White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Leeds.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/158017/

Version: Accepted Version

Article:
Boudara, VAH, Read, DJ and Ramírez, J (2020) REPTATE rheology software: Toolkit for the analysis of theories and experiments. Journal of Rheology, 64 (3). pp. 709-722. ISSN: 0148-6055

https://doi.org/10.1122/8.0000002

© 2020 The Society of Rheology. This article may be downloaded for personal use only. Any other use requires prior permission of the author and AIP Publishing. The following article has been accepted for publication in Boudara, VAH, Read, DJ and Ramírez, J RepTate rheology software: toolkit for the analysis of theories and experiments. Journal of Rheology 64:3, 709-722. ISSN 0148-6055 and will be found at https://sor.scitation.org/doi/10.1122/8.0000002 when published. Uploaded in accordance with the publisher's self-archiving policy.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

RepTate rheology software: toolkit for the analysis of theories and experiments

Victor A. H. Boudara and Daniel J. Read

School of Mathematics, University of Leeds, LS2 9JT Leeds, U.K.

Jorge Ramírez

Department of Chemical Engineering,

Universidad Politécnica de Madrid, Madrid, Spain

Abstract

We present the new, free and open source, RepTate [Rheology of Entangled Polymers: Toolkit for Analysis of Theory and Experiment] software package for viewing, exchanging and analyzing rheological and associated data. The main idea of RepTate is to propose a powerful and user-friendly platform which can be installed on the same computer as, e.g., the rheometer, and which makes comparing experiments with classical, or latest, theories easy – without the need of a theoretician. The new RepTate software offers full compatibility with different operating systems (Windows, Mac, and Linux). We demonstrate the use of RepTate by reproducing predictions of recently published articles, from entangled, mono- and polydisperse, linear chains to branch-on-branch polymer systems, in the linear and nonlinear rheology regimes.

I. INTRODUCTION

In any scientific discipline, it is very common to desire to analyse experimental data and to compare these against the latest theories: either to extract parameters representing the data, or to test the theory against the reality of experiment. In many cases, parameters extracted from one experiment might then be applied to the modelling of a different experiment on the same system or material. The field of rheology is no different: we perform multiple different types of experiments on the same material (linear viscoelasticity, nonlinear shear, extension, MAOS, LAOS, etc.), often accompanied by structural probes such as scattering, hoping to learn about the material and to describe all experiments with a consistent set of theories and parameters. The purpose of this paper is to introduce a software tool, "RepTate", designed to facilitate this process of viewing and processing data from standard rheology experiments and of comparing the latest rheology theories to the data.

The original Reptate software ("Old-RepTate") was initially developed as a part of the Microscale Polymer Processing 2 (μPP^2) project funded by the UK Engineering and Physical Sciences Research Council, in 2006. The development of the software was instigated by Alexei Likhtman, who was at the time a newly appointed lecturer with recent memories of his postdoctoral work in the first μPP project. He observed that, as the postdoctoral researcher in charge of theoretical developments, a substantial amount of his time was spent performing relatively simple data fitting exercises for experimentalists who did not have easy access to the codes for calculating theories. Keen to avoid a similar fate for his own postdoctoral charges, he conceived the idea of a software tool which could be delivered to the experimental groups, allowing them to use the theories for themselves (and to test multiple theories against the same data set). The resulting original RepTate software was created using Borland Delphi (a Pascal based coding language) by Jorge Ramirez and Alexei Likhtman with additional contributions from many collaborators at the University of Leeds, UK. Some improvements and advanced features were added to the code while the authors were at the University of Reading. Many members of the muPP^2 project (Richard Graham, Dietmar Auhl, Daniel Read, Kamakshi Jagannathan and Bart Vorselaars) added particular modules and theories to Old-RepTate. The name RepTate ("Rheology of Entangled Polymers: Toolkit for Analysis of Theory & Experiment") can be attributed to Sathish Sukumaran. Beyond the muPP^2 project, the Old-RepTate software acquired a small user community

and gained use (for example) in education of postgraduate researchers in various doctoral training programmes, allowing them to work with the state-of-the-art theories in learning rheological concepts.

It may be noted that Alexei Likhtman's idea was not completely unique. For example, Winter and co-workers [1] created the IRIS "Rheo-Hub" rheology tool-kit, specifically for addressing linear and nonlinear viscoelastic data, which includes theories such as Milner-McLeish dynamic dilution [2, 3], Larson's hierarchical model [4–6], Wagner's MSF model [7], and the NAPLES slip-link code [8]. The IRIS software, however, requires a licence to access, with separate licences for each individual module or theory. Even so, Old-RepTate was similar in its underlying philosophy to IRIS in that it provides a common framework within which multiple theoretical tools can be applied to data. In a different vein, Doi and co-workers created the OCTA suite of programmes [9], but there the specific focus was on simulation methods rather than on analysing and fitting data.

More recently, Larson's Hierarchical model [4–6] and the "Branch-on-Branch" (BoB) model [10–12] have been released as freely-available software for predicting rheology of entangled branched polymers, using tube model concepts. Despite powerful predictive capability, these are command-line driven codes with no built-in capacity for viewing or analysing experimental data. Furthermore, they are codes written to perform a single, albeit complex, task (rheology prediction for branched polymers) using a single theory. As will be seen below, RepTate aims at a much broader goal: it provides a framework within which experimental data (of many different types) can be viewed and compared against multiple candidate theories.

A substantial number of theories (including an implementation of BoB [10–12]) have already been added to the code, but we hope and expect that more will be included. It should be noted that in seeking to include many different theories within the same package, we have not aimed to make all the theories in the package perfectly consistent with one another. For example, the Rolie-Double-Poly [13] and BoB models do not make exactly the same prediction for linear rheology of polydisperse linears, and both differ from the Likhtman-McLeish theory [14] in prediction for monodisperse linear polymers. Such variations are inevitable where different theories make different assumptions or approximations. Likewise, some theories assume the dilution exponent $\alpha = 1$ (e.g. theories based on double reptation, such as Rolie-Double-Poly) while others are coded such that it is a parameter that can be

varied (e.g. BoB or the Dynamic Tube Dilution theory for stars [2]). As more theories are added, it is likely that such variety will increase. In our view, this approach allows the user freedom to compare the different models in the most transparent way.

So, we consider that there remains strong potential for software such as RepTate to be useful to the rheological community and beyond. However, in seeking to update and maintain Old-RepTate, we faced two related problems with the original code: firstly it would only run on Windows based machines (whilst a cross-platform solution would be preferable) and secondly, due to changes in operating systems and software we could no longer compile the original code, let alone update it. Therefore, we have rebuilt the whole code structure of Old-RepTate into a modern and portable form. We use Python 3 programming language [15], and PyQt5 for the graphical user interface [16], which guarantees full compatibility and a "native" look on the different operating systems (Windows, Mac and Linux). For the numerical operations, we use NumPy and SciPy libraries [17], and to speed-up the calculations for the theories requiring heavy computations we use C or C++ code compiled into shared libraries. In addition, we developed a command line interface that allows to run calculations as batch processes. This is useful for theories that need long CPU times. We use Matplotlib library [18] to achieve high quality, publishable, figures: all data figures presented in this article are produced with RepTate.

Finally, we have made RepTate available in a free and open-source form, under a GPL v3 licence [19]. We believe that the software will be most useful if the whole community is able to access and update the code, adding new functionality, modules and theories. Although born within the rheology community, there is nothing that would prevent modules being created for many other forms of data (indeed there are already modules for molecular weight distribution, scattering and dielectric data).

In the remainder of this paper, we first outline the overall design and typical workflow for the RepTate software. We then present examples illustrating the use of the software to recreate results from several previously published works on entangled polymers.

II. DESIGN AND WORK FLOW

The online user manual for RepTate (`http://reptate.readthedocs.io/`) contains a significant amount of detail on the installation and operation of the software. Here, our

intention is to briefly summarise the main features and typical workflow.

A. Brief Description of RepTate's main window

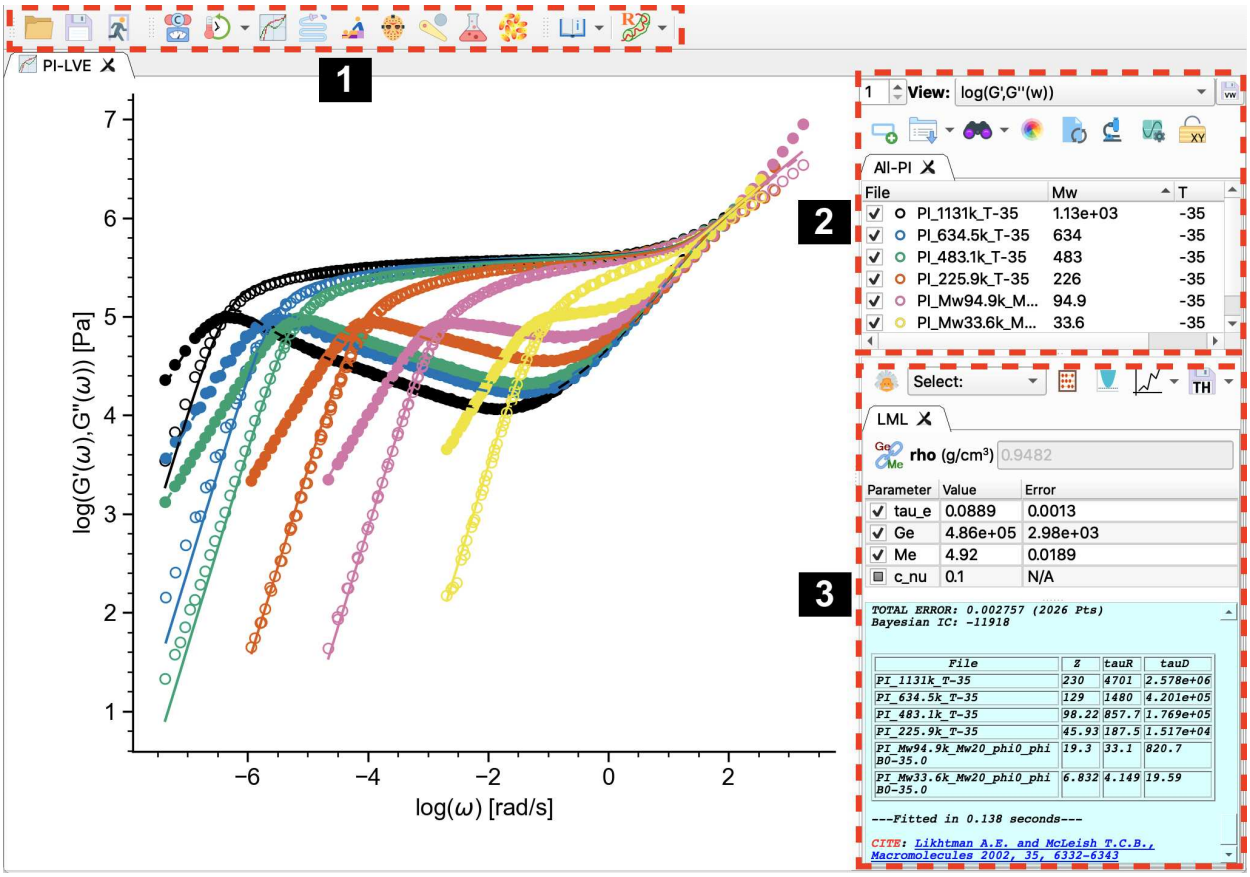

FIG. 1. RepTate window. Applications are found in the area "1", datasets in area "2", and the theories in area "3".

The upper part of the main RepTate window, zone 1 in Fig. 1, contains a toolbar giving access to several different *Applications* (such as Molecular Weight Distribution, Linear Viscoelasticity (LVE), Start-up of Shear or Extension, etc.). See Table I for the full list of *Applications*. The purpose of each *Application* is (i) to be able to display the typical experimental data associated with an experimental technique, (ii) to be able to view the data in different representations (e.g. LVE data can be viewed as storage and loss modulus, or complex viscosity), and (iii) to model the data using theories. Each *Application* contains three main panels: in the centre, a main plot panel in which the data are viewed; to the upper right a panel for loading and manipulating the data from files which can be grouped

into *Datasets* (zone 2 in Fig. 1); and to the bottom right a panel for *Theories* (zone 3 in Fig. 1). To the left, but hidden by default, a further panel can open up which allows the user to inspect the contents of data files, as well as to apply different filters to the data (called *Tools* in RepTate), and to access the *Materials Database*.

The internal modular data structure in RepTate reflects the underlying philosophy of the software, which is based on the idea that *theories are compared to experimental data*. Hence, *Datasets* are "contained" and viewed within *Applications*, and *Theories* are "contained" within *Datasets*, i.e. a *Theory* is only ever calculated in the context of an associated *Dataset*, to which it can be compared. Nevertheless, under some circumstances a user may wish to explore the behaviour of a theory in the absence of data. For this we provide the facility of "dummy files", as described in more detail in Section IV.

TABLE I. List of Applications in the current version of RepTate

Name	**Description**	**Theories**
MWD	Molecular weight distributions	Discretization, GEX, LogNormal
TTS	SAOS data and TTS	Automatic, WLF
LVE	SAOS master curves	Maxwell modes, Likhtman-McLeish, Carreau-Yasuda Rouse, Dynamic Tube Dilution (stars), BoB Rolie-Double-Poly
NLVE	Nonlinear flow	Rolie-Poly, UCM, Giesekus, Pom-pom Rolie-Double-Poly, BoB, PETS, GLaMM
Gt	Stress relaxation	Maxwell modes, Rouse, DTD (stars)
Creep	Creep experiments	Retardation modes
SANS	Small-angle neutron scattering	Debye
React	Monte Carlo polymerisation	Tobita batch, Tobita CSTR, Multi-Met CSTR React Mix, BoB, Diene CSTR
Dielectric	Dielectric spectroscopy	Debye modes, Havriliak-Negami modes, KWW modes

B. Typical workflow

Datasets are loaded into each *Application* from data files with a default file format specified for each application based on a columnar text data structure. At the header of each file it is possible (and sometimes required) to specify parameter values (such as temperature, molecular weight, chemistry) associated with the data. Alternatively, the user can load data from Excel spreadsheets (.xls or .xlsx format).

A typical workflow in using RepTate, to address data for a given material, would then be to open a series of *Applications*, and to load data files into each *Application* (thus creating *Datasets*). The user may then examine the data, for example by zooming into different regions of the data, or viewing the data with different representations: to facilitate this each *Application* has a standard set of *Views* which manipulates the data from the input file to produce a graph (e.g. in LVE, the input storage and loss moduli can be converted to many representations such as loss angle, compliance or complex viscosity). The user may also choose to represent the data in up to four simultaneous views, by changing the value at the left of the *Views* menu. Work is in progress to extend the *Views* functionality so that users can add their preferred *Views* to RepTate using simple algebraic expressions.

Following this the user might then apply *Theories* to those *Datasets*, perhaps determining theory parameters which best fit the data. By default, a theory's x-range is identical to the x-range of the data file it is applied to, but sometimes it is visually appealing or instructive to extend the theory prediction beyond the range of the data, and this option is available.

Having determined best fit parameters from one *Theory*, applied to a given *Dataset* in one *Application*, it might be these parameters should then be transferred and be used in evaluating another *Theory* in a different *Application*. RepTate facilitates this operation. A simple example of this would be in transferring parameters from linear to nonlinear viscoelasticity: in many nonlinear multimode models (e.g. multimode pom-pom [20, 21]) the standard procedure is to fit linear viscoelastic data using Maxwell modes (obtaining a discrete spectrum $\{g_i, \tau_i\}$ of modes). Following this, one would associate a nonlinear model with each Maxwell mode (e.g. in the pom-pom model requiring two further parameters $\{q_i, \tau_{s,i}\}$ - the priority and stretch relaxation time) in a fit to nonlinear data. Within RepTate, the $\{g_i, \tau_i\}$ parameters from a multimode Maxwell fit in the LVE Application can either be saved, and reloaded into a nonlinear viscoelastic theory, or (even more conveniently) passed

directly between Theories in the two different *Applications* within an open RepTate window.

C. How we deal with the fitting procedure

Some (but not all) *Theories* in RepTate are sufficiently fast to calculate that it is possible to automatically fit data by adjusting parameters of the *Theory*. RepTate provides the facility to do this, using a Trust Region Reflective algorithm [22], which is a generally robust method for large problems with bounds. It is possible to select which parameters in a *Theory* will be automatically adjusted to achieve the fit, i.e. the fit can be achieved by adjusting a subset of the parameters in a given *Theory*. By double-clicking on a parameter name, users can configure how each parameter is handled during the minimization procedure (by setting bounds, a weight factor, or a set of discrete values that the parameter can take).

As with any fitting procedure, a function should be minimised in order to determine the best fit. At present in RepTate we use a simple least-squares procedure, based on minimising the sum of the squares of the deviation between theory and experiment. This deviation is calculated based on the y-values of theory and experimental data as shown in the **current** *View* of the *Application* (when multiple *Views* are displayed, only the first one is used for fitting). For example, in the LVE application, if the current view is $\log(G'(\omega))$, the deviation is calculated as $(\log(G'_{\text{exp}} - \log(G'_{\text{theory}}))^2$, whereas if the current view is G' it is calculated as $(G'_{\text{exp}} - G'_{\text{theory}})^2$. It is also possible to set x and y ranges (with respect to the current View) that will be considered in the calculation of the error by clicking on the "Change limits for fit" button in the *Theory* region of the RepTate window (area 3 in Fig. 1). Note that the best fit parameters for a theory do depend, to some extent, on the current *View* and fitting range in RepTate. In the future, this may be improved by implementing a method recently outlined by the Ewoldt group for carrying known measurement errors between different representations of the same data [23], and invoking Bayesian inference to decide which parameters, and which theory, best fits a given data set [24]. For the time being, RepTate prints in the *Theory* text box (area 3 of Fig. 1) the Bayesian information criterion (BIC), every time a theory is calculated. The BIC is defined by the number of data points, n, the residual sum of squares (RSS), and the number of free fitting parameters, p, as

$$\text{BIC} = n \log(\text{RSS}/n) + p \log(n). \tag{1}$$

In general, the model with the lowest BIC value is preferred.

D. Exporting publication quality figures

All the visual elements in the plot (axes, labels, symbols, legend, comments and lines) can be customised in RepTate. Most of these features are accessible from a menu button. When the user is satisfied with a certain *View* of the experimental data and the fitted theories, the plot can be exported as a high resolution bitmap or a vector quality picture (e.g. eps or pdf format), ready for publication. Also conveniently, the software allows to copy the current graph to the clipboard, so it can be pasted in a word processing or presentation software.

All the data figures in the current paper were produced with RepTate.

E. Inspecting and manipulating the data

The experimental data loaded from files can be inspected and manipulated by opening the data inspector panel. In the panel, the data of the currently selected file in the *Dataset* is displayed in a spreadsheet-like table, where individual data points can be eliminated. The complete data table, or a subset of it, can be selected, copied, and pasted into other software.

Within the data inspector panel, the user can activate a "data shift" operation, which allows shifting the data of the selected file, as shown in the current view, vertically and/or horizontally, by dragging the data points with the mouse. The resulting shift factor is shown in a small table at the top of the data inspection panel. This feature can be very useful, for example, to extract the shift factors of experimental data when none of the standard time-temperature superposition methods (i.e. Arrhenius or WLF equations) work. The shift factors can be exported, copied or saved to a separate file for later use.

F. Tools and Materials Database

In the data inspector panel, below the data table, there is a *Tools* panel that allows the user to apply different filters to the data before it is represented or fitted in the current *View*. There are several tools available ranging from a simple filter that allows to set upper and lower bounds to the data (in both x and y axes) to more sophisticated ones, such as

an integral or a peak finder. See Table II for a full list. The *Tools* can also be applied to results of *Theories*. It is important to note that the tool filters affect the fit in RepTate. For example, if we set a lower bound to the frequency in a plot of the complex modulus, the experimental (and theoretical) data points whose frequencies are below the threshold will not be shown in the plot and will be discarded during the fitting procedure.

A particular *Tool*, the *Materials Database*, deserves a more detailed description. This tool stores the values of well established materials parameters so they can be used during the fitting procedure. For example, in the Time-Temperature Superposition *Application*, if we open a set of small-angle oscillatory shear data files of a polybutadiene sample measured at different temperatures (the chemistry *PBd* can be specified in the header of each data file), and then open the Williams-Landel Ferry theory to perform TTS and build the master curve. Then the stored WLF parameters for polybutadyene will be loaded automatically from the Materials Database. If the parameters for a particular material are not available in the Materials Database, the user can manually introduce their values and save them for later use.

TABLE II. List of Tools contained in the current version of RepTate

Tool	Description
Bounds	Remove points outside some prescribed boundaries.
Evaluate expression	Calculate algebraic expressions of x and y.
Find Peaks	Find maxima or minima in the current view.
Gradient	Calculate the derivative of y with respect to x in the current view.
Integral	Calculate the integral of y with respect to x in the current view.
Smooth	Smooths the current view data by applying a Savitzky-Golay filter [25].
Materials Database	Stored values of theory parameters for particular polymers.

G. Interlink between *Theories*

The current version of RepTate has 9 *Applications*, listed on Table I and depicted schematically in Fig. 2. Some applications can exchange data in the form of master curves, transformed experimental data or theory parameters. For example, starting from experimental

oscillatory shear data, and using the TTS module, one can build a master curve that can be imported in the LVE module; then, Maxwell modes can be fitted to the moduli, G' and G'', and the modes can be exported so they can be used in the NLVE module to predict, or fit, start-up of shear flow. Another interesting example of data exchange between RepTate *Applications* is the transformation from the time to the frequency domain (i.e. relaxation modulus $G(t)$ to complex moduli $G'(\omega)$ and $G''(\omega)$) using the i-Rheo algorithm [26], also implemented in RepTate. The details about the contents and structure of the files that can be read by each one of the applications can be found in the official documentation of RepTate (`http://reptate.readthedocs.io/`), and the theories included in each application in the current version of RepTate are listed in Table I.

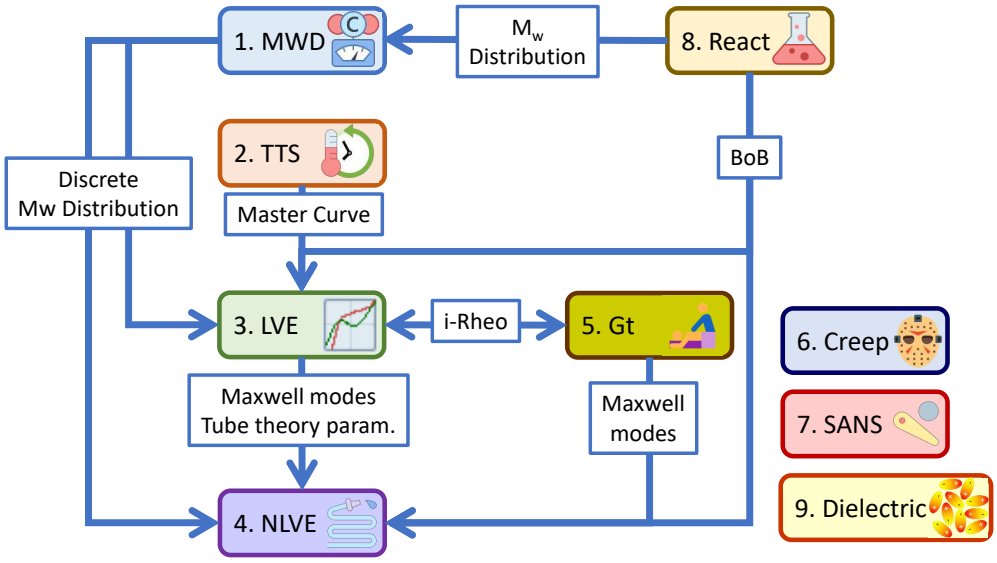

FIG. 2. List of the applications included in the current version of RepTate, along with some of the data that can be exchanged between them.

H. Project files

When working with RepTate, it is common to have several *Applications* open, each of them containing one or more *Datasets* to which theories have been applied. It is sometimes convenient to save the current workspace to a file so it can be reopened later or shared with collaborators. This can be done by clicking the "Save project" button in the main RepTate window, zone 1 of Fig. 1. The project file contains information to restore all the *Applications*,

Datasets, data files, visualization options, *Theories* and theory parameters that are open in the current RepTate session. Even though the project file is written in binary form, it can be exchanged between different operating systems.

III. RESULTS

A. Linear and nonlinear rheology of polyisoprene melts from Auhl *et al.* [27]

As a way to explore the modules **MWD**, **TTS**, **LVE**, and **NLVE** of RepTate, we demonstrate, in this section, reproduction of the results presented in Auhl *et al.*, where the authors studied nearly monodisperse, entangled linear polyisoprene (PI) polymers of various molecular weights.

In particular we will present: (i) the determination of the number- and weight-average molecular mass from size-exclusion chromatography (SEC) data; (ii) the application of the time-temperature superposition (TTS) principle to small amplitude oscillatory shear (SAOS) data to produce mastercurves; (iii) the fit of the Likhtman-McLeish theory [14] to the mastercurve, obtained in (ii), of all the PI samples, simultaneously, to get the material parameters (entanglement modulus G_e, entanglement relaxation time τ_e, and average molecular weight between entanglement M_e), and (iv) the GLaMM model [28] predictions of the nonlinear shear rheology using the entanglement parameters determined in (iii).

1. Molecular weight distribution

In Fig. 3a, we present the molecular weight distribution of twelve PI samples with 2 kg/mol $< M_w <$ 10^3 kg/mol. In Fig. 3b, we exemplify the discretization of a molar mass distributions using the 200 kg/mol PI sample with the "Discretize MWD" theory of the "MWD" application in RepTate. The results of the discretization are shown as a bar plot where

(i) We chose to have 10 molecular weight bins;

(ii) The bin edges (red triangles) have been adjusted manually to avoid unnecessary bins at the distribution tails (by default, bins are equally spaced on log-scale);

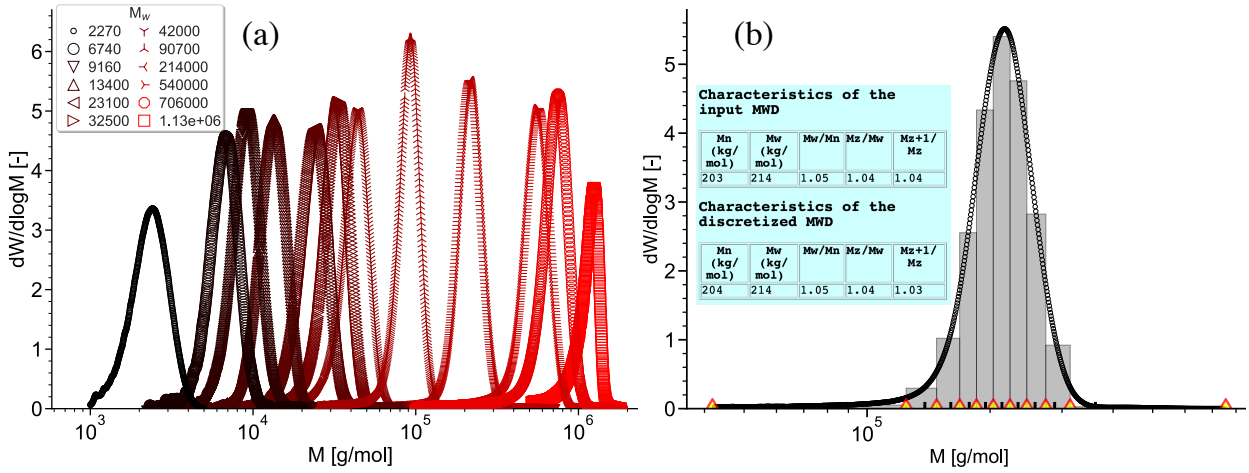

FIG. 3. (a) Experimental differential molar mass distributions $\mathrm{d}W/\mathrm{d}\log_{10} M$ from RALS signal versus molar mass M from Ref. [27] for various PI samples, as indicated in the legend (in g/mol). (b) Discretized molecular weight distribution (bar plot) of the SEC data (symbols) of the PI-200k sample. The insert shows the information printed in the *Theory* text box.

(iii) The area of the ith bin (height times width in log-scale) is equal to the area under the curve delimited by the bin edges. This defines the volume fraction, ϕ_i, of the ith component.

(iv) The molecular mass, M_i, is the weight-averaged molecular mass calculated over the bin width. It is represented by the thick tick-mark on the x-axis of Fig. 3b.

RepTate also displays additional information in the theory text-box, see insert of Fig. 3b: the number- and weight- average molecular mass of the discretized and input data, as well as the higher order moments of the molecular weight distribution, M_z and M_{z+1}, where $M_{z+k} \equiv \sum_i \phi_i M_i^{k+2}$, which characterize the high molecular weight tail of the distribution.

2. *Discussion of TTS*

The classic Williams-Landel-Ferry (WLF) parameters [29], C_1 and C_2, are not constants for a particular material but rather depend on the reference temperature to which the data is shifted (i.e. shifting to a different temperature requires different C_1 and C_2). In RepTate,

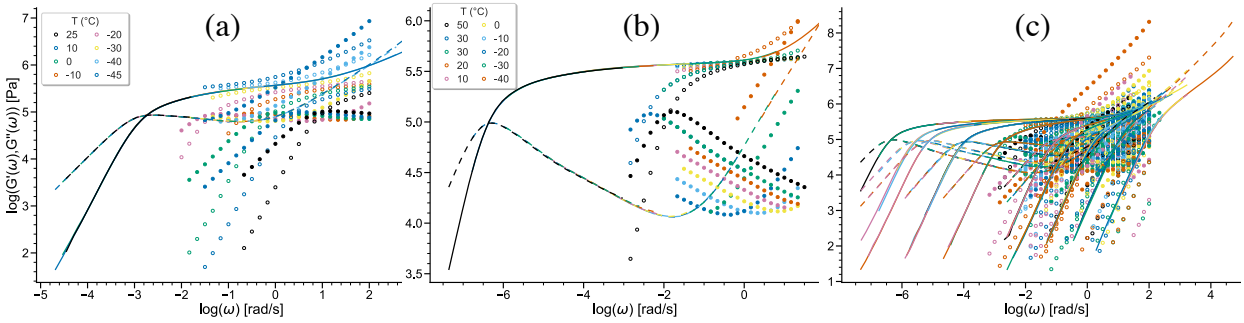

FIG. 4. Storage modulus G' and loss modulus G'' (full and open symbols, respectively) at different temperatures from [27] and result of the "WLF shift" theory, using unique material parameters $B_1 = 692$ and $B_2 = 114$ °C, to a reference temperature $T_{\text{ref}} = -35$ °C (dashed and solid lines, respectively). (a) PI88k sample, (b) PI1000k sample, and (c) all molecular weight PI samples shifted simultaneously.

we have defined two new parameters

$$B_1 = C_1 \times C_2, \tag{2}$$

$$B_2 = C_2 - T_{\text{ref}}, \tag{3}$$

where T_{ref} is the reference temperature to which the experimental data is shifted. These new parameters, B_1 and B_2, are independent of the reference temperature and, therefore, can be considered as material parameters. We can rewrite the classic WLF equations using the new temperature-independent parameters

$$\omega(T) = a_T \omega(T_{\text{ref}}), \tag{4}$$

$$G(T) = b_T G(T_{\text{ref}}), \tag{5}$$

$$\tag{6}$$

with

$$\log_{10} a_T = \frac{-B_1(T - T_{\text{ref}})}{(B_2 + T_{\text{ref}})(B_2 + T)}, \tag{7}$$

$$b_T = \frac{\rho(T_{\text{ref}})T_{\text{ref}}}{\rho(T)T} = \frac{(1 + \alpha T)(T_{\text{ref}} + 273.15)}{(1 + \alpha T_{\text{ref}})(T + 273.15)}, \tag{8}$$

$$T_g = T_g^{\infty} - \frac{C_{T_g}}{M_w}, \tag{9}$$

where α is the thermal expansion coefficient of the polymer at 0 °C, and C_{T_g} is a material parameter that describes the molecular weight dependence of the glass transition, T_g, and

M_w is the weight average molecular weight. The molecular weight dependence of T_g must be considered when shifting data of very short chains or very broad molecular weight distributions containing a significant fraction of short chains [27, 30]. This change is roughly related to the amount of free ends present in the sample. In order to take the effect into account, we use the following phenomenological expression, which was shown to fit the experimental data:

$$\log_{10} a_T = \frac{-B_1(T - T_{\text{ref}} + C_{T_g}/M_w)}{(B_2 + T_{\text{ref}})(B_2 + T + C_{T_g}/M_w)}, \tag{10}$$

In most cases, when shifting the data of well entangled, nearly-monodisperse polymers, this effect can be discarded by setting $C_{T_g} = 0$ in the theory box.

The "WLF shift" theory in RepTate shifts LVE data to a reference temperature using the above WLF equations to determine the shifting factors a_T and b_T. This is in contrast to other approaches in which data is first shifted via factors a_T and b_T chosen freely for each experimental temperature so as to achieve the best data overlap, then the shift factors are interpreted according to the WLF equations. In situations where the WLF theory is expected to work (e.g. homopolymer melts) the "WLF shift" theory in RepTate provides a more constrained and controlled shifting procedure with fewer fitting parameters, and this would be our preferred choice in such cases (Likhtman insisted on this procedure in creating the old-RepTate). Nevertheless, there are situations where WLF theory is not appropriate, and RepTate also provides a theory that allows to find, either manually or automatically, the temperature dependent shift factors a_T and b_T that provide the best data overlap. The found values of the shift factors can be exported to a file for later use.

In Fig. 4, we present the linear rheology data (G' and G'' as full and open symbols, respectively) at -40 °C $< T <$ 50 °C of each of the PI samples, 2.4 kg/mol $< M_w <$ 1131 kg/mol and the mastercurves (solid and dashed lines) resulting from the "WLF shift" theory in RepTate, where we have used $B_1 = 692$ and $B_2 = 114$ °C, independently of the choice of T_{ref}. We can save these mastercurves and open them in the LVE application in RepTate.

3. LVE and LM theory

Once an LVE mastercurve has been constructed, a range of options already exist within RepTate to compare the mastercurve with theory. These include fitting Maxwell modes to the curve; Dynamic Tube Dilution theory for stars [2]; the linear rheology version of the

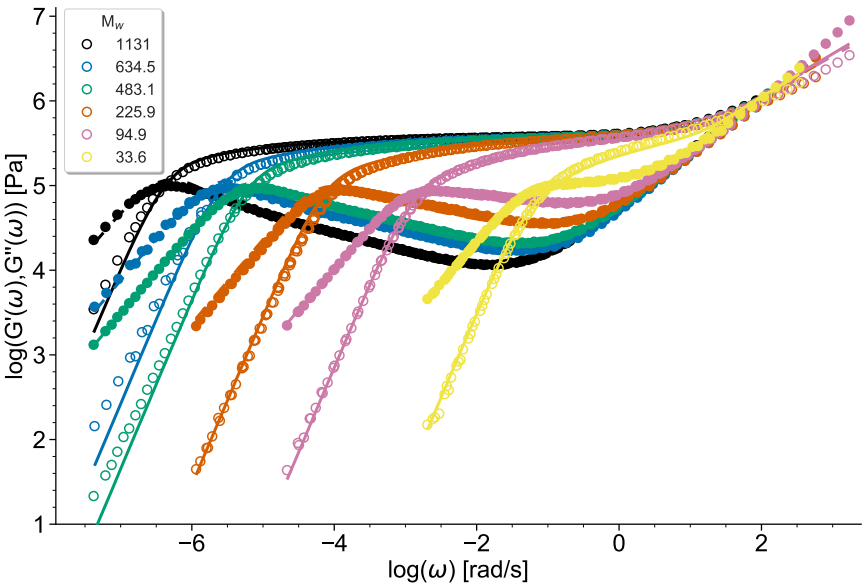

FIG. 5. Linear viscoelasticity of various PI samples at a reference temperature $T_{\rm ref} = -35$ °C (same as Fig. 4) [27] together with predictions of the Likhtman-McLeish theory [14] (lines) for all entangled samples ($M_w > 10$ kg/mol) as indicated in the legend (in kg/mol).

Rolie-Double-Poly [13] model; the BoB model for polydisperse linear or branched polymers [10–12]. Different theories may apply to different materials, and may involve different levels of approximation. We hope that the range of available theories can be expanded by ourselves and other contributors. In the present case, we use the mastercurves of Fig. 4 to extract the entanglement parameters of PI using the Likhtman-McLeish (LM) theory [14]. This is considered to be the state-of-the-art tube theory for near monodisperse linear entangled polymers, but we would not recommend its use for bidisperse or polydisperse materials, which are beyond its regime of applicability. The LM theory includes contour length fluctuation (CLF) and constraint release (CR) mechanisms and has four parameters: entanglement modulus $G_{\rm e}$, the average molecular weight between entanglements $M_{\rm e}$, the Rouse time of one entanglement segment $\tau_{\rm e}$, and a dimensionless parameter c_ν which is related to the effectiveness of the CR mechanism.

Figure 5 shows the best simultaneous fit (over all the PI samples) obtained in RepTate with the LM theory [14], fixing the constraint release parameter $c_\nu = 0.1$ and using three fitting parameters: $G_{\rm e}$, $\tau_{\rm e}$, and $M_{\rm e}$. The fitting procedure results in the following values (at $T = -35$ °C): $G_{\rm e} = 0.485$ MPa, $\tau_{\rm e} = 0.0894$ s, and $M_{\rm e} = 4.92$ kg/mol. It may be noted that

here we have treated the parameters G_e and M_e as separate fitting parameters, even though they are expected to be related via $G_e = \rho RT/M_e$, with the plateau modulus defined as $G_N^0 = \frac{4}{5}G_e$ within the LM theory. In the literature, this relationship is not always rigorously enforced: this approach may be supported by the argument that the $\frac{4}{5}$ prefactor in the plateau modulus is a result of a very specific model for the entanglement tube as a series of affinely deforming but rigid tube segments (whilst other models give different prefactors) and that the elasticity might not be fully entropic in origin. Nevertheless, we provide the option in RepTate to fix the relation $G_e = \rho RT/M_e$ exactly and so reduce the number of fitting paramters by one; this of course results in a slightly poorer quality of fit.

4. NLVE and SCCR theory

Using the entanglement parameters produced by the LM theory fit in the linear rheology regime previously ($G_e = 0.485$ MPa, $\tau_e = 0.0894$ s, $M_e = 4.92$ kg/mol and $c_\nu = 0.1$; the additional numerical prefactor R_S, related to the strength of the retraction rate, is set to 2), it is possible to predict, with no free parameter, the nonlinear rheology, in shear and elongation, of the entangled PI samples with the GLaMM model [28]. The GLaMM model has the advantage of being consistent with the LM model: it includes the same relaxation mechanisms (such as reptation, CLF and CR), it is based on similar approximations, and uses the same set of parameters (G_e, τ_e, M_e and c_ν).

In Fig. 6 we present the nonlinear shear rheology data (symbols) for the PI14k, PI30k, PI90k, and PI200k samples, together with the "GLaMM" theory calculated in RepTate (within a few minutes on a conventional laptop). Our intention here is to reproduce previously published results, and the predictions shown in Fig. 6 are identical to those presented in Ref. [27]. Nevertheless, we note that the predictions do not match experiment in every regard: there are deviations from the LVE for the lower molecular weight samples (GLaMM does not perfectly reproduce the Likhtman-McLeish LVE) and the higher flow rate data especially are typically overpredicted by the model.

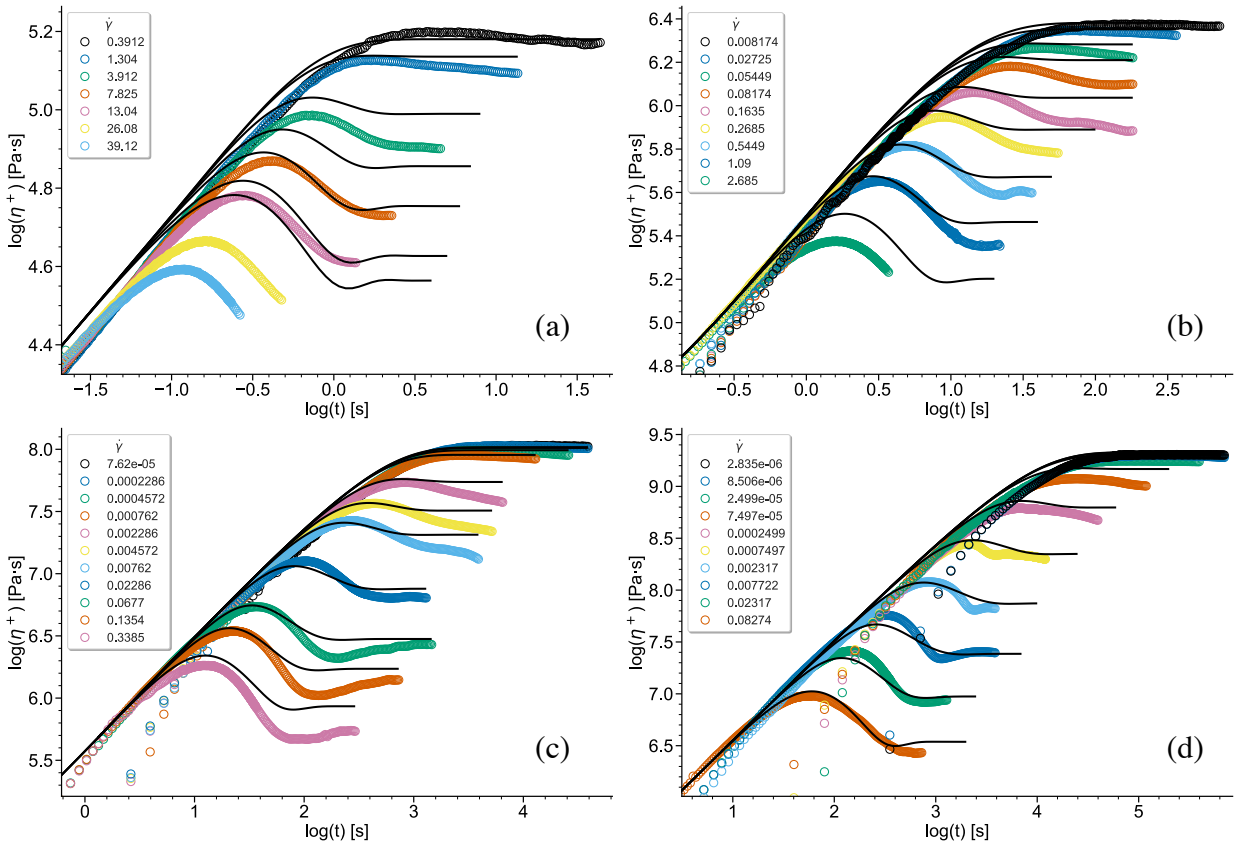

FIG. 6. Logarithm of the transient shear stress growth coefficient, $\eta^+(t)$, as a function of logarithm of time, t, at shear rates, $\dot{\gamma}$, indicated in the legend (in s^{-1}) for (a) PI14k, (b) PI30k, (c) PI90k, and (d) PI200k samples at $T = -35$ °C (symbols) from Ref. [27], together with the predictions of the GLaMM model [28] (lines) from RepTate.

B. LDPE structure and rheology predictions of Read *et al.* [31]

In Ref. [31], a number of low density polyethylene (LDPE) resins were characterised in terms of structural parameters (molecular weight distribution and g-factor from light scattering radius of gyration), LVE and nonlinear viscoelasticity (transient shear and extensional flow). A predictive scheme was proposed in which a Monte-Carlo algorithm was used to simulate the creation of molecules in a tubular reactor, and then BoB rheology software [10–12] was utilised to predict the linear and nonlinear rheology. Here, we recreate those predictions for one of the materials (LDPE2 of [31]), making use of the **React** *Application*, and BoB theory embedded within the **LVE** and **NLVE** *Applications* of RepTate.

The Monte Carlo simulation was performed using the algorithm due to Tobita [32] for

batch synthesis of LDPE, as an approximation of industrial tubular reactors. The parameters used in the algorithm are based on rates of free-radical reaction processes: polymerization (rate R_p, rate constant k_p); termination by disproportionation (rate R_{td}) and combination (rate R_{tc}); and chain transfer by long-chain branching (rate constant k_b), to small molecules (rate R_f), and by scission (rate constant k_s). Algorithm parameters are obtained from ratios of these rates, or rate constants:

$$\tau = (R_{td} + R_f)/R_p, \tag{11}$$

$$\beta = R_{tc}/R_p, \tag{12}$$

$$C_b = k_b/k_p, \tag{13}$$

$$C_s = k_s/k_p. \tag{14}$$

A further parameter, the conversion x_s, makes five parameters in total. The Supporting Information in Ref. [31] gives two recipes for simulating the LDPE2 material, making use of a weighted combination of Tobita batch simulations. The "alternative" parameterisation uses two such simulations ("Sim 1" and "Sim 2"), with parameters given in Table III. The last column of Table III gives the weight of each simulation in the combination.

TABLE III. Tobita simulation parameters for LDPE2 material.

	τ	β	C_b	C_s	x_s	Weight
Sim 1	1.1×10^{-3}	1.6×10^{-4}	1.8×10^{-2}	0	3.2×10^{-2}	0.2
Sim 2	1.1×10^{-3}	1.6×10^{-4}	1.8×10^{-2}	0	1.85×10^{-1}	0.8

Within the **React** *Application* the "Tobita Batch" theory allows these simulations to be performed. The simulation creates a requested number of molecules, evaluating the contribution of all of them to the molecular weight distribution. However it "saves" only a certain number of the molecules (evenly distributed across the molecular weight distribution) and writes them into an input file for BoB rheology. It does this by splitting the molecular weight axis into a number of "bins" and then saving molecules as they are created, up to a maximum number per bin. By adjusting these binning settings, it is possible to control the number of molecules saved into BoB input files. In the present exercise, the binning settings were adjusted so that roughly 800 polymers from Sim 1 and 1500 polymers from Sim 2 were saved. The "React Mix" theory of RepTate then permits the outputs from the two

simulations to be combined with the weights given in Table III, into a combined molecular weight distribution, shown in Fig. 7. The saved molecules can be output into a polymer configuration file for BoB software [10–12].

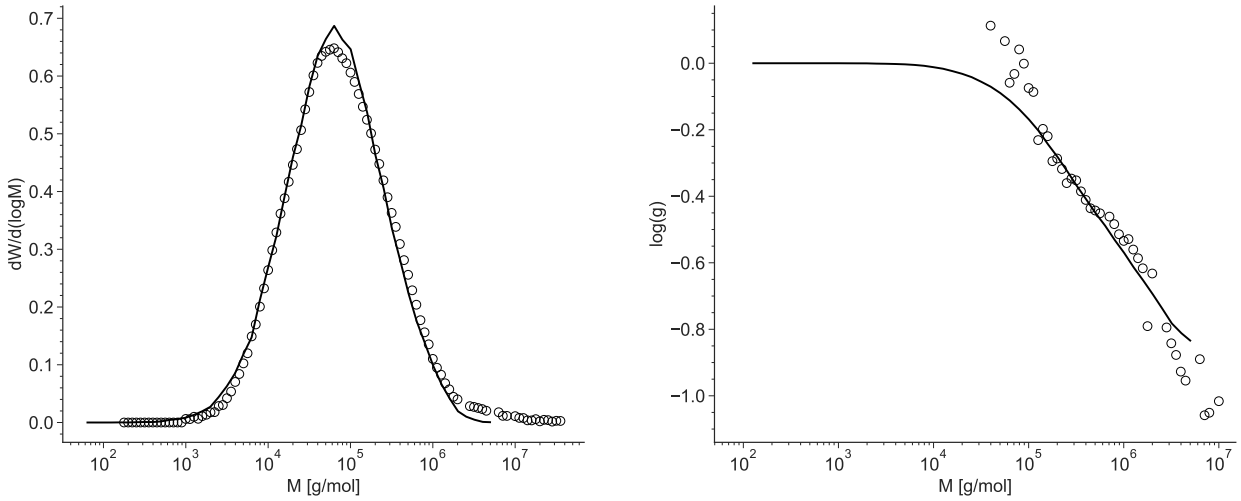

FIG. 7. Molecular weight distribution (left) and radius of gyration contraction factor, g as a function of molecular weight (right) for the LDPE2 material of Ref. [31] (symbols), together with predictions using Tobita batch and Mixture theories within the **React** *Application* (lines).

The **LVE** and **NLVE** *Applications* contain an implementation of the BoB algorithm, for prediction of linear and nonlinear viscoelasticity. Using the published material parameters for polyethylene [31] (M_e = 1.6 kg/mol, $\tau_e = 5.8 \times 10^{-7}$ s at T = 150 °C, with density 800 kg/m^3), we can use the BoB algorithm, together with the polymer configuration file generated from the **React** *Application*, to predict the linear rheology (Fig. 8) and transient extensional and shear stress growth coefficients (Fig. 9). For the polymer configuration file used (containing roughly 2300 molecules, in which the larger LDPE molecules may contain hundreds of branch points) calculation of the linear rheology is relatively fast, taking less than 3 minutes on a conventional laptop. Calculation of the full set of NLVE curves is more time consuming (roughly 2 hours on a laptop) since the nonlinear BoB code requires the algorithm to be run twice for each flow rate. However, once the rheological parameters are determined for linear rheology prediction, no further fitting is required for NLVE: the BoB algorithm has no further free parameters to adjust.

In general, the BoB model is designed to be predictive of the linear and non-linear rheology for entangled linear and branched polymers, both monodispese and polydisperse, and is often

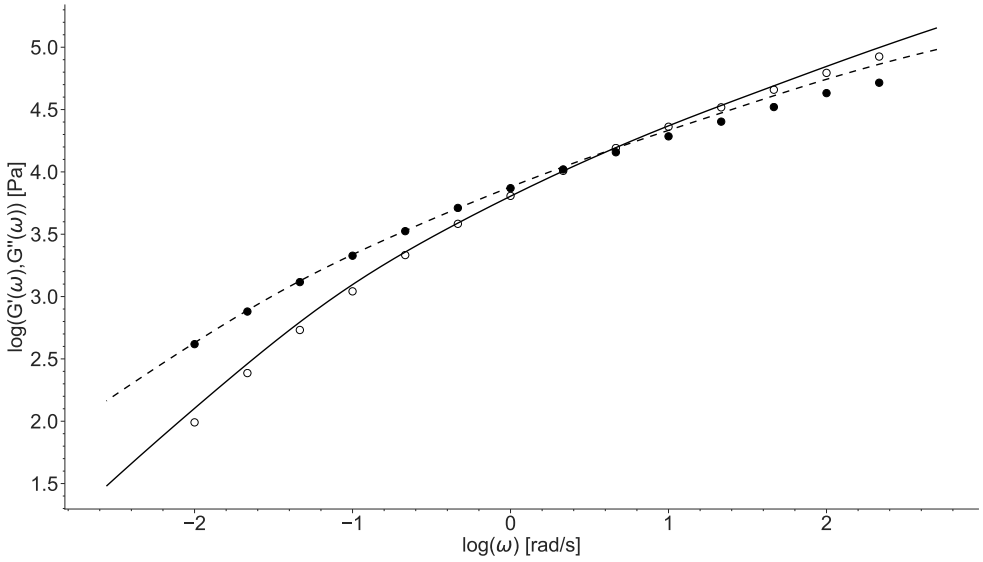

FIG. 8. Linear viscoleasticity at 150 °C for the LDPE2 material of Ref. [31] (symbols), together with prediction using BoB theory within the **LVE** Application (lines).

successful in doing this. It is nevertheless an approximate scheme which makes assumptions and approximations and is not universally applicable (one example of this is that the non-linear theory does not include stretching of linear chains, and so it will never successfully predict extension hardening in such materials - see the next section for an appropriate theory in that case). Recent reviews detailing some of the limitations and approximations of BoB and similar schemes can be found in Refs. [33, 34].

C. Polydisperse linear entangled polymers, Rolie-Double-Poly model [13]

In Ref [13], a range of experimental nonlinear shear and extensional rheology data of bidisperse or polydisperse linear entangled polystyrene and polyisoprene material, together with predictions using the proposed "Rolie-Double-Poly" (RDP) model, were discussed. In particular, the molecular weight distribution of a polydisperse linear entangled polystyrene "PSIV", described in Ref. [35], was discretized using 20 bins equally distributed on log-scale, as shown in Fig. 10. The discretization procedure of the "MWD theory" in RepTate is described in Section III A.

The experimental data of transient stress growth coefficient, η_E^+, for the "PSIV" sample of Ref. [35] at Hencky strain rates $\dot{\varepsilon} = 0.00075, 0.0001, 0.0015, 0.07, 0.3\ \mathrm{s}^{-1}$ are then imported

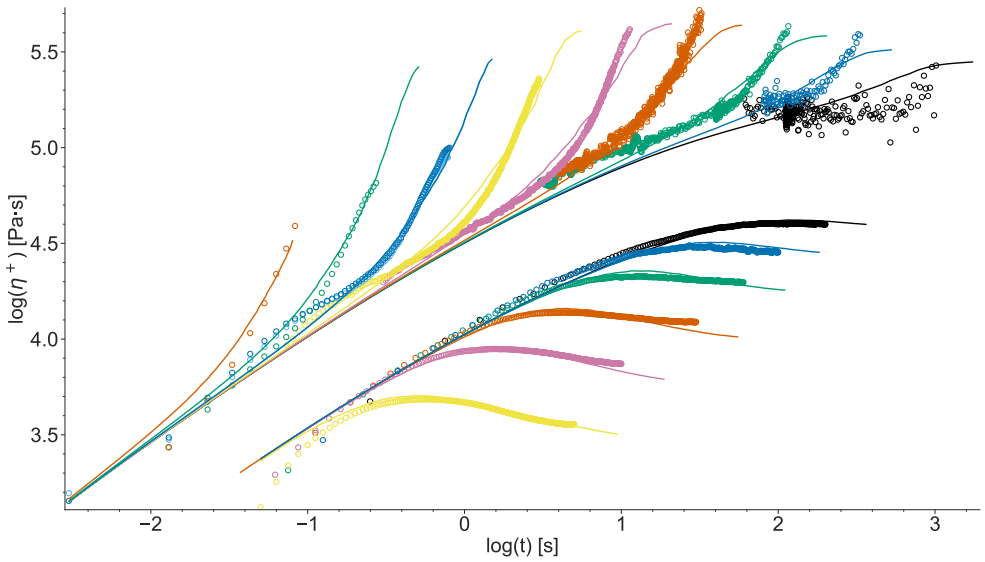

FIG. 9. Transient extensional stress growth coefficient (upper curves) and shear stress growth coefficient (lower curves) at 150 °C for the LDPE2 material of Ref. [31] (symbols), together with prediction using BoB theory within the **NLVE** Application (lines). Extension rates are $\dot{\varepsilon} = 0.003$, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30 s^{-1}. Shear rates are $\dot{\gamma} = 0.03$, 0.1, 0.3, 1, 3, 10 s^{-1}.

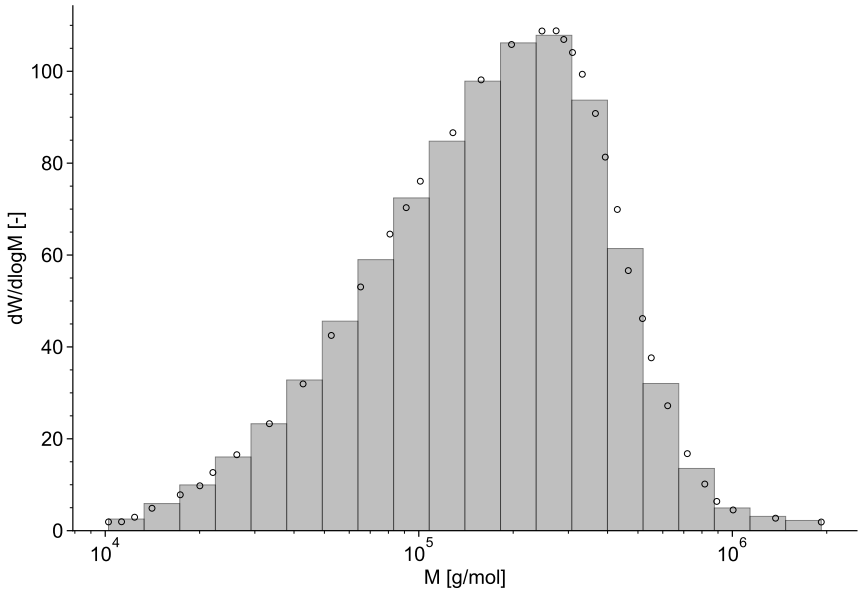

FIG. 10. Molecular weight distribution of the "PSIV" sample of Ref [35]. Experimental data and discretization, using 20 bins equally distributed on log-scale, are the symbols and bar plot, respectively.

to a "NLVE" *Application* in RepTate and a "Rolie-Double-Poly" *Theory* is opened. This theory allows the user to import the discretised molecular weight distribution as input to the calculation. This requires two tube parameter values for polystyrene: i) the average molecular weight between entanglements, which we choose to be $M_e = 16.6$ kg/mol, and ii) the entanglement relaxation time, which we choose to be $\tau_e = 3.4 \times 10^{-3}$ s at 160 °C. These parameters are as used in Ref. [13].

This allows the software to calculate the reptation and stretch relaxation times. This calculation accounts for both static dilution (due to the unentangled chains) and dynamic dilution (shorter entangled chains release their entanglements on longer chains, acting as solvent for the longer chains) as described in [13]. We set the entanglement plateau modulus $G_N^0 = 0.2$ MPa. Clicking the "Modulus Correction" button ensures that the theory includes contour length fluctuation corrections to this modulus (arising from primitive path fluctuations which reduce the terminal modulus because of the shortened tube) [14, 36]. No further input is required to perform the calculations.

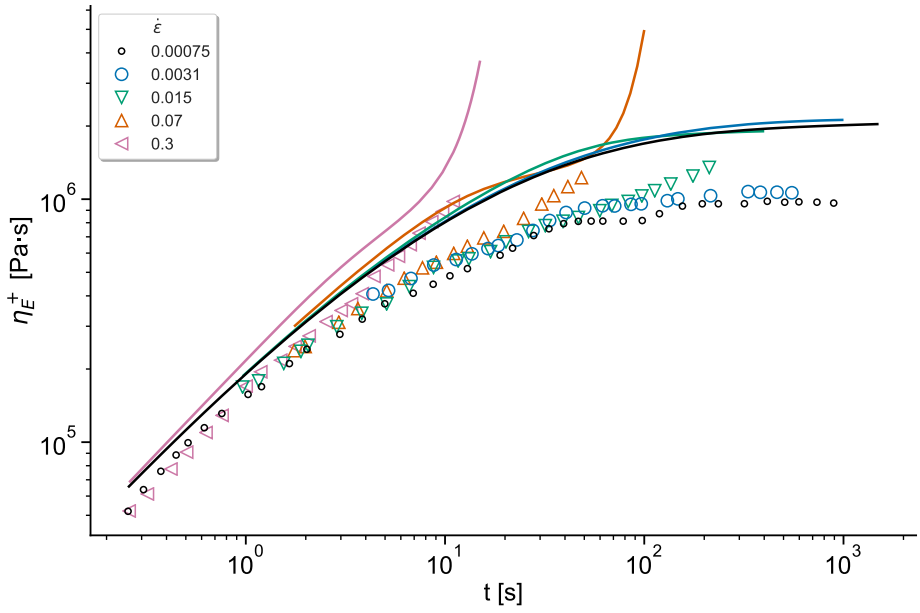

FIG. 11. Transient stress growth coefficient, η_E^+, as a function of time, t, for the "PSIV" sample of Ref. [35] at Hencky strain rates $\dot{\varepsilon} = 0.00075$, 0.0001, 0.0015, 0.07, 0.3 s^{-1} at 160 °C. Experimental data [35] and RDP [13] predictions are in symbols and lines, respectively

The results are shown in Fig. 11. Note that we extended the time range ("x-range" in the plot) for the theory prediction beyond the range of the experimental data, as noted

in Section II B. Having done this, the theory curves show more clearly the predicted onset of extension hardening at the fastest two extension rates. Again, our intention here is to reproduce previously published results, and the predictions shown in Fig. 11 are identical to those presented in Ref. [13]. Nevertheless, we note that the predictions do not perfectly match experiment. In particular, it was noted in Ref. [13] that the LVE was not perfectly predicted by the theory: this may be due to the approximate calculation used to determine reptation times within the model, or may be due to the chosen tube model parameters G_e, τ_e or M_e not being perfect for the material. Nevertheless, an important aspect of the Rolie-Double-Poly theory, emphasised in Ref. [13], is its non-trivial prediction of the onset of extension hardening rate in bidisperse and polydisperse materials, and the RDP model's success in doing this is illustrated in Fig. 11.

IV. USE OF DUMMY FILES IN THE ABSENCE OF EXPERIMENTAL DATA

Although the structure of RepTate is designed so that *Theories* are compared with experimental *Datasets*, we recognise that in some cases a user may wish to investigate the behaviour of a theory without reference to experimental data. For such cases it is possible to create "dummy files" within RepTate. Dummy files are empty files (they contain no real data) in which the parameters (such as molecular weight, temperature, shear rate, etc.) can be set to span a certain range, either in linear or logarithmic scale. Then, any theory can be applied to the dummy files exactly in the same way as with real data files. Dummy files allow the user to explore features of theories very easily.

For example, in Fig. 12, a set of 5 dummy files with molecular weights ranging from 5 to 500 in logarithmic scale has been created. Then, two instances of the Likhtman-McLeish theory [14] have been applied to the dummy files, using parameters $\tau_e = 0.0001$, $G_e = 1000$ and $M_e = 1$. In one of the instances of the theory, the constraint release (CR) parameter c_ν has been set to 0 (no CR at all) whereas in the other it has been set to 1 (very effective CR mechanism). The whole procedure takes a couple of minutes in RepTate and allows the user to gain an intuition about the effect of the CR parameter in the predictions of the theory.

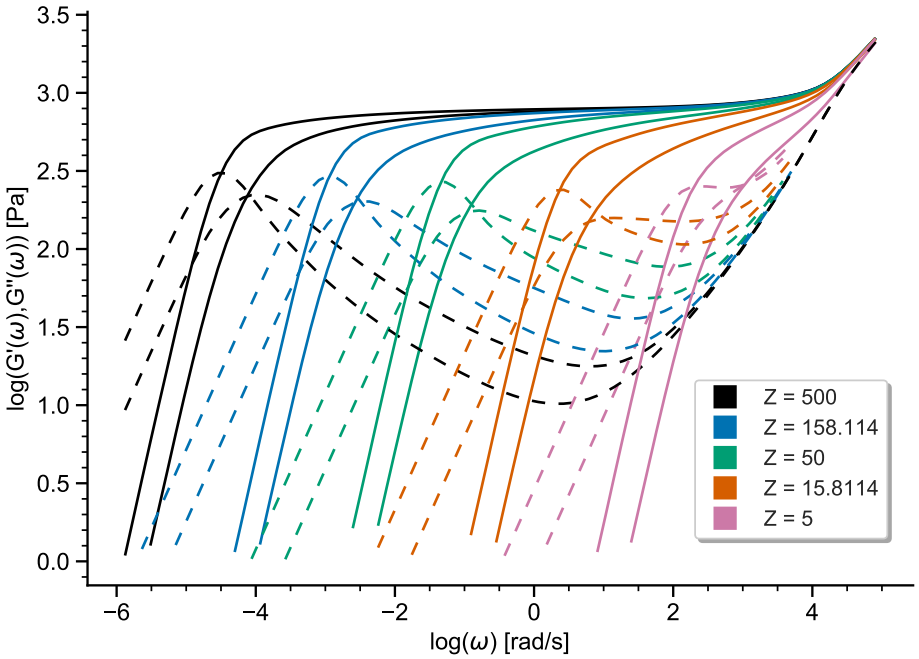

FIG. 12. Exploring the effect of the constraint release parameter c_ν of the Likhtman-McLeish theory [14]. When $c_\nu = 0$ (no CR) the terminal region is sharper and occurs later than in the case of very effective CR ($c_\nu = 1$).

V. CONCLUSIONS

In this paper we have presented the new RepTate software, and illustrated some of its functions by recreating the predictions of a number of recent articles. These examples should be considered illustrative rather than demonstrating the full range of *Applications* and *Theories* presently available within the software. We might have included further examples such as, viewing stress relaxation following small step strain or fitting LVE data using a multi-mode Maxwell model, then using that as a basis for multi-mode pom-pom [21, 37] or Rolie-Poly [14] fits to NLVE data. There are also developing *Applications* for viewing SANS data, and dielectric relaxation data.

Furthermore, our hope is that this software will continue to develop. The design of the software, and our open source model for distributing it, indicates our wish that other members of the community might feel inspired to develop their own *Applications* and *Theories* within the RepTate environment, or to improve upon aspects of what we have created. We highlight that although the applications discussed in the paper are all related to rheology,

there is nothing particularly unique to rheology in the structure of RepTate. It is coded with enough flexibility so that users can add new *Applications*, *Theories* and *Tools* related to any scientific area. What we have developed remains imperfect, both in some of the implementation and documentation, and we are conscious of our own time constraints in making further improvements. Nevertheless, we are presently developing further *Applications* and *Theories* (e.g. for LAOS, and flow accelerated crystallisation) which we expect to release shortly. The GPL v3 licence permits users to freely use the software and make their own private (and perhaps confidential) improvements, whilst committing users to an open source model for distribution of any modified code.

The software is documented at https://reptate.readthedocs.io/. This gives detailed instructions on installation, together with a user manual describing the main *Applications* and *Theories*, and a limited number of tutorials. Importantly, a section "RepTate for Developers" gives detailed instructions and examples for creating new *Applications* and *Theories* within RepTate. We are, of course, keen to hear of any use of the software, suggestions for improvements, and especially to hear of modifications users have made and which they are willing to make available to others.

ACKNOWLEDGEMENTS

We are grateful to the School of Mathematics at the University of Leeds for providing a six month "Impact Fellowship" to VAHB for development of the RepTate software. JR acknowledges funding from Grant FIS2016-78847-P of the MEC.

As described in the Introduction, RepTate has a history, and we would like to acknowledge the invaluable contributions and suggestions made by many μPP^2 colleagues, in particular, Tom McLeish, Choon Chai, Ian Robinson, Sathish K. Sukumaran, David Hoyle, Suneel Kunamaneni, Dietmar Auhl, Richard Graham, Kamakshi Jagannathan and Bart Vorselaars; beyond muPP^2, Maksim Shivokhin provided many useful comments.

Finally, this paper and software are dedicated to the memory of Alexei Likhtman, who was the original inspiration behind and instigator of the RepTate project.

[1] Winter, H. H., and M. Mours, "The cyber infrastructure initiative for rheology," *Rheol. Acta* **45**, 331–338 (2006).

[2] Milner, S. T., and T. C. McLeish, "Parameter-free theory for stress relaxation in star polymer melts," *Macromolecules* (1997).

[3] Milner, S. T., and T. C. B. McLeish, "Reptation and Contour-Length Fluctuations in Melts of Linear Polymers," *Phys. Rev. Lett.* **81**, 725–728 (1998).

[4] Larson, R. G., "Combinatorial Rheology of Branched Polymer Melts," *Macromolecules* **34**, 4556–4571 (2001).

[5] Park, S. J., S. Shanbhag, and R. G. Larson, "A hierarchical algorithm for predicting the linear viscoelastic properties of polymer melts with long-chain branching," *Rheol. Acta* **44**, 319–330 (2005).

[6] Wang, Z., X. Chen, and R. G. Larson, "Comparing tube models for predicting the linear rheology of branched polymer melts," *J. Rheol.* **54**, 223–260 (2010).

[7] Rolón-Garrido, V. H., and M. H. Wagner, "The MSF model: relation of nonlinear parameters to molecular structure of long-chain branched polymer melts," *Rheol. Acta* **46**, 583–593 (2007).

[8] Masubuchi, Y., J.-I. Takimoto, K. Koyama, G. Ianniruberto, G. Marrucci, and F. Greco, "Brownian simulations of a network of reptating primitive chains," *J. Chem. Phys.* **115**, 4387–4394 (2001).

[9] Doi, M., "OCTA (Open Computational Tool for Advanced material technology)," *Macromolecular Symposia* **195**, 101–108 (2003).

[10] Das, C., N. J. Inkson, D. J. Read, M. a. Kelmanson, and T. C. B. McLeish, "Computational linear rheology of general branch-on-branch polymers," *J. Rheol.* **50**, 207–234 (2006).

[11] Das, C., D. J. Read, M. A. Kelmanson, and T. C. B. McLeish, "Dynamic scaling in entangled mean-field gelation polymers," *Phys. Rev. E* **74**, 011404 (2006).

[12] Chambon, P., C. M. Fernyhough, K. Im, T. Chang, C. Das, J. Embery, T. C. B. McLeish, and D. J. Read, "Synthesis, Temperature Gradient Interaction Chromatography, and Rheology of Entangled Styrene Comb Polymers," *Macromolecules* **41**, 5869–5875 (2008).

[13] Boudara, V. A. H., J. D. Peterson, L. G. Leal, and D. J. Read, "Nonlinear rheology of polydisperse blends of entangled linear polymers: Rolie-Double-Poly models," *J. Rheol.* **63**,

71–91 (2019).

[14] Likhtman, A. E., and T. C. B. McLeish, "Quantitative Theory for Linear Dynamics of Linear Entangled Polymers," *Macromolecules* **35**, 6332–6343 (2002).

[15] Kiusalaas, J., *Numerical methods in engineering with Python 3*, (Cambridge university press, 2013).

[16] Summerfield, M., *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*, (Pearson Education, 2007).

[17] Bressert, E., *SciPy and NumPy: an overview for developers*, (" O'Reilly Media, Inc.", 2012).

[18] Hunter, J. D., "Matplotlib: A 2D graphics environment," *Computing in science & engineering* **9**, 90 (2007).

[19] "GNU General Public License, version 3,". http://www.gnu.org/licenses/gpl.html (2007). Last retrieved 2019-11-15.

[20] Inkson, N. J., T. C. B. McLeish, O. G. Harlen, and D. J. Groves, "Predicting low density polyethylene melt rheology in elongational and shear flows with pom-pom constitutive equations," *J. Rheol.* **43**, 873–896 (1999).

[21] McLeish, T. C. B., and R. G. Larson, "Molecular constitutive equations for a class of branched polymers: The pom-pom polymer," *J. Rheol.* **42**, 81–110 (1998).

[22] Branch, M. A., T. F. Coleman, and Y. Li, "A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems," *SIAM Journal on Scientific Computing* **21**, 1–23 (1999).

[23] Singh, P. K., J. M. Soulages, and R. H. Ewoldt, "On fitting data for parameter estimates: residual weighting and data representation," *Rheol. Acta* **58**, 341–359 (2019).

[24] Freund, J. B., and R. H. Ewoldt, "Quantitative rheological model selection: Good fits versus credible models using Bayesian inference," *J. Rheol.* **59**, 667–701 (2015).

[25] Savitzky, A., and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures.," *Analytical Chemistry* **36**, 1627–1639 (1964).

[26] Tassieri, M., J. Ramírez, N. C. Karayiannis, S. K. Sukumaran, and Y. Masubuchi, "i-Rheo GT: Transforming from Time to Frequency Domain without Artifacts," *Macromolecules* **51**, 5055–5068 (2018).

[27] Auhl, D., J. Ramirez, A. E. Likhtman, P. Chambon, and C. Fernyhough, "Linear and nonlinear shear flow behavior of monodisperse polyisoprene melts with a large range of molecular

weights," *J. Rheol.* **52**, 801 (2008).

[28] Graham, R. S., A. E. Likhtman, T. C. B. McLeish, and S. T. Milner, "Microscopic theory of linear, entangled polymer chains under rapid deformation including chain stretch and convective constraint release," *J. Rheol.* **47**, 1171–1200 (2003).

[29] Williams, M. L., R. F. Landel, and J. D. Ferry, "The Temperature Dependence of Relaxation Mechanisms in Amorphous Polymers and Other Glass-forming Liquids," *J. Am. Chem. Soc.* **77**, 3701–3707 (1955).

[30] Colby, R. H., L. J. Fetters, and W. W. Graessley, "The melt viscosity-molecular weight relationship for linear polymers," *Macromolecules* **20**, 2226–2237 (1987).

[31] Read, D. J., D. Auhl, C. Das, J. den Doelder, M. Kapnistos, I. Vittorias, and T. C. B. McLeish, "Linking Models of Polymerization and Dynamics to Predict Branched Polymer Structure and Flow," *Science* **333**, 1871–1874 (2011).

[32] Tobita, H., "Simultaneous long-chain branching and random scission: I. Monte Carlo simulation," *Journal of Polymer Science Part B: Polymer Physics* **39**, 391–403 (2001).

[33] Read, D. J., "From reactor to rheology in industrial polymers," *J. Polym. Sci., Part B: Polym. Phys.* **53**, 123–141 (2015).

[34] Dealy, J. M., D. J. Read, and R. G. Larson, *Structure and Rheology of Molten Polymers From Structure to Flow Behavior and Back Again*, (Hanser, 2018).

[35] Münstedt, H., "Dependence of the Elongational Behavior of Polystyrene Melts on Molecular Weight and Molecular Weight Distribution," *J. Rheol.* **24**, 847–867 (1980).

[36] Doi, M., "Explanation for the 3.4-power law for viscosity of polymeric liquids on the basis of the tube model," *Journal of Polymer Science: Polymer Physics Edition* **21**, 667–684 (1983).

[37] Blackwell, R. J., T. C. B. McLeish, and O. G. Harlen, "Molecular Drag-Strain Coupling in Branched Polymer Melts," *MRS Proceedings* **629**, FF3.6 (2000).