ReSpect v3 Manual

Arsia Takeh and Sachin Shanbhag

August 7, 2024

Contents

1 Introduction

`ReSpect` infers continuous and discrete relaxation spectra from dynamic moduli data obtained from small-angle oscillatory shear (SAOS) experiments. These experiments yield the frequency-dependent dynamic moduli, $G^*(\omega) = G'(\omega) + iG''(\omega)$. The relaxation spectra and complex modulus are mathematically linked by:

$$G'(\omega) = \int_{-\infty}^{\infty} \frac{\omega^2\tau^2}{1+\omega^2\tau^2} h(\tau)\, d\ln\tau = \sum_{i=1}^{N} g_i \frac{\omega^2\tau_i^2}{1+\omega^2\tau_i^2} \tag{1}$$

$$G''(\omega) = \int_{-\infty}^{\infty} \frac{\omega\tau}{1+\omega^2\tau^2} h(\tau)\, d\ln\tau = \sum_{i=1}^{N} g_i \frac{\omega\tau_i}{1+\omega^2\tau_i^2} \tag{2}$$

where $h(\tau)$ is the continuous relaxation spectrum (CRS), and the N modes $\{g_i, \tau_i\}$ describe the discrete relaxation spectrum (DRS).

Note: In the literature, $H(\tau)$ is often used to denote the CRS.[1]

1.1 Source

The basic algorithms used in this paper are described in the paper:

- Shanbhag, S., "Relaxation spectra using nonlinear Tikhonov regularization with a Bayesian criterion", Rheol. Acta, 2020, 59, 509.
- Shanbhag, S., "pyReSpect: A Computer Program to Extract Discrete and Continuous Spectra from Stress Relaxation Experiments", Macromolecular Theory and Simulations, 2019, 1900005.
- Takeh A, Shanbhag S: A Computer Program to Extract the Continuous and Discrete Relaxation Spectra from Dynamic Viscoelastic Measurements, Appl. Rheol. 23 (2013) 24628.

Please cite these paper if you use `ReSpect` and find it helpful.

2 History and Log of Changes

The first full release of ReSpect was in Feb 2013.

2.1 v2

In the second version (ReSpect v2, Dec 2015), several changes were made including bug-fixes, tweaks to the basic algorithm, and addition of new capability.

(i) **Bug-Fixes**

- Looks for repeated frequencies in input data, and removes duplicates. This often happens due to time-temperature superposition of data gathered at different frequencies (in `common/GetExpData.m`).
- In the GUI, the discrete spectrum was incorrectly plotted (in `gui/updateplot`). This is now fixed.
- The default plotting setting for `discSpec` have been changed to show the spectrum. Note that when the modes g_i are negative, you may see warnings about inability to represent negative numbers on a log-log plot.

[1] The $H(\tau) = \log h(\tau)$ used in ReSpect has a different meaning.

(ii) **Tweaks to the Algorithms**

- The residual and smoothness constraints are now normalized by the number of data points in the input file n, and the number of grid-points for representing the CRS n_l, respectively.
- The method used to determine the optimal λ_c has been tweaked. We define a function based on the normalized $\rho - \eta$ curve,

  $$\epsilon(\lambda) = \frac{\rho(\lambda)}{\max \rho(\lambda)} + \frac{\eta(\lambda)}{\max \eta(\lambda)},$$

  and choose a λ_c that minimizes $\epsilon(\lambda)$. In general, the new criterion leads to somewhat larger values of λ_c, and hence smoother spectra.

(iii) **New Features**

- By default, ReSpect uses Douglas Schwarz's SuperSmoother program[2] to filter input data (in `common/GetExpData.m`). This is particularly handy when the input data is noisy.
- Introduced an independent directory `Gt` to infer $h(\tau)$ from $G(t)$ data via the equation:

  $$G(t) = \int_{-\infty}^{\infty} h(\tau) e^{-t/\tau} d\ln\tau = \sum_{i=1}^{N} g_i e^{-t/\tau} \tag{3}$$

  The original program to infer $h(\tau)$ from $G^*(\omega)$ is now housed in the subdirectory `Gstar`
- Introduced a directory structure to the programs

2.2 v3

Since about 2016, I basically switched from coding in Matlab/Octave to Python. As I rewrote ReSpect in Python, I made **significant algorithmic changes** which not only reduced computational time, but also improved the inference. These projects are better maintained, and I strongly encourage users of ReSpect to use these programs instead. They are available at:

- $G^*(\omega) \to h(\tau)$: `https://github.com/shane5ul/pyReSpect-freq`
- $G(t) \to h(\tau)$: `https://github.com/shane5ul/pyReSpect-time`

Consequently, ReSpect-v2 languished over the past decade. I noticed that people still used the Matlab program. It probably has seen more use since it became outdated. I did not have the bandwidth to update ReSpect.

That changed recently. By 2023, AI and LLMs had become quite good at code translation. At the back of my mind, I always thought I might be able to translate my Python projects back to Matlab.

[2] `http://www.mathworks.com/matlabcentral/fileexchange/17986-supersmoother/content/supsmu.m`

This release which used chatGPT, Claude, and Google Gemini to rewrite functions is a result of this actions.

The result is code that is more reliable, and about 10x faster than ReSpect v2.

For faster turnaround, I decided to drop the GUI and $G(t) \rightarrow h(\tau)$ capability for now.

3 Program

`ReSpect` v3 contains a single folder `Gstar/` to extract $h(\tau)$ from $G^*(\omega)$. `ReSpect v3` can only be accessed either from the command line. Support for a graphical user interface (GUI) is discontinued.

To use `ReSpect`, the user needs to provide an **input file** which contains the $G^*(\omega)$ data, and **specify parameters** by modifying the function file `SetParameters`.

3.1 Using the Program

Enter the sub-directory `Gstar/`.

By calling `contSpec` and `discSpec`, we get the CRS and DRS, respectively.

Note that CRS has to be computed before DRS, since DRS is an approximation built on the CRS. Default settings can be adjusted by modifying the `SetParameters` function.

3.1.1 Octave Note

If you want to use Octave, then you will have to install and load the `optim` package. See here for detailed instructions, and the github repository for available packages and appropriate version numbers. Typically, this is easy and involves issuing the following commands at the Octave prompt.[3]

```
% You need to load it every time you open a new Octave session
> pkg load optim                    % load optim in current session
```

This is required to use the nonlinear regression method `lsqnonlin`. This step is not required (and not possible) if you are using Matlab.

3.1.2 Input file

The experimental data $G^*(\omega)$ is reported at a set of discrete frequencies $\{\omega_i, G'(\omega_i), G''(\omega_i)\}$. It is saved as an ASCII text file[4] with 3 columns. The first, second and third columns contain ω, $G'(\omega)$ and $G''(\omega)$ respectively. This text file is labeled as `Gst.dat` by default, although this can be changed if required through the `SetParameters` function. Thus, an example of `Gst.dat` looks like:

[3] It may be easiest to use your system's package manager (`apt-get` on Debian/Ubuntu) to install `octave-optim`.

[4] a plain text file made with Notepad, for example

```
0.0370  12.4048  226.2025
0.0406  14.6341  246.8058
0.0455  17.2642  269.2831
0.0489  20.3672  293.8066
0.0536  24.0280  320.5642
0.0588  28.3462  349.7587
0.0645  33.4399  381.6140
0.0708  39.4517  416.3851
0.0777  46.5402  454.2931
0.0852  54.8993  495.6178
   .        .         .
   .        .         .
   .        .         .
```

3.1.3 Specifying Parameters

The parameters can be set either by modifying the `SetParameters.m` function file, or the GUI. We will first describe the parameters that appear in `SetParameters` file, and comment on the default settings. The GUI allows us to modify the default settings, without having to open the `SetParameters` file directly.

(i) **Overall parameters**:

- `GstFile` = "Gst.dat"

  Default name for experimental complex moduli data, 3 columns ASCII file contains [ω,G',G''].

- `plateau` = 0

  Is there a residual plateau in the data? ON (=1) or OFF (0).

- `verbose` = 1

  Turns ON printing to screen and files (OFF = 0).

- `plotting` = 1

  Turns ON plotting functions (OFF = 0).

(ii) **Continuous spectrum**:

- `ns` = 100

  Number of grid points to represent the continuous spectrum.

- `lamC` $= 0$

  Use a new and improved Bayesian criterion to determine λ_c (if not 0, that value is used as λ_c).

- `SmFacLam` $= 0$

  Smoothing factor, indirect way of controlling λ_c, relative to the one inferred using the Bayesian criterion. Set between -1 (lowest λ explored) and 1 (highest λ explored). Default setting ($= 0$) uses λ_c determined automatically.

- `FreqEnd` $= 1$

  Range of τ in $H(\tau)$ extracted from $G^*(\omega)$. Strictly, if the ω ranges from $\omega_{\min}$ to $\omega_{\max}$, then the "trustable" range of τ is $[e^{\pi/2}/\omega_{\max}, e^{-\pi/2}/\omega_{\min}]$. However, the smoothness criterion imposed on $H(\tau)$ allows us to be less careful.

choice	range of τ
3	$[e^{\pi/2}/\omega_{\max}, e^{-\pi/2}/\omega_{\min}]$
2	$[1/\omega_{\max}, 1/\omega_{\min}]$
1	$[e^{-\pi/2}/\omega_{\max}, e^{\pi/2}/\omega_{\min}]$

(iii) **Discrete spectrum**

- `minTauSpacing` $= 1.25$

  How close do successive modes (τ_{i+1}/τ_i) have to get before we try to merge them?

- `MaxNumModes` $= 0$ Set $= 0$, if you want ReSpect to automatically determine the optimal number of modes; otherwise this parameter sets the maximum possible number of modes.

3.1.4 Output

Depending on whether the `verbose` and `printing` flags are turned on or off, some of the output will be directed to the screen. Regardless, all the necessary output files are stored in the subfolder `output`. This includes the spectrum (`H.dat` for CRS which contains $H(\tau) = \log h(\tau)$, and `dmodes.dat` for DRS), the $G^*(\omega)$ corresponding to the spectrum (`Gfit.dat` for CRS, and `Gfitd.dat` for DRS) as plain text files.

The 'L-curve' corresponding to the CRS is stored in the `rho-eta.dat` as three columns, $\{\lambda_i, \rho_i, \eta_i\}$.

The file `logP.dat` contains two columns, the regularization parameter explored (λ), and the probability distribution using the corresponding Bayesian criterion.

4 Example

In this example, we demonstrate the extraction of the CRS and DRS from $G^*(\omega)$ data. As an example, we use an experimental sample on binary linear-linear blends which spans over seven decades in frequency from Watanabe et al [5]. Experimental dynamic moduli data for this example is provided as the file (`Gst.dat`).

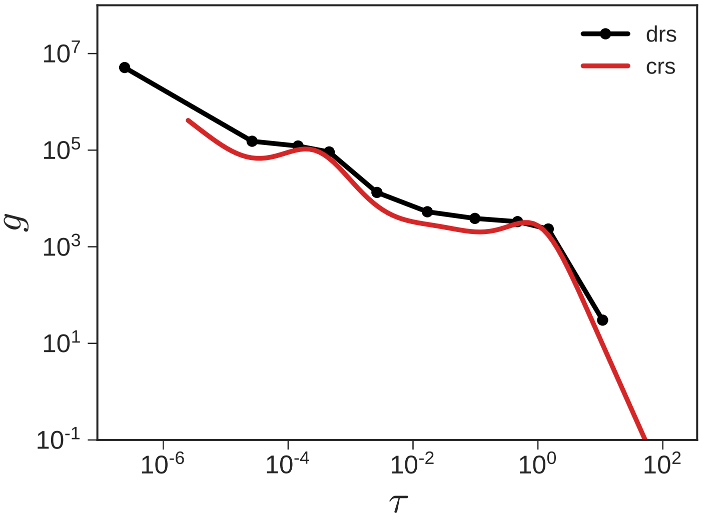


Figure 1: The CRS $h(\tau) = e^{H(\tau)}$ (where $H(\tau) = \log h(\tau)$ is stored in `H.dat`) is shown by the red line. The DRS (`dmodes.dat`) is shown by the circles and black line.

We first ensure we are in the appropriate directory (`Gstar/`) and start an Octave or Matlab session.

At the command line, call the CRS function:

```
>> H = contSpec();
```

If we call the function without any parameters, it reads the parameters from the function `SetParameters`. This is the preferred mode of operation.

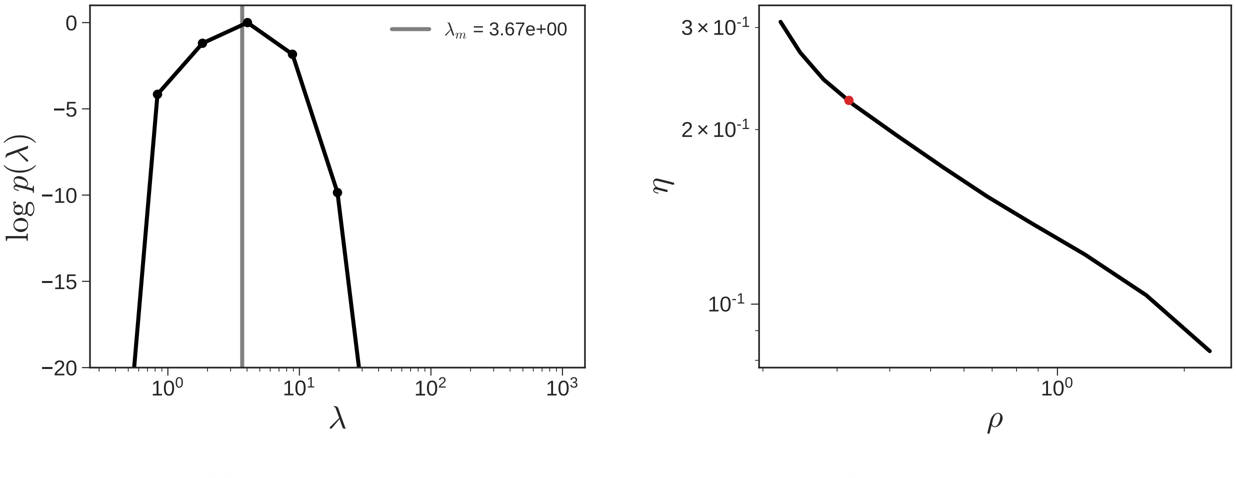
Unlike ReSpect v2 which took about a minute to run, this version runs quite fast. It typically takes less than 10 s on a modern CPU. This one took 2 s on my computer.

Output is written into separate files: spectrum data (`H.dat`), L-curve data $\rho - \eta$ (`rho-eta.dat`), $p(\lambda)$ (`logP.dat`), and dynamic modulus from the corresponding CRS (`Gfit.dat`). All the output files are stored in the `Gstar/output/` subfolder.

The spectrum is shown in fig. 1. Contents of the other three files are plotted and explained in figures 2 and 3.

[5] Watanabe H, Ishida S, Matsumiya Y, Inoue T: Test of full and partial tube dilation pictures in entangled blends of linear polyisoprenes, Macromolecules 37 (2004) 6619-6631

The $\rho - \eta$ curve is presented in figure 2. The location corresponding to the default λ_c chosen by the algorithm is also specially marked.



(a) `logP.dat` (b) `rho-eta.dat`

Figure 2: (a) Probability of different regularization constant values according to the Bayesian criterion, and (b) $\rho - \eta$ curve (columns 2 and 3 of `rho-eta.dat`). The (a) vertical gray line, and (b) red circle marks the optimal value of λ_c.

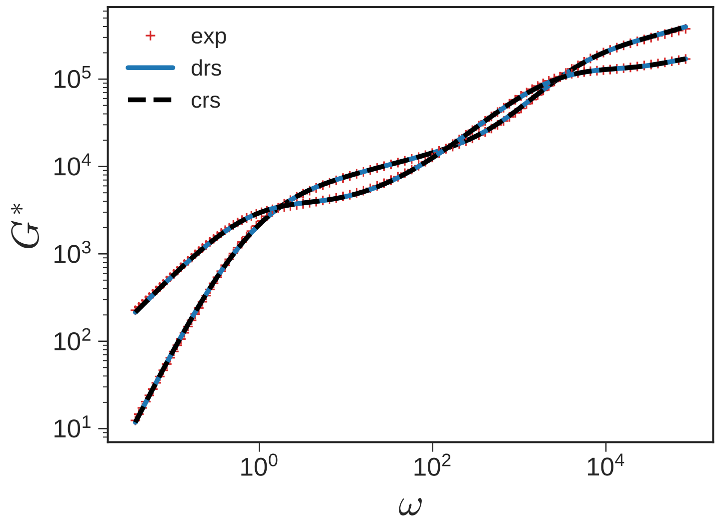


Figure 3: The circles represent the experimental $G'(\omega)$ and $G''(\omega)$ (`Gst.dat`). The overlapping dashed and solid lines represent the corresponding dynamic modulus from the corresponding DRS and CRS (`Gfitd.dat` and `Gfit.dat`), respectively.

To compute the DRS, after the CRS has been determined, use the following command.

```
>> discSpec()
```

If we call the function without any parameters, it reads the parameters from the function `SetParameters`. This is the preferred mode of operation.

Outputs are written into separate files, spectrum data (`dmodes.dat`), the file which contains error and conditioning of the problem, used to determine the optimal number of modes (`Nopt.dat`) and dynamic modulus from the corresponding DRS (`Gfitd.dat`). All the output files are stored in the `Gstar/output/` subfolder.

Using the default setting, the inferred DRS had 12 modes. The spectrum from file `dmodes.dat` is presented in figure 1.