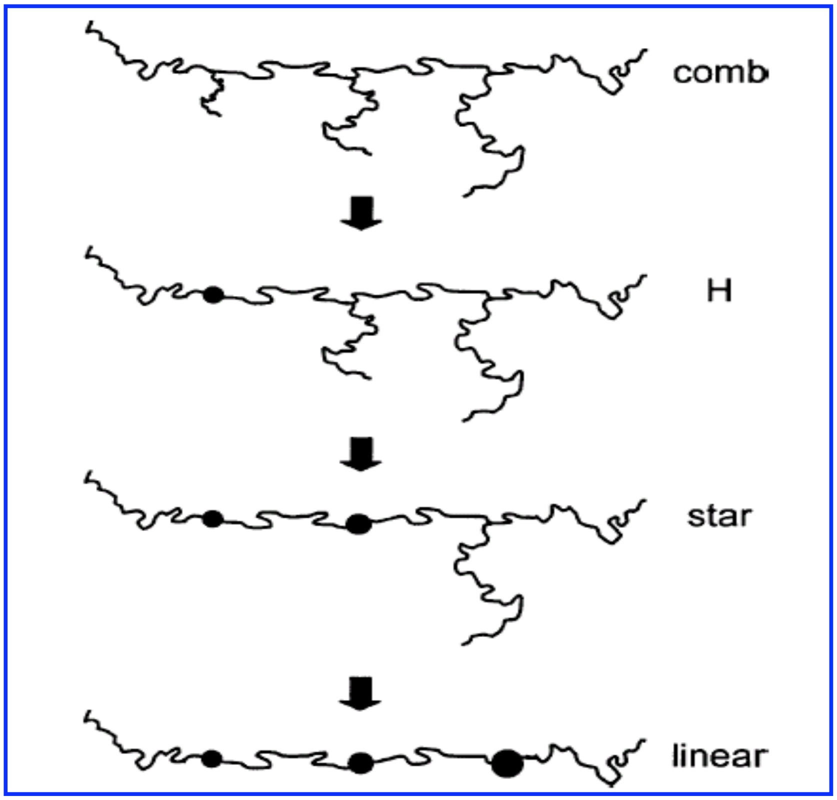
Hierarchical-3.0



January 26, 2010

1. Introduction

The hierarchical model was first developed by Larson (2001) for predicting the linear viscoelasticity of general mixtures of branched polymers, which generalizes the theory of Milner and McLeish for blends of monodisperse star and linear polymers (Milner and McLeish, 1997; Milner *et al.*, 1998). This original work was later modified by Park *et al.* (2005) to include the "early-time fluctuations" and other refinements from the theories of Milner and McLeish, allowing for more quantitative predictions. The original hierarchical model (Larson, 2001) and its modified version (Park *et al.*, 2005) are named as the "hierarchical-1.0" model and the "hierarchical-2.0" model, respectively. The "hierarchical-3.0" model is an updated version of the hierarchical-2.0 model, which incorporates several refinements of the relaxation mechanisms and also shows a higher computational efficiency due to the use of a logarithmic time stepping method for calculating the time evolution of the arm-end retraction (Wang *et al.*, 2009).

The physical basis of the hierarchical model can be found in the related publications (Larson, 2001; Park *et al.*, 2005; Wang *et al.*, 2009). Here we provide an introduction on how to use the hierarchical-3.0 code, which is based on the hierarchical-3.0 model, to predict the linear rheology of branched polymer melts with arbitrary distributions of molecular weights and branching architectures. So far, this code still could not handle the branch-on-branch structures, but will incorporate this feature in a later release. The hierarchical-3.0 code has an alternative version, called the "hierarchical-3.0b" code, which uses the "bob" methods (Das *et al.*, 2006), rather than the hierarchical methods, to handle the retraction dynamics of compound arms and the reptation time of effective linear chains. Our purpose of having the hierarchical-3.0b code is to facilitate the comparison of the different relaxation mechanisms employed in the hierarchical and bob models on their predictions of the viscoelastic properties of various branched polymers; see Wang *et al.*, 2009. The hierarchical-3.0 code should be taken as the representative of the standard hierarchical model.

References

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

Larson, R. G., "Combinatorial rheology of branched polymer melts," Macromolecules **34**, 4556–4571 (2001).

Milner, S. T., and T. C. B. McLeish, "Parameter-free theory for stress relaxation in star polymer melts", Macromolecules **30**, 2159–2166 (1997).

Milner, S. T., T. C. B. McLeish, R. N. Young, A. Hakiki, and J. M. Johnson, "Dynamic dilution, constraint release, and star-linear blends," Macromolecules **31**, 9345–9353 (1998).

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).

Wang, Z. W., X. Chen and R. G. Larson, "Comparing tube models for predicting the linear rheology of branched polymer melts", J. Rheol., in press (2010).

2. Source Codes

Due to historical reasons, the hierarchical codes are written in Fortran. In principle, these codes are compatible to most Fortran compilers, such as g77, g90, g95, f77, f90 and xlf, and various operation platforms, including Unix, Linux, Mac OS X and Microsoft Windows.

2.1 Hierarchical codes

Two sets of hierarchical codes are available: (see Introduction)

Hierarchical-3.0: *hierarchical_3.f, hierarchical_3.h*

Hierarchical-3.0b: *hierarchical_3b.f, hierarchical_3b.h*

In each case, the user can simply put the **.f* and **.h* files in the same directory, compile the code using the standard Fortran compiler as

```
g77 –Wall –O3 hierarchical_3.f
```

and then run it by

```
./a.out
```

2.2 Code for generating polydisperse ensemble

We provide a Fortran code, *ensemble_bob_to_hier.f*, which transforms the polydisperse polymer ensembles generated by the Bob-2.3 code (http://sourceforge.net/projects/bob-rheology) into the Input format of the hierarchical-3.0 codes; see Sec.5.2.

3. Input and Output Files

The file name is defined as *file_name* in lines 13 and 15 in *hierarchical_3.f* , and lines 16 and 18 in *hierarchical_3b.f*, respectively. One can give any file name preferred, but remember to define the correct length of the character string (line 13 in *hierarchical_3.f* and line 16 in *hierarchical_3b.f*).

Then the Input and Output files will be:

File name	Description	Content
Input_*file_name*.dat	Input file	Input parameters and polymer ensemble; see Sections 4 and 5.

Mod_*file_name*.dat	Output file
	Stress relaxation moduli
	omeg, Gp, Gpp
		Omeg: frequency;
		Gp: storage modulus;
		Gpp: loss modulus.

Phi_*file_name*.dat	Output file
	Time dependence of volume fractions
	cur_time, PHI, PHIST, PHI_Tube
		cur_time: time;
		PHI: volume fraction of unrelaxed material;
		PHIST: supertube volume fraction;
		PHI_Tube: volume fraction used to define the tube diameter in which the arms retract,.

4. System and Input Parameters

4.1 Parameters controlling relaxation mechanisms

The hierarchical-3.0 code provides the following options to choose different relaxation mechanisms, which can be changed in *hierarchical_3.h*. It is suggestive to use the default options, especially for Sche_CR_Crit and Sche_Rep_Time.

Parameters	Options	Description
Sche_Arm_Retract	1: no-arm-rectraction 2: arm-retraction in thin tube (default) 3: arm-retraction in fat tube	Arm retraction mechanisms in the constraint release (CR) Rouse regime

Sche_CR_Crit	1: hierarchical method (default, activating after the drop of any polymer) 2: bob method (judging at each time step)	Criterion for activating the CR-Rouse regime.
Sche_Rep_Time	1: hierarchical method [default, using $\Phi_{Tube}(\tau_{rep})$] 2: bob method [using $\Phi_{Tube}(t=0)=1$]	Evaluating the Reptation time of linear chains.
Alpha	1 4/3	Dilution exponent α
P2	1/12 (Park *et al.*, 2005) 1/40 (Das et al., 2006) Other values < 1	P^2: Fraction of branch point hopping distance relative to the tube diameter

In the hierarchical-3.0b code, only Sche_Arm_Retract, alpha and p2 could be changed in *hierarchical_3b.h*, due to the use of the bob methods for other options.

In *hierarchical_3.h* and in *hierarchical_3b.h*, one can also change the following integer values for defining the dimensions of the arrays:
maxm: maximum number of polymer molecules
maxna: maximum number of branch arms per molecule
maxt_step: maximum number of time steps

4.2 Input Parameters

The input parameters are defined in the first two lines of the Input file (see Sec.3):

Line 1: *omeg_i, omeg_f ,nomeg, GN0,m0*
omeg_i and *omeg_f* define the range of the frequencies interested, in the units of S^{-1};
nomeg: Number of data points to be generated in this frequency range;
GN0: Plateau modulus G_N^0, in the unit of *Pa*;

m0: Molar weight of monomers, in the unit of *g/mol.*

Line 2: *taue, me, tinit, xx ,nt ,np, samin*

taue: Equilibration or entanglement time τ_e, in the unit of second (*S*)

me: Entanglement spacing M_e (molecular weight per entanglement), in the unit of *g/mol*;

tinit: Initial time (time zero t_0) measured in *taue,* by default set to 10^{-4}, giving $t_0=10^{-4}\tau_e$;

xx: Time step size multiplier (m_{st}), so that time grows as $t_{n+1} = m_{st}t_n$. In general, a value of *xx*=1.001~1.005 is small enough to obtain converged results.

nt: Total number of time steps. The exact value of *nt* is not important, as far as it's big enough (e.g., 50000000 by default) to cover the relaxation time of the system.

np: Every *np* time steps, save the volume fractions Φ and Φ_{ST} for calculating the stress moduli. By fault, *np=1*, meaning that the saving is performed at each time step.

samin: Disentanglement threshold $S_{a,min}$. Values of *samin=1* and *3* have been used in hierarchical model predictions (Park *et al.*, 2005; Wang *et al.*, 2009). To switch off the disentanglement relaxation, set *samin* to *0* or a tiny value like *1.e-15*.

5. Polymer Ensemble

5.1 Input format

The polymer ensemble is defined in the Input file (see Sec.3) starting from Line 3.

Line 3: *nmole,* total number of polymer molecules

Line 4 and after: giving the characterization of each polymer in the following order:

Number of arms in each molecule: The first *nmole* integer numbers, staring from Line 4, give the number of arms, *na(i)*, in each of the *nmole* polymers, i.e.,

na(1), na(2), ...na(i), ...na(nmole)

By default, the number of backbone segments, *nb(i)*, of each polymer is automatically taken to be *nb(i)= na(i)-3* (*nb(i)=0,* if *na(i)≤3*) in the code, since the branch-on-branch structure is not yet considered in the hierarchical-3.0 codes.

Molecular weights of arms and total volume fraction of each molecule: After defined *na(i)*, each of the following *nmole* lines provides the molecular weights of the arms, *ma(i,j)*, and also the total volume fraction, *phit(i)*, of one polymer, in the increasing order of the polymer index *i* (from *1* to *nmole*), i.e.,

ma(1,1), ma(1,2),…, ma(1,na(1)), phit(1)

….

ma(i,1), ma(i,2),…, ma(i,na(i)), phit(i)

….

ma(nmole,1), ma(nmole,2),…, ma(nmole,na(nmole)), phit(nmole)

Molecular weights of backbone segments of each molecule: The remaining part of the Input file gives the molecular weights of the backbone segments, *mb(i,j)*, in the polymers again in the order of the polymer index *i*, i.e.,

mb(1,1), mb(1,2), …,mb(1, nb(1)), …, mb(i,j), ….mb(nmole, nb(nomle))

If *nb(i)=0* for polymer *i*, there is simply no input of *mb* for that polymer. There is no specific requirement for how many numbers should be put in one line.

5.2 Generating polymer ensembles

a) Simple melt systems: For single-component or mixtures of mondisperse branched polymers, like monodisperse linear, star, H and comb polymers, or binary blends of star-linear or linear-linear polymers, one can simply set up the Input file by hand, referring to the sample Input files given below.

b) Polydisperse branched polymers: For melt systems with certain polymer molecular weight and architecture distributions, such as the metallocene-catalyzed polyethylene resins, we take the polymer ensembles generated by the Bob-2.3 code as our input for the hierarchical-3.0 codes. The Bob-2.3 code was developed by C. Das and is available online (http://sourceforge.net/projects/bob-rheology), which contains a growing number of methods of generating branched polymer ensembles appropriate to various synthesis methods.

We provide a Fortran code, named *ensemble_bob_to_hier.f*, which transforms the output polymer ensemble generated by the Bob-2.3 code into the proper Input file for the hierarchical-3.0 codes. In that code (*ensemble_bob_to_hier.f*), the input file is the polymer configuration file, by default named as *polyconf.dat*, generated by the Bob-2.3

code. Its output file is consequently the Input file, Input*_file_name*.dat (see Sec.3), for the hierarchical-3.0 codes. When running *ensemble_bob_to_hier.f* , the fraction of each type of polymers in the ensemble is also printed out on the screen, in the order of number of arms they have, i.e., 2 for linear chains, 3 for stars, 4 for H polymers, and ≥5 for combs. Since the hierarchical code still can not handle the branch-on-branch polymers, the polymer with hyperbranching structures, but satisfying the relation of *nb=na-3*, generated by the Bob-2.3 code will be automatically transformed into a comb polymer with the same number of branch arm and backbone segments. If the relation between the number of arms and number of backbone segments is not satisfied, a warning will be printed and a new ensemble needs to be generated by the Bob code.

Note that if *ensemble_bob_to_hier.f* is used to generate the Input file for the hierarchical-3.0 codes, all the Input parameter values (see Sec.4.2) for the related polymer system need to be defined in that code (from Line 109 to Line 120), so that the generated Input file can be used directly for running the hierarchical-3.0 codes without any further adjustment.

6. Samples for Input Files

We provide the following sample input files for the polymer melt systems studied in the Wang *et al.*, 2010 J. Rheol. paper, which is also available upon request.

Note: For using the Park parameters, set α=4/3 and p^2=1/12

For using the Das parameters, set α=1 and p^2=1/40

6.1 Bidisperse 1,4-polybutadiene star-linear blends (Figures 2-6)

Input_LS_PBd_Park.dat : using Park parameters
Input_LS_PBd_Das.dat : using Das parameters

To switch on the disentanglement relaxation mechanism (Figure 4d), change the *samin* value (the last parameter in Line 2 of the Input files) to *3*.

To change the composition of the two components: keep the first two lines fixed in the

Input files and change the following lines:

Monodisperse linear chains (Figure 3):

```
1
2
50000 50000       1.0
```

Monodisperse symmetric stars (Figure 3):

```
1
3
42333 42333 42333 1.0
```

Bidisperse star-linear blends with star fraction of 0.2 (Figure 4-6)

```
2
3  2
42333 42333 42333 0.1
50000 50000       0.9
```

6.2 Bidisperse 1,4-polybutadiene linear-linear blends (Figure 7)

Input_PBd_LL_Park.dat : using Park parameters

Input_PBd_LL_Das.dat : using Das parameters

6.3 Polyisoprene asymmetric stars (AS37) (Figure 8)

Input_PI_AS37_Park_mono.dat : monodisperse, using Park parameters

Input_PI_AS37_Park_poly.dat : polydisperse, using Park parameters

Input_PI_AS37_Das_mono.dat : monodisperse, using Das parameters

Input_PI_AS37_Das_poly.dat : polydisperse, using Das parameters

6.4 Polyisoprene H polymers (H110B20A) (Figure 9)

Input_PI_H110B20A_Park_mono.dat : monodisperse, using Park parameters

Input_PI_H110B20A_Park_poly.dat : polydisperse, using Park parameters

Input_PI_H110B20A_Das_mono.dat : monodisperse, using Das parameters

Input_PI_H110B20A_Das_poly.dat : polydisperse, using Das parameters

6.5 1,4-polybutadiene combs (PBC11) (Figure 10)

Input_PBd_PBC11_Park.dat : using Park parameters

Input_PBd_PBC11_Das.dat : using Das parameters

6.6 Metallocene-catalyzed polyethylene resins (Figure 11)

Using $\alpha=1$ and $p^2=1/40$

Input_mPE_HDB1_alpha1.dat

Input_mPE_HDB2_alpha1.dat

Input_mPE_HDB3_alpha1.dat

Using $\alpha=4/3$ and $p^2=1/12$

Input_mPE_HDB1_alpha43.dat

Input_mPE_HDB2_alpha43.dat

Input_mPE_HDB3_alpha43.dat

7. Contact

Please send any questions or suggestions concerning the hierarchical-3.0 codes to:

Dr. Zuowei Wang, zuowwang@yahoo.com

Dr. Ronald G. Larson, rlarson@umich.edu