Mech Time-Depend Mater (2013) 17:53–82
DOI 10.1007/s11043-012-9176-y

Interconversion of linearly viscoelastic material functions expressed as Prony series: a closure

Jacques Luk-Cyr · Thibaut Crochon · Chun Li · Martin Lévesque

Received: 3 December 2011 / Accepted: 14 June 2012 / Published online: 6 July 2012

Abstract Interconversion of viscoelastic material functions is a longstanding problem that has received attention since the 1950s. There is currently no accepted methodology for interconverting viscoelastic material functions due to the lack of stability and accuracy of the existing methods. This paper presents a new exact, analytical interconversion method for linearly viscoelastic material functions expressed as Prony series. The new algorithm relies on the equations of the thermodynamics of irreversible processes used for defining linearly viscoelastic constitutive theories. As a result, interconversion is made possible for unidimensional and tridimensional materials for arbitrary material symmetry. The algorithm has been tested over a broad range of cases and was found to deliver accurate interconversion in all cases. Based on its accuracy and stability, the authors believe that their algorithm provides a closure to the interconversion of linearly viscoelastic constitutive theories expressed with Prony series.

Keywords Viscoelasticity · Interconversion · Creep · Relaxation

1 Introduction

Interconversion of linearly viscoelastic material properties consists of determining the creep compliance $\mathbf{S}(t)$ from a given relaxation modulus $\mathbf{C}(t)$, or vice-versa. For practical rea-

J. Luk-Cyr · T. Crochon · M. Lévesque (✉)
Laboratory for MultiScale Mechanics (LM2), CREPEC, Département de Génie Mécanique,
École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville H3C 3A7, Canada
e-mail: martin.levesque@polymtl.ca

J. Luk-Cyr
e-mail: jacques.luk-cyr@polymtl.ca

T. Crochon
e-mail: thibaut.crochon@polymtl.ca

C. Li
National Research Council, Aerospace, 1200 Montréal Road, Building M14, Ottawa K1A 0R6, Canada
e-mail: chun.li@nrc-cnrc.gc.ca

sons, tridimensional viscoelastic properties are often obtained from creep-recovery tests (Lévesque et al. 2008). However, finite element packages usually require knowing $\mathbf{C}(t)$ for numerical implementation (Crochon et al. 2010). An accurate and computationally efficient interconversion method is therefore of practical importance. The topic has been examined by many authors since the 1950s (Hopkins and Hamming 1957).

Most of the approaches found in the literature aim to determine either $\mathbf{C}(t)$ or $\mathbf{S}(t)$ from a given $\mathbf{S}(t)$ or $\mathbf{C}(t)$ through the following relationship:

$$\int_0^t \mathbf{C}(t-\tau)\cdot\mathbf{S}(\tau)\,\mathrm{d}\tau = t\mathbf{I}, \tag{1}$$

where $t\mathbf{I}$ is the t-multiplied identity. Equation (1) is a Volterra equation of the first kind and is generally ill-posed. As a result, its numerical integration can be convergent, though not necessarily towards the solution (Sorvari and Malinen 2006). A few methodologies have been proposed to circumvent this ill-posedness, and they can be grouped into two main strategies. The first (i) strategy involves the direct numerical integration of experimental data associated with either $\mathbf{C}(t)$ or $\mathbf{S}(t)$. This approach eludes assuming a mathematical representation for the viscoelastic functions. The ill-posedness of the problem is bypassed by using regularization methods. The second (ii) strategy consists of assuming possible shapes for $\mathbf{C}(t)$ and $\mathbf{S}(t)$ and compute their parameters so that Eq. (1) is verified.

In the sequel, the known function is referred to as *source*, while the unknown function (i.e. that for which the interconversion is sought) is referred to as *target*. The objective of approach (i) is to obtain a system of equations whose solution provides the expression of the target function so that Eq. (1) is met. Since the target function is unknown, one of the key issues with that approach is to devise an efficient numerical integration scheme using the experimental data for Eq. (1). Hopkins and Hamming (1957) discretized Eq. (1) into subintervals. In each subinterval, the target function was considered as constant. This led to a recursive relation from which they computed discrete values of the target function. Knoff and Hopkins (1972) subsequently improved this method by considering the source and target functions as piecewise linear for each subinterval. Tschoegl (1989) developed an approach, similar to that of Hopkins and Hamming (1957), where the value of the target function was evaluated at the midpoint of the time interval. Then, applying the trapezoidal rule to the convolution integral, he obtained a similar recursive relation. These recursive relations can each be expressed as a simple system of linear algebraic equations. However, they are ill-posed, and small variations in the experimental data, such as noise or discrepancies between the applied and assumed loading, can lead to significant errors in the numerical interconversion (Sorvari and Malinen 2006). To overcome this difficulty, the problem must be regularized. Examples of such regularizations can be found in Sorvari and Malinen (2006), who used an L-curve-based regularization method on the system of linear algebraic equations. Nikonov et al. (2005) proposed measuring neither $\mathbf{C}(t)$ nor $\mathbf{S}(t)$ but an intermediate response (e.g. force or deformation) from which they recovered both $\mathbf{C}(t)$ and $\mathbf{S}(t)$ through Volterra's integral equations of the second kind, which is a well-posed problem.[1] More recently, Yaoting et al. (2011) transformed Volterra's integral equation of the first kind to that of the second kind and applied Tikhonov's L-curve-based regularization for obtaining a well-posed system of algebraic equations from which they deduced the target function.

[1] Volterra's integral equation of the second kind can be obtained, for example, by differentiating Eq. (1) with respect to t.

This mathematics-based approach yields a set of numerical data for the target function from a set of experimental data for the source function. Even though some authors consider this approach being the most fundamental since no assumptions are made on the form of the source and target functions (Sorvari and Malinen 2006), regularization involves fitting experimental data. This is an entirely mathematical approach and may yield a solution that is inconsistent with underlying physical principles. In addition, if stress calculation is the final objective, some kind of model must be fitted for the target and source functions. These models will obviously introduce hypotheses and will fit the data with a certain degree of accuracy. If a thermodynamically acceptable linearly viscoelastic model is unable to fit the experimental data associated with the source function a priori, it is very likely that the same problem will be met with the target function. The authors believe that under these circumstances, more effort should be spent on a nonlinearly viscoelastic model that fits the data adequately. Then, a numerical algorithm for predicting the target response, like the Newton–Raphson method usually used when implementing Schapery's creep compliances in finite element packages (Crochon et al. 2010), could be used. Finally, this mathematical approach has only been applied for the interconversion of unidimensional linearly viscoelastic properties.

In approach (ii), $\mathbf{C}(t)$ and $\mathbf{S}(t)$ are given thermodynamically admissible shapes. When $\mathbf{C}(t)$ and $\mathbf{S}(t)$ are represented as Prony series, Eq. (1) becomes a well-posed problem and can be solved for the unknown target function coefficients. Let $C(t)$ be an uniaxial relaxation modulus, and $S(t)$ an uniaxial creep relaxation modulus. Baumgaertel and Winter (1989) developed exact analytical expressions relating $C(t)$ and $S(t)$ in the Laplace domain. Let N be the number of time constants in the Prony series used to define the source function. Their approach involves finding the N roots of a polynomial function that leads to the N time constants of the target function. Let us define as *intensities* the terms multiplying the exponential functions in the Prony series. Baumgaertel and Winter (1989) derived analytical expressions for the intensities. Similarly, Park and Schapery (1999) devised a series of approximate methods where the time constants are chosen a priori, e.g. evenly distributed on a logarithmic scale, and where the intensities are computed by solving a linear system of equations. Let $f^*(p)$ be the Laplace–Carson transform of $f(t)$.[2] Their linear system of equations can be obtained by collocating $C^*(p)$ to $(S^*)^{-1}(p)$ at carefully chosen discrete values of the Laplace–Carson variable p. Alternatively, a least square criterion in the Laplace domain can also be used to define a different set of linear equations. Therefore, depending on the choices made for defining the system of equations, different solutions to the same problem can be obtained. Finally, they also developed an approach where the time constants are computed from the roots of a polynomial similar to that of Baumgaertel and Winter (1989) and the intensities are computed by solving a linear system of equations. In that specific case, the solutions of Baumgaertel and Winter (1989) and Park and Schapery (1999) should be identical.

The principal drawback of the Baumgaertel and Winter (1989) and the Park and Schapery (1999) approaches is that numerical instabilities leading to inaccurate results can be encountered when the source function has numerous time constants spread over many decades. For example, Fernandez et al. (2011) have tested the Park and Schapery (1999) algorithm for the interconversion of unidimensional viscoelastic properties of polymethyl methacrylate (PMMA). They experimentally measured the uniaxial creep compliance and relaxation moduli, which they both interconverted using the Park and Schapery (1999) algorithm. The

[2]The Laplace–Carson transform of $f(t)$ is defined as $f^*(p) = p\int_0^\infty f(t)\exp[-pt]\,\mathrm{d}t$.

interconverted functions were compared with the experimental results, and they obtained relative errors of 8.5 % for the interconversion of creep to relaxation and 21 % for the interconversion of relaxation to creep. In addition, these approaches do not impose any restrictions on the intensities. As a result, negative terms can be obtained. This violates the restrictions imposed by thermodynamics (for example, when considering axial modulus) and prevents using the results in a finite element code. Finally, as experienced in this paper, numerical singularities (i.e. division by numerical zeros) can be encountered, and the approach can fail to provide any results at all.

One way to circumvent such difficulties would be to define the interconversion as a constrained optimization problem. Consider for example the case where the interconversion is performed in the Laplace–Carson domain by virtue of the correspondence principle. Consider further the case where the source function is $C(t)$ expressed as a Prony series and that the unknown target function is $S(t)$, also expressed as a Prony series. Then, one optimization problem could be to determine the unknown set of parameters $\mathcal{P}$, defining $S(t)$, a solution of the following problem:

$$\min_{\mathcal{P}} \int_0^\infty \left[\left(C^*\right)^{-1}(p) - S^*(p) \right]^2 \mathrm{d}p \tag{2}$$

The optimization problem thus defined is in fact a least square minimization in the Laplace–Carson domain. In a different context, Cost and Becker (1970) introduced such ideas for the numerical inversion of Laplace transforms. The problem with such an approach is that no restrictions are imposed on $\mathcal{P}$. Let $\mathcal{Q}$ be the set of parameters defining $S(t)$ that are also thermodynamically admissible. Lévesque (2004) and Lévesque et al. (2007) proposed solving the following problem in the case of homogenization problems

$$\min_{\mathcal{Q}\in\mathcal{P}} \int_0^\infty \left[\tilde{S}^*(p) - S^*(p) \right]^2 \mathrm{d}p, \tag{3}$$

where $\tilde{S}^*(p)$ is the function to be inverted and only known in the Laplace–Carson domain.[3] It should be noted that Lévesque (2004) and Lévesque et al. (2007) provided algorithms for both unidimensional and tridimensional cases. By definition, the solution of such a problem will always lead to thermodynamically admissible interconversions, and the problems associated with the numerical singularities described above are avoided. It should be further noted that constrained optimization can also be used to obtain the set of parameters $\mathcal{Q}$ (or those defining a relaxation modulus) from experimental data (see among others Lévesque 2004; Lévesque et al. 2007; Knauss and Zhao 2007). Although these approaches elude the problems associated with numerical instabilities and singularities, they inherit the difficulties of optimization problems, like the possibility of reaching a local minimum. As a result, and especially in the tridimensional case (Lévesque et al. 2007), the accuracy of such methods can be limited, and it is very unlikely that they will provide the exact inverse expression. To the knowledge of the authors, no authors formulated the specific problem of interconverting viscoelastic properties as a constrained optimization problem.

Most of the interconversion algorithms found in the literature were developed for unidimensional creep or relaxation expressions. Such algorithms could be applied to each of the individual components of a creep compliance or relaxation modulus tensor, or alternatively, to their tensorial projections in the case of isotropy, cubic symmetry or transverse

[3] If expression (3) was used for the interconversion of viscoelastic properties, then $\tilde{S}^*(p)$ would be set to $(C^*)^{-1}(p)$.

isotropy. Interconverting each of the individual components of the viscoelastic properties is very likely to lead to interconverted materials that do not meet the thermodynamics restrictions. Except for Lévesque et al. (2007) in the context of homogenization problems and to the knowledge of the authors, no authors ever studied the interconversion of viscoelastic properties expressed as their tridimensional tensorial expression.

The purpose of this work is to develop and rigorously test a new algorithm that eludes any numerical instabilities and provides exact solutions for the interconversion of $\mathbf{C}(t)$ and $\mathbf{S}(t)$ when the linearly viscoelastic functions are expressed as Prony series. As was shown by many authors (Biot 1954; Bouleau 1992, 1999; Lévesque et al. 2008), Prony series and their extension to spectra are thermodynamically justified and can converge to other expressions such as power laws, power exponential, etc. In addition, Prony series can lead to computationally efficient finite element implementations (Crochon et al. 2010). In the context that linearly viscoelastic properties are to be used for computations, the authors believe that approach (ii), involving Prony series, shall cover all the applications in linear viscoelasticity. Furthermore, and more importantly, the algorithm is developed for fully anisotropic materials, rather than for unidimensional functions.

The paper is organized as follows: Sect. 2 recalls background information upon which the new algorithm is based and also the algorithm of Park and Schapery (1999), used for comparison purposes; Sect. 3 introduces the new interconversion method developed from the thermodynamics of irreversible processes. Section 4 gives theoretical examples of application of the proposed algorithm. Finally, Sect. 5 presents an error estimation of the interconversion algorithm.

In this paper, the summation of repeated indices is adopted, unless specified otherwise. Stresses or strains and stiffnesses are represented as vectors and matrices, according to the modified Voigt notation recalled in Lévesque et al. (2008). A vector is denoted by a boldfaced lower case letter, (i.e. $\boldsymbol{a}$ and $\boldsymbol{\alpha}$), and a matrix is denoted by a boldfaced capital Roman letter (i.e. $\mathbf{A}$). The simply contracted product is represented by '$\cdot$' (e.g. $\mathbf{A}\cdot\mathbf{B} = A_{ij}B_{jk} = C_{ik}$; $\mathbf{A}\cdot\boldsymbol{b} = A_{ij}b_j = c_i$), and the dyadic product is represented by '$\otimes$' (e.g. $\boldsymbol{a}\otimes\boldsymbol{b} = a_i b_j = C_{ij}$).

2 Background

2.1 Linearly viscoelastic constitutive theories through the thermodynamics of irreversible processes

Biot (1954) has shown that the thermodynamics of irreversible processes can be used for obtaining the most general thermodynamically admissible mechanical responses of linearly viscoelastic materials. Recently, Lévesque et al. (2008), among others, have shown that after some manipulations and for applied strains, the thermodynamics of irreversible processes leads to the following set of equations:

$$\boldsymbol{\sigma}(t) = \mathbf{L}^{(1)}\cdot\boldsymbol{\varepsilon}(t) + \mathbf{L}^{(2)}\cdot\boldsymbol{\chi}(t) \tag{4a}$$

$$\mathbf{B}\cdot\dot{\boldsymbol{\chi}}(t) + \mathbf{L}^{(3)}\cdot\boldsymbol{\chi}(t) + \left(\mathbf{L}^{(2)}\right)^{\mathrm{T}}\cdot\boldsymbol{\varepsilon}(t) = \mathbf{0} \tag{4b}$$

where $\mathbf{B}$, $\mathbf{L}^{(1)}$, $\mathbf{L}^{(2)}$ and $\mathbf{L}^{(3)}$ are the so-called internal matrices, and $\dot{f}$ represents time differentiation of f. The $\mathbf{L}^{(i)}$ are sub-matrices of the matrix

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}^{(1)} & \mathbf{L}^{(2)} \\ (\mathbf{L}^{(2)})^{\mathrm{T}} & \mathbf{L}^{(3)} \end{bmatrix} \tag{5}$$

that must be positive definite (denoted herein by $\mathbf{L} > 0$) for thermodynamics reasons. As a consequence, $\mathbf{L}^{(1)} > 0$ and $\mathbf{L}^{(3)} > 0$. Likewise, $\mathbf{B}$ must be positive semi-definite (denoted herein as $\mathbf{B} \geq 0$). Furthermore, matrices $\mathbf{B}$, $\mathbf{L}^{(1)}$ and $\mathbf{L}^{(3)}$ are symmetric matrices by construction (Lévesque et al. 2008). $\boldsymbol{\chi}$ are called hidden variables and represent internal phenomena (e.g. polymer chain movement).

It should be noted that, $\mathbf{L}^{(1)}$ is a 6×6 square matrix in the tridimensional case. Furthermore, $\mathbf{L}^{(3)}$ is an $R \times R$ square matrix where R is the number of hidden variables. Finally, $\mathbf{L}^{(2)}$ is a $6 \times R$ matrix.

Since $\mathbf{B}$ and $\mathbf{L}^{(3)}$ are symmetric, $\mathbf{L}^{(3)} > 0$ and $\mathbf{B} \geq 0$, it is always possible to find a basis where both matrices are diagonal. Without loss of generality, it can be assumed that $\boldsymbol{\chi}$ is already expressed in this basis. Then, the set of R coupled equations (4b) becomes a set of R uncoupled first-order differential equations that can easily be solved, leading to

$$\chi_r(t) = -\frac{L_{ir}^{(2)}}{L_{rr}^{(3)}} \int_0^t \left(1 - \exp\left[-\frac{L_{rr}^{(3)}}{B_{rr}}(t-\tau)\right]\right) \frac{\mathrm{d}\varepsilon_i}{\mathrm{d}\tau}\,\mathrm{d}\tau \quad \text{(no sum on } r\text{)} \tag{6}$$

where $1 \leq i \leq 6$ and $1 \leq r \leq R$ in the tridimensional case. Combining Eqs. (4a), (4b) and (6) leads to

$$\sigma_i(t) = \left(L_{ij}^{(1)} - \frac{L_{ir}^{(2)} L_{jr}^{(2)}}{L_{rr}^{(3)}}\right)\varepsilon_j(t) + \frac{L_{ir}^{(2)} L_{jr}^{(2)}}{L_{rr}^{(3)}} \int_0^t \exp\left[-\frac{L_{rr}^{(3)}}{B_{rr}}(t-\tau)\right] \frac{\mathrm{d}\varepsilon_j}{\mathrm{d}\tau}\,\mathrm{d}\tau \tag{7}$$

which can finally be reduced to the familiar expression:

$$\begin{aligned} \boldsymbol{\sigma}(t) &= \mathbf{C}^{(0)} \cdot \boldsymbol{\varepsilon}(t) + \int_0^t \sum_{n=1}^{N} \mathbf{C}^{(n)} \exp\left[-\rho_n(t-\tau)\right] \cdot \frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}\tau}\,\mathrm{d}\tau \\ &= \int_0^t \mathbf{C}(t-\tau) \cdot \frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}\tau}\,\mathrm{d}\tau \end{aligned} \tag{8}$$

where $\mathbf{C}^{(0)}$ is the equilibrium relaxation tensor, $\mathbf{C}^{(n)}$ are the transient relaxation tensors associated with the so-called inverted relaxation times ρ_n, and $\mathbf{C}(t)$ is given by

$$\mathbf{C}(t) = \mathbf{C}^{(0)} + \sum_{n=1}^{N} \mathbf{C}^{(n)} \exp(-t\rho_n) \tag{9}$$

By construction, $\mathbf{C}^{(0)} > 0$, $\mathbf{C}^{(n)} > 0$ and $\rho_n > 0$. It should be noted that Eq. (8) has been developed for the most generalized tridimensional case and is not restricted to any degree of material symmetry.

The same development can be accomplished for applied stresses (Schapery 1969) and leads to

$$\boldsymbol{\varepsilon}(t) = \mathbf{A}^{(1)} \cdot \boldsymbol{\sigma}(t) - \mathbf{A}^{(2)} \cdot \boldsymbol{\xi}(t) \tag{10a}$$

$$\mathbf{B} \cdot \dot{\boldsymbol{\xi}}(t) + \mathbf{A}^{(3)} \cdot \boldsymbol{\xi}(t) + \left(\mathbf{A}^{(2)}\right)^{\mathrm{T}} \cdot \boldsymbol{\sigma}(t) = \mathbf{0} \tag{10b}$$

where $\boldsymbol{\xi}$ are hidden variables, $\mathbf{B}$, $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$ are the new internal matrices of the material (Lévesque et al. 2008). Matrices $\mathbf{B}$, $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(3)}$ are symmetric and $\mathbf{A}^{(1)} > 0$, $\mathbf{A}^{(3)} > 0$

and $\mathbf{B} \geq 0$. After the same mathematical manipulations as for the relaxation behavior, the creep behavior is expressed as

$$\varepsilon_i(t) = A_{ij}^{(1)}\sigma_j(t) + \frac{A_{ir}^{(2)}A_{jr}^{(2)}}{A_{rr}^{(3)}}\int_0^t \left(1 - \exp\left[-\frac{A_{rr}^{(3)}}{B_{rr}}(t-\tau)\right]\right)\frac{\mathrm{d}\sigma_j}{\mathrm{d}\tau}\,\mathrm{d}\tau \tag{11}$$

leading to the familiar expression

$$\begin{aligned}\boldsymbol{\varepsilon}(t) &= \mathbf{S}^{(0)}\cdot\boldsymbol{\sigma}(t) + \int_0^t \sum_{m=1}^{M}\mathbf{S}^{(m)}\left(1-\exp\left[-\lambda_m(t-\tau)\right]\right)\cdot\frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}\tau}\,\mathrm{d}\tau \\ &= \int_0^t \mathbf{S}(t-\tau)\cdot\frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}\tau}\,\mathrm{d}\tau\end{aligned} \tag{12}$$

where $\mathbf{S}^{(0)}$ is the instantaneous creep compliance tensor, $\mathbf{S}^{(m)}$ are the transient compliance tensors associated to the inverted retardation times λ_m, and $\mathbf{S}(t)$ is given by

$$\mathbf{S}(t) = \mathbf{S}^{(0)} + \sum_{m=1}^{M}\mathbf{S}^{(m)}\left[1-\exp(-t\lambda_m)\right] \tag{13}$$

$\mathbf{S}^{(0)} > 0$, $\mathbf{S}^{(m)} > 0$ and $\lambda_m > 0$ by construction.

When testing a material, it is more convenient to determine the relaxation tensors and times (or compliance tensors and retardation times) than the internal matrices since they measure the effect of undetermined quantities on material behavior. Crochon et al. (2010) recently proposed an algorithm to compute a possible set of internal matrices $\mathbf{B}$ and $\mathbf{L}^{(i)}$ (or $\mathbf{A}^{(i)}$) from $\mathbf{C}(t)$ (or $\mathbf{S}(t)$). Using Laplace transforms of Eqs. (4a), (4b) and (8), the following relationships between the internal and relaxation matrices[4] are obtained:

$$\mathbf{L}^{(1)} = \mathbf{C}^{(0)} + \sum_{n=1}^{N}\mathbf{C}^{(n)} \tag{14a}$$

$$\mathbf{L}^{(2)}\cdot\left(p\mathbf{B}+\mathbf{L}^{(3)}\right)^{-1}\cdot\left(\mathbf{L}^{(2)}\right)^{\mathrm{T}} = \sum_{n=1}^{N}\frac{\rho_n}{p+\rho_n}\mathbf{C}^{(n)} \tag{14b}$$

where p is the Laplace coordinate. Matrix $\mathbf{L}^{(1)}$ can be directly evaluated from Eq. (14a). Equation (14b), on the contrary, is ill-defined, and an infinity of solutions exist. A possible solution consists of choosing $\mathbf{B}$ as the identity matrix and $\mathbf{L}^{(3)}$ as the block diagonal matrix:[5]

$$\mathbf{L}^{(3)} = \bigoplus_{n=1}^{N}[\rho_n\mathbf{I}] \tag{15}$$

where each block $[\rho_n\mathbf{I}]$ is a 6×6 diagonal matrix with ρ_n as the non-zero elements for the tridimensional case. $\mathbf{L}^{(2)}$ can be written as a block matrix

$$\mathbf{L}^{(2)} = \left(\mathbf{T}_1|\ldots|\mathbf{T}_n|\ldots|\mathbf{T}_N\right) \tag{16}$$

[4] A similar relationship between internal and creep matrices can be found in Crochon et al. (2010).

[5] The direct sum (symbol $\oplus$) of two matrices A and B is defined as $\mathrm{A}\oplus\mathrm{B} = \left[\begin{smallmatrix} A & 0 \\ 0 & B \end{smallmatrix}\right]$.

and

$$\mathbf{T}_n = \mathcal{C}_{\mathcal{L}}\left(\rho_n \mathbf{C}^{(n)}\right) \tag{17}$$

where $\mathcal{C}_L$ computes the lower triangular matrix obtained from the Cholesky decomposition of $\rho_n \mathbf{C}^{(n)}$.

2.2 Park and Schapery (1999) interconversion method

For the unidimensional case, the relaxation modulus and the creep compliance can be expressed similarly to Eqs. (9) and (13):

$$C(t) = C_0 + \sum_{n=1}^{N} C_n \exp(-t/\tau_n) \tag{18a}$$

and

$$S(t) = S_0 + \sum_{m=1}^{M} S_m \left[1 - \exp(-t/\omega_m)\right] \tag{18b}$$

where C_0, C_n, S_0 and S_m are scalars, and τ_n and ω_m are the relaxation and retardation times, respectively. It is worth mentioning that for unidimensional cases, if $C(t)$ has N distinct relaxation times, $S(t)$ will also have $M = N$ retardation times such as (Tschoegl 1989)

$$\tau_1 < \omega_1 < \cdots < \tau_n < \omega_n < \cdots < \tau_N < \omega_N \tag{19}$$

Considering the interconversion from $C(t)$ to $S(t)$, Park and Schapery (1999) have derived from Eq. (1) a system of linear, algebraic equations of the form

$$G_{km} S_m = b_k \tag{20}$$

where

$$G_{km} = \begin{cases} C_0(1 - e^{-(t_k/\omega_m)}) + \sum_{n=1}^{N} \frac{\tau_n C_n}{\tau_n - \omega_m}(e^{-(t_k/\tau_n)} - e^{-(t_k/\omega_m)}) & \text{if} \quad \tau_n \neq \omega_m \\ \text{or} & \\ C_0(1 - e^{-(t_k/\omega_m)}) + \sum_{n=1}^{N} \frac{t_k C_n}{\omega_m}(e^{-(t_k/\tau_n)}) & \text{if} \quad \tau_n = \omega_m \end{cases} \tag{21a}$$

and

$$b_k = 1 - \left(C_0 + \sum_{n=1}^{N} C_n e^{-(t_k/\tau_n)}\right) \Big/ \left(C_0 + \sum_{n=1}^{N} C_n\right) \tag{21b}$$

where t_k $(k = 1, \ldots, M)$ denotes discrete times. Finally, S_0 is computed using the following relationship:

$$S_0 = 1 \Big/ \left(C_0 + \sum_{n=1}^{N} C_n\right) \tag{22}$$

In Eqs. (21a), (21b), C_0, C_n and τ_n are known, and ω_m are calculated using the following relationship derived from the Laplace–Carson transform of Eq. (1):

$$\lim_{p \to -(1/\omega_m)} C_0 + \sum_{n=1}^{N} \frac{p\tau_n C_n}{p\tau_n + 1} = 0 \tag{23}$$

Equation (23) is a nonlinear equation with M solutions and can be solved using a root-finding algorithm. In this paper, we have used the Golden Ratio Search method and a convergence criterion ψ, representing the length of the subinterval, as $\psi \leq 10^{-8}\tau_n$.

The sampling points t_k were set to $t_k = \omega_k$, as suggested by Park and Schapery (1999). Different sets of t_k will therefore lead to different sets of equations. As a result, the numerical solution will be sensitive to their choice.

2.3 Tensor decomposition

Typical interconversion algorithms, such as that of Park and Schapery (1999), were developed for unidimensional expressions. They can however be extended to tridimensional expressions with tensorial projections, when applicable.

Isotropic, cubic and transversely isotropic tensors can be expressed as per the following decompositions:

$$\mathbf{W}^{\text{iso}} = \alpha\mathbf{J} + \beta\mathbf{K} \tag{24a}$$

$$\mathbf{W}^{\text{cub}} = \alpha\mathbf{J} + \beta\mathbf{K}_a + \gamma\mathbf{K}_b \tag{24b}$$

$$\mathbf{W}^{\text{isT}} = \alpha\mathbf{E}_L + \beta\mathbf{J}_T + \gamma\mathbf{F} + \gamma'\mathbf{F}^{\text{T}} + \delta\mathbf{K}_T + \delta'\mathbf{K}_L \tag{24c}$$

where $\mathbf{J}$, $\mathbf{K}$, $\mathbf{K}_a$, $\mathbf{K}_b$, $\mathbf{E}_L$, $\mathbf{J}_T$, $\mathbf{F}$, $\mathbf{F}^{\text{T}}$, $\mathbf{K}_T$ and $\mathbf{K}_L$ are defined in Appendix 7, and $\mathbf{W}^{\text{iso}}$, $\mathbf{W}^{\text{cub}}$ and $\mathbf{W}^{\text{isT}}$ are isotropic, cubic and transversely isotropic tensors respectively.

2.4 Randomly generated anisotropic tensors

Lévesque et al. (2007) proposed a method to generate fourth-order symmetric tensors with the following decomposition:

$$\mathbf{Y} = \mathbf{Q}^{\text{T}} \cdot \mathbf{U} \cdot \mathbf{Q} \tag{25}$$

where $\mathbf{U}$ is the diagonal matrix containing the eigenvalues of $\mathbf{Y}$, and $\mathbf{Q}$ is an orthogonal matrix containing the eigenvectors. In our specific case, $\mathbf{Q}$ is defined as

$$\mathbf{Q} = \prod_{q=1}^{15} \mathbf{Q}(\theta_q) \tag{26}$$

where each $\mathbf{Q}(\theta_q)$ is a plane rotation. For example, the rotation in the 1–2 plane can be given as

$$\mathbf{Q}(\theta_1) = \begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{27}$$

Randomly generating fourth-order symmetric positive semi-definite anisotropic tensors is achieved by generating the 15 angles (θ_q) and the 6 positive eigenvalues (denoted U_{ii}) randomly, assembling the rotation matrices according to Eq. (26) and computing the tensors using Eq. (25).

3 Analytical interconversion algorithm

Assuming that $\mathbf{C}^{(0)}$, $\mathbf{C}^{(n)}$ and ρ_n are known and $\mathbf{S}^{(0)}$, $\mathbf{S}^{(m)}$ and λ_m are sought, then $\mathbf{L}^{(i)}$ and $\mathbf{B}$ can be computed using Eqs. (14a), (15) and (16). Recalling Eq. (4a), $\boldsymbol{\varepsilon}$ can be expressed as

$$\begin{aligned}\boldsymbol{\varepsilon}(t) &= \left(\mathbf{L}^{(1)}\right)^{-1} \cdot \left[\boldsymbol{\sigma} - \mathbf{L}^{(2)} \cdot \boldsymbol{\chi}\right] \\ &= \mathbf{A}^{(1)} \cdot \boldsymbol{\sigma} - \mathbf{A}^{(2)} \cdot \boldsymbol{\chi} \\ &= \mathbf{A}^{(1)} \cdot \boldsymbol{\sigma} - \mathbf{A}^{(2)} \cdot \boldsymbol{\xi}\end{aligned} \tag{28}$$

where $\boldsymbol{\chi}$ has been replaced by $\boldsymbol{\xi}$ since the internal variables are now expressed as functions of $\boldsymbol{\sigma}$. Injecting Eq. (28) into Eq. (4b) leads to

$$\begin{aligned}\mathbf{0} &= \mathbf{B} \cdot \dot{\boldsymbol{\chi}} + \mathbf{L}^{(3)} \cdot \boldsymbol{\chi} + \left(\mathbf{L}^{(2)}\right)^{\mathrm{T}} \cdot \left(\mathbf{L}^{(1)}\right)^{-1} \cdot \left[\boldsymbol{\sigma} - \mathbf{L}^{(2)} \cdot \boldsymbol{\chi}\right] \\ &= \mathbf{B} \cdot \dot{\boldsymbol{\chi}} + \left[\mathbf{L}^{(3)} - \left(\mathbf{L}^{(2)}\right)^{\mathrm{T}} \cdot \left(\mathbf{L}^{(1)}\right)^{-1} \cdot \mathbf{L}^{(2)}\right] \cdot \boldsymbol{\chi} + \left(\mathbf{L}^{(2)}\right)^{\mathrm{T}} \cdot \left(\mathbf{L}^{(1)}\right)^{-1} \cdot \boldsymbol{\sigma} \\ &= \mathbf{B} \cdot \dot{\boldsymbol{\chi}} + \mathbf{A}^{(3)} \cdot \boldsymbol{\chi} + \left(\mathbf{A}^{(2)}\right)^{\mathrm{T}} \cdot \boldsymbol{\sigma} \\ &= \mathbf{B} \cdot \dot{\boldsymbol{\xi}} + \mathbf{A}^{(3)} \cdot \boldsymbol{\xi} + \left(\mathbf{A}^{(2)}\right)^{\mathrm{T}} \cdot \boldsymbol{\sigma}\end{aligned} \tag{29}$$

where the fact that $\mathbf{L}^{(1)}$ is symmetric has been used.

As in Sect. 2, $\boldsymbol{\xi}(t)$ can be obtained from Eq. (29) and injected into Eq. (28) to obtain $\boldsymbol{\varepsilon}(t)$. In this specific case, $\mathbf{A}^{(3)}$ cannot be chosen to be arbitrarily diagonal since it is specifically expressed as a function of the $\mathbf{L}^{(i)}$. Considering that $\mathbf{L} > 0$, it can easily be shown that $\mathbf{A}^{(3)} > 0$. It can therefore be diagonalized using a particular case of the Singular Value Decomposition theorem:[6]

$$\mathbf{A}^{(3)} = \mathbf{P} \cdot \mathbf{D} \cdot \mathbf{P}^{\mathrm{T}} \tag{30}$$

where $\mathbf{D}$ contains the eigenvalues of $\mathbf{A}^{(3)}$ along its diagonal, and $\mathbf{P}$ has the orthonormal eigenvectors along its columns (Trefethen and Bau 1997). Now, let $\boldsymbol{\xi} = \mathbf{P} \cdot \boldsymbol{\xi}^*$ and multiply both sides of Eq. (29) by $\mathbf{P}^{\mathrm{T}}$. Then:

$$\begin{aligned}\mathbf{0} &= \mathbf{P}^{\mathrm{T}} \cdot \mathbf{B} \cdot \mathbf{P} \cdot \dot{\boldsymbol{\xi}}^* + \mathbf{P}^{\mathrm{T}} \cdot \mathbf{A}^{(3)} \cdot \mathbf{P} \cdot \boldsymbol{\xi}^* + \mathbf{P}^{\mathrm{T}} \cdot \left(\mathbf{A}^{(2)}\right)^{\mathrm{T}} \cdot \boldsymbol{\sigma} \\ &= \mathbf{B} \cdot \dot{\boldsymbol{\xi}}^* + \mathbf{A}^{(3*)} \cdot \boldsymbol{\xi}^* + \left(\mathbf{A}^{(2*)}\right)^{\mathrm{T}} \cdot \boldsymbol{\sigma}\end{aligned} \tag{31}$$

$\mathbf{B}$ remains the identity matrix, and $\mathbf{A}^{(3*)}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}^{(3)}$. Finally, Eq. (28) becomes

$$\boldsymbol{\varepsilon}(t) = \mathbf{A}^{(1)} \cdot \boldsymbol{\sigma} - \mathbf{A}^{(2*)} \cdot \boldsymbol{\xi}^* \tag{32}$$

Equations (31) are similar to Eqs. (10a), (10b) and can be solved using the same approach in order to obtain expressions for $\mathbf{S}^{(0)}$, $\mathbf{S}^{(m)}$ and λ_m.

[6] If $\boldsymbol{\Sigma}$ is a $u \times u$ matrix, then it can be written using a so-called singular value decomposition of the form $\boldsymbol{\Sigma}_{[u\times u]} = \boldsymbol{\Gamma}_{[u\times u]} \cdot \boldsymbol{\Delta}_{[u\times u]} \cdot \boldsymbol{\Lambda}^{\mathrm{T}}_{[u\times u]}$ where $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ have the orthonormal eigenvectors along their columns so that $\boldsymbol{\Gamma}^{\mathrm{T}} \cdot \boldsymbol{\Gamma} = \mathbf{I}$ and $\boldsymbol{\Lambda}^{\mathrm{T}} \cdot \boldsymbol{\Lambda} = \mathbf{I}$ and $\boldsymbol{\Delta}$ has entries only along its diagonal.

$\mathbf{S}^{(0)}$, $\mathbf{S}^{(m)}$ and λ_m are now related to the initially known $\mathbf{C}^{(0)}$, $\mathbf{C}^{(n)}$ and ρ_n. It should be noted that the procedure is independent of the degree of symmetry and is applicable to unidimensional, bidimensional and tridimensional cases alike. For the unidimensional case, Eqs. (14a), (15) and (16) become

$$L_1 = C_0 + \sum_{n=1}^{N} C_n \tag{33a}$$

$$\boldsymbol{l}^{(2)} = [\sqrt{\rho_1 C_1}, \sqrt{\rho_2 C_2}, \ldots, \sqrt{\rho_N C_N}] \tag{33b}$$

$$\mathbf{L}^{(3)} = \bigoplus_{n=1}^{N} [\rho_n \mathbf{I}] \tag{33c}$$

where L_1 is a scalar, and $\boldsymbol{l}^{(2)}$ is a $1 \times n$ vector. The internal matrices associated with $S(t)$ are computed as follows:

$$A_1 = (L_1)^{-1} \tag{34a}$$

$$\left(\mathbf{a}^{(2)}\right)^{\mathrm{T}} = \left(\boldsymbol{l}^{(2)}\right)^{\mathrm{T}} (L_1)^{-1} \tag{34b}$$

$$\mathbf{A}^{(3)} = \mathbf{L}^{(3)} - (L_1)^{-1}\left(\boldsymbol{l}^{(2)} \otimes \boldsymbol{l}^{(2)}\right) \tag{34c}$$

Furthermore, the same methodology can be applied for expressing the relaxation tensors as functions of the creep compliance tensors. From the $\mathbf{S}^{(0)}$, $\mathbf{S}^{(m)}$ and λ_m, the results of Crochon et al. (2010) deliver the $\mathbf{A}^{(i)}$ matrices. Then, the $\mathbf{L}^{(i)}$ matrices are expressed as functions of the $\mathbf{A}^{(i)}$ matrices and the $\boldsymbol{\chi}$ solved by diagonalization. Algorithms 1 and 2 list the different steps for the interconversion from $C(t)$ to $S(t)$ and from $S(t)$ to $C(t)$, while Algorithms 3 and 4 list the steps for the interconversion from $\mathbf{C}(t)$ to $\mathbf{S}(t)$ and from $\mathbf{S}(t)$ to $\mathbf{C}(t)$, respectively.

4 Examples of interconversion

This section provides complete theoretical examples of applications of Algorithms 1 and 3. Details are given to allow other researchers to validate their own implementations and to exemplify the procedures.

4.1 Unidimensional case

Assume a shear relaxation modulus $\mu(t)$ with two relaxation times expressed as

$$\mu(t) = 10 + 3\exp(-2t) + 4\exp(-t/35) \tag{35}$$

The first step is to compute the internal matrices according to Algorithm 1 (Lines 1–5), which leads to

$$L_1 = 17; \qquad \boldsymbol{l}^{(2)} = \begin{bmatrix} \sqrt{6} & 2/\sqrt{35} \end{bmatrix}; \qquad \mathbf{L}^{(3)} = \begin{bmatrix} 2 & 0 \\ 0 & 1/35 \end{bmatrix}; \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{36}$$

Algorithm 1 Interconversion from $C(t)$ to $S(t)$

1: *Compute the internal matrices associated with the relaxation modulus*
2: $L_{1[1\times 1]} = C_0 + \sum_{n=1}^{N} C_n$
3: $\boldsymbol{l}^{(2)}_{[1\times N]} = [\,\sqrt{\rho_1 C_1}\ \sqrt{\rho_2 C_2}\ \ldots\ \sqrt{\rho_N C_N}\,]$
4: $\mathbf{L}^{(3)}_{[N\times N]} = \begin{bmatrix} \rho_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \rho_N \end{bmatrix}$
5: $\mathbf{B}_{[N\times N]}$ is the identity matrix
6: *Compute the internal matrices associated with the creep compliance*
7: $A_1 = (L_1)^{-1}$
8: $(\mathbf{a}^{(2)})^{\mathrm{T}} = (\boldsymbol{l}^{(2)})^{\mathrm{T}}\,(L_1)^{-1}$
9: $\mathbf{A}^{(3)} = \mathbf{L}^{(3)} - (L_1)^{-1}(\boldsymbol{l}^{(2)} \otimes \boldsymbol{l}^{(2)})$
10: *Diagonalization*
11: Compute the eigenvectors $\mathbf{P}$ and eigenvalues $\mathbf{D}$ of $\mathbf{A}^{(3)}$ with singular value decomposition
12: $\mathbf{A}^{(3*)} = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{A}^{(3)} \cdot \mathbf{P}$
13: $(\mathbf{a}^{(2*)})^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} \cdot (\mathbf{a}^{(2)})^{\mathrm{T}}$
14: *Obtain the S_0, S_m and λ_m*
15: $S_0 = A_1$
16: **for** $m = 1$ to $M = N$ **do**
17: $\quad S_m = (a_m^{(2*)})^2 / A_{mm}^{(3*)}$ (no sum on m)
18: $\quad \lambda_m = A_{mm}^{(3*)} / B_{mm}$
19: **end for**

A preliminary set of internal matrices is computed (Lines 6–10):

$$A_1 = 1/17; \qquad \mathbf{a}^{(2)} = \left[\,\sqrt{6}/17 \quad 2/(17\sqrt{35})\,\right]; \qquad \mathbf{A}^{(3)} = \begin{bmatrix} 28/17 & -\frac{2}{17}\sqrt{\frac{6}{35}} \\ -\frac{2}{17}\sqrt{\frac{6}{35}} & 13/595 \end{bmatrix} \tag{37}$$

The eigenvalues and eigenvectors of $\mathbf{A}^{(3)}$ are then computed (Lines 10–13):

$$\lambda_m = \{0.0204, 1.649\}; \qquad \mathbf{P} = \begin{bmatrix} -0.02993 & -0.9996 \\ -0.9996 & 0.02993 \end{bmatrix} \tag{38}$$

Computation of the final set of internal matrices leads to

$$\mathbf{a}^{(2*)} = \left[\,-0.0242 \quad -0.1434\,\right]; \quad \mathbf{A}^{(3*)} = \begin{bmatrix} 0.0204 & 0 \\ 0 & 1.649 \end{bmatrix} \tag{39}$$

Finally, the shear compliance $\mu^{-1}(t)$ is given by (Lines 14–19):

$$10^2\mu^{-1}(t) = 5.88 + 2.87\big(1 - \exp[-0.0204t]\big) + 1.248\big(1 - \exp[-1.649t]\big) \tag{40}$$

For verification purposes, $\mu(t)$ was also interconverted to $\mu^{-1}(t)$ with the correspondence principle. Expressing Eq. (35) in the Laplace–Carson domain leads to

$$\mu^*(p) = 10 + \frac{3}{p+2} + \frac{4}{p+1/35} \tag{41}$$

Algorithm 2 Interconversion from $S(t)$ to $C(t)$

Compute the internal matrices associated with the creep compliance
$A_{1[1\times 1]} = S_0$
$\mathbf{a}^{(2)}_{[1\times M]} = [\,\sqrt{\lambda_1 S_1}\ \sqrt{\lambda_2 S_2}\ \ldots\ \sqrt{\lambda_M S_M}\,]$
$\mathbf{A}^{(3)}_{[M\times M]} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_M \end{bmatrix}$
$\mathbf{B}_{[M\times M]}$ is the identity matrix
Compute the internal matrices associated with the relaxation modulus
$L_1 = (A_1)^{-1}$
$(\boldsymbol{l}^{(2)})^{\mathrm{T}} = (\mathbf{a}^{(2)})^{\mathrm{T}}\,(A_1)^{-1}$
$\mathbf{L}^{(3)} = \mathbf{A}^{(3)} + (A_1)^{-1}(\mathbf{a}^{(2)} \otimes \mathbf{a}^{(2)})$
Diagonalization
Compute the eigenvectors $\mathbf{P}$ and eigenvalues $\mathbf{D}$ of $\mathbf{L}^{(3)}$ with singular value decomposition
$\mathbf{L}^{(3*)} = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{L}^{(3)} \cdot \mathbf{P}$
$(\boldsymbol{l}^{(2*)})^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} \cdot (\boldsymbol{l}^{(2)})^{\mathrm{T}}$
Obtain the C_0, C_n and ρ_n
$C_0 = L_1 - \sum_{n=1}^{N} C_n$
for $n = 1$ to $N = M$ **do**
    $C_n = (l_n^{(2*)})^2 / L_{nn}^{(3*)}$ (no sum on n)
    $\rho_n = L_{nn}^{(3*)} / B_{nn}$
end for

and

$$\left(\mu^*\right)^{-1}(p) = \frac{(35p+1)(p+2)}{595p^2+993p+20} = \frac{(35p+1)(p+2)}{595(p+\omega_1)(p+\omega_2)} \tag{42}$$

where $\omega_1 = 0.0204$ and $\omega_2 = 1.649$ are the retardation times associated to the shear compliance $\mu^{-1}(t)$. Then, decomposing Eq. (42) into partial fractions yields:

$$\begin{aligned}\left(\mu^*\right)^{-1}(p) &= \frac{35}{595} + \frac{(1-35\omega_1)(2-\omega_2)}{595(\omega_2-\omega_1)(p+\omega_1)} + \frac{(1-35\omega_2)(2-\omega_2)}{595(\omega_1-\omega_2)(p+\omega_2)} \\ &= 5.88\times 10^{-2} + \frac{2.87\times 10^{-2}}{(p+\omega_1)} + \frac{1.248\times 10^{-2}}{(p+\omega_2)}\end{aligned} \tag{43}$$

Finally, transforming back Eq. (43) to the time domain leads to the same expression of the shear compliance found in Eq. (40).

4.2 Isotropic, cubic symmetry and transverse isotropy

For materials having isotropic, cubic or transversely isotropic symmetries, the proposed method consists of first determining the parameters $\alpha(t)$, $\beta(t)$, $\gamma(t)$, $\gamma'(t)$, $\delta(t)$ and $\delta'(t)$. Then, interconversion is performed with either Algorithm 1 or 2 on each of these unidimensional functions. Finally, the target material functions are computed with Eqs. (24a), (24b) and the interconverted functions. For these material symmetries, the tridimensional interconversion reduces to 2, 3 or 5 unidimensional interconversions.

Algorithm 3 Interconversion from $\mathbf{C}(t)$ to $\mathbf{S}(t)$

Compute the internal matrices associated with the relaxation modulus
$\mathbf{L}^{(1)}_{[6\times 6]} = \mathbf{C}^{(0)} + \sum_{n=1}^{N} \mathbf{C}^{(n)}$
$\mathbf{L}^{(2)}_{[6\times 6N]} = [\, \mathcal{C}_{\mathcal{L}}(\rho_1 \mathbf{C}^{(1)}) |\, \mathcal{C}_{\mathcal{L}}(\rho_2 \mathbf{C}^{(2)}) |\, \ldots |\, \mathcal{C}_{\mathcal{L}}(\rho_N \mathbf{C}^{(N)}) \,]$
$\mathbf{L}^{(3)}_{[6N\times 6N]} = \begin{bmatrix} [\rho_1]_6 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & [\rho_N]_6 \end{bmatrix}$
$\mathbf{B}_{[6N\times 6N]}$ is the identity matrix
Compute the internal matrices associated with the creep compliance
$\mathbf{A}^{(1)} = (\mathbf{L}^{(1)})^{-1}$
$(\mathbf{A}^{(2)})^{\mathrm{T}} = (\mathbf{L}^{(2)})^{\mathrm{T}} \cdot (\mathbf{L}^{(1)})^{-1}$
$\mathbf{A}^{(3)} = \mathbf{L}^{(3)} - (\mathbf{L}^{(2)})^{\mathrm{T}} \cdot (\mathbf{L}^{(1)})^{-1} \cdot \mathbf{L}^{(2)}$
Diagonalization
Compute the eigenvectors $\mathbf{P}$ and eigenvalues $\mathbf{D}$ of $\mathbf{A}^{(3)}$ with singular value decomposition
$\mathbf{A}^{(3*)} = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{A}^{(3)} \cdot \mathbf{P}$
$(\mathbf{A}^{(2*)})^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} \cdot (\mathbf{A}^{(2)})^{\mathrm{T}}$
Obtain the $\mathbf{S}^{(0)}$, $\mathbf{S}^{(m)}$ *and* λ_m
$\mathbf{S}^{(0)} = \mathbf{A}^{(1)}$
for $m = 1$ to $M = 6N$ **do**
    $S^{(m)}_{ij} = \frac{A^{(2*)}_{im} A^{(2*)}_{jm}}{A^{(3*)}_{mm}}$ (no sum on m)
    $\lambda_m = \frac{A^{(3*)}_{mm}}{B_{mm}}$
end for

4.3 Anisotropic materials

Consider the following randomly generated anisotropic material:

$$\mathbf{C}^{(0)} = \begin{bmatrix} 136.8 & 1.546 & 0.996 & 1.209 & 0.408 & 0.026 \\ 1.546 & 100.9 & -3.455 & -4.792 & 5.692 & 1.367 \\ 0.996 & -3.455 & 51.18 & 48.78 & -20.04 & -15.27 \\ 1.209 & -4.792 & 48.78 & 75.10 & -37.96 & 4.752 \\ 0.408 & 5.692 & -20.04 & -37.96 & 235.3 & 12.08 \\ 0.026 & 1.367 & -15.27 & 4.752 & 12.08 & 189.1 \end{bmatrix} 10^{-1} \tag{44a}$$

$$\mathbf{C}^{(1)} = \begin{bmatrix} 44.76 & 0.560 & 1.741 & 0.173 & 2.033 & 2.467 \\ 0.560 & 116.9 & -27.11 & 35.68 & 11.87 & -10.81 \\ 1.741 & -27.11 & 55.8 & -3.398 & 5.29 & -9.681 \\ 0.173 & 35.68 & -3.398 & 51.97 & -30.22 & 18.29 \\ 2.033 & 11.87 & 5.285 & -30.22 & 95.46 & -50.67 \\ 2.467 & -10.81 & -9.681 & 18.29 & -50.67 & 88.94 \end{bmatrix} 10^{-1} \tag{44b}$$

$$\mathbf{C}^{(2)} = \begin{bmatrix} 25.73 & -12.29 & -9.826 & -6.928 & -1.113 & -2.627 \\ -12.29 & 119.2 & -53.04 & -57.93 & -4.033 & 18.40 \\ -9.826 & -53.04 & 147.4 & -31.91 & -56.97 & -16.33 \\ -6.928 & -57.93 & -31.91 & 198.1 & 61.87 & -32.13 \\ -1.113 & -4.033 & -56.97 & 61.87 & 155.0 & 23.37 \\ -2.627 & 18.40 & -16.33 & -32.13 & 23.37 & 97.59 \end{bmatrix} 10^{-1} \tag{44c}$$

Algorithm 4 Interconversion from $\mathbf{S}(t)$ to $\mathbf{C}(t)$

1: *Compute the internal matrices associated with the creep compliance*
2: $\mathbf{A}^{(1)}_{[6\times 6]} = \mathbf{S}^{(0)}$
3: $\mathbf{A}^{(2)}_{[6\times 6M]} = [\, \mathcal{C}_{\mathcal{L}}(\lambda_1 \mathbf{S}^{(1)}) | \; \mathcal{C}_{\mathcal{L}}(\lambda_2 \mathbf{S}^{(2)}) | \; \ldots | \; \mathcal{C}_{\mathcal{L}}(\lambda_M \mathbf{S}^{(M)}) \,]$
4: $\mathbf{A}^{(3)}_{[6M\times 6M]} = \begin{bmatrix} [\lambda_1]_6 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & [\lambda_M]_6 \end{bmatrix}$
5: $\mathbf{B}_{[6M\times 6M]}$ is the identity matrix
6: *Compute the internal matrices associated with the relaxation modulus*
7: $\mathbf{L}^{(1)} = (\mathbf{A}^{(1)})^{-1}$
8: $(\mathbf{L}^{(2)})^{\mathrm{T}} = (\mathbf{A}^{(2)})^{\mathrm{T}} \cdot (\mathbf{A}^{(1)})^{-1}$
9: $\mathbf{L}^{(3)} = \mathbf{A}^{(3)} + (\mathbf{A}^{(2)})^{\mathrm{T}} \cdot (\mathbf{A}^{(1)})^{-1} \cdot \mathbf{A}^{(2)}$
10: *Diagonalization*
11: Compute the eigenvectors $\mathbf{P}$ and eigenvalues $\mathbf{D}$ of $\mathbf{L}^{(3)}$ with singular value decomposition
12: $\mathbf{L}^{(3*)} = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{L}^{(3)} \cdot \mathbf{P}$
13: $(\mathbf{L}^{(2*)})^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} \cdot (\mathbf{L}^{(2)})^{\mathrm{T}}$
14: *Obtain the* $\mathbf{C}^{(0)}$, $\mathbf{C}^{(n)}$ *and* ρ_n
15: $\mathbf{C}^{(0)} = \mathbf{L}^{(1)} - \sum_{n=1}^{N} \mathbf{C}^{(n)}$
16: **for** $n = 1$ to $N = 6M$ **do**
17: $\quad C^{(n)}_{ij} = \frac{L^{(2*)}_{in} L^{(2*)}_{jn}}{L^{(3*)}_{nn}}$ (no sum on n)
18: $\quad \rho_n = \frac{L^{(3*)}_{nn}}{B_{nn}}$
19: **end for**

with the inverted relaxation times

$$\rho_n = \{14.514, 368.88\} \tag{44d}$$

Following the methodology proposed in Algorithm 3 (Lines 1–5), the internal matrices are computed as

$$\mathbf{L}^{(1)} = \begin{bmatrix} 207.3 & -10.18 & -7.089 & -5.545 & 1.328 & -0.134 \\ -10.18 & 337.1 & -83.61 & -27.05 & 13.52 & 8.960 \\ -7.089 & -83.61 & 254.4 & 13.46 & -71.72 & -41.28 \\ -5.545 & -27.05 & 13.46 & 325.1 & -6.298 & -9.086 \\ 1.328 & 13.52 & -71.72 & -6.298 & 485.8 & -15.21 \\ -0.134 & 8.960 & -41.28 & -9.086 & -15.21 & 375.6 \end{bmatrix} 10^{-1} \tag{45a}$$

$$\mathbf{L}^{(2)} = \begin{bmatrix} 8.06 & 0 & 0 & 0 & 0 & 0 & 30.8 & 0 & 0 & 0 & 0 & 0 \\ 0.10 & 13.0 & 0 & 0 & 0 & 0 & -14.7 & 64.6 & 0 & 0 & 0 & 0 \\ 0.31 & -3.02 & 8.47 & 0 & 0 & 0 & -11.7 & -32.9 & 64.9 & 0 & 0 & 0 \\ 0.03 & 3.97 & 0.835 & 7.67 & 0 & 0 & -8.29 & -34.9 & -37.3 & 68.0 & 0 & 0 \\ 0.36 & 1.31 & 1.36 & -6.54 & 9.59 & 0 & -1.33 & -2.60 & -33.9 & 13.4 & 66.1 & 0 \\ 0.44 & -1.20 & -2.10 & 4.31 & -4.27 & 9.28 & -3.14 & 9.78 & -4.88 & -15.4 & 13.9 & 55.0 \end{bmatrix} \tag{45b}$$

$$\mathbf{L}^{(3)} = \begin{bmatrix} [14.514]_6 & \\ & [368.88]_6 \end{bmatrix} \tag{45c}$$

The internal matrices $\mathbf{A}^{(i)}$ are computed (Lines 6–9) as

$$\mathbf{A}^{(1)} = \begin{bmatrix} 484.235 & 20.320 & 20.406 & 9.190 & 1.311 & 2.206 \\ 20.320 & 325.796 & 109.318 & 23.209 & 7.475 & 5.115 \\ 20.406 & 109.318 & 456.465 & -6.788 & 65.775 & 50.069 \\ 9.190 & 23.209 & -6.788 & 310.099 & 2.544 & 6.308 \\ 1.311 & 7.475 & 65.775 & 2.544 & 215.867 & 15.856 \\ 2.206 & 5.115 & 50.069 & 6.308 & 15.856 & 272.411 \end{bmatrix} 10^{-4} \tag{46a}$$

$$\mathbf{A}^{(2)} = \begin{bmatrix} 39.1 & 2.39 & 1.78 & 0.715 & 0.03 & 0.20 & 143 & 3.40 & 9.27 & 6.09 & 1.18 & 1.22 \\ 2.37 & 40.1 & 9.45 & 1.51 & 0.49 & 0.47 & -56.7 & 166 & 59.6 & 16.0 & 5.66 & 2.82 \\ 3.65 & 0.43 & 38.5 & -2.67 & 4.17 & 4.65 & -65.4 & -74.0 & 274 & -3.52 & 50.5 & 27.6 \\ 0.87 & 15.5 & 1.92 & 23.9 & -0.02 & 0.58 & -25.7 & -90.5 & -121 & 210 & 2.57 & 3.48 \\ 1.18 & 1.74 & 8.20 & -13.23 & 20.0 & 1.47 & -12.0 & -21.8 & -32.2 & 28.3 & 145 & 8.74 \\ 1.61 & -3.68 & -1.23 & 11.2 & -10.1 & 25.3 & -15.3 & 10.9 & 11.5 & -35.7 & 48.6 & 150 \end{bmatrix} 10^{-2} \tag{46b}$$

$$\mathbf{A}^{(3)} = \begin{bmatrix} 11.3 & -0.22 & -0.29 & -0.05 & -0.04 & -0.14 & -11.1 & -0.15 & -1.56 & -0.51 & -1.01 & -0.88 \\ -0.22 & 8.62 & -0.26 & -0.91 & -0.32 & 0.34 & 6.41 & -20 & 5.92 & -11.4 & -0.63 & 2.03 \\ -0.29 & -0.26 & 11.1 & 0.44 & -0.83 & 0.11 & 5.6 & 7.55 & -21.5 & -2.6 & -5.26 & 0.67 \\ -0.05 & -0.91 & 0.44 & 11.3 & 1.75 & -1.04 & 1.85 & 5.06 & 6.71 & -12.8 & 7.2 & -6.17 \\ -0.04 & -0.32 & -0.83 & 1.75 & 12.2 & 0.94 & 0.5 & 2.55 & 3.58 & -4.24 & -11.8 & 5.58 \\ -0.14 & 0.34 & 0.11 & -1.04 & 0.94 & 12.2 & 1.42 & -1.01 & -1.07 & 3.32 & -4.51 & -13.9 \\ -11.1 & 6.41 & 5.6 & 1.85 & 0.5 & 1.42 & 306 & 7.36 & 28 & 16.8 & 10.1 & 8.41 \\ -0.15 & -20 & 7.55 & 5.06 & 2.55 & -1.01 & 7.36 & 203 & 7.44 & 66.2 & 12.9 & -5.98 \\ -1.56 & 5.92 & -21.5 & 6.71 & 3.58 & -1.07 & 28 & 7.44 & 135 & 88.6 & 19.7 & -6.32 \\ -0.51 & -11.4 & -2.6 & -12.8 & -4.24 & 3.32 & 16.8 & 66.2 & 88.6 & 216 & -13.7 & 19.6 \\ -1.01 & -0.63 & -5.26 & 7.2 & -11.8 & -4.51 & 10.1 & 12.9 & 19.7 & -13.7 & 266 & -26.8 \\ -0.88 & 2.03 & 0.67 & -6.17 & 5.58 & -13.9 & 8.41 & -5.98 & -6.32 & 19.7 & -26.8 & 286 \end{bmatrix} \tag{46c}$$

The eigenvalues of $\mathbf{A}^{(3)}$ as well as matrix $\mathbf{P}$ are computed in Matlab as per singular value decomposition algorithm (Lines 10–13) as

$$\begin{aligned} \lambda_m = \{&2.1494, 7.3787, 8.9574, 10.815, 10.919, 12.282, \\ &72.326, 177.34, 245.93, 281.99, 310.22, 339.59\} \end{aligned} \tag{47a}$$

$$\mathbf{P} = \begin{bmatrix} -2.65 & 0.61 & -2.58 & -0.37 & -0.20 & 0.35 & -8.38 & -95.8 & 26.1 & 6.13 & 1.59 & 2.57 \\ -1.65 & 0.24 & 6.56 & -0.11 & -8.73 & -7.55 & -2.15 & 7.62 & 23.8 & 24.8 & 68.6 & -62.0 \\ -0.80 & 1.19 & 2.49 & 1.96 & 11.1 & 18.8 & -5.10 & -3.40 & -34.2 & 48.8 & 50.2 & 58.2 \\ -0.71 & -3.82 & 1.26 & -1.14 & 2.50 & -19.4 & -2.99 & -15.4 & -62.2 & 49 & -35.1 & -42.7 \\ 0.11 & 3.47 & 0.56 & 2.47 & 1.37 & -12.9 & 62.4 & -20.8 & -45.9 & -48.4 & 31.4 & -5.57 \\ -0.29 & -2.56 & -0.83 & 5.23 & -1.00 & 5.54 & 77 & 6.07 & 36.7 & 46 & -21.1 & 6.62 \\ 68.4 & -16.4 & 67.2 & 19 & 10.1 & 5.05 & -0.62 & -3.71 & 1.26 & -2.44 & -2.79 & -0.86 \\ 31.1 & -7.05 & -43 & -0.29 & 79.2 & -26.7 & -1.02 & 2.76 & 9.69 & 0.49 & 5.28 & -3.16 \\ 33.5 & -7.62 & -22.1 & -0.22 & -50 & -71.1 & -3.50 & 2.40 & 4.97 & 7.79 & 6.31 & 24.8 \\ 52.7 & 10 & -53.3 & 7.50 & -29.4 & 54.9 & 1.61 & -3.07 & -10.1 & -0.53 & -0.55 & -14.7 \\ 6.95 & -60.2 & 0.08 & -78.2 & -4.05 & 12.3 & 4.96 & -1.16 & -1.50 & -1.83 & 2.32 & -0.25 \\ 19.4 & 76.5 & 15.2 & -58.5 & 5.03 & -6.41 & 2.86 & 0.36 & 2.25 & 4.22 & -2.47 & 1.32 \end{bmatrix} 10^{-2} \tag{47b}$$

$\mathbf{A}^{(2*)}$ and $\mathbf{A}^{(3*)}$ are computed as

$$\mathbf{A}^{(2*)} = \begin{bmatrix} 104 & -23.3 & 88.8 & 25.9 & 10.7 & 3.3 & -4.3 & -42.7 & 11.8 & 1.5 & -0.4 & 0.3 \\ 41.6 & -6.5 & -128 & -16.1 & 89.4 & -82.1 & -3.7 & 7.8 & 22.7 & 22.2 & 46.0 & -12.3 \\ 30.7 & -14.0 & -65.9 & -67.6 & -198 & -168 & -1.2 & 1.6 & -4.8 & 40.8 & 36.7 & 95.2 \\ 24.9 & 41.1 & -61.7 & 7.1 & -75.9 & 219 & 8.2 & -14.3 & -47.8 & 6.6 & -9.8 & -76.6 \\ 0.9 & -70.6 & -5.0 & -117 & -15.4 & 62.5 & 22.8 & -6.8 & -11.2 & -15.8 & 16.1 & -3.3 \\ 10.3 & 81.5 & 24.0 & -130 & 17.5 & -34.5 & 18.5 & 2.4 & 14.6 & 27.6 & -16.2 & 8.8 \end{bmatrix} 10^{-2} \tag{48a}$$

$$\mathbf{A}^{(3*)} = \begin{bmatrix} 339 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 310 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 282 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 246 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 177 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 72.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8.96 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.38 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.15 \end{bmatrix} \tag{48b}$$

Finally, the compliance matrices $\mathbf{S}^{(0)}$ and $\mathbf{S}^{(m)}$ are given as follows:

$$\mathbf{S}^{(0)} = \begin{bmatrix} 484 & 20.3 & 20.4 & 9.1 & 1.31 & 2.2 \\ 20.3 & 325 & 109 & 23.2 & 7.4 & 5.1 \\ 20.4 & 109 & 456 & -6.7 & 65.7 & 50 \\ 9.1 & 23.2 & -6.7 & 310 & 2.5 & 6.8 \\ 1.3 & 7.4 & 65.7 & 2.5 & 215 & 15.8 \\ 2.2 & 5.1 & 50 & 6.3 & 15.8 & 272 \end{bmatrix} 10^{-4}$$

$$\mathbf{S}^{(1)} = \begin{bmatrix} 321 & 128.0 & 94.3 & 76.6 & 2.75 & 31.8 \\ 128 & 51.0 & 37.6 & 30.5 & 1.10 & 12.7 \\ 94.3 & 37.6 & 27.7 & 22.5 & 0.81 & 9.33 \\ 76.6 & 30.5 & 22.5 & 18.3 & 0.66 & 7.58 \\ 2.75 & 1.10 & 0.81 & 0.66 & 0.02 & 0.27 \\ 31.8 & 12.7 & 9.33 & 7.58 & 0.27 & 3.14 \end{bmatrix} 10^{-5};$$

$$\mathbf{S}^{(2)} = \begin{bmatrix} 1.75 & 0.49 & 1.05 & -3.09 & 5.31 & -6.13 \\ 0.49 & 0.14 & 0.29 & -0.86 & 1.47 & -1.70 \\ 1.05 & 0.29 & 0.63 & -1.85 & 3.18 & -3.67 \\ -3.09 & -0.86 & -1.85 & 5.45 & -9.36 & 10.8 \\ 5.31 & 1.47 & 3.18 & -9.36 & 16.1 & -18.6 \\ -6.13 & -1.70 & -3.67 & 10.80 & -18.6 & 21.4 \end{bmatrix} 10^{-4};$$

$$\mathbf{S}^{(3)} = \begin{bmatrix} 27.9 & -40.5 & -20.7 & -19.4 & -1.56 & 7.56 \\ -40.5 & 58.6 & 30.0 & 28.1 & 2.26 & -11.0 \\ -20.7 & 30.0 & 15.4 & 14.4 & 1.16 & -5.61 \\ -19.4 & 28.1 & 14.4 & 13.5 & 1.09 & -5.25 \\ -1.56 & 2.26 & 1.16 & 1.09 & 0.09 & -0.42 \\ 7.56 & -11.0 & -5.61 & -5.25 & -0.42 & 2.05 \end{bmatrix} 10^{-4};$$

$$\mathbf{S}^{(4)} = \begin{bmatrix} 2.72 & -1.70 & -7.12 & 0.75 & -12.4 & -13.7 \\ -1.70 & 1.06 & 4.44 & -0.47 & 7.73 & 8.57 \\ -7.12 & 4.44 & 18.6 & -1.96 & 32.4 & 35.9 \\ 0.75 & -0.47 & -1.96 & 0.21 & -3.42 & -3.79 \\ -12.4 & 7.73 & 32.4 & -3.42 & 56.4 & 62.5 \\ -13.7 & 8.57 & 35.9 & -3.79 & 62.5 & 69.3 \end{bmatrix} 10^{-4};$$

$$\mathbf{S}^{(5)} = \begin{bmatrix} 0.64 & 5.37 & -11.9 & -4.56 & -0.93 & 1.05 \\ 5.37 & 45.1 & -99.9 & -38.3 & -7.77 & 8.81 \\ -11.9 & -99.9 & 221 & 84.8 & 17.2 & -19.5 \\ -4.56 & -38.3 & 84.8 & 32.5 & 6.60 & -7.48 \\ -0.93 & -7.77 & 17.2 & 6.60 & 1.34 & -1.52 \\ 1.05 & 8.81 & -19.5 & -7.48 & -1.52 & 1.72 \end{bmatrix} 10^{-4};$$

$$\mathbf{S}^{(6)} = \begin{bmatrix} 0.15 & -3.74 & -7.68 & 9.99 & 2.85 & -1.57 \\ -3.74 & 93.2 & 191 & -249 & -70.9 & 39.2 \\ -7.68 & 191 & 393 & -512 & -145.7 & 80.5 \\ 9.99 & -249 & -512 & 666 & 190 & -105 \\ 2.85 & -70.9 & -146 & 190 & 54.0 & -29.8 \\ -1.57 & 39.2 & 80.5 & -105 & -29.8 & 16.5 \end{bmatrix} 10^{-4};$$

$$\mathbf{S}^{(7)} = \begin{bmatrix} 1.52 & 1.31 & 0.43 & -2.87 & -8.02 & -6.52 \\ 1.31 & 1.13 & 0.37 & -2.48 & -6.93 & -5.63 \\ 0.43 & 0.37 & 0.12 & -0.80 & -2.25 & -1.83 \\ -2.87 & -2.48 & -0.80 & 5.43 & 15.2 & 12.3 \\ -8.02 & -6.93 & -2.25 & 15.2 & 42.4 & 34.4 \\ -6.52 & -5.63 & -1.83 & 12.3 & 34.4 & 28.0 \end{bmatrix} 10^{-4}$$

$$\mathbf{S}^{(8)} = \begin{bmatrix} 167 & -30.3 & -6.40 & 55.9 & 26.4 & -9.19 \\ -30.3 & 5.50 & 1.16 & -10.2 & -4.79 & 1.67 \\ -6.40 & 1.16 & 0.25 & -2.15 & -1.01 & 0.35 \\ 55.9 & -10.2 & -2.15 & 18.8 & 8.85 & -3.08 \\ 26.4 & -4.79 & -1.01 & 8.85 & 4.17 & -1.45 \\ -9.19 & 1.67 & 0.35 & -3.08 & -1.45 & 0.51 \end{bmatrix} 10^{-4}$$

$$\mathbf{S}^{(9)} = \begin{bmatrix} 12.9 & 24.8 & -5.26 & -52.1 & -12.2 & 16.0 \\ 24.8 & 47.7 & -10.1 & -100 & -23.6 & 30.7 \\ -5.26 & -10.1 & 2.15 & 21.3 & 5.00 & -6.52 \\ -52.1 & -100 & 21.3 & 211 & 49.6 & -64.6 \\ -12.2 & -23.6 & 5.0 & 49.6 & 11.6 & -15.2 \\ 16.0 & 30.7 & -6.52 & -64.6 & -15.2 & 19.8 \end{bmatrix} 10^{-4}$$

$$\mathbf{S}^{(10)} = \begin{bmatrix} 0.26 & 3.79 & 6.97 & 1.12 & -2.69 & 4.71 \\ 3.79 & 55.08 & 101 & 16.3 & -39.1 & 68.4 \\ 6.97 & 101 & 186 & 29.9 & -71.8 & 126 \\ 1.12 & 16.3 & 29.9 & 4.82 & -11.6 & 20.2 \\ -2.69 & -39.1 & -71.8 & -11.6 & 27.7 & -48.5 \\ 4.71 & 68.4 & 126 & 20.2 & -48.5 & 84.94 \end{bmatrix} 10^{-4}$$

$$\mathbf{S}^{(11)} = \begin{bmatrix} 0.02 & -2.46 & -1.96 & 0.52 & -0.86 & 0.87 \\ -2.46 & 287 & 229 & -61.0 & 101 & -101 \\ -1.96 & 229 & 182 & -48.6 & 80.1 & -80.7 \\ 0.52 & -61.0 & -48.6 & 13 & -21.4 & 21.5 \\ -0.86 & 101 & 80.1 & -21.4 & 35.2 & -35.5 \\ 0.87 & -101 & -80.7 & 21.5 & -35.5 & 35.7 \end{bmatrix} 10^{-4}$$

$$\mathbf{S}^{(12)} = \begin{bmatrix} 0.05 & -1.97 & 15.2 & -12.3 & -0.52 & 1.41 \\ -1.97 & 70.4 & -545 & 439 & 18.6 & -50.5 \\ 15.2 & -545 & 4219 & -3395 & -144 & 391 \\ -12.3 & 439 & -3395 & 2732 & 116 & -314 \\ -0.52 & 18.6 & -144 & 116 & 4.92 & -13.3 \\ 1.41 & -50.5 & 391 & -314 & -13.3 & 36.2 \end{bmatrix} 10^{-4}$$

Note that the matrices' elements have not been written with their full precision due to space restrictions and terms smaller than 10^{-13}, in absolute value, were set to 0. A numerical example for the interconversion from $\mathbf{S}(t)$ to $\mathbf{C}(t)$ is given in Appendix A.

Recall that for unidimensional cases, $M = N$. However, for generally anisotropic materials, if $\mathbf{C}(t)$ has N distinct relaxation times, $\mathbf{S}(t)$ will have at most $M = 6N$ distinct retardation times. The number of distinct retardation times will vary as a function of material symmetry.

5 Numerical case studies

5.1 Error definition

The right-hand side of Eq. (1), evaluated with the source and target functions, was used to define a measure of the proposed algorithm accuracy. Injecting Eqs. (9) and (13) into Eq. (1) leads to

$$t\mathbf{I} = \int_0^t \left(\mathbf{C}^{(0)} + \sum_{n=1}^{N} \mathbf{C}^{(n)} \exp\left[-(t-\tau)\rho_n\right] \right) \cdot \left(\mathbf{S}^{(0)} + \sum_{m=1}^{M} \mathbf{S}^{(m)} \left(1 - \exp[-\tau\lambda_m]\right) \right) \mathrm{d}\tau \quad (49)$$

Integrating Eq. (49) and differentiating with respect to t yields

$$\mathbf{I} = \mathbf{C}^{(0)} \cdot \mathbf{S}^{(\infty)} - \mathbf{C}^{(0)} \cdot \left(\sum_{m=1}^{M} \mathbf{S}^{(m)} \exp[-t\lambda_m] \right) + \left(\sum_{n=1}^{N} \mathbf{C}^{(n)} \exp[-t\rho_n] \right) \cdot \mathbf{S}^{(\infty)} + \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\mathbf{C}^{(n)} \cdot \mathbf{S}^{(m)}}{\rho_n - \lambda_m} \left[\lambda_m \exp[-t\lambda_m] - \rho_n \exp[-t\rho_n] \right] \quad (50)$$

Table 1 Parameters for the unidimensional relaxation modulus

		a	b	c
1	ρ_n	$[10^{-2}, 10^{3}]$	$[10^{-2}, 10^{5}]$	$[10^{-2}, 10^{8}]$
2	C_n	$[1, 10^{1.5}]$	$[1, 10^{2.5}]$	$[1, 10^{4}]$
3	N	5	10	20

where $\mathbf{S}^{(\infty)} = \sum_{m=0}^{M} \mathbf{S}^{(m)}$. Regrouping the exponential terms, Eq. (50) becomes

$$\begin{aligned}\mathbf{I} &= \mathbf{C}^{(0)} \cdot \mathbf{S}^{(\infty)} + \sum_{m=1}^{M} \left[-\mathbf{C}^{(0)} \cdot \mathbf{S}^{(m)} + \sum_{n=1}^{N} \frac{\mathbf{C}^{(n)} \cdot \mathbf{S}^{(m)}}{\rho_n - \lambda_m} \lambda_m \right] \exp[-t\lambda_m] \\ &\quad + \sum_{n=1}^{N} \left[\mathbf{C}^{(n)} \cdot \mathbf{S}^{(\infty)} - \sum_{m=1}^{M} \frac{\mathbf{C}^{(n)} \cdot \mathbf{S}^{(m)}}{\rho_n - \lambda_m} \rho_n \right] \exp(-t\rho_n) \\ &= \mathbf{C}^{(0)} \cdot \mathbf{S}^{(\infty)} + \sum_{m=1}^{M} \mathbf{X}^{(m)} \exp[-t\lambda_m] + \sum_{n=1}^{N} \mathbf{H}^{(n)} \exp(-t\rho_n) \end{aligned} \tag{51}$$

It can readily be seen that $X_{ij}^{(m)} = H_{ij}^{(n)} \cong 0$ and that $C_{ik}^{(0)} S_{kj}^{(\infty)} - I_{ij} \cong 0$. Therefore,

$$\epsilon = \max_{i,j} \left[\log_{10} \left\{ \left| \mathbf{C}^{(0)} \cdot \mathbf{S}^{(\infty)} - \mathbf{I} \right| + \sum_{m=1}^{M} \left| \mathbf{X}^{(m)} \right| + \sum_{n=1}^{N} \left| \mathbf{H}^{(n)} \right| \right\} \right] \tag{52}$$

bounds the maximum error, in line with Eq. (50), introduced by our interconversion algorithm.

5.2 Unidimensional case

Numerous interconversions of $C(t)$ to $S(t)$ were simulated using random values for the Prony coefficients C_n and for the inverted relaxation time constants ρ_n. The ρ_n were generated according to $\rho_n = 10^{\phi}$, where ϕ was a random number in the interval $[-2, \upsilon]$. The C_n were generated according to $C_n = 10^{\nu}$, where ν was a random number in the interval $[0, \eta]$. ϕ and ν were generated using the uniform distribution random number generator of Matlab. Moreover, the number of inverted relaxation times N took three different values. The parameters and their values are tabulated in Table 1. In the sequel, the various parameter combinations are referred to as 'Tag x1-x2-x3' were x1, x2 and x3 are linked to the columns of Table 1. For example, Tag a-b-c is for $\rho_n \in [10^{-2}, 10^{3}]$, $C_n \in [1, 10^{2.5}]$ and $N = 20$.

Equation (52) has been evaluated for 10^6 randomly generated relaxation moduli according to the parameters of Table 1, for both our method and that of Park and Schapery (1999). Figure 1 shows histograms for ϵ when $\rho_n \in [10^{-2}, 10^{3}]$, $C_n \in [1, 10^{1.5}]$ and $N = 5$ (Tag a–a–a). Figure 1 also shows the 99th percentile of ϵ. The 99th percentile of ϵ, denoted herein as ϵ^{99th}, was considered as a relevant measure of our algorithm accuracy since it allows us to state that 99 % of the time, the accuracy of the algorithm is at least of $10^{\epsilon^{99th}}$. The highest value of ϵ could have also been used, but as it is the case for Monte Carlo simulations, it is dependent on the sample size. The 99th percentile was assumed as a more stable criterion for evaluating the algorithm performance. When considering ϵ^{99th}, it can be seen that the

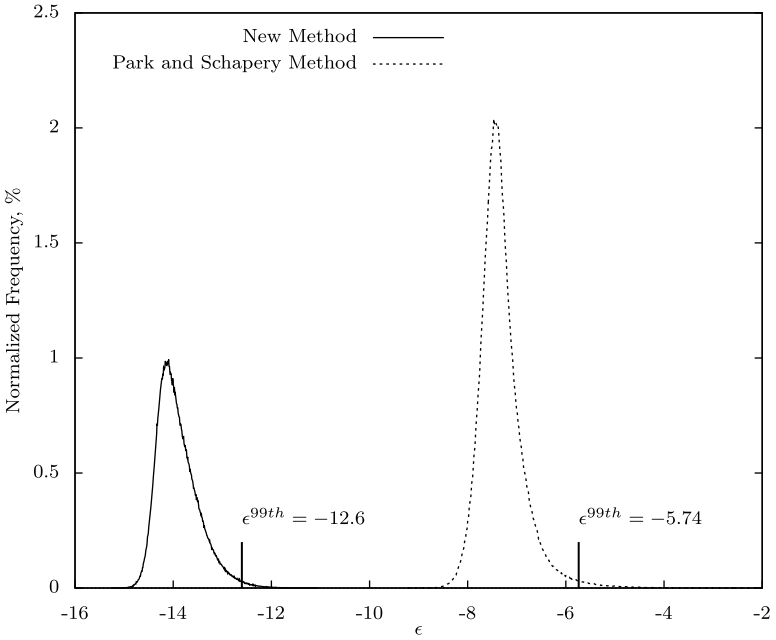

Fig. 1 Distribution of ϵ for 10^6 interconversions from $C(t)$ to $S(t)$ for materials generated according to Tag a–a–a with our New Method and that of Park and Schapery (1999)

Table 2 ϵ^{99th} computed from interconversions from $C(t)$ to $S(t)$ using our New Method and that of Park and Schapery (1999)

Tag	New	Schapery	Tag	New	Schapery	Tag	New
a-a-a	−12.6	−5.74	a-a-b	−11.9	−2.76	a-a-c	−11.5
b-a-a	−11.5	−6.05	b-a-b	−10.3	−3.87	b-a-c	−9.80
c-a-a	−9.59	−6.29	c-a-b	−7.89	−4.84	c-a-c	−7.09
a-b-a	−12.5	−5.62	a-b-b	−12.0	−2.76	a-b-c	−11.6
b-b-a	−11.4	−5.79	b-b-b	−10.4	−3.87	b-b-c	−9.79
c-b-a	−9.59	−5.89	c-b-b	−7.89	−4.81	c-b-c	−7.09
a-c-a	−11.7	−5.07	a-c-b	−11.5	−2.81	a-c-c	−11.2
b-c-a	−11.3	−5.18	b-c-b	−10.4	−3.89	b-c-c	−9.89
c-c-a	−9.49	−5.25	c-c-b	−7.99	−4.66	c-c-c	−7.29

new approach (for Tag a-a-a) is approximately 10^7 times more accurate than the method of Park and Schapery (1999) and leads to ϵ close to machine precision.

Table 2 lists the ϵ^{99th} for all 27 combinations of $\{\rho_n, C_n, N\}$ for both our method and that of Park and Schapery (1999). It can be seen that, in the worst case, the new method is approximately 10^3 times more accurate than that of Park and Schapery (1999) and that the worst value of ϵ^{99th} is $10^{-7.09}$, which is quite accurate. It can also be seen that no results are given for the Park and Schapery (1999) method when $N = 20$. This comes from the fact that Eq. (20) can become ill-conditioned as N increases. No such problem was encountered with our new method, which demonstrates its stability.

The influence of each parameter on ϵ was investigated through an analysis of variance. For each parameter value, the ϵ^{99th} of Table 2 were grouped, and 95 % confidence inter-

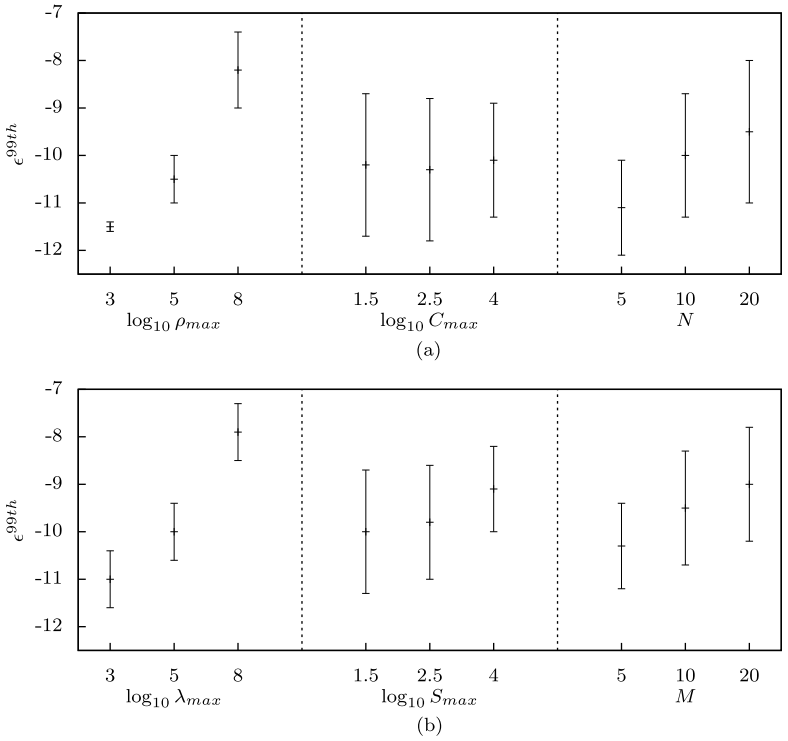

Fig. 2 Effect of the parameters on the mean value of ϵ^{99th} for the interconversion from (**a**) $C(t)$ to $S(t)$ and (**b**) $S(t)$ to $C(t)$. The interval represents a 95 % confidence interval for the mean of ϵ^{99th}

Table 3 ϵ^{99th} computed from interconversions from $S(t)$ to $C(t)$ using the New Method

Tag	New	Tag	New	Tag	New
a-a-a	−12.0	a-a-b	−11.7	a-a-c	−11.3
b-a-a	−11.1	b-a-b	−10.2	b-a-c	−9.72
c-a-a	−9.0	c-a-b	−7.60	c-a-c	−6.97
a-b-a	−11.5	a-b-b	−11.2	a-b-c	−10.7
b-b-a	−10.8	b-b-b	−10.2	b-b-c	−9.71
c-b-a	−9.03	c-b-b	−7.69	c-b-c	−7.06
a-c-a	−10.5	a-c-b	−10.1	a-c-c	−9.55
b-c-a	−10.0	b-c-b	−9.41	b-c-c	−8.79
c-c-a	−8.67	c-c-b	−7.73	c-c-c	−7.14

vals were calculated. For example, the first column of Table 2 lists the values of ϵ^{99th} when $N = 5$. From these 9 values, a 95 % confidence interval was calculated. Figure 2(a) illustrates the confidence intervals and the mean for ϵ^{99th} for each parameter value. It can be observed that all the confidence intervals for the ranges of C_n and those of N overlap. The algorithm seems to be insensitive to these parameters. However, the range of ρ_n seems to have a significant effect on ϵ^{99th}. The algorithm seems therefore to be sensitive to this parameter.

Table 4 Parameters for the anisotropic relaxation modulus

		a	b	c
1	ρ_n	$[10^{-2}, 10^{3}]$	$[10^{-2}, 10^{5}]$	$[10^{-2}, 10^{8}]$
2	U_{ii}	$[1, 10^{1.5}]$	$[1, 10^{2.5}]$	$[1, 10^{4}]$
3	N	5	10	20

Table 5 ϵ^{99th} computed from interconversions from $\mathbf{C}(t)$ to $\mathbf{S}(t)$ using the New Method

Tag	New	Tag	New	Tag	New
a-a-a	−10.8	a-a-b	−10.28	a-a-c	−9.65
b-a-a	−8.93	b-a-b	−8.40	b-a-c	−8.16
c-a-a	−6.2	c-a-b	−5.52	c-a-c	−5.27
a-b-a	−10.2	a-b-b	−9.57	a-b-c	−9.30
b-b-a	−8.41	b-b-b	−7.43	b-b-c	−7.17
c-b-a	−5.7	c-b-b	−4.56	c-b-c	−4.27
a-c-a	−9.07	a-c-b	−8.06	a-c-c	−7.82
b-c-a	−7.33	b-c-b	−6.20	b-c-c	−5.92
c-c-a	−4.68	c-c-b	−3.35	c-c-c	−3.04

The same analysis has been performed for the interconversion from $S(t)$ to $C(t)$ and led to similar results. Table 3 lists the distribution of ϵ^{99th}, while Fig. 2(b) illustrates the confidence intervals and the mean for ϵ^{99th}. Previous work such as Knoff and Hopkins (1972) and Anderssen et al. (2008) concluded that when no assumptions were made on the shape of the material functions, numerical interconversion from $S(t)$ to $C(t)$ was more problematic than the interconversion from $C(t)$ to $S(t)$. However, our new method has proven to be as efficient for both cases.

5.3 Anisotropic case

Similar tests have been performed for the interconversion from $\mathbf{C}(t)$ to $\mathbf{S}(t)$ for anisotropic materials. The anisotropic tensors have been randomly generated according to Sect. 2.4. The parameters were generated as in Sect. 5.2 and are given in Table 4. In addition, the θ_q were randomly generated numbers within the interval [0, 360]. The distribution of ϵ^{99th} is given in Table 5. In the worst case, ϵ^{99th} was −3.04 (Tag c-c-c). Recall that Tag c-c-c was for $N = 20$, which leads to potentially 120 different retardation times. The number of numerical operations is much more important for the tridimensional case than for the unidimensional case, which could explain the relatively low accuracy. It also represents a theoretical worst case scenario: the authors are unaware of published tridimensional and anisotropic relaxation stiffnesses for 20 relaxation times spread over 10 decades. Tag b-b-b could be more representative of a potential practical relaxation stiffness. For this case, $\epsilon^{99th} = 10^{-7.43}$, which is quite acceptable considering the complexity of the problem. Figure 3(a) shows the confidence intervals and the mean for ϵ^{99th} for each parameter value. Still, the range of ρ_n seems to be the parameter with the most significant effect on ϵ^{99th}. The same analysis has been performed for the interconversion from $\mathbf{S}(t)$ to $\mathbf{C}(t)$ and led to similar conclusions (see Table 6 and Fig. 3(b)).

Table 6 ϵ^{99th} computed from interconversions from $\mathbf{S}(t)$ to $\mathbf{C}(t)$ using the New Method

Tag	New	Tag	New	Tag	New
a-a-a	−10.3	a-a-b	−10.1	a-a-c	−9.9
b-a-a	−8.5	b-a-b	−8.2	b-a-c	−7.2
c-a-a	−5.6	c-a-b	−5.3	c-a-c	−5.1
a-b-a	−9.5	a-b-b	−9.12	a-b-c	−8.8
b-b-a	−7.6	b-b-b	−7.3	b-b-c	−7.0
c-b-a	−4.8	c-b-b	−4.4	c-b-c	−3.2
a-c-a	−8.0	a-c-b	−6.5	a-c-c	−6.2
b-c-a	−6.2	b-c-b	−4.6	b-c-c	−4.3
c-c-a	−3.2	c-c-b	−2.7	c-c-c	−2.3

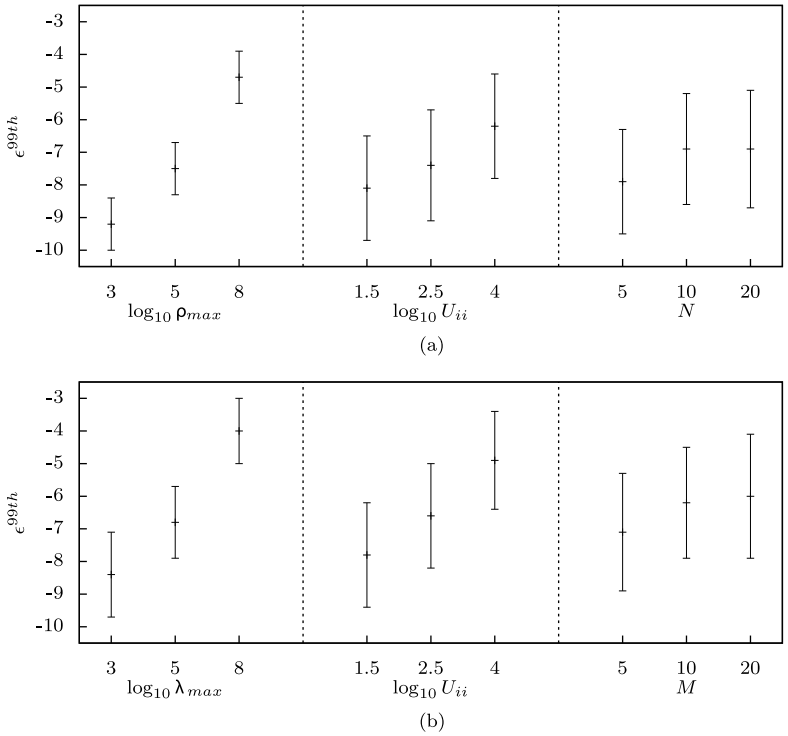

Fig. 3 Effect of the parameters on the mean value of ϵ^{99th} for the anisotropic interconversion from (**a**) $\mathbf{C}(t)$ to $\mathbf{S}(t)$ and (**b**) $\mathbf{S}(t)$ to $\mathbf{C}(t)$. The interval represents a 95 % confidence interval for the mean of ϵ^{99th}

6 Discussion

This work showed that our new algorithm is more accurate and stable than that of Park and Schapery (1999), even though it should theoretically lead to the same, exact, results. We provide below some leads for explaining this result.

The main differences between the two algorithms reside in the kind of mathematical/numerical problems they are built on. In the new algorithm, a potential source of inaccuracies and instabilities is the computation of the eigenvalues and eigenvectors for matrices

$\mathbf{A}^{(3*)}$ or $\mathbf{L}^{(3*)}$. The problem of numerically computing eigenvalues and eigenvectors has received considerable attention over the years. Very accurate and stable algorithms have been developed for their computation. The new algorithm proposed in this paper benefited from those progresses since it relies on the very efficient singular value decomposition algorithm implemented in Matlab. The accuracy and stability of this kind of algorithms depend only on the condition number of the matrix to decompose. This explains why the range of retardation (or relaxation) times seems to be the only parameter having an effect on the accuracy of the new algorithm. Conversely, computation of the $\mathbf{C}^{(n)}$ and $\mathbf{S}^{(m)}$ relies only on matrix products and sums. This eludes the problems of matrix inversion associated with other algorithms.

On the other hand, the algorithm of Park and Schapery (1999) presents more potential sources of instabilities and inaccuracies. First, determination of the retardation times from the relaxation times (or vice versa) requires computing the roots of the polynomial in Eq. (23), which can be a very daunting task (Park and Schapery 1999 even suggested to arbitrarily set $\tau_n = \omega_m$ instead of computing the roots). When considering more than four retardation times (or relaxation times), no closed-form solution to this problem exists, and an iterative algorithm must be used. Furthermore, this kind of problem can become ill-conditioned when roots are, either very close or well-separated. Also, in Park and Schapery (1999) algorithm, the linear equation system (20) needs to be solved. However, matrix $\mathbf{G}$ becomes ill-conditioned when there are numerous retardation (or relaxation) times distributed over many decades. Similar round-off errors will also be encountered with the exact expressions of Baumgaertel and Winter (1989).

We emphasize that our new algorithm leads, by construction, to thermodynamically acceptable solutions. This comes from the fact that, instead of interconverting directly $\mathbf{C}^{(n)}$ into $\mathbf{S}^{(m)}$ (or vice-versa), the interconversion is performed on the internal matrices. Since $\mathbf{L}^{(3*)}$ and $\mathbf{A}^{(3*)}$ are positive-definite matrices, their eigenvalues are positive and lead to positive retardation and relaxation times as well as positive definite matrices $\mathbf{C}^{(n)}$ or $\mathbf{S}^{(m)}$. Such construction prevents any unacceptable temporal oscillations in the target function and hence promotes the algorithm's accuracy.

Finally, our new algorithm provides direct expressions for interconverting tensorial relaxation modulus and creep compliances. The existing methods could potentially be tailored for tensorial expressions, though our preliminary attempts revealed that this task would be awkward and would most likely lead to thermodynamically inadmissible solutions.

7 Conclusion

The main contributions of this study are as follows:

1. A new exact analytical method for the interconversion of linearly viscoelastic material functions has been proposed. This method is applicable for both unidimensional and tridimensional cases, and for any degree of material symmetry. Furthermore, the new method leads to thermodynamically admissible interconversions. These are major improvements over all the existing methods.
2. A rigorous procedure has been developed for evaluating the accuracy and stability of the new algorithm. A broad range of artificial materials was tested. It was found that the new algorithm delivers interconversions of high accuracy and is a few orders of magnitude more accurate than that of Park and Schapery (1999). Finally, the algorithm is quite stable. All of the 126 million interconversions attempted were successfully completed using Matlab 2011.

Considering its stability and accuracy, the authors believe that their algorithm provides a closure to the interconversion of linearly viscoelastic properties expressed as Prony series. Matlab implementations of the algorithms will be provided upon request.

Acknowledgements Fruitful correspondence with W.G. Knauss is gratefully acknowledged. Financial support from Pratt and Whitney Canada, Rolls Royce Canada, Consortium for Research and Innovation in Aerospace in Québec and Natural Sciences and Engineering Research Council of Canada (through Collaborative Research and Development grants) is gratefully acknowledged. Some of the computations were performed on supercomputers financed by the Canadian Foundation for Innovation and hosted by the Fluids Dynamics Laboratory (LADYF) of École Polytechnique de Montréal.

Appendix: Definition of the orthogonal projectors

A fourth-order isotropic tensor can be decomposed using the following orthogonal projectors:

$$\mathbf{J} = \frac{1}{3}\boldsymbol{i} \otimes \boldsymbol{i} \quad \text{and} \quad \mathbf{K} = \mathbf{I} - \mathbf{J} \tag{53}$$

For cubic symmetry,

$$\mathbf{Z} = \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 \otimes \boldsymbol{e}_2 + \boldsymbol{e}_3 \otimes \boldsymbol{e}_3 \otimes \boldsymbol{e}_3 \otimes \boldsymbol{e}_3 \tag{54a}$$

$$\mathbf{K}_a = \mathbf{Z} - \mathbf{J} \quad \text{and} \quad \mathbf{K}_b = \mathbf{K} - \mathbf{K}_a = \mathbf{I} - \mathbf{Z} \tag{54b}$$

When the axis of transverse isotropy is $\boldsymbol{e}_3$, a transversely isotropic tensor can be expressed with the following matrices:

$$\boldsymbol{n} = \{0, 0, 1\}; \qquad \boldsymbol{i}_T = \boldsymbol{i} - \boldsymbol{n} \otimes \boldsymbol{n}; \qquad \mathbf{E}_L = \boldsymbol{n} \otimes \boldsymbol{n} \otimes \boldsymbol{n} \otimes \boldsymbol{n}; \qquad \mathbf{J}_T = \frac{1}{2}\boldsymbol{i}_T \otimes \boldsymbol{i}_T \tag{55a}$$

$$\mathbf{I}_T = \begin{bmatrix} \boldsymbol{i}_T & 0 \\ 0 & \boldsymbol{n} \otimes \boldsymbol{n} \end{bmatrix}; \qquad \mathbf{K}_E = \frac{1}{6}(2\boldsymbol{n} \otimes \boldsymbol{n} - \boldsymbol{i}_T) \otimes (2\boldsymbol{n} \otimes \boldsymbol{n} - \boldsymbol{i}_T) \tag{55b}$$

$$\mathbf{K}_T = \mathbf{I}_T - \mathbf{J}_T; \qquad \mathbf{K}_L = \mathbf{K} - \mathbf{K}_T - \mathbf{K}_E; \tag{55c}$$

$$\mathbf{F} = \frac{\sqrt{2}}{2}(\boldsymbol{i}_T \otimes \boldsymbol{n} \otimes \boldsymbol{n}); \qquad \mathbf{F}^{\mathrm{T}} = \frac{\sqrt{2}}{2}(\boldsymbol{n} \otimes \boldsymbol{n} \otimes \boldsymbol{i}_T) \tag{55d}$$

Appendix A: Interconversion from tensor $\mathbf{S}(t)$ to $\mathbf{C}(t)$

Consider the following anisotropic material:

$$\mathbf{S}^{(0)} = \begin{bmatrix} 210.174 & 48.162 & 2.118 & 0.423 & 0.537 & 0.071 \\ 48.162 & 195.498 & 7.986 & 1.111 & 1.036 & -1.379 \\ 2.118 & 7.986 & 18.481 & -15.804 & -6.611 & 17.307 \\ 0.423 & 1.111 & -15.804 & 228.036 & -67.011 & 33.105 \\ 0.537 & 1.036 & -6.611 & -67.011 & 177.019 & 35.857 \\ 0.071 & -1.379 & 17.307 & 33.105 & 35.857 & 147.184 \end{bmatrix} 10^{-1} \tag{56a}$$

$$\mathbf{S}^{(1)} = \begin{bmatrix} 16.997 & 1.452 & 3.006 & 0.409 & 0.749 & 0.059 \\ 1.452 & 102.517 & -39.342 & 8.715 & -22.003 & 28.447 \\ 3.006 & -39.342 & 32.762 & -3.286 & 1.264 & -14.781 \\ 0.409 & 8.715 & -3.286 & 95.569 & -45.699 & -36.615 \\ 0.749 & -22.003 & 1.264 & -45.699 & 74.399 & 18.321 \\ 0.059 & 28.447 & -14.781 & -36.615 & 18.321 & 87.530 \end{bmatrix} 10^{-1} \tag{56b}$$

with the inverted retardation time

$$\lambda_1 = \{0.043145\} \tag{56c}$$

Following the methodology proposed in Algorithm 4, the internal matrices are

$$\mathbf{A}^{(1)} = \begin{bmatrix} 210.174 & 48.162 & 2.118 & 0.423 & 0.537 & 0.071 \\ 48.162 & 195.498 & 7.986 & 1.111 & 1.036 & -1.379 \\ 2.118 & 7.986 & 18.481 & -15.804 & -6.611 & 17.307 \\ 0.423 & 1.111 & -15.804 & 228.036 & -67.011 & 33.105 \\ 0.537 & 1.036 & -6.611 & -67.011 & 177.019 & 35.857 \\ 0.071 & -1.379 & 17.307 & 33.105 & 35.857 & 147.184 \end{bmatrix} 10^{-1} \tag{57a}$$

$$\mathbf{A}^{(2)} = \begin{bmatrix} 27.080 & 0 & 0 & 0 & 0 & 0 \\ 2.314 & 66.466 & 0 & 0 & 0 & 0 \\ 4.789 & -25.705 & 27.016 & 0 & 0 & 0 \\ 0.651 & 5.634 & -0.002 & 63.962 & 0 & 0 \\ 1.194 & -14.325 & -11.822 & -29.577 & 44.596 & 0 \\ 0.094 & 18.463 & -6.055 & -26.326 & 4.588 & 51.815 \end{bmatrix} 10^{-2} \tag{57b}$$

$$\mathbf{A}^{(3)} = \begin{bmatrix} 0.043145 \end{bmatrix}_6 \tag{57c}$$

The $\mathbf{L}^{(i)}$ are computed as

$$\mathbf{L}^{(1)} = \begin{bmatrix} 50.427 & -12.399 & -0.545 & -0.109 & -0.138 & -0.018 \\ -12.399 & 55.768 & -35.632 & -5.235 & -5.044 & 7.125 \\ -0.545 & -35.632 & 822.119 & 108.166 & 101.444 & -146.050 \\ -0.109 & -5.235 & 108.166 & 67.686 & 37.199 & -37.055 \\ -0.138 & -5.044 & 101.444 & 37.199 & 82.587 & -40.463 \\ -0.018 & 7.125 & -146.050 & -37.055 & -40.463 & 103.375 \end{bmatrix} 10^{-3} \tag{58a}$$

$$\mathbf{L}^{(2)} = \begin{bmatrix} 13.340 & -8.091 & -0.130 & -0.024 & -0.062 & -0.010 \\ -3.862 & 47.970 & -9.461 & -3.732 & -1.923 & 3.692 \\ 40.177 & -270.411 & 218.953 & 77.631 & 38.539 & -75.676 \\ 5.880 & -39.640 & 27.067 & 42.046 & 14.889 & -19.200 \\ 5.894 & -46.634 & 20.092 & 10.019 & 34.974 & -20.966 \\ -7.462 & 65.072 & -40.933 & -38.948 & -13.302 & 53.564 \end{bmatrix} 10^{-3} \tag{58b}$$

$$\mathbf{L}^{(3)} = \begin{bmatrix} 48.69 & -14.78 & 10.61 & 3.98 & 2.29 & -3.87 \\ -14.78 & 161.00 & -71.4 & -28.6 & -17.81 & 33.72 \\ 10.61 & -71.48 & 102.40 & 22.15 & 7.08 & -21.21 \\ 3.98 & -28.69 & 22.15 & 77.33 & 2.68 & -20.18 \\ 2.29 & -17.81 & 7.08 & 2.68 & 58.13 & -6.89 \\ -3.87 & 33.72 & -21.21 & -20.18 & -6.89 & 70.90 \end{bmatrix} 10^{-3} \tag{58c}$$

The eigenvalues of $\mathbf{L}^{(3)}$ as well as matrix $\mathbf{P}$ are computed in Matlab as per singular value decomposition as

$$\rho_n = \{0.23444, 0.07411, 0.06072, 0.05156, 0.05108, 0.04655\} \tag{59a}$$

$$\mathbf{P} = \begin{bmatrix} -10.22 & 4.97 & -4.58 & -8.63 & 9.99 & -98.36 \\ 76.40 & -32.82 & -20.60 & 5.20 & 51.17 & -3.90 \\ -51.19 & 12.07 & -56.09 & 7.74 & 61.87 & 14.14 \\ -24.98 & -79.08 & -17.17 & -50.98 & -14.96 & 2.36 \\ -11.31 & 8.10 & 68.21 & -45.70 & 54.79 & 7.98 \\ 26.19 & 49.34 & -38.23 & -71.77 & -15.08 & 6.31 \end{bmatrix} 10^{-2} \tag{59b}$$

$\mathbf{L}^{(2*)}$ and $\mathbf{L}^{(3*)}$ are computed as

$$\mathbf{L}^{(2*)} = \begin{bmatrix} -7.469 & 3.311 & 1.094 & -1.535 & -2.918 & -12.831 \\ 44.005 & -12.461 & -6.481 & 2.226 & 17.257 & 0.581 \\ -366.357 & 21.573 & -27.054 & -3.450 & 22.011 & 2.123 \\ -61.959 & -24.946 & 2.992 & -14.932 & 1.810 & 0.558 \\ -58.467 & 2.589 & 28.218 & -7.422 & 9.980 & 0.567 \\ 96.696 & 29.479 & -12.969 & -11.651 & -2.308 & 0.415 \end{bmatrix} 10^{-3} \tag{60a}$$

$$\mathbf{L}^{(3*)} = \begin{bmatrix} 0.2344 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0741 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0607 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0516 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0511 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0466 \end{bmatrix} \tag{60b}$$

Finally, the relaxation matrices $\mathbf{C}^{(0)}$ and $\mathbf{C}^{(n)}$ are given as follows:

$$\mathbf{C}^{(0)} = \begin{bmatrix} 46.27 & -9.11 & -10.95 & -1.21 & -2.12 & 1.61 \\ -9.11 & 38.79 & 26.56 & 2.55 & 6.32 & -6.17 \\ -10.95 & 26.56 & 221.48 & 18.14 & 17.07 & -9.11 \\ -1.21 & 2.55 & 18.14 & 38.37 & 18.72 & -4.23 \\ -2.12 & 6.32 & 17.07 & 18.72 & 51.78 & -12.58 \\ 1.61 & -6.17 & -9.11 & -4.23 & -12.58 & 46.26 \end{bmatrix} 10^{-3}$$

$$\mathbf{C}^{(1)} = \begin{bmatrix} 0.2 & -1.4 & 11.7 & 2.0 & 1.9 & -3.1 \\ -1.4 & 8.3 & -68.8 & -11.6 & -11.0 & 18.1 \\ 11.7 & -68.8 & 572.5 & 96.8 & 91.4 & -151.1 \\ 2.0 & -11.6 & 96.8 & 16.4 & 15.5 & -25.6 \\ 1.9 & -11.0 & 91.4 & 15.5 & 14.6 & -24.1 \\ -3.1 & 18.1 & -151.1 & -25.6 & -24.1 & 39.9 \end{bmatrix} 10^{-3};$$

$$\mathbf{C}^{(2)} = \begin{bmatrix} 1.5 & -5.6 & 9.6 & -11.1 & 1.2 & 13.2 \\ -5.6 & 21.0 & -36.3 & 41.9 & -4.4 & -49.6 \\ 9.6 & -36.3 & 62.8 & -72.6 & 7.5 & 85.8 \\ -11.1 & 41.9 & -72.6 & 84.0 & -8.7 & -99.2 \\ 1.2 & -4.4 & 7.5 & -8.7 & 0.9 & 10.3 \\ 13.2 & -49.6 & 85.8 & -99.2 & 10.3 & 117.3 \end{bmatrix} 10^{-4};$$

$$\mathbf{C}^{(3)} = \begin{bmatrix} 0.2 & -1.2 & -4.9 & 0.5 & 5.1 & -2.3 \\ -1.2 & 6.9 & 28.9 & -3.2 & -30.1 & 13.8 \\ -4.9 & 28.9 & 120.5 & -13.3 & -125.7 & 57.8 \\ 0.5 & -3.2 & -13.3 & 1.5 & 13.9 & -6.4 \\ 5.1 & -30.1 & -125.7 & 13.9 & 131.1 & -60.3 \\ -2.3 & 13.8 & 57.8 & -6.4 & -60.3 & 27.7 \end{bmatrix} 10^{-4};$$

$$\mathbf{C}^{(4)} = \begin{bmatrix} 0.46 & -0.66 & 1.03 & 4.44 & 2.21 & 3.47 \\ -0.66 & 0.96 & -1.49 & -6.45 & -3.20 & -5.03 \\ 1.03 & -1.49 & 2.31 & 9.99 & 4.97 & 7.80 \\ 4.44 & -6.45 & 9.99 & 43.24 & 21.49 & 33.74 \\ 2.21 & -3.20 & 4.97 & 21.49 & 10.68 & 16.77 \\ 3.47 & -5.03 & 7.80 & 33.74 & 16.77 & 26.33 \end{bmatrix} 10^{-4}$$

$$\mathbf{C}^{(5)} = \begin{bmatrix} 1.67 & -9.86 & -12.57 & -1.03 & -5.70 & 1.32 \\ -9.86 & 58.31 & 74.37 & 6.11 & 33.72 & -7.80 \\ -12.57 & 74.37 & 94.86 & 7.80 & 43.01 & -9.95 \\ -1.03 & 6.11 & 7.80 & 0.64 & 3.54 & -0.82 \\ -5.70 & 33.72 & 43.01 & 3.54 & 19.50 & -4.51 \\ 1.32 & -7.80 & -9.95 & -0.82 & -4.51 & 1.04 \end{bmatrix} 10^{-4}$$

$$\mathbf{C}^{(6)} = \begin{bmatrix} 353.7 & -16.0 & -58.5 & -15.4 & -15.6 & -11.4 \\ -16.0 & 0.73 & 2.65 & 0.70 & 0.71 & 0.52 \\ -58.5 & 2.65 & 9.69 & 2.55 & 2.58 & 1.89 \\ -15.4 & 0.70 & 2.55 & 0.67 & 0.68 & 0.50 \\ -15.6 & 0.71 & 2.58 & 0.68 & 0.69 & 0.51 \\ -11.4 & 0.52 & 1.89 & 0.50 & 0.51 & 0.37 \end{bmatrix} 10^{-5}$$

Note that the matrices' elements have not been written with their full precision due to space restrictions and terms smaller than 10^{-13}, in absolute value, were set to 0.

References

Anderssen, R.S., Davies, A.R., de Hoog, F.R.: On the Volterra integral equation relating creep and relaxation. Inverse Probl. **24**(3), 035,009 (2008)

Baumgaertel, M., Winter, H.H.: Determination of discrete relaxation and retardation time spectra from dynamic mechanical data. Rheol. Acta **28**, 511–519 (1989)

Biot, M.A.: Theory of stress-strain relations in anisotropic viscoelasticity and relaxation phenomena. J. Appl. Phys. **25**(11), 1385–1391 (1954)

Bouleau, N.: Interprétation probabiliste de la viscoélasticité linéaire. Mech. Res. Commun. **19**, 15–20 (1992)

Bouleau, N.: Visco-elasticité et processus de Lévy. Potential Anal. **11**, 289–302 (1999)

Cost, T.L., Becker, E.B.: A multidata method of approximate Laplace transform inversion. Int. J. Numer. Methods Eng. **2**, 207–219 (1970)

Crochon, T., Schönherr, T., Li, C., Lévesque, M.: On finite-element implementation strategies of Schapery-type constitutive theories. Mech. Time-Depend. Mater. **14**, 359–387 (2010)

Fernandez, P., Rodriguez, D., Lamela, M.J., Fernandez-Canteli, A.: Study of the interconversion between viscoelastic behavior functions of PMMA. Mech. Time-Depend. Mater. **15**(2), 169–180 (2011)

Hopkins, I.L., Hamming, R.W.: On creep and relaxation. J. Appl. Phys. **28**(8), 906–909 (1957)

Knauss, W.G., Zhao, J.: Improved relaxation time coverage in ramp-strain histories. Mech. Time-Depend. Mater. **11**(3–4), 199–216 (2007)

Knoff, W.F., Hopkins, I.L.: An improved numerical interconversion for creep compliance and relaxation modulus. J. Appl. Polym. Sci. **16**, 2963–2972 (1972)

Lévesque, M.: Modélisation du comportement mécanique de matériaux composites viscoélastiques non linéaires par une approche d'homogénéisation. Ph.D. thesis, École Nationale Supérieure d'Arts et Métiers, Paris, in English (2004). Download at pastel.paristech.org/archive/00001237/

Lévesque, M., Derrien, K., Baptiste, D., Gilchrist, M.D., Bouleau, N.: Numerical inversion of the Laplace–Carson transform applied to homogenization of randomly reinforced linear viscoelastic media. Comput. Mech. **40**(4), 771–789 (2007)

Lévesque, M., Derrien, K., Baptiste, D., Gilchrist, M.D.: On the development and parameter identification of Schapery-type constitutive theories. Mech. Time-Depend. Mater. **12**, 95–127 (2008)

Nikonov, A., Davies, A.R., Emri, I.: The determination of creep and relaxation functions from a single experiment. Rheol. Acta **49**(6), 1193–1211 (2005)

Park, S.W., Schapery, R.A.: Methods of interconversion between linear viscoelastic material functions. Part I. A numerical method based on Prony series. Int. J. Solids Struct. **36**, 1653–1675 (1999)

Schapery, R.A.: Further Development of a Thermodynamic Constitutive Theory: Stress Formulation. Purdue University, School of Aeronautics (1969)

Sorvari, J., Malinen, M.: Numerical interconversion between linear viscoelastic material functions with regularization. Int. J. Solids Struct. **44**, 1291–1303 (2006)

Trefethen, L.N., Bau, D.: Numerical Linear Algebra. SIAM, Philadelphia (1997)

Tschoegl, N.W.: The Phenomenological Theory of Linear Viscoelastic Behavior. Springer, Berlin (1989)

Yaoting, Z., Lu, S., Huilin, X.: L-curve based Tikhonov's regularization method for determining relaxation modulus from creep test. J. Appl. Mech. **78**, 031002 (2011)