1 Introduction

This version builds on bob2.3 to calculate non-linear flow properties (uniaxial extension and continuous shear) of entangled polymer melts. While calculating the stress relaxation after small strain, the previous versions of bob already calculated the tube escape time and Rouse time of a particular segment. By assuming that each small segment of a polymer can be treated as a pompom molecule, these two times become the orientational and stretch relaxation times of the pompom constitutive equation. To get the number of pompom arms, we use a generalized concept of the 'priority' variable. To the zeroth approximation, priority of a segment is the lower of the number of free-ends on the two sides of the segment. Flow modifies this 'geometric' priority.

To describe the linear rheology, we need both relaxation by tube escape (ϕ) and constraint release (ϕ_{ST}). Thus, each polymer segment further gives rise to two kinds of pompom modes.

The present version is meant to be run with slightly different inputs, twice for each rate. In the first pass, it resolves the linear relaxation moduli to a set of Maxwell modes. Also, it assigns the relaxation times to the segments as the molecules relax by tube escape. The next run uses these information to calculate the proper priority for the segments and resolves the molecules to a set of pompom modes having different weights, relaxation times and number of pompoms, at each of the Maxwell times. At the end, these modes are solved with pompom constitutive equations to predict the transient uniaxial extension and shear stress.

A short description of the method is in *Science*, **333**, 1871-1874 (2012). A longer description will be communicated at a later date.

As always, the code evolved over several years. In the actual 'research' phase, code fragments were stitched together, completely separate programs worked in a production line. This is a sanitized version, which is supposed to be stand-alone. There is no guarantee that we have not introduced new bugs in the process. Also, some of the routines will make little sense - the physical picture evolved over the years, but the variables retained their original historical name.

The rest of this document will outline the main changes introduced compared to bob2.3. You may need the documentation of bob2.3 to make sense of the following. (To find the documentation of bob2.3, look at `sourceforge.net/projects/bob-rheology/files/bob-rheology/bob2.3`)

2 Linear rheology

Changes

- Some instances of hardcoded $\alpha = 1$ have been modified for generic α.
- bob.rc accepts a variable named *SlavePhiToPhiST*. The default is 'no'. If set to 'yes', during supertube relaxation, ϕ follows ϕ_{ST} (i.e. the diameter of the thin tube grows continuously as it explores inside the fat tube). ϕ is considered to be fixed till ϕ_{ST} reaches the ϕ_{true} in bob2.3.

  For small fraction of slowly relaxing branched material, setting *SlavePhiToPhiST=yes*, may give better results. Though you may need to fit all the other parameters.

To use bob2.5 for linear rheology prediction, use the bob.rc options
CalcNlin=no
NlinPrep=no
FlowPriority=no

Run bob2p5 in the same way as the bob2.3 executable.

3 Nonlinear rheology

3.1 computing a non-linear flow prediction

3.1.1 Step 1.

For non-linear predictions, create the polymer input file as in the linear case.
use bob.rc options:
CalcNlin=no
NlinPrep=yes
FlowPriority=no
FlowTime=1/rate

You need to put the numerical value for SnipTime.

Run bob2p5 once.

3.2 Step 2.

If you had generated polymers during the previous step, you will need to use exactly the same polymers. So, set the number of components to zero in the input file (6th line).

Create a file called nlin.inp

The first entry of the file is number of pompom calculations you want to do. Since the bulk of the computational time goes in resolving in the pom-pom modes, we usually ask for both the shear and extensional predictions and hence we enter 2 in the first line. The next lines each should have 4 entries. First entry is 0 for shear or 1 or extension. The second entry is the rate. The 3rd entry is the starting time from which to calculate the transient stress. And the 4th entry is the largest time until which to calculate the transients. Remember to enter the 3rd variables as a positive number, since the prediction will be in log-scale.

A sample file for the rate being 10 s^{-1} would be something like:

2
0 10.0 1.0e-4 5.0
1 10.0 1.0e-4 1.0

Change the bob.rc options to

CalcNlin=yes
NlinPrep=yes
FlowPriority=yes
FlowTime=1/rate

3.2.1 Automating the computation

On Unix like platforms, you can create a file called rates.dat with the different rates you want, one in each line and with no line break. The rates must be in decimal format (i.e. 0.001 and not 1.0e-3).

You will find a shell script called nlin_run in the directory bob2.5/code/nlin_run. Change the directory name of the source code in that script (first line). It assumes that the executable is in bob2.5/code/src.

Now, if you run the script, it should run bob twice for each rate, modifying the bob.rc, inp.dat and rates.dat as needed. It will store the results in files with names to reflect the rates.

3.3 Output format

If OutMode is set to zero, the outputs are in ascii format. For the shear data, the four columns are
(i) time since starup, (ii) stress, (iii) Normal stress N_1, (iv) rate.
For the extension data, the three columns are
(i) time since startup, (ii) stress, (iii) rate.

4 Where it is unlikely to work and why

- The resolving to the pompom modes assumes that the stress relaxation in linear deformation is smooth enough to have smooth derivatives. Thus for highly polydisperse system like industrial resins, one can sample at discrete intervals during the stress relaxation and predict the flow behavior. For melts with low polydispersity, the program is likely to give completely wrong predictions. Fortunately, when it does so, the predictions are qualitatively wrong. If the non-linear transients fall over the linear rheology data at short times, you are probably ok. If not, increasing the number of polymers (assuming that you are not working with strictly monodisperse polymers) often helps.

- The file inputs assume '\ n' as end-of-line character. On some platforms, the program is likely to read completely wrong information. Unfortunately, it will carry on and probably crash somewhere else. It is always a good idea to try to run the code on *NUX platform in that case and check with *file* command to see if the input files are UNIX format. This affects all version of bob.

5 How to find the executable

For *NIX system, go to bob2.5/code/src/obj and type make. This should create the executable bob2p5

For Win32 systems (Windows XP and possibly 32 bit Windows 7), you may download bob2p5.exe from the sourceforge site.

Because the predictions need repeated usage of the program, automating them with shell script is quite handy, you may want to install minGW/msys or cygwin on windows platform.