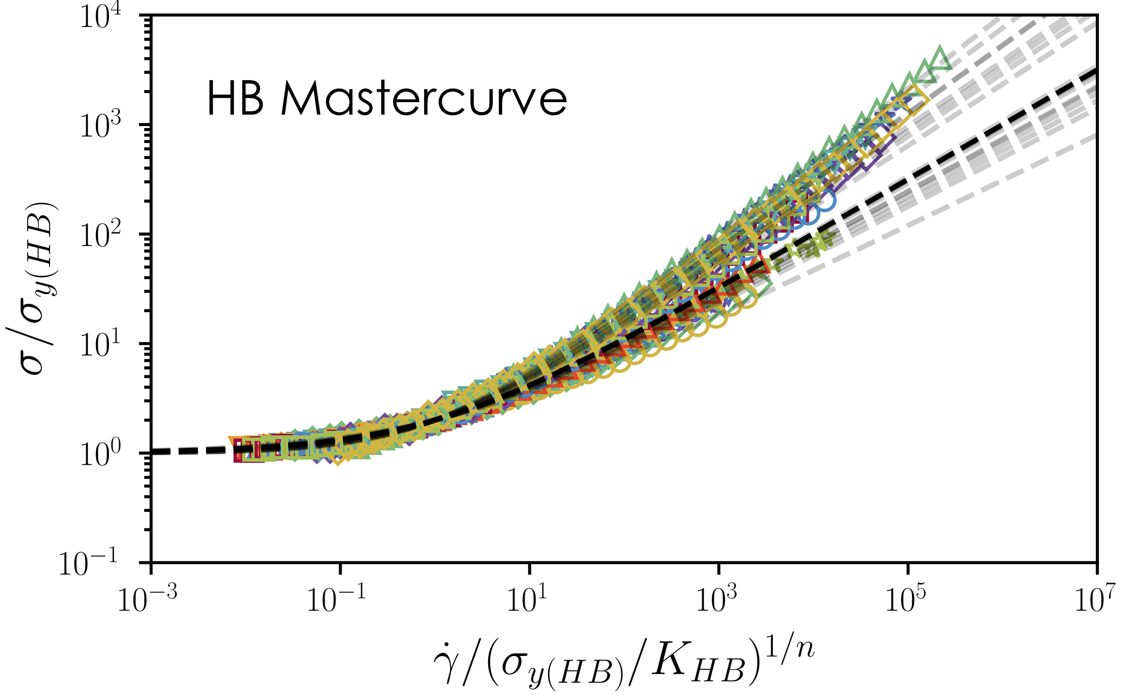


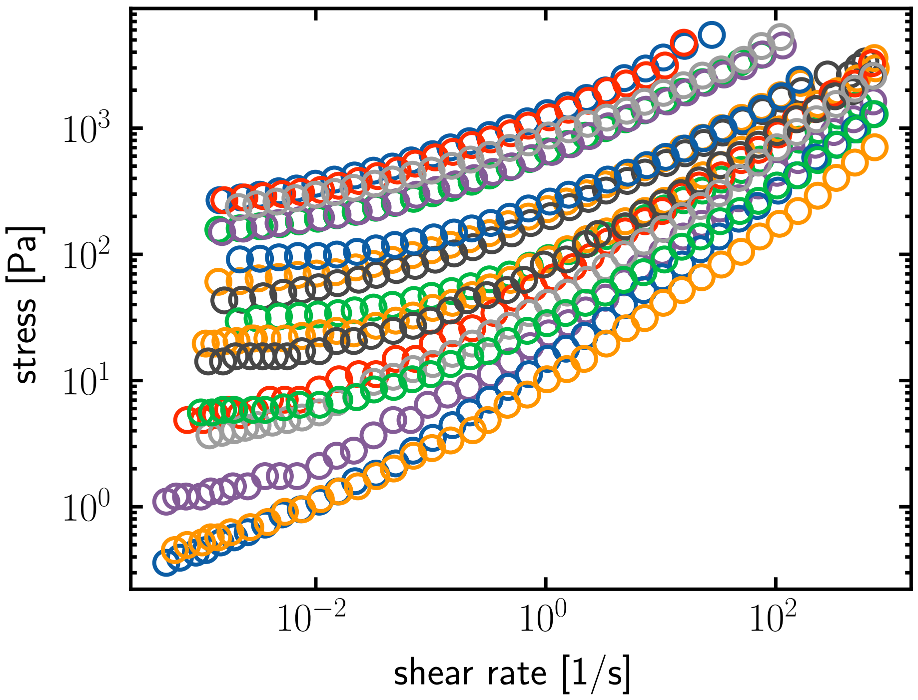


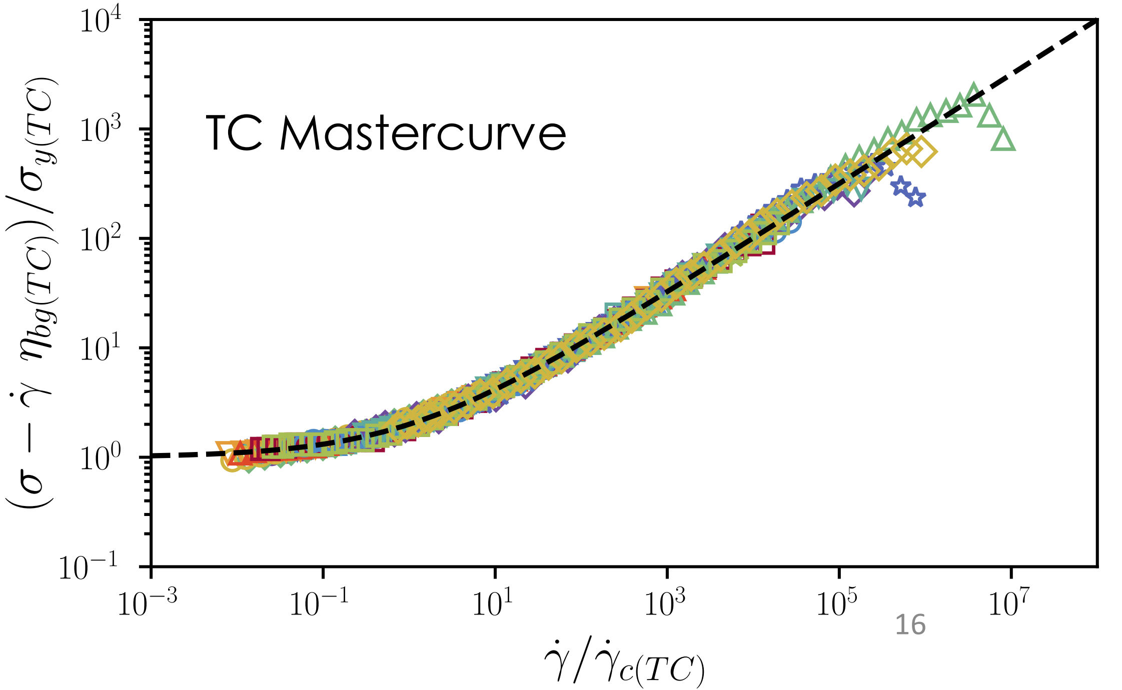
More, challenging, cases

Migliozzi, Simona, et al. "Investigation of the swollen state of Carbopol molecules in non-aqueous solvents through rheological characterization." Soft Matter 16.42 (2020): 9799-9815.

Carbopol Glycerin, PEG, and PEG/Gly
Multiple datasets as a function Carbopol concentration 1-8%
And solvent



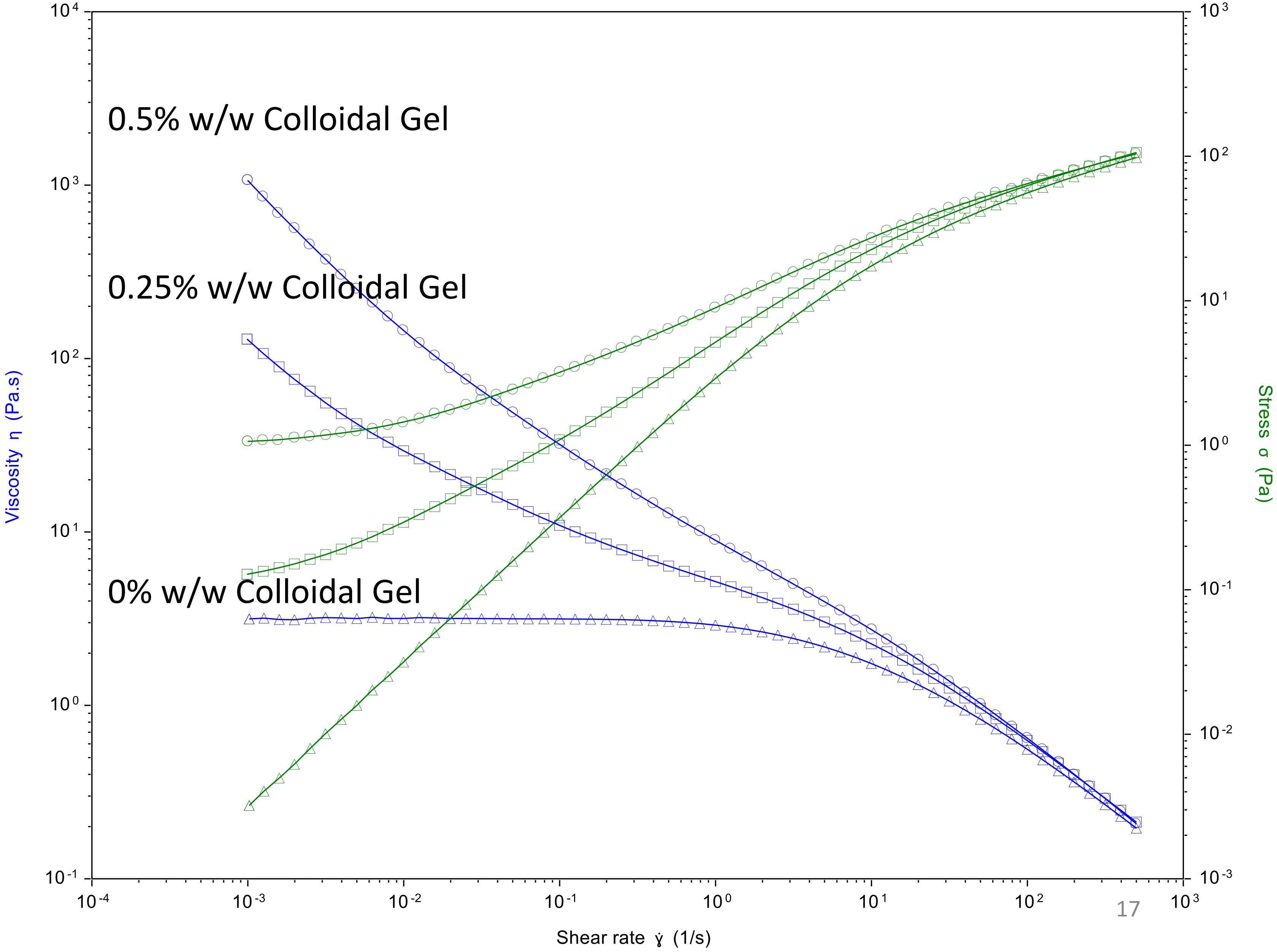




More, challenging, cases

Colloidal gel added to non-Newtonian fluid

Subtraction of a non-Newtonian background



Conclusions

- Mastercurves provide important physical insights: fundamental and practical
- ML tool represent objective ways to explore and judge different model/approaches: TC seems more reasonable than HB
- Open-source tool-chain: interesting new way to develop and share code
- Many opportunities to extend ML mastercurve to more complex cases