**Noname manuscript No.**
(will be inserted by the editor)

# Relaxation Spectra using Nonlinear Tikhonov Regularization with a Bayesian Criterion

**Sachin Shanbhag**

Received: date / Accepted: date

**Abstract** Nonlinear Tikhonov regularization within a Bayesian framework is incorporated into a computer program called pyReSpect, which infers the continuous and discrete relaxation spectra from oscillatory shear experiments. It uses Bayesian inference to provide uncertainty estimates for the continuous spectrum $h(\tau)$ by propagating the uncertainty in the regularization parameter $\lambda$. The new algorithm is about 6-9 times faster than an older version of the program (ReSpect) in which the optimal $\lambda$ was determined by the L-curve method. About half of the speedup arises from the Bayesian formulation by restricting the window of $\lambda$ explored. The other half arises from the nonlinear formulation for which the spectrum is a weak function of $\lambda$, allowing us to use a coarse mesh for $\lambda$. The

Sachin Shanbhag
Department of Scientific Computing,
Florida State University, Tallahassee, FL 32306, USA
Tel.: +1-850-6446548
E-mail: sshanbhag@fsu.edu

program is tested on three examples, a synthetic spectrum, a H-polymer, and an elastomer with a nonzero terminal plateau.

**Keywords** relaxation spectrum · Bayesian inference · Tikhonov regularization · algorithm · software

## 1 Introduction

The relaxation spectrum of a material $h(\tau)$ is a fundamental property. All linear viscoelastic functions such as time and frequency dependent moduli and compliances can be calculated from it [12]. The utility of $h(\tau)$ extends beyond the linear viscoelastic regime; for example, it informs constitutive models for large or fast deformation [20–22, 27].

Unfortunately, the relaxation spectrum cannot be measured directly; instead, it has to be inferred from linear viscoelastic measurements, such as small amplitude oscillatory shear experiments. These experiments yield the frequency-dependent dynamic moduli, $G^*(\omega) = G'(\omega) + iG''(\omega)$, where $G'(\omega)$ and $G''(\omega)$ are the storage and loss modulus, respectively, and $\omega$ is the frequency of deformation.

Mathematically, $G^*(\omega)$ is related to the continuous relaxation spectrum (CRS) or $h(\tau)$ via,

$$\begin{aligned} G'(\omega) &= G_0 + \int_{-\infty}^{\infty} \frac{\omega^2\tau^2}{1+\omega^2\tau^2}\, h(\tau)\, d\log\tau, \\ G''(\omega) &= \int_{-\infty}^{\infty} \frac{\omega\tau}{1+\omega^2\tau^2}\, h(\tau)\, d\log\tau, \end{aligned} \tag{1}$$

where $G_0$ is the terminal plateau. In gels, for example, $G_0 \neq 0$. Practically, a nonzero $G_0$ is also useful to model materials where a relaxation mode lies outside the experimental observation window.

### 1.1 Status and Scope

Extracting $h(\tau)$ from $G^*(\omega)$ is a difficult inverse problem, and the determination of a unique $h(\tau)$ is challenging [11,29]. Nevertheless, the problem of deducing $h(\tau)$ from $G^*(\omega)$ has a rich history, due to the sheer importance of the problem in mechanical characterization of materials [1,3,4,7,9,10,18,26,30–32,40–42,45]. Most of these attempts seek to constrain the spectrum by appealing to the principles of parsimony and interpretability. Parsimony steers us away from complex and oscillatory solutions towards simple and smooth solutions. Interpretability discards spurious solutions, such as negative values of the CRS. Although the "true spectrum" is elusive, meaningful approximations to it are within reach.

A cursory look at the software landscape reveals that while the number of algorithms published in the literature on this topic is truly staggering, only a small fraction of these have been translated into software that is simultaneously accessible and extensible [18,19,33,34,38,39,42–44]. In this context, *accessibility* implies freely availability for experimentalists to use off-the-shelf on any operating system. *Extensibility* implies sufficient transparency of the algorithm and implementation, so that a developer interested in tinkering or modifying the code is not intimidated. Our Matlab/Octave program, ReSpect, was an attempt in this direction [42]. Maintaining a single program that worked with both Matlab and GNU Octave involved uncomfortable trade-offs. Thus, we reimplemented a significantly improved version in python, called pyReSpect. The program to extract $h(\tau)$ from stress relaxation experiments which yield the stress modulus $G(t)$ is called pyReSpect-time and is available on GitHub (`https://github.com/shane5ul/pyReSpect-time`) [39]. This paper describes a companion program called pyReSpect-freq which brings the same advancements to the analysis of $G^*(\omega)$ (`https://github.com/shane5ul/pyReSpect-freq`). When there is no room for confusion, we use the label pyReSpect to refer to both these programs, since the underlying computational engines are similar.

The principal goal of this work is to describe the significant new additions to the algorithms underlying pyReSpect. Like ReSpect, and NLREG before it [19], it uses nonlinear Tikhonov regularization to determine $h(\tau)$. In Tikhonov regularization, the optimal value of the regularization parameter $\lambda^*$ can be found using a variety of methods [14,15,24]. Earlier, we used a "L-curve" method that was layered with complicated heuristics to account for difficult corner cases. Here, we propose a Bayesian framework for determining $\lambda^*$; it is less arbitrary, provides uncertainty estimates for $h(\tau)$, and runs 6-9 times faster than the older version. A similar criterion was championed by Hansen and conveniently made available for use as a web-interface (`bayesrelax.org`) [16,17]. The key difference between pyReSpect and BayesRelax is that the latter poses a linear Tikhonov regularization problem, which leads to difficulties for some types of $G^*(\omega)$ described later. Furthermore, BayesRelax cannot be used when $G_0 \neq 0$. Thus, pyReSpect can be thought of as a nonlinear Tikhonov regularization algorithm (implemented in NLREG and ReSpect) that uses a Bayesian criterion (implemented in BayesRelax) to determine $\lambda^*$ to characterize uncertainty in $h(\tau)$. In addition, pyReSpect can also compute the discrete relaxation spectrum (DRS); that, however, is not the focus of this paper.

## 2 Methods

Experimental data $\{\omega_i, G'_e(\omega_i), G''_e(\omega_i)\}$ are available at a set of $n$ discrete frequencies $\omega_i$, where $i = 1, \cdots, n$. We stack the storage and loss moduli into a $(2n \times 1)$ column vector $\mathbf{D}$, and the frequencies into a $n \times 1$ vector $\boldsymbol{\omega}$. The top half of $\mathbf{D}$ contains the storage modulus $D_i = G'_e(\omega_i)$, while the bottom half contains the loss modulus $D_{n+i} = G''_e(\omega_i)$. Here, and elsewhere, bold symbols denote vectors and matrices, while regular symbols with subscripts denote components of these quantities (e.g., $\mathbf{D}$ and $D_i$).

We follow the nonlinear Tikhonov regularization strategy of Honerkamp and Weese used in ReSpect [19,42]. We substitute $h(\tau) := e^{H(\tau)}$ in eqn. 1, which makes the problem nonlinear and harder to solve. Nevertheless, the substitution confers three advantages: (i) it allows us to deal with data defined over a large frequency range, (ii) it automatically ensures $h(\tau) \geq 0$ [19], and, as shown later, (iii) provides a recipe for speeding up the determination of $\lambda^*$.

We discretize $H(\tau)$ by dividing the domain between $\tau_{\text{min}}$ to $\tau_{\text{max}}$ into $n_\tau$ equally spaced grid points (on a logarithmic scale) by,

$$\tau_i = \tau_{\text{min}} \left(\frac{\tau_{\text{max}}}{\tau_{\text{min}}}\right)^{\frac{i-1}{n_\tau - 1}}. \tag{2}$$

By default, we set $\tau_1 = \tau_{\text{min}} = e^{-\pi/2}/\omega_{\text{max}}$ and $\tau_{n_\tau} = \tau_{\text{max}} = e^{\pi/2}/\omega_{\text{min}}$, where $\omega_{\text{min}}$ and $\omega_{\text{max}}$ define the frequency window over which $G^*_e(\omega)$ is acquired. Strictly, the domain of $\tau$ over which $H(\tau)$ is reliable is smaller by a factor of $e^\pi$ on either end, i.e, $e^{\pi/2}/\omega_{\text{max}} \leq \tau \leq e^{-\pi/2}/\omega_{\text{min}}$ [11]. However, the smoothness imposed on $H(\tau)$ may increase the apparent range.

The $n_\tau \times 1$ column vector $\mathbf{H} = [H_1, H_2, ..., H_{n_\tau}]^T$ is used to store the value of the discretized spectrum at $[\tau_1, ..., \tau_{n_\tau}]^T$. Typically, we set $n_\tau$ so that there are

5 - 10 grid points per decade. For a given $\mathbf{H}$, the resulting $G^*(\omega)$ is numerically evaluated by integrating eqn. 1 as,

$$\begin{aligned} G'(\omega; \mathbf{H}) &\approx G_0 + \sum_{j=1}^{n_\tau} w_j e^{H_j} \frac{\omega^2 \tau_j^2}{1+\omega^2\tau_j^2}, \\ G''(\omega; \mathbf{H}) &\approx \sum_{j=1}^{n_\tau} w_j e^{H_j} \frac{\omega \tau_j}{1+\omega^2\tau_j^2}, \end{aligned} \tag{3}$$

denote the storage and loss moduli that are computed from $\mathbf{H}$. The quadrature weights $w_i$ incorporate the trapezoidal rule,

$$w_j = \begin{cases} (\Delta \log \tau)/2 & \text{for } j = 1 \text{ and } n_\tau \\ (\Delta \log \tau) & \text{elsewhere,} \end{cases} \tag{4}$$

and $\Delta \log \tau = (\log \tau_{\text{max}} - \log \tau_{\text{min}})/(n_\tau - 1)$. In pyReSpect, the kernel matrix $\mathbf{K}$ is a $2n \times n_\tau$ matrix where the storage ($\mathbf{K}'$) and loss kernel matrices ($\mathbf{K}''$) are stacked vertically. The $(i, j)$ element corresponding to $(\omega_i, \tau_j)$ for the kernel matrices $\mathbf{K}'$ and $\mathbf{K}''$ are, respectively,

$$K'_{ij} = w_j \frac{\omega_i^2 \tau_j^2}{1+(\omega_i\tau_j)^2}, \quad K''_{ij} = w_j \frac{\omega_i \tau_j}{1+(\omega_i\tau_j)^2}, \tag{5}$$

for $1 \leq i \leq n, 1 \leq j \leq n_\tau$. Thus, the dynamic moduli corresponding to $\mathbf{H}$ at frequencies $\boldsymbol{\omega}$ is given by the $2n \times 1$ column vector,

$$\mathbf{G} = G_0\mathbf{u} + \mathbf{K}\mathbf{h}, \tag{6}$$

where $\mathbf{h} = [e^{H_1}, \cdots e^{H_{n_\tau}}]^T$, and $\mathbf{u} = [1, \cdots, 1, 0, \cdots, 0]^T$ is a $2n \times 1$ vector, with $u_i = 1$ for $1 \leq i \leq n$, and $u_i = 0$ for $n < i \leq 2n$. For $G_0 = 0$, $\mathbf{G} = \mathbf{K}\mathbf{h}$ yields $G^*(\omega)$ at $\boldsymbol{\omega}$, which is comparable with $\mathbf{D}$. For $G_0 \neq 0$, the $G_0\mathbf{u}$ term in eqn. 6 adds the terminal plateau to the storage modulus.

### 2.1 Tikhonov Regularization

We seek a CRS for which $\mathbf{G} \approx \mathbf{D}$. To measure the distance between $\mathbf{G}$ and $\mathbf{D}$, we define $\rho^2$, a sum of squared residuals,

$$\rho^2(\mathbf{H}) = \sum_{i=1}^{2n} r_i^2 \tag{7}$$

where the residuals $r_i$ are given by,

$$r_i = \left( \frac{D_i - G_i(\mathbf{H})}{D_i} \right), \quad 1 \leq i \leq 2n. \tag{8}$$

$\rho^2$ represents the mismatch between the experimental and inferred $G^*(\omega)$. Due to ill-conditioning, simply minimizing $\rho^2$ leads to spurious oscillations in $\mathbf{H}$. In order avoid such over-fitting, we incorporate a curvature penalty term that penalizes oscillations,

$$\eta^2(\mathbf{H}) = ||\mathbf{L}\mathbf{H}||^2, \tag{9}$$

where $\mathbf{L}$ is a $n_l \times n_\tau$ tridiagonal matrix that encodes the difference operator for the second derivative,

$$L_{ij} = \begin{cases} -2 & j = i \\ 1 & j = i \pm 1 \\ 0 & \text{elsewhere.} \end{cases} \tag{10}$$

The curvature condition is applied at only the $n_l = n_\tau - 2$ internal points $H_2$, ..., $H_{n_\tau - 1}$. $\eta^2$ is small for smooth $\mathbf{H}$, and is equal to zero when $\mathbf{H}$ is linear in $\tau$.

In Tikhonov regularization, the relative importance of $\rho^2$ and $\eta^2$ is controlled by the regularization parameter $\lambda$. For a given $\lambda$, we find the optimal $\mathbf{H}_\lambda$ by minimizing the cost function $V = \rho^2 + \lambda\eta^2$,

$$\mathbf{H}_\lambda = \min_{\mathbf{H}} V(\mathbf{H}, \lambda) = \rho^2(\mathbf{H}) + \lambda\eta^2(\mathbf{H}). \tag{11}$$

When $\lambda$ is too large, the smoothness condition dominates the cost function, and we get an overly smooth CRS that fits the experimental data poorly (small $\eta^2$, and large $\rho^2$). When $\lambda$ is too small, the smoothness condition is essentially ignored, and we end up with the original non-regularized problem, which is ill-conditioned. This leads to an oscillatory $\mathbf{H}$ that is sensitive to noise in the data (large $\eta^2$, and small $\rho^2$).

The idea behind Tikhonov regularization is to choose an optimal $\lambda = \lambda^*$ that is approximately midway between these extremes. There are several different methods of choosing $\lambda^*$ [14,15,24]. While the value of $\lambda^*$ depends on the method used; the estimates of $H(\tau)$ are often similar. One of these methods, called the L-curve method, is employed in ReSpect. We plot $\eta^2$ versus $\rho^2$ obtained by solving the minimization problem (eqn 11) for a range of $\lambda$ between $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$. Often, this yields an "L-shaped" curve (see inset in fig. 3). In the L-curve method, a suitable point near the corner of the "L" is used to determine $\lambda_c \approx \lambda^*$. The subscript "$c$" stands for "corner". When a clear corner cannot be discerned (see inset in figure 5), additional heuristics are required to determine $\lambda_c$.

In pyReSpect, we solve the minimization problem (eqn. 11) using the nonlinear least-squares minimizer "least_squares()" from scipy, which uses a trust-region reflective method [8,25]. This requires us to furnish subroutines that supply residuals and the Jacobian. A detailed description of these quantities is provided because they are also useful in Bayesian analysis.

Before that, it is useful to consider the size of the computational problem. When $G_0 = 0$ the number of unknowns $(H_1, ..., H_{n_\tau})$ is $n_\tau$. When $G_0 \neq 0$ also has to be determined, the number of unknowns increases to $n_\tau + 1$. The number of residuals, $2n + n_l$, is the same in either case. The size of the Jacobian matrix for

$G_0 = 0$ is $(2n+n_l) \times n_\tau$. When $G_0 \neq 0$, it has an extra column $(2n+n_l) \times (n_\tau+1)$ to account for the additional unknown $(G_0)$.

### *2.1.1 Residuals and Jacobian*

The objective function (eqn 11) can be written as a sum of squared residuals,

$$V(\mathbf{H}, \lambda) = \sum_{i=1}^{2n+n_l} r_i^2 = \sum_{i=1}^{2n} r_i^2 + \sum_{i=2n+1}^{2n+n_l} r_i^2,$$

where the first $2n$ residuals correspond to $\rho^2$ (eqn. 8). The subsequent $n_l$ residuals correspond to $\lambda\eta^2$, and are given by,

$$r_{2n+i} = \sqrt{\lambda}\left(H_{i+1} - 2H_i + H_{i-1}\right), \quad 2 \leq i \leq n-1. \tag{12}$$

These can be conveniently built from the vector $\mathbf{LH}$.

The Jacobian $\mathbf{J}$ is defined as $J_{ij} = \partial r_i / \partial H_j$. Let us first consider the simpler case with $G_0 = 0$. For the first $2n$ rows (eqn. 8),

$$\frac{\partial r_i}{\partial H_j} = -\frac{1}{D_i}\frac{\partial G_i(\mathbf{H})}{\partial H_j} = -\frac{\nabla G_{ij}}{D_i}. \tag{13}$$

From eqn 3, the derivative of the kernel can be computed; all the terms in the summation, except the particular $H_j$, drop off. The $2n \times n_\tau$ matrix $\nabla \mathbf{G}$ is,

$$\nabla G_{ij} = e^{H_j} K_{ij}. \tag{14}$$

Note that apart from the additional $e^{H_j}$ factor, $\nabla \mathbf{G}$ is identical to the kernel matrix $\mathbf{K}$. Thus, the derivative of the residual is,

$$J_{ij} = \frac{\partial r_i}{\partial H_j} = -\frac{e^{H_j} K_{ij}}{D_i}, \quad 1 \leq i \leq 2n, 1 \leq j \leq n_\tau. \tag{15}$$

The last $n_l$ rows of the Jacobian are are simply a rescaled form of $\mathbf{L}$.

$$J_{2n+i,j} = \sqrt{\lambda} L_{ij}, \quad 1 \leq i \leq n_l, 1 \leq j \leq n_\tau. \tag{16}$$

If $G_0 \neq 0$, then the Jacobian has an additional column, which is zero everywhere, except,

$$J_{i,n_\tau+1} = \frac{\partial r_i}{\partial G_0} = -\frac{1}{D_i}\frac{\partial G_i(\mathbf{H})}{\partial H_j} = -\frac{1}{D_i}, \quad 1 \leq i \leq n. \tag{17}$$

## 2.2 Bayesian Inference

In a Bayesian framework, we treat $\lambda$ as a random variable characterized by a probability distribution function $\pi(\lambda)$. The posterior distribution $\pi(\lambda|\mathbf{D})$ is proportional to the product of the likelihood or evidence $\pi(\mathbf{D}|\lambda)$ and the prior,

$$\pi(\lambda|\mathbf{D}) \sim \pi(\mathbf{D}|\lambda)\,\pi(\lambda).$$

Here, we assume a simple exponential prior $\pi(\lambda) = e^{-\lambda}$. The evidence is the integral over all possible spectra $\mathbf{H}$,

$$\pi(\mathbf{D}|\lambda) = \int \pi(\mathbf{D}|\mathbf{H},\lambda)\,\pi(\mathbf{H}|\lambda)\,d\mathbf{H}. \tag{18}$$

The components of the objective function in eqn 11 and the evidence (eqn 18) are closely related. In particular, the first term in the integral, $\pi(\mathbf{D}|\mathbf{H},\lambda)$ can be related to $\rho^2$ via,

$$\pi(\mathbf{D}|\mathbf{H},\lambda) \sim e^{-\rho^2} = \mathcal{N}(\mathbf{G}, \mathbf{D}/\sqrt{2}). \tag{19}$$

This asserts that the observed data are normally distributed around the predicted values $\mathbf{G}(\mathbf{H})$ with a standard deviation proportional to $\mathbf{D}$ (see eqn. 8). Note that this term does not depend on $\lambda$ explicitly, and the prefactor or normalization constant is independent of $\mathbf{H}$.

The second term in eqn 18, $\pi(\mathbf{H}|\lambda)$ corresponds to the regularization constraint or $\eta^2$,

$$\pi(\mathbf{H}|\lambda) = \frac{e^{-\lambda\eta^2(\mathbf{H})}}{Z_{\eta^2}} \tag{20}$$

where the curvature penalty term, $\eta^2(\mathbf{H}) = ||\mathbf{L}\mathbf{H}||^2$. $Z_{\eta^2}$ is the normalization constant, which is determined shortly.

In Tikhonov regularization, the cost function is $V(\mathbf{H}, \lambda) = \rho^2(\mathbf{H}) + \lambda\eta^2(\mathbf{H})$. Thus, eqns. 19 and 20 imply that the evidence can be written as,

$$\pi(\mathbf{D}|\lambda) = \int \frac{e^{-V(\mathbf{H},\lambda)}}{Z_{\eta^2}}\, d\mathbf{H}. \tag{21}$$

We can use the saddle point or Laplace approximation to estimate the normalization constant. This means that if $\mathbf{H}_\lambda$ is the optimal solution at a given $\lambda$, we can integrate over all spectra $\mathbf{H}$,

$$\int \frac{e^{-V(\mathbf{H},\lambda)}}{Z_{\eta^2}}\, d\mathbf{H} \approx \frac{e^{-V(\mathbf{H}_\lambda)}}{Z_{\eta^2}} \sqrt{\frac{(2\pi)^{n_\tau}}{\det(\lambda\mathbf{A} + \mathbf{B})}}, \tag{22}$$

to obtain the evidence for $\mathbf{D}$ at a particular $\lambda$. In the expression above, the Hessians $\mathbf{A} := \nabla\nabla\eta^2$ and $\mathbf{B} := \nabla\nabla\rho^2$ are evaluated at the optimum $\mathbf{H}_\lambda$. We can define the Hessian corresponding to the cost function as $\mathbf{C} := \nabla\nabla V = \lambda\mathbf{A} + \mathbf{B}$.

Unlike the normalization constant for $\rho^2$, $Z_{\eta^2}$ cannot be neglected because it depends on $\lambda$. Nevertheless, we can use the saddle point approximation again to estimate it. From eqn. 20,

$$Z_{\eta^2} = \int e^{-\lambda\eta^2(\mathbf{H})} d\mathbf{H} = e^{-\lambda\eta^2(\mathbf{H}_0)} \sqrt{\frac{(2\pi)^{n_\tau}}{\det(\lambda\mathbf{A})}} \tag{23}$$

For $\lambda \gg 1$, the $\mathbf{H}$ that maximizes $e^{-\lambda\eta^2}$, $\mathbf{H}_0$, corresponds to $\eta^2 = 0$. Thus, it is safe to assume that $e^{-\lambda\eta^2(\mathbf{H}_0)} = 1$. In pyReSpect, the level of discretization $n_\tau$ is held fixed throughout; therefore, the saddle point approximation for the posterior can be written as,

$$\pi(\lambda|\mathbf{D}) \sim \pi(\mathbf{D}|\lambda)\pi(\lambda) \sim e^{-V(\mathbf{H}_\lambda)} \sqrt{\frac{\det(\lambda\mathbf{A})}{\det(\mathbf{C})}}\, e^{-\lambda} \tag{24}$$

Since $\det(\lambda\mathbf{A}) = \lambda^{n_\tau}\det(\mathbf{A})$ where $n_\tau$ is the size of matrix $\mathbf{A}$,

$$\log\pi(\lambda|\mathbf{D}) \sim -V(\mathbf{H}_\lambda) + \frac{1}{2}\left[\log\det(\mathbf{A}) + n_\tau\log\lambda - \log\det(\mathbf{C})\right] - \lambda, \tag{25}$$

### *2.2.1 Hessians*

We now seek to explicitly resolve the matrices $\mathbf{A}$ and $\mathbf{B}$, starting with $\mathbf{A} = \nabla\nabla\eta^2$.

$$\eta^2(\mathbf{H}) = (\mathbf{LH})^T(\mathbf{LH}) = \mathbf{H}^T\mathbf{L}^T\mathbf{LH}. \tag{26}$$

The Hessian of the quadratic form is a square $(n_\tau \times n_\tau)$ matrix,

$$\mathbf{A} = \nabla\nabla\eta^2 = \frac{\partial^2(\mathbf{H}^T\mathbf{L}^T\mathbf{LH})}{\partial\mathbf{H}\,\partial\mathbf{H}^T} = 2\mathbf{L}^T\mathbf{L}. \tag{27}$$

Note that $\mathbf{A}$ is constant and does not depend on $\lambda$.

To compute $\mathbf{B} = \nabla\nabla\rho^2$, we start by considering the $j^{\text{th}}$ element of the vector $\nabla\rho^2$,

$$\frac{\partial\rho^2}{dH_j} = \sum_{k=1}^{2n} \frac{\partial}{dH_j} r_k^2 = \sum_{k=1}^{2n} 2r_k \frac{\partial r_k}{\partial H_j}, \quad 1 \leq j \leq n_\tau. \tag{28}$$

The derivative of the residual was previously evaluated (eqn. 15) to determine the Jacobian, $\partial r_k/\partial H_j = J_{kj}$, for $1 \leq k \leq 2n, 1 \leq j \leq n_\tau$. Thus, eqn. 28 may be simplified as,

$$\frac{\partial\rho^2}{dH_j} = 2\sum_{k=1}^{2n} r_k \frac{\partial r_k}{\partial H_j} = 2\sum_{k=1}^{2n} r_k J_{kj} \tag{29}$$

To find the Hessian $\mathbf{B} = \nabla\nabla\rho^2$, we take the partial with respective to $H_i$,

$$\begin{aligned} B_{ij} &= \frac{\partial}{\partial H_i}\frac{\partial\rho^2}{\partial H_j} \\ &= 2\sum_k \left(\frac{\partial r_k}{\partial H_i} J_{kj} + r_k \frac{\partial J_{kj}}{\partial H_i}\right) \\ &= 2\sum_k \left(J_{ki}J_{kj} + r_k J_{ki}\delta_{ij}\right). \end{aligned} \tag{30}$$

The Kronecker delta function in the second term of the summation ensures that it is only active along the diagonal of the Hessian $(i = j)$. Note that size of the matrix $\mathbf{B} = \nabla\nabla\rho^2$ is also $n_\tau \times n_\tau$. In matrix form, we can write,

$$\begin{aligned}\mathbf{A} &= 2\mathbf{L}^T\mathbf{L}\\ \mathbf{B} &= 2\left(\mathbf{J}^T\mathbf{J} + \mathrm{diag}(\mathbf{r}^T\mathbf{J})\right),\end{aligned} \tag{31}$$

where $\mathbf{J}$ is the $2n \times n_\tau$ part of Jacobian matrix, and $\mathbf{r}$ is the $2n \times 1$ residual vector that corresponds to $\rho^2$. The advantage of expressing $\mathbf{A}$ and $\mathbf{B}$ in matrix form is that the equations translate seamlessly even when $G_0 \neq 0$.

### *2.2.2 Algorithm in pyReSpect*

We precompute the matrix $\mathbf{A} = \mathbf{L}^T\mathbf{L}$. The prefactor of two (eqn 31) can be thrown away from both $\mathbf{A}$ and $\mathbf{B}$ since it cancels out eventually. We scan through a range of $\lambda \in [\lambda_{\mathrm{min}}, \lambda_{\mathrm{max}}]$. At each $\lambda$,

(i) compute $\mathbf{H}_\lambda$ by minimizing $V(\mathbf{H}, \lambda)$ and store it.

(ii) compute $\rho^2$, $\eta^2$, $V(\lambda) = \rho^2 + \lambda\eta^2$.

(iii) compute the residual $\mathbf{r}$ and Jacobian $\mathbf{J}$.

(iv) compute $\mathbf{B} = \left(\mathbf{J}^T\mathbf{J} + \mathrm{diag}(\mathbf{r}^T\mathbf{J})\right)$ and hence $\mathbf{C} = \lambda\mathbf{A} + \mathbf{B}$.

(v) compute log posterior probability,

$$\log \pi_\lambda = -V(\mathbf{H}_\lambda) + \frac{1}{2}\left[\log|\mathbf{A}| + n_\tau \log\lambda - \log|\mathbf{C}|\right] - \lambda$$

We normalize $\log \pi_\lambda$ to get the posterior $\pi(\lambda)$ (the dependence on $D$ in $\pi(\lambda|\mathbf{D})$ is dropped henceforth for brevity), and find the mean (subscript "$m$")

$$\lambda^* \approx \lambda_m = E[\lambda] = \exp\left(\sum_\lambda \log\lambda \cdot \log\pi_\lambda\right). \tag{32}$$

Finally, we compute the mean spectrum

$$\bar{\mathbf{H}} = E[\mathbf{H}_\lambda] = \int \mathbf{H}_\lambda\, \pi(\lambda)\, d\lambda, \tag{33}$$

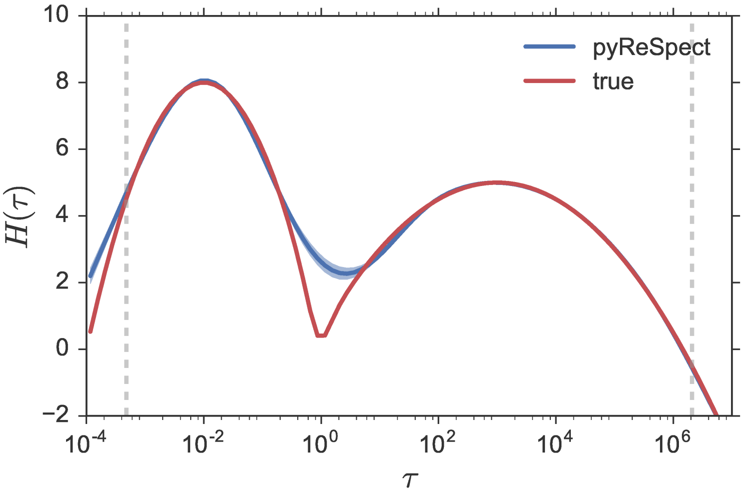

**Fig. 1** The true spectrum is given by eqn 35. The spectrum inferred by pyReSpect captures the two peaks. Due to the smoothness constraint, it is unable to resolve the sharp cusp near $\tau \approx 1$. The shaded region around the inferred spectrum represents the uncertainty due to $\lambda$. The dashed vertical lines mark the domain $e^{\pi/2}/\omega_{\max} \leq \tau \leq e^{-\pi/2}/\omega_{\min}$ in which the inferred spectrum is reliable [11].

and the error estimate,

$$\mathrm{var}(\mathbf{H}) = E[(\mathbf{H}_\lambda - \bar{\mathbf{H}})^2]. \tag{34}$$

Typically, the mean spectrum $\bar{\mathbf{H}}$ is approximately equal to the spectrum computed at the mean $\lambda$; i.e. $\bar{\mathbf{H}} \approx \mathbf{H}_{\lambda^*}$.

### 2.3 Test Cases

We consider three test cases in this paper: (i) a synthetic dataset generated from a spectrum with two unequal peaks, (ii) an H-polymer [28,36], and (iii) an elastomer with nonzero $G_0$ [13,27].

(i) **Synthetic Data from Spectrum with Unequal Peaks**: This dataset is fashioned after the example used in Honerkamp and Weese (their figure 1a) [19]. The underlying spectrum is given by a weighted sum of two quadratic components,

$$\begin{aligned} H_1(\tau) &= 8 - 2\left(\log_{10}\tau + 2\right)^2 \\ H_2(\tau) &= 5 - 0.5\left(\log_{10}\tau - 3\right)^2 \\ H_e(\tau) &= w(\tau) H_1(\tau) + \left(1 - w(\tau)\right) H_2(\tau), \end{aligned} \tag{35}$$

where the weight function $w(\tau) = 1/(1 + e^{H_2 - H_1})$ preferentially overweights the component ($H_1$ or $H_2$) with the larger magnitude at any given $\tau$. The spectrum is shown in figure 1. The first (taller) peak corresponds to $H_1$ while the second (shorter) peak corresponds to $H_2$. Synthetic data is generated by adding 2.5% noise to the $G^*(\omega)$ computed numerically from $H_e(\tau)$ using eqn. 1 at $n = 100$ logarithmically equispaced points. It extends over 11 orders of magnitude, and is chosen to represent a polymer in terminal, plateau, transition, and glassy regions (see figure 2).

For a similar example, Honerkamp and Weese showed that using a linear least squares method (like FTIKREG) results in serious difficulties due to the contributions corresponding to the two peaks. At small $\tau$, the higher peak contributes significantly to $G^*(\omega)$ in the transition and glassy regions, while at large $\tau$, the lower peak contributes modestly to the linear rheology in the terminal and plateau regions. It is difficult for FTIKREG to resolve both peaks simultaneously.

(ii) **H-polymer**: This is the H3A1A polystyrene H-polymer which was synthesized, and first studied by Roovers and co-workers [36, 37]. The nominal molecular weight of the arms and the backbone are 132 kDa and 123 kDa, respectively. The linear viscoelasticity was restudied by Lentzakis [28], using a more advanced rheometer and resolving low-frequency measurements more carefully.

This data is presented in figure 4. It was used by Ankiewicz et al. to discuss the advantages of using a CRS to identify power-laws and relaxation regimes that are not distinctly visible in the experimental $G^*(\omega)$ [2]. In particular, they were able to recognize a $\tau^{-1/2}$ Rouse regime at short time scales in the spectrum (not visible in $G''(\omega)$ due to contamination from glassy modes), and $\tau^{-1/4}$ regime at intermediate time corresponding to the relaxation of the arms, prior to terminal relaxation.

(iii) **Elastomer with Terminal Plateau**: This test case (shown in figure 6) corresponds to Sorbothane 70, a cross-linked polyurethane rubber, at 20°C, by superposing data obtained at 16 different temperatures [13, 27]. The elastomer has good shock-absorption properties because high energy dissipation occurs in the transition zone between rubbery and glassy behavior at frequencies ($10^2 - 10^4$ Hz) that are typical of impacts.

There are two reasons for selecting this as a test case: (i) $G^*(\omega)$ data extends over 18 orders of magnitude, and (ii) nonzero terminal plateau, $G_0 \neq 0$, which is visible as a plateau in $G'(\omega)$ at low frequencies.

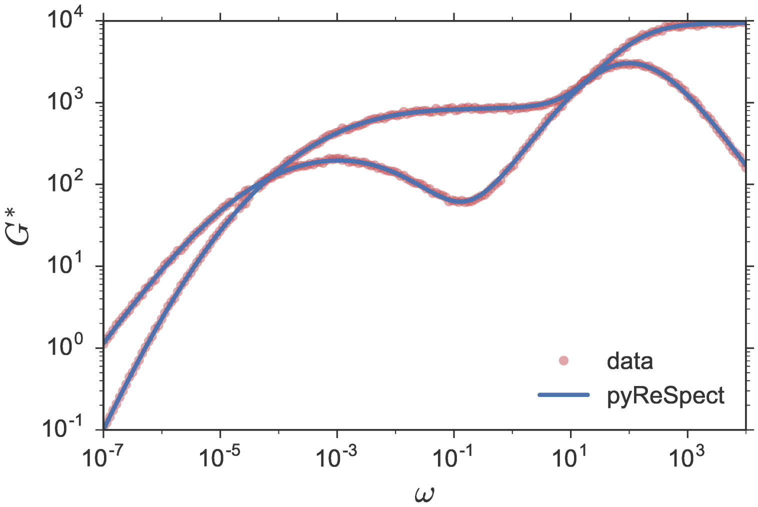

**Fig. 2** Symbols are data generated from the true spectrum shown in fig. 1, with 2.5% noise added. Lines depict $G^*(\omega)$ resulting from the inferred spectrum.

## 3 Results and Discussion

For all the examples below, unless explicitly mentioned otherwise, the CRS was determined by using default settings in the pyReSpect program. Unlike the original ReSpect, these defaults are quite robust and reliable.

### 3.1 Synthetic Spectrum

The true (eqn 35) and inferred spectra are compared in figure 1. For a similar spectrum, Honerkamp and Weese found that a linear least squares method was unable to simultaneously resolve both peaks. This motivated the $h = e^H$ substitution used in NLREG (and this work). A similar problem was reported by Hansen using BayesRelax, which uses a Bayesian framework without the $h = e^H$ substitu-

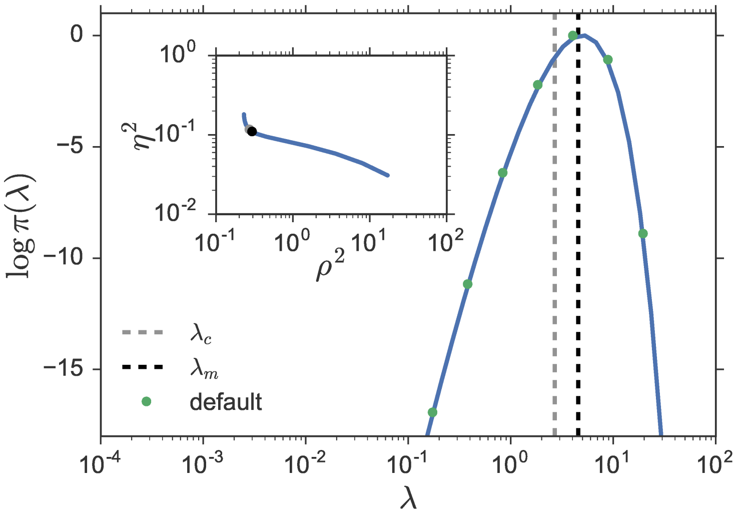

**Fig. 3** Logarithm of the posterior distribution of $\lambda$. Symbols (green) denote the discrete values of $\lambda$ explored using default settings in pyReSpect. The mean value $\lambda_m$ is indicated by the dashed black line. The dashed gray line indicates $\lambda_c$ - the "corner" value using the older heuristic method. The inset shows the $\eta^2$ v/s $\rho^2$ curve, and the location of $\lambda_m$ and $\lambda_c$ on it (gray and black circles).

tion [17]. Here, the agreement between the true and inferred spectra is quite good near and around the two peaks.

The difference between them is most pronounced near the sharp cusp ($\tau \approx 1$) where the two quadratic modes in the true spectrum intersect. Due to the smoothness constraint placed on the spectrum, such sharp corners in $H(\tau)$ are heavily penalized. The $\eta^2$ for the inferred spectrum is an order of magnitude smaller than that for the true spectrum. This tradeoff is especially understandable, since the fit between the experimental data and $G^*(\omega)$ resulting from the inferred $H(\tau)$ (figure 2) is remarkably good. Note that it is possible to incorporate more flexible regularization terms, such as those implemented in the program GENEREG [38].

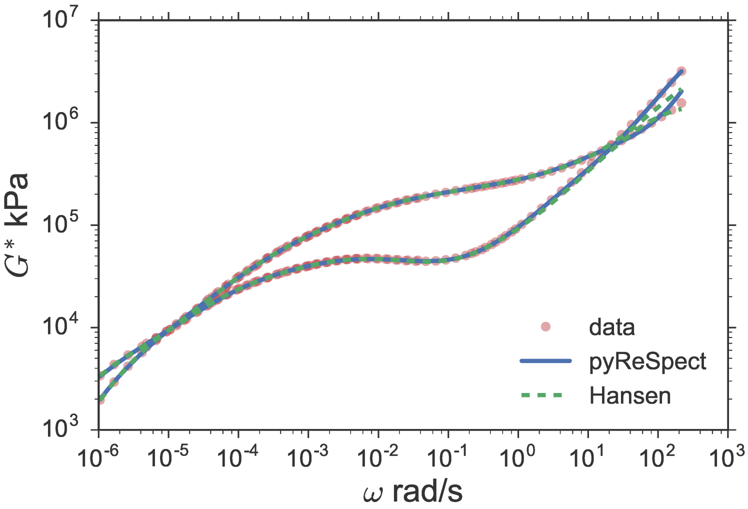

**Fig. 4** $G^*(\omega)$ for the H-polymer data H3A1A [2,28]. The solid line shows the fit obtained from pyReSpect. The dashed green line shows the prediction using the program BayesRelax [17].

The inset to figure 3 shows the competition between $\rho^2$ and $\eta^2$. As $\lambda$ is increased $\eta^2$ decreases, and $\rho^2$ increases. The older heuristic method was designed to find the optimal $\lambda^*$ as the corner of this $\eta^2$ v/s $\rho^2$ curve. It yields $\lambda_c = 2.68$ as the corner. We can use the Bayesian inference framework to find the mean of the posterior distribution $\pi(\lambda)$ (fig. 3). This analysis yields, $\lambda_m = 4.55$. Although $\lambda_m > \lambda_c$ in this case, they essentially overlap on the $\eta$ v/s $\rho$ curve, and lead to nearly identical spectra; i.e., $H_{\lambda_c}(\tau) \approx H_{\lambda_m}(\tau)$.

### 3.2 H-Polymer

The H-polymer data H3A1A is shown in figure 4 [28]. This dataset was studied by Ankiewicz et al. using the program NLREG [2,19]. They argued that it was easier to detect power-law behavior in $H(\tau)$ than in the $G^*(\omega)$ data from which it was inferred.

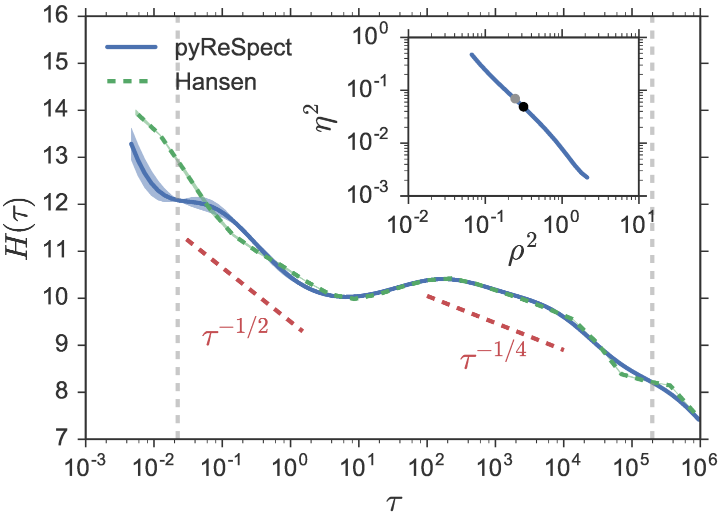

**Fig. 5** The solid blue line shows the spectrum obtained from pyReSpect, while the dashed green line shows $H(\tau)$ obtained from BayesRelax. The power-laws at short ($\tau^{-1/2}$) and intermediate times ($\tau^{-1/4}$) are identified. The spectrum is reliable in the region between the dashed gray lines. The inset shows the $\eta^2$ versus $\rho^2$ curve, which does not have a well-defined corner. Similar to figure 1, the black and the gray symbols locate $\lambda_m$ and $\lambda_c$, respectively.

Figure 5 depicts $H(\tau)$ obtained using pyReSpect with default settings. At short times between $\tau = 10^{-2} - 10^{0}$s, the signature of Rouse relaxation $h(\tau) = \tau^{-1/2}$ is visible, even though it is unclear in the corresponding region of the frequency plot [2]. The $G''(\omega) \sim \omega^{1/2}$ is not clearly visible, because the terminal relaxation of the glassy modes interferes with $G''(\omega)$. In the intermediate time region $\tau = 10^2 - 10^4$s, we observe a weaker power-law close to the $h(\tau) = \tau^{-1/4}$ expected for relaxation of star arms. Most of the terminal relaxation of the backbone lies beyond the reliable limit.

Figure 5 also depicts the spectrum obtained using BayesRelax [17]. The program has a convenient web-interface (bayesrelax.org), where data can be uploaded.

The Fortran 77 computer program which carries out these calculations is also available at that site. Like pyReSpect, BayesRelax uses Bayesian analysis to obtain a distribution of $\pi(\lambda)$, from which $h(\tau)$ is determined. Unlike pyReSpect or NLREG, however, it solves a linear least squares problem to determine $h(\tau)$. As a result the spectrum is not guaranteed to be positive, and suffers from limitations similar to FTIKREG. Nevertheless, in this particular case, the spectrum generated closely matches the $H(\tau)$ obtained by pyReSpect, except perhaps at short $\tau$.

The $G^*(\omega)$ corresponding to the spectra obtained using pyReSpect and BayesRelax are compared with the experimental data in figure 4. The fits agree with the data everywhere, except at large frequencies (corresponding to short $\tau$), where the mismatch between the data and $G^*(\omega)$ obtained from BayesRelax is more pronounced. Even the $G^*(\omega)$ obtained using pyReSpect shows some deviations from the last $G''(\omega)$ data-point. It is unclear whether this arises from a break-down of Kramers-Kronig relationship, that is sometimes observed when data from several different experiments are superposed to obtain a master curve. Ankiewicz et al. did not report the fit with $G^*(\omega)$ in their paper [2], but we expect the outcome to be similar.

The $\eta^2$ versus $\rho^2$ curve is shown in the inset to figure 5. Over the range of $\lambda$ explored, it is a monotonically decreasing curve. Since there is no obvious corner on this "L-curve", the original version of pyReSpect used a series of complicated heuristics to determine $\lambda_c$. For this example, $\lambda_c = 2.81$, which is quite close to $\lambda_m = 4.92$, determined from the posterior distribution $\pi(\lambda)$. Due to the proximity of $\lambda_c$ and $\lambda_m$, they yield very similar $H(\tau)$. Nevertheless, the criterion used to determine $\lambda_m$ is simpler, easier to articulate and comprehend, and generalizes quite well. When the L-curve has a corner (fig 1), $\lambda_m$ is close to it. In other cases,

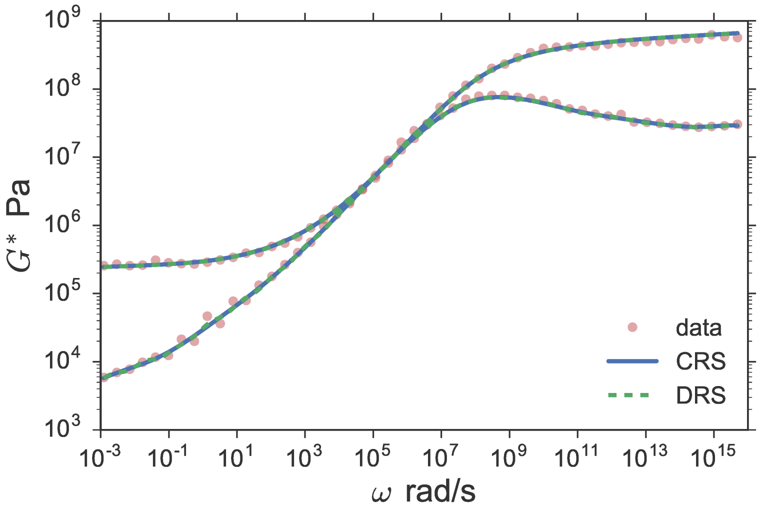

**Fig. 6** Symbols show experimental $G^*(\omega)$ for Sorbothane 70 shifted to 20°C. The solid and dashed lines show the fits using the CRS and DRS, respectively. These spectra are shown in fig. 7.

when $\eta^2$ versus $\rho^2$ does not have a sharp corner, it finds a reasonable $\lambda_m$ that is close to the possibly over-engineered heuristic used for $\lambda_c$.

### 3.3 Terminal Plateau

The final example, shown in figure 6, is an elastomer. Unlike previous cases, it has a nonzero $G_0$. In pyReSpect, we can turn the "plateau" flag on in the input file to find $H(\tau)$ for such datasets. In this case, pyReSpect returns $G_0 = 0.24$ MPa.

pyReSpect can also be used to determine the discrete relaxation spectrum (DRS) [5,6,23,35] which consists of pairs of relaxation times and strengths $\{\tau_i, g_i\}$, with $i = 1, 2, ..., N$, where $N$ is the number of modes in the spectrum. pyReSpect automatically determines an optimal number of modes based on an information criterion [39]. The relationship between the DRS and $G^*(\omega)$ is given by,

$$\begin{aligned} G'(\omega) &= G_0 + \sum_{i=1}^{N} g_i \frac{\omega^2 \tau_i^2}{1+\omega^2\tau_i^2} \\ G''(\omega) &= \sum_{i=1}^{N} g_i \frac{\omega \tau_i}{1+\omega^2\tau_i^2} \end{aligned} \tag{36}$$

Figure 7 depicts the CRS and DRS for this example. The fit of the $G^*(\omega)$ corresponding to these spectra are shown in figure 6. For this case, $\lambda_m = 7.39$ is smaller than $\lambda_c = 178$ obtained using the older method. However, it turns out that $H_\lambda$ is somewhat insensitive to $\lambda$ for this example. This is evident in figure 7; in addition to the CRS determined at $\lambda = \lambda_m$ (shown by the thick red line), $H_\lambda$s for $\lambda \in [0.4, 1000]$ are shown by thin gray lines. These curves essentially superpose, except near the ends, outside the region demarcated by the dashed gray lines, where they flare out mildly. Thus, for this example, the inferred CRS at $\lambda_m$ and $\lambda_c$ are essentially indistinguishable. However, this remarkable insensitivity of $H_\lambda$ to $\lambda$ is not a general feature. In the appendix, figure 8 shows a similar plot for the H-polymer, where somewhat greater dispersion in $H_\lambda$ can be observed.

Here, the DRS has $N = 27$ modes, and closely mimics the shape of the CRS. In ref. 27, the authors fit a $N = 14$ and $N = 15$ mode DRS, in conjunction with a viscous mode, to selectively fit the data at frequencies between $10^{-2} - 10^{4}$ rad/s. Since the primary object of investigation in their study was the impact response of elastomers in drop tests, matching $G^*(\omega)$ for frequencies higher than $10^5$ rad/s was deemed unnecessary. The duration of impact is of the order of several milliseconds, and modes faster than $10^5$ rad/s contribute little to the viscoelastic impact response [27]. Nevertheless, the DRS inferred from pyReSpect offers a compelling description of the entire $G^*(\omega)$ response with a sparse mode density of approximately 1.4 modes/decade.

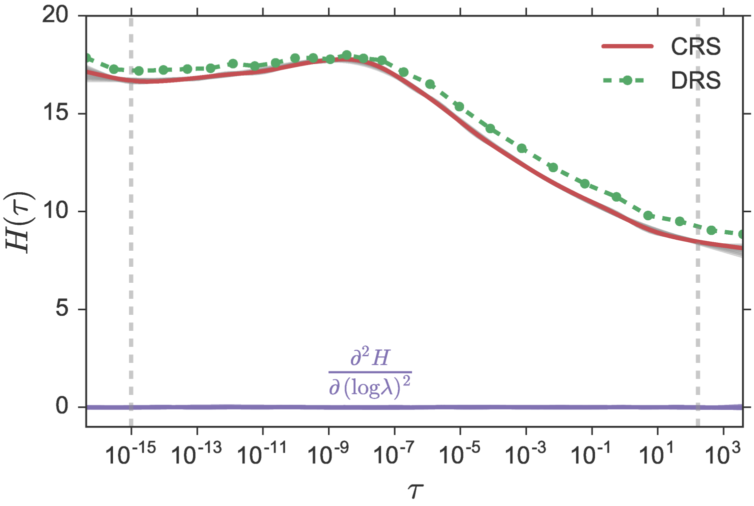

**Fig. 7** The CRS and DRS are represented by the solid and circle-dashed lines, respectively. $H_\lambda(\tau)$ determined at different values of $\lambda$ is overlaid in gray. Here, these curves essentially superpose, except near the edges. The gray vertical dashed lines mark the region in which the spectrum is reliable. The second derivative of the spectrum with respect to $\lambda$ is also shown by purple solid lines.

### 3.4 Advantages of the Nonlinear Bayesian Formulation

The Bayesian formulation for determining $\lambda^*$ confers several advantages. It is simple and robust. More importantly, it provides error-bars for $H(\tau)$ that propagate the uncertainty in determining $\lambda^*$. This is a major improvement over previous versions of ReSpect and pyReSpect which offered no such uncertainty quantification. Surprisingly, the Bayesian formulation also leads to a major improvement in efficiency.

Table 1 depicts the computational cost using the Bayesian framework (labeled "Bayes"), and the older method (labeled "Old") using the complicated heuristic to determine $\lambda^*$. Calculations were performed on a desktop computer with an

| | **Bayes** | | **Old** | |
|---|---|---|---|---|
| **sample** | total | $\lambda$-scan | total | $\lambda$-scan |
| synthetic spectrum | 3.2 | 1.5 | 10.1 | 8.8 |
| H-polymer | 3.4 | 1.4 | 14.2 | 12.1 |
| elastomer | 3.1 | 1.8 | | |

**Table 1** Computational time in seconds to run pyReSpect with (Bayes) and without (old) the Bayesian framework, for the three cases studied here. The old method cannot be used for the last case since $G_0 \neq 0$.

Intel i7-6700 3.4GHz CPU. The total cost of the calculation, which includes the overhead: I/O, setup, initialization, printing and plotting, is shown in addition to the cost of the most expensive step, viz., scanning a range of $\lambda$ to determine $\lambda^*$. The overhead cost is about 1.5–2s for both the old and new methods. However, we observe a significant speedup in the computational cost for scanning $\lambda$. For the first two cases (synthetic spectrum and H-polymer), speeds ups in the range of 6-9x are obtained with the Bayesian formulation. The old program was not geared to handle nonzero terminal plateau ($G_0 \neq 0$), and hence a direct comparison cannot be made for the last case.

The efficiency arises from both the Bayesian, and the nonlinear formulation of Tikhonov regularization. The Bayesian framework allows us to limit the range of $\lambda$ explored, and the nonlinear formulation allows us to use a sparse distribution of $\lambda$s within this limited range. Let us explore both these advantages in more detail.

By default, pyReSpect scans $\lambda \in [10^{-10}, 10^{3}]$, starting from large values, with a density of 3 modes per decade. As seen in figure 3, $\log \pi(\lambda)$ initially increases as $\lambda$ decreases registering a maximum (between $\lambda \approx 10^0 - 10^1$, in this case), before decreasing further. In the program, once $\pi(\lambda)$ falls sufficiently below the maximum (8 orders of magnitude), we truncate the $\lambda$-scan. Smaller values of $\lambda$ are associated with negligible probabilities. Compared to the older method, this results in significant computational savings, since the entire $\eta^2$ versus $\rho^2$ curve does not have to be constructed. In figure 3 for example, the $\lambda$-scan is terminated after $\lambda = 7.8 \times 10^{-2}$, instead of going all the way to $\lambda_{\min} = 10^{-10}$. This implies that only a third of the potential $\approx 40$ discrete values of $\lambda \in [10^{-10}, 10^3]$ are explored. This directly results in savings of a factor of $\approx 3$, over the original program.

In figure 3, green circles indicate the values of $\lambda$ explored using the default setting in pyReSpect. The blue line depicts $\log \pi(\lambda)$ obtained with a higher resolution of $\lambda$. The $\lambda_m$ inferred from the (default) coarse spacing of $\lambda$ is remarkably consistent with that obtained using the finer resolution. Interestingly, and fortunately, this turns out to be a general observation that arises from the nonlinear formulation of the problem. In general, we observe that for $\lambda_1 \leq \lambda \leq \lambda_2$,

$$H_\lambda(\tau) \approx (1 - s_\lambda) H_{\lambda_1}(\tau) + s_\lambda H_{\lambda_2}(\tau), \tag{37}$$

where $s_\lambda = (\log \lambda - \log \lambda_1)(\log \lambda_2 - \log \lambda_1)$. Differentiating eqn 37, we find that the first derivative is independent of $\lambda$,

$$\frac{\partial H_\lambda}{\partial \log \lambda} \approx \frac{H_{\lambda_2} - H_{\lambda_1}}{\log \lambda_2 - \log \lambda_1}. \tag{38}$$

In other words, the second derivative $\partial^2 H_\lambda / \partial(\log \lambda)^2 \approx 0$. This is depicted in figure 7 by the solid purple lines using a centered difference formula to approximate the second derivative at $\lambda = \lambda_j$,

$$\frac{\partial^2 H_\lambda(\tau)}{\partial(\log \lambda)^2} \approx \frac{H(\tau, \lambda_{j+1}) - 2H(\tau, \lambda_j) + H(\tau, \lambda_{j-1})}{(\Delta \log \lambda)^2}, \tag{39}$$

with $\Delta \log \lambda = \log \lambda_{j+1} - \log \lambda_j$. Despite minor oscillations, the second derivative (fig. 7) is negligible. This implies that the spectrum $H_\lambda(\tau)$ is approximately linear in $\log \lambda$. Practically, this means that the density of $\lambda$s can be quite sparse. Often about 10 different values of $\lambda$ are sufficient to determine $\lambda_m$, compared to 30-100 values for BayesRelax [17]. Appendix 5 explores the origin of $\partial^2 H_\lambda / \partial(\log \lambda)^2 \approx 0$ using a toy model.

Thus, the combination of the Bayesian (clip the range of $\lambda$) and nonlinear (sparse distribution of $\lambda$) formulations greatly reduce the effort required to find the optimal $\lambda^*$. Each of these steps contributes a 2-3x cost saving, leading to a combined savings of 6-9x. Since this is the slowest step in the algorithm, it reduces the computational effort greatly.

## 4 Summary

We developed a computer program called pyReSpect to infer the continuous and discrete relaxation spectra from oscillatory shear experiments. An older version of this program used nonlinear Tikhonov regularization with a complicated heuristic to determine the optimal amount of regularization. In this work, we replaced it with a method based on Bayesian inference. The new and improved program provides uncertainty estimates for $H(\tau)$ by considering the distribution of $\pi(\lambda)$ that arises from Bayesian analysis.

The Bayesian formulation improves the performance by a factor of 2-3x by focusing on a limited window of $\lambda$ for which $\pi(\lambda)$ is non-negligible. The nonlinear formulation results in $\partial^2 H_\lambda / \partial(\log \lambda)^2 \approx 0$. This linearity allows us to consider a coarse mesh of $\lambda$, thereby improving the performance by an additional factor of 2-3x. For the examples considered, combined efficiency gains of order 6-9x are observed, consistent with these sources.

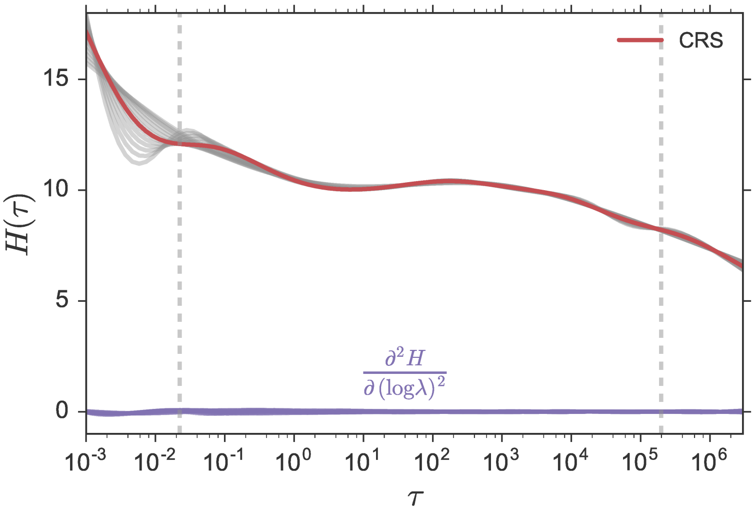

**Fig. 8** The CRS for the H-polymer is represented by the red line. $H_\lambda(\tau)$ determined at different values of $\lambda$ is overlaid in gray. These curves show some variation at small $\tau$. The gray vertical dashed lines mark the region in which the spectrum is reliable. The second derivative of the spectrum with respect to $\lambda$ is also shown by purple solid lines.

pyReSpect differs from the Hansen's Bayesian analysis in that it considers a nonlinear Tikhonov regularization problem via $H = e^h$ [17]. This makes the method more robust to wide spectra with modes of different heights, and requires a smaller mesh density of $\lambda$, since $\partial H_\lambda/\partial(\log\lambda) \approx$ constant. The program is also able to handle cases when $G_0 \neq 0$.

## 5 Appendix: Linearity of the CRS with regularization parameter

What property of the problem leads to the general observation $\partial^2 H_\lambda/\partial(\log\lambda)^2 \approx 0$? Since $H = \log h$, we intuitively understand that deviations in $H$ are mild compared to deviations in $h$. Nevertheless, we can show using a toy example that the linearity of $H(\log\lambda)$ stems from the form of the cost function.

For simplicity, and without loss of generality, suppose that $H$ is a variable rather than a function $H(\tau)$. Let, $V(\lambda, H) = \rho^2(H) + \lambda\eta^2(H)$ be a cost function that is linear in $\lambda$, but nonlinear in $H$. Furthermore, suppose

$$\begin{aligned} H^{+} &= \min_{H} V(\lambda, H) \\ H^{*} &= \min_{H} V(\lambda^{*}, H), \end{aligned}$$

where $\lambda^*$ is any particular $\lambda$, not necessarily the optimal $\lambda$. Let $R(H) = d\rho^2(H)/dH$ and $N(H) = d\eta^2(H)/dH$, be the first derivatives. Optimality conditions imply,

$$\begin{aligned} &R(H^{+}) + \lambda N(H^{+}) = 0 \\ &R(H^{*}) + \lambda^{*} N(H^{*}) = 0. \end{aligned}$$

Let $R^{+} = R(H^{+})$, $N^{+} = N(H^{+})$, $R^{*} = R(H^{*})$, and $N^{*} = N(H^{*})$. For slowly varying functions, a Taylor series expansion implies,

$$R^{+} = R^{*} + (H^{+} - H^{*}) \left.\frac{dR}{dH}\right|_{H^{*}} + \mathcal{O}(\Delta H^2) \tag{40}$$

$$N^{+} = N^{*} + (H^{+} - H^{*}) \left.\frac{dN}{dH}\right|_{H^{*}} + \mathcal{O}(\Delta H^2) \tag{41}$$

Ignoring quadratic and higher order terms, and setting $c_1 = dR(H^*)/dH$ and $c_2 = dN(H^*)/dH$,

$$\begin{aligned} R^{+} + \lambda N^{+} &= R^{*} + \lambda N^{*} + (H^{+} - H^{*})(c_1 + \lambda c_2) \\ 0 &= (\lambda - \lambda^{*}) N^{*} + (H^{+} - H^{*})(c_1 + \lambda c_2). \end{aligned}$$

If $c_2\lambda \gg c_1$, then $(H^{+} - H^{*})/(\lambda - \lambda^{*}) = \Delta H/\Delta\lambda \approx -N^{*}/(c_2\lambda)$. Thus,

$$\frac{\Delta H}{\Delta \log \lambda} \approx -\frac{N^{*}}{c_2} = \text{ constant.} \tag{42}$$

## Conflict of interest

The authors declare that they have no conflict of interest.

**Acknowledgements** This work is based in part upon work supported by the National Science Foundation under grant no. NSF DMR-1727870.

## References

1. Anderssen, R.S., Davies, A.R.: Simple moving-average formulae for the direct recovery of the relaxation spectrum. J. Rheol. **45(1)**, 1–27 (2001)
2. Ankiewicz, S., Orbey, N., Watanabe, H., Lentzakis, H., Dealy, J.: On the use of continuous relaxation spectra to characterize model polymers. J. Rheol. **60**(6), 1115–1120 (2016). DOI 10.1122/1.4960334
3. Bae, J.E., Cho, K.S.: Logarithmic method for continuous relaxation spectrum and comparison with previous methods. J. Rheol. **59**, 1081 (2015)
4. Baumgaertel, M., Derosa, M.E., Machado, J., Masse, M., Winter, H.H.: The relaxation-time spectrum of nearly monodisperse polybutadiene melts. Rheol. Acta **31**, 75–82 (1992)
5. Baumgaertel, M., Winter, H.: Interrelation between continuous and discrete relaxation time spectra. J.Non-Newt. Fluid Mech. **44**(0), 15 – 36 (1992). DOI 10.1016/0377-0257(92) 80043-W
6. Baumgaertel, M., Winter, H.H.: Determination of discrete relaxation and retardation time spectra from dynamic mechanical data. Rheol. Acta **28**(6), 511–519 (1989). DOI 10.1007/ BF01332922
7. Brabec, C.J., Rogl, H., Schausberger, A.: Investigation of relaxation properties of polymer melts by comparison of relaxation time spectra calculated with different algorithms. Rheol. Acta **36**, 667–676 (1997)
8. Branch, M., Coleman, T., Li, Y.: A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. SIAM J. Sci. Comput. **21**(1), 1–23 (1999). DOI 10.1137/S1064827595289108

9. Cho, K.S.: A simple method for determination of discrete relaxation time spectrum. Macromol. Res. **18**(4), 363–371 (2010). DOI 10.1007/s13233-010-0413-4

10. Cho, K.S., Park, G.W.: Fixed-point iteration for relaxation spectrum from dynamic mechanical data. J. Rheol. **57**(2), 647–678 (2013). DOI 10.1122/1.4789786

11. Davies, A.R., Anderssen, R.S.: Sampling localization in determining the relaxation spectrum. J. Non-Newton. Fluid Mech. **73**(1-2), 163–179 (1997). DOI 10.1016/S0377-0257(97)00056-6

12. Ferry, J.: Viscoelastic Properties of Polymers, $3^{rd}$ edn. John Wiley & Sons Inc, Hoboken, NY (1980)

13. Goyal, S., Larson, R.G., Aloisio, C.J.: Quantitative prediction of impact forces in elastomers. J. Engg. Mat. Tech. **121**(3), 294–304 (1999). DOI 10.1115/1.2812378

14. Hansen, P.: Analysis of discrete ill-posed problems by means of the L-curve. SIAM Rev. **34**(4), 561–580 (1992)

15. Hansen, P.C., O'Leary, D.P.: The use of the Lhul-curve in the regularization of discrete ill-posed problems. SIAM J. Sci. Comp. **14**(6), 1487–1503 (1993). DOI 10.1137/0914086

16. Hansen, S.: Bayesian estimation of hyperparameters for indirect Fourier transformation in small-angle scattering. J. Appl. Cryst. **33**(6), 1415–1421 (2000). DOI 10.1107/S0021889800012930

17. Hansen, S.: Estimation of the relaxation spectrum from dynamic experiments using Bayesian analysis and a new regularization constraint. Rheol. Acta **47**, 169–178 (2008). DOI 10.1007/s00397-007-0225-4

18. Honerkamp, J., Weese, J.: Determination of the relaxation spectrum by a regularization method. Macromolecules **22**(11), 4372–4377 (1989). DOI 10.1021/ma00201a036

19. Honerkamp, J., Weese, J.: A nonlinear regularization method for the calculation of relaxation spectra. Rheol. Acta **32**(1), 65–73 (1993). DOI 10.1007/BF00396678

20. Hulsen, M.A., van der Zanden, J.: Numerical simulation of contraction flows using a multimode Giesekus model. J. Non-Newton. Fluid Mech. **38**(2), 183 – 221 (1991). DOI https://doi.org/10.1016/0377-0257(91)83005-O

21. Ianniruberto, G., Marrucci, G.: A multi-mode CCR model for entangled polymers with chain stretch. J. Non-Newton. Fluid Mech. **102**(2), 383 – 395 (2002). DOI https://doi.org/10.1016/S0377-0257(01)00188-4

22. Inkson, N.J., McLeish, T.C.B., Harlen, O.G., Groves, D.J.: Predicting low density polyethylene melt rheology in elongational and shear flows with "pom-pom" constitutive equations. J. Rheol. **43**(4), 873–896 (1999). DOI 10.1122/1.551036

23. Jensen, E.A.: Determination of discrete relaxation spectra using simulated annealing. J. Non-Newton. Fluid Mech. **107(1-3)**, 1–11 (2002)

24. Johnston, P., Gulrajani, R.: Selecting the corner in the L-curve approach to Tikhonov regularization. IEEE Trans. Biomed. Engg. **47**(9), 1293–1296 (2000)

25. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001). URL `http://www.scipy.org/`. [Online; accessed ¡today¿]

26. Kaschta, J., Stadler, F.: Avoiding waviness of relaxation spectra. Rheol. Acta **48**(6), 709–713 (2009). DOI 10.1007/s00397-009-0370-z

27. Larson, R.G., Goyal, S., Aloisio, C.: A predictive model for impact response of viscoelastic polymers in drop tests. Rheol. Acta **35**(3), 252–264 (1996). DOI 10.1007/BF00366912

28. Lentzakis, H.: Linear and non-linear rheological architecture of architecturally complex polymers. Ph.D. thesis, University of Crete, Greece (2014)

29. Macdonald, J.R.: On relaxation-spectrum estimation for decades of data: Accuracy and sampling-localization considerations. Inverse Probl. **16**, 1561–1583 (2000)

30. McDougall, I., Orbey, N., Dealy, J.M.: Inferring meaningful relaxation spectra from experimental data. J. Rheol. **58**, 779 (2014)

31. Mead, D.W.: Determination of molecular-weight distributions of linear flexible polymers from linear viscoelastic material functions. J. Rheol. **38**, 1797–1827 (1994)

32. Orbey, N., Dealy, J.M.: Determination of the relaxation spectrum from oscillatory shear data. J. Rheol. **35(6)**, 1035–1049 (1991)

33. Provencher, S.W.: An eigenfunction expansion method for the analysis of exponential decay curves. J. Chem. Phys. **64**(7), 2772–2777 (1976). DOI 10.1063/1.432601

34. Provencher, S.W.: CONTIN: A general purpose constrained regularization program for inverting noisy linear algebraic and integral equations. Comp. Phys. Comm. **27**(3), 229 – 242 (1982). DOI 10.1016/0010-4655(82)90174-6

35. Ronca, G.: Frequency spectrum and dynamic correlations of concentrated polymer liquids. J. Chem. Phys. **79**(2), 1031–1043 (1983)

36. Roovers, J.: Melt rheology of h-shaped polystyrenes. Macromolecules **17**(6), 1196–1200 (1984). DOI 10.1021/ma00136a016

37. Roovers, J., Toporowski, P.M.: Preparation and characterization of H-shaped polystyrene. Macromolecules **14**(5), 1174–1178 (1981). DOI 10.1021/ma50006a007

38. Roths, T., Marth, M., Weese, J., Honerkamp, J.: A generalized regularization method for nonlinear ill-posed problems enhanced for nonlinear regularization terms. Comp. Phys. Comm. **139**(3), 279 – 296 (2001). DOI 10.1016/S0010-4655(01)00217-X

39. Shanbhag, S.: pyReSpect: A computer program to extract discrete and continuous spectra from stress relaxation experiments. Macromol. Theory Simul. **28**(3), 1900005 (2019). DOI 10.1002/mats.201900005

40. Shanbhag, S., Park, S.J., Wang, Z.: Superensembles of linear viscoelastic models of polymer melts. J. Rheol. **56**(2), 279–303 (2012). DOI 10.1122/1.3679469

41. Stadler, F., Bailly, C.: A new method for the calculation of continuous relaxation spectra from dynamic-mechanical data. Rheol. Acta **48**(1), 33–49 (2009). DOI 10.1007/s00397-008-0303-2

42. Takeh, A., Shanbhag, S.: A computer program to extract the continuous and discrete relaxation spectra from dynamic viscoelastic measurements. Appl. Rheol. **23**(2), 24628 (2013)

43. Weese, J.: A reliable and fast method for the solution of Fredholm integral equations of the first kind based on Tikhonov regularization. Comp. Phys. Comm. **69**(1), 99 – 111 (1992). DOI 10.1016/0010-4655(92)90132-I

44. Weese, J.: A regularization method for nonlinear ill-posed problems. Comp. Phys. Comm. **77**(3), 429 – 440 (1993). DOI DOI:10.1016/0010-4655(93)90187-H

45. Winter, H.H.: Analysis of dynamic mechanical data: Inversion into a relaxation time spectrum and consistency check. J. Non-Newton. Fluid Mech. **68**, 225–239 (1997)