Internship at the Institute for Mechanics, TU Berlin 03.05.12 - 13.07.12

Coupled Temperature-Deformation Computations for Viscoelastic Materials with Fractional Derivatives

Aditya Desai

Supervisors - B. Emek Abali, *M. Sc.* & Prof. Dr. rer. nat. Wolfgang H. Müller

1 Introduction

Viscoelastic materials are those which display a combination of elastic and viscous behaviour. The stresses and their rate of change in such materials are dependent on strains as well as strain rates. Unlike elastic materials, a viscoelastic substance loses energy when a load is applied, then removed. Hysteresis is observed in the stress-strain curve, with the area of the loop being equal to the energy lost during the loading cycle. When a stress is applied to a viscoelastic material such as a polymer, parts of the long polymer chain change position. This movement or rearrangement is due to the viscous effects. Polymers remain a solid material even when these parts of their chains are rearranging in order to accompany the stress, and as this occurs, it creates a back stress in the material. When the back stress is the same magnitude as the applied stress, the body reaches mechanical equilibrium. When the original stress is taken away, the accumulated back stresses will cause the polymer to return to its original form.

Viscoelastic behaviour is observed in a variety of different materials like rubbers, glasses and synthetic polymers (particularly polymer melts). The aim of this work is to implement a generic constitutive model for viscoelastic materials to develop a finite element code which can be used for simulating the rheological response of such materials.

2 Formulation

The objective is to find the displacement and temperature fields for every time slice t, in a continuum body $\mathcal{B}_0$ expressed with a Cartesian coordinate system in the material frame, *i.e.* initial framework.

The coordinate system labels particles X_i and this does not change in time. The configuration (relationship between particles) is defined with the mesh and form functions, *i.e.* it is known *a priori*. The particles have their positions initially at X_i and after deformation at x_i, the difference between them is the sought displacement field, $u_i = x_i - X_i$. Positions of particles are changed, therefore also the configuration is changed and the relation between the reference and actual state can be measured:

$$C_{jk} = \frac{\partial x_i}{\partial X_j}\frac{\partial x_i}{\partial X_k} \ , \ C_{jk} = C_{kj} \ , \tag{1}$$

which is known as the right Cauchy-Green deformation tensor. Here and henceforth we use the Einstein summation.

To be able to express in a variational form, we need to get an invariant term which does not change under coordinate transformation. This is indeed the case in a length $\mathrm{d}S$ between neighbouring particles $X_i^1 - X_i^2 = \mathrm{d}X_i$

before the loading, which changes to $\mathrm{d}s$ afterwards. We can express the length in terms of $\mathrm{d}X_i$ as:

$$
\begin{aligned}
(\mathrm{d}S)^2 &= \frac{\partial S}{\partial X_i}\,\mathrm{d}X_i \frac{\partial S}{\partial X_j}\,\mathrm{d}X_j = \delta_{ij}\,\mathrm{d}X_i\,\mathrm{d}X_j \ , \\
(\mathrm{d}s)^2 &= \frac{\partial s}{\partial X_i}\,\mathrm{d}X_i \frac{\partial s}{\partial X_j}\,\mathrm{d}X_j = C_{ij}\,\mathrm{d}X_i\,\mathrm{d}X_j \ ,
\end{aligned} \tag{2}
$$

where the change of length is a measure of strains:

$$
(\mathrm{d}l)^2 = (\mathrm{d}s)^2 - (\mathrm{d}S)^2 = (C_{ij} - \delta_{ij})\,\mathrm{d}X_i\,\mathrm{d}X_j \ . \tag{3}
$$

The numerical value of $\mathrm{d}l$ remains the same, independent of the chosen frame. We have chosen the Cartesian coordinates in the initial configuration. This leads to the idea of a coordinate transformation named as deformation gradient:

$$
F_{ij} = \frac{\partial x_i}{\partial X_j} \ , \ \left(F^{-1}\right)_{ji} = \frac{\partial X_j}{\partial x_i} \ , \ F_{ij}\left(F^{-1}\right)_{jk} = \delta_{ik} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}
$$

Since we can write the displacements in terms of X_i:

$$
\begin{aligned}
u_i(X_i) &= x_i(X_i) - X_i \\
\frac{\partial u_i}{\partial X_j} &= F_{ij} - \delta_{ij} \ , \\
(\mathrm{d}l)^2 = \big(C_{ij}(X_k) - \delta_{ij}\big)\,\mathrm{d}X_i\,\mathrm{d}X_j &= \left(\frac{\partial x_i}{\partial X_j}\frac{\partial x_i}{\partial X_k} - \delta_{jk}\right)\mathrm{d}X_j\,\mathrm{d}X_k \ .
\end{aligned}
$$

This leads to the aforementioned Cauchy-Green deformation tensor:

$$
C_{jk} = \frac{\partial x_i}{\partial X_j}\frac{\partial x_i}{\partial X_k} = \left(\frac{\partial u_i}{\partial X_j} + \delta_{ij}\right)\left(\frac{\partial u_i}{\partial X_k} + \delta_{ik}\right) = \frac{\partial u_i}{\partial X_j}\frac{\partial u_i}{\partial X_k} + \frac{\partial u_k}{\partial X_j} + \frac{\partial u_j}{\partial X_k} + \delta_{jk} = 2E_{jk} + \delta_{jk} \ ,
$$

which is employed to get the Green-Lagrange strains E_{jk}, quadratic in displacement gradients.

As we have chosen to work in the material framework, the balance equations need to be transformed from the actual state to the reference state where the framework is described. A volume element $\mathrm{d}V$ and an area element $\mathrm{d}a_i$ is transformed as [2]:

$$
\mathrm{d}V = J\,\mathrm{d}V_0 \ , \tag{5}
$$

$$
\mathrm{d}a_i = n_i\,\mathrm{d}a = \left(\frac{\partial x_i}{\partial X_r}\right)^{-1} JN_r\,\mathrm{d}A_0 = \left(F^{-1}\right)_{ri} JN_r\,\mathrm{d}A_0 \ . \tag{6}
$$

The displacement field can be obtained by solving the balance of linear momentum. The balance equation in the actual frame is:

$$
\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{B}} \rho v_i\,\mathrm{d}V = \int_{\partial\mathcal{B}} n_j\sigma_{ji}\,\mathrm{d}A + \int_{\mathcal{B}} \rho f_i\,\mathrm{d}V. \tag{7}
$$

This equation can be transformed to the material frame by using the Eqs.(5) and (**??**):

$$
\int_{\mathcal{B}_0} \frac{\rho_0}{J}\frac{\partial v_i}{\partial t} J\,\mathrm{d}V_0 = \int_{\partial\mathcal{B}_0} \sigma_{ji}(F^{-1})_{rj} JN_r\,\mathrm{d}A_0 + \int_{\mathcal{B}_0} \frac{\rho_0}{J} f_i J\,\mathrm{d}V_0 \ , \tag{8}
$$

The conservation of mass is implicitly satisfied: $\mathrm{d}m = \rho_0\,\mathrm{d}V_0 = \rho\,\mathrm{d}V = \rho J\,\mathrm{d}V_0$. By defining

$$
P_{ri} = \sigma_{ji}(F^{-1})_{rj} J = \sigma_{ji}(F_{jr})^{-1} J \tag{9}
$$

we will spare some notation on the following. The material frame does not change in time, *i.e.*, $\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{d}V_0 = 0$, thus we can build the residual and transform it via Gauß -Ostrogradsky theorem into a form:

$$\int_{\mathcal{B}_0} \left(\rho_0 \frac{\partial v_i}{\partial t} - \frac{\partial}{\partial X_r}(P_{ri}) - \rho_0 f_i \right) \mathrm{d}V_0 = 0 \tag{10}$$

The solution of the momentum equation requires discretisation in time and space. Time discretisation is done using the finite difference method while, space discretisation is done by weighting with a test function and summing over all the elements. The time discretisation for the instationary term:

$$\frac{\partial v_i}{\partial t} = \frac{\partial^2 u_i}{\partial t^2} = \frac{1}{\Delta t \Delta t}\left(u_i - 2u_i^0 + u_i^{00} \right) \tag{11}$$

will be implemented and moreover be extended. From Eq. (10) and Eq. (11), we have:

$$\int_{\mathcal{B}_0} \left(\frac{1}{\Delta t \Delta t} \rho_0 (u_i - 2u_i^0 + u_i^{00}) - \frac{\partial}{\partial X^r}\left(\sigma_{ji}(F^{-1})_{rj} J \right) - \rho f_i \right) \mathrm{d}V_0 = 0 \ . \tag{12}$$

This can be weighted with test functions δu_i:

$$\int_{\mathcal{B}_0} \left(\frac{1}{\Delta t \Delta t} \rho_0 (u_i - 2u_i^0 + u_i^{00}) - \frac{\partial}{\partial X_r}\left(\sigma_{ji}(F^{-1})_{rj} J \right) - \rho_0 f_i \right) \delta u_i \, \mathrm{d}V_0 = 0 \ , \tag{13}$$

to get the minimization problem to be solved. Out of these terms, the body force, f_i is generally known and is often neglected, as it is small in magnitude. The equation can thus be solved for the displacement field, u_i, if σ_{ij} is defined.

The stress derivative may not give the correct results as the elements of the mesh are linear CG elements. Thus, the stress term can be transformed using the Gauß -Ostrogradsky theorem, which gives:

$$\int_{\mathcal{B}_0} \left(\frac{1}{\Delta t \Delta t} \rho_0 (u_i - 2u_i^0 + u_i^{00}) \delta u_i + \sigma_{ji}(F^{-1})_{rj} J \frac{\partial \delta u_i}{\partial X^r} - \rho f_i \delta u_i \right) \mathrm{d}V_0 - \oint_{\partial B_0} \sigma_{ji}(F^{-1})_{rj} J \delta u_i N_r \, \mathrm{d}A_0 = 0 \ . \tag{14}$$

The solution of the balance of momentum can give the required displacement field in the body for the isothermal deformation of the body. A more general solution, however, needs to account for the coupling of thermal and mechanical effects. The temperature field is thus, also required. This can be obtained from the balance of energy. The balance of total energy, consists of the internal energy density U and the kinetic energy density K, in the actual frame:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int_B (U + K) \, \mathrm{d}V &= - \int_{\partial B} q_i n_i \, \mathrm{d}A + \int_{\partial B} \sigma_{ji} v_i n_j \, \mathrm{d}A + \int_B \rho r \, \mathrm{d}V + \int_B \rho f_i v_i \, \mathrm{d}V \ , \\ \frac{\mathrm{d}}{\mathrm{d}t} \int_B (\rho u + \frac{\rho}{2} v_i v_i) \, \mathrm{d}V &= - \int_{\partial B} q_i n_i \, \mathrm{d}A + \int_{\partial B} \sigma_{ji} v_i n_j \, \mathrm{d}A + \int_B \rho r \, \mathrm{d}V + \int_B \rho f_i v_i \, \mathrm{d}V \ , \end{aligned} \tag{15}$$

changed into the material reference frame reads:

$$\begin{aligned} \frac{d}{\mathrm{d}t} \int_{B_0} (\rho_0 u + \frac{\rho_0}{2} v_i v_i) \, \mathrm{d}V_0 = - \int_{\partial B_0} q_i \left(\frac{\partial x_i}{\partial X_r} \right)^{-1} J N_r \, \mathrm{d}A_0 + \int_{\partial B_0} \sigma_{ji} v_i \left(\frac{\partial x_i}{\partial X_r} \right)^{-1} J N_r \, \mathrm{d}A_0 \\ + \int_{B_0} \rho_0 r \, \mathrm{d}V_0 + \int_{B_0} \rho_0 f_i v_i \, \mathrm{d}V_0 \ . \end{aligned}$$

By renaming:

$$P_{ri} = \sigma_{ji}(F^{-1})_{rj} J = \sigma_{ji}(F_{jr})^{-1} J \ , \ Q_r = q_i \left(\frac{\partial x_i}{\partial X_r} \right)^{-1} J \ ,$$

and using the Gauß- Ostrograskiy theorem the balance of energy reads:

$$\int_{B_0} \left(\rho_0 \frac{\partial u}{\partial t} + \rho_0 v_i \frac{\partial v_i}{\partial t} + \frac{\partial Q_r}{\partial X_r} - \frac{\partial}{\partial X_r}(P_{ri} v_i) - \rho_0 r - \rho_0 f_i v_i \right) dV_0 = 0 \tag{16}$$

We assume that there is no radiative heat input to the system and that the body forces are negligible. Additionally using the balance of momentum as: Eq.(16)-$v_i\times$ Eq.(10), gives:

$$\int_{B_0}\left(\rho_0\frac{\partial u}{\partial t}+\frac{\partial Q_r}{\partial X_r}-P_{ri}\frac{\partial v_i}{\partial X_r}\right)dV_0=0\ . \tag{17}$$

This is the balance of internal energy energy, *i.e.* the 1$^{\text{st}}$ Law, and is used by contracting with the test function for temperature δT, as:

$$\int_{B_0}\left(\rho_0\frac{\partial u}{\partial t}+\frac{\partial Q_r}{\partial X_r}-P_{ri}\frac{\partial v_i}{\partial X_r}\right)\delta T dV_0=0\ . \tag{18}$$

3 Constitutive Relations

For the equations of balance of linear momentum and energy to be solved, three different constitutive relation are required, *i.e.* explicit relations in terms of the displacements and temperatures for the stresses, σ_{ij}, specific internal energy, u, and the heat transfer in the body, q_i.

3.1 Stress relation

A viscoelastic material (like polycarbonate) also has strain rate dependency in its model. The constitutive relation governing the mechanical response is given by the fractional Zener model [3]:

$$\sigma_{ij}+\tau_0^\alpha\frac{\mathrm{d}^\alpha\sigma_{ij}}{\mathrm{d}t^{\,\alpha}}=G_e\left(\varepsilon_{ij}+\tau_0^\alpha\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\,\alpha}}\right)+G_0\tau_0^\beta\frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^{\,\beta}} \tag{19}$$

An important thing to note here is that the orders of differentiation α and β need not be integers, but any positive reals. The fractional Zener model is general in nature and reduces to other models under special values of α and β.

Case 1: $\alpha=\beta=0$:
This reduces the Eq.(19) to:

$$\sigma_{ij}=\left(G_e+\frac{G_0}{2}\right)\varepsilon_{ij}\ .$$

There is no rate dependence and the model is reduced to one similar to an isotropic elastic model.

Case 2: $\alpha=0\ ,\ \beta=1$:
This reduces the Eq.(19)to:

$$\sigma_{ij}=G_e\varepsilon_{ij}+\frac{G_0\tau_0}{2}\frac{\mathrm{d}\varepsilon_{ij}}{\mathrm{d}t}\ .$$

This situation represents the 'regular' Zener model.

Case 3: $\alpha=1\ ,\ \beta=0$:
This reduces the Eq.(19) to:

$$\sigma_{ij}+\tau_0\frac{\mathrm{d}\sigma_{ij}}{\mathrm{d}t}=(G_e+G_0)\varepsilon_{ij}+G_e\tau_0\frac{\mathrm{d}\varepsilon_{ij}}{\mathrm{d}t}\ .$$

This situation represents the Maxwell model.

The values of all the parameters in the equation(viz. $G_0, G_e, \tau_0, \alpha, \beta$) can be obtained through experiments. The experiments to determine the model parameters can be performed on a rotary viscometer. This consists of two plates, within which the sample to be tested is held. The upper plate can rotate while, the lower plate is held stationary. The rotation of the upper plate induces a shear strain in the system. The force required to hold the lower plate stationary is measured, from which the shear stress can be calculated.

There are two geometrical configurations of the two plates that are commonly used. They are the cone-plate and plate-plate(or parallel plate) configurations. The experimental data used here has been obtained by using the plate-plate geometry. Hence, we have used a rectangular box in the simulation.

Figure 1: Rotary viscometer from Lab at the Institute for Mechanics, TU Berlin

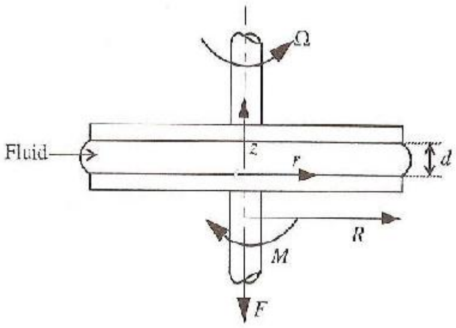

Figure 2: Schematic diagram of parallel plate viscometer

The upper plate is subjected to an oscillatory motion in order to study the frequency response of the material, *i.e.* to calculate the complex moduli of the material.

From the fractional ZENER model, the storage and loss moduli are given by [3]:

$$G' = G_e + G_0 \frac{y^\beta \left\{\cos(\beta\pi/2) + y^\alpha \cos\left[(\beta-\alpha)\pi/2\right]\right\}}{1 + 2y^\alpha \cos(\alpha\pi/2) + y^{2\alpha}} \ , \tag{20}$$

$$G'' = G_0 \frac{y^\beta \left\{\sin(\beta\pi/2) + y^\alpha \sin\left[(\beta-\alpha)\pi/2\right]\right\}}{1 + 2y^\alpha \cos(\alpha\pi/2) + y^{2\alpha}} \ , \tag{21}$$

$$y = \omega\tau_0$$

The rotary viscometer returns the values of the storage and loss moduli for different frequencies. This can then be fit to the above expressions, to get appropriate values of the model parameters.

Fractional Derivatives

The process of repeated differentiation and integration is well known and is commonly represented by the notation $\frac{\mathrm{d}^n}{\mathrm{d}t^{\,n}}$ and $\int\ ...\int \mathrm{d}t_{\,1}....\mathrm{d}t_{\,n}$.A logical extension of these operators is the case where n is not required to be a positive integer but can be any real number[1]. Only fractional differentiation is considered here. The

[1] In a letter dated September 30th, 1695 L'HOPITAL wrote to LEIBNIZ asking him about a particular notation he had used in his publications for the nth-derivative of the linear function $f(x) = x$, $\frac{D^n x}{D x^n}$. L'HOPITAL posed the question to Leibniz, what would the result be if n = 1/2. Leibniz's response: "An apparent paradox, from which one day useful consequences will be drawn." In these words fractional calculus was born.

GRÜNWALD-LETNIKOV definition of the fractional derivative is [7]:

$$\frac{d^\alpha f}{\mathrm{d}t^{\,\alpha}} = \lim_{h\to 0}\frac{1}{h^\alpha}\sum_{m=0}^{\frac{t}{h}}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\,\Gamma|_{(\alpha-m+1)}}\;f|_{(t-mh)} \tag{22}$$

where h is the time step and

$$\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}\,\mathrm{d}t \quad .$$

The Gamma function can be thought of as a generalization of the factorial to all reals. Here we shall use this definition to discretise the fractional derivatives using a finite time step, Δt. Note that there have been a number of different interpretations given in literature to the idea of the fractional derivative. Although mathematicians prefer to use the RIEMANN-LOUVILLE definition of the fractional derivative, we have used the GRÜNWALD-LETNIKOV definition as it is easier to implement in numerical calculations. The two definitions can, however, be proved to be equivalent[7].

Applying this definition to Eq. (19):

$$\sigma_{ij} + \tau_0^\alpha\sum_{m=0}^{t/\Delta t}\frac{1}{\Delta t^\alpha}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\,\Gamma|_{(\alpha-m+1)}}\;\sigma_{ij}|_{(t-m\Delta t)} = G_e\left(\varepsilon_{ij}+\tau_0^\alpha\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\,\alpha}}\right) + G_0\tau_0^\beta\frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^{\,\beta}}\;,$$

$$\sigma_{ij} + \frac{\tau_0^\alpha\sigma_{ij}}{\Delta t^\alpha} + \tau_0^\alpha\sum_{m=1}^{t/\Delta t}\frac{1}{\Delta t^\alpha}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\,\Gamma|_{(\alpha-m+1)}}\;\sigma_{ij}|_{(t-m\Delta t)} = G_e\left(\varepsilon_{ij}+\tau_0^\alpha\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\,\alpha}}\right) + G_0\tau_0^\beta\frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^{\,\beta}}\;,$$

$$\sigma_{ij}(1+\frac{\tau_0^\alpha}{\Delta t^\alpha}) = -\tau_0^\alpha\sum_{m=1}^{t/\Delta t}\frac{1}{\Delta t^\alpha}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\,\Gamma|_{(\alpha-m+1)}}\;\sigma_{ij}|_{(t-m\Delta t)} + G_e\left(\varepsilon_{ij}+\tau_0^\alpha\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\,\alpha}}\right) + G_0\tau_0^\beta\frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^{\,\beta}}\;,$$

$$\sigma_{ij} = \frac{1}{(1+\frac{\tau_0^\alpha}{\Delta t^\alpha})}\left(-\tau_0^\alpha\sum_{m=1}^{t/\Delta t}\frac{1}{\Delta t^\alpha}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\,\Gamma|_{(\alpha-m+1)}}\;\sigma_{ij}|_{(t-m\Delta t)} + G_e\left(\varepsilon_{ij}+\tau_0^\alpha\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\,\alpha}}\right) + G_0\tau_0^\beta\frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^{\,\beta}}\right)\;, \tag{23}$$

where:

$$\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^\alpha} = \frac{1}{\Delta t^\alpha}\sum_{m=0}^{t/\Delta t}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\,\Gamma|_{(\alpha-m+1)}}\;\varepsilon_{ij}|_{(t-m\Delta t)}\;, \tag{24}$$

$$\frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^\beta} = \frac{1}{\Delta t^\beta}\sum_{m=1}^{t/\Delta t}(-1)^m\frac{\Gamma|_{(\beta+1)}}{m!\,\Gamma|_{(\beta-m+1)}}\;\varepsilon_{ij}|_{(t-m\Delta t)}\;. \tag{25}$$

3.2 Internal energy relation

In order to solve the Eq. (18) we need an explicit expression for the specific internal energy, u. One possible way to derive this needs an assumption: The internal energy is fully recoverable. Then the phenomenological relation:

$$\mathrm{d}U = \mathrm{d}W + \mathrm{d}Q\;, \tag{26}$$

defines the change of the internal energy density, $\mathrm{d}U$, as being due to the work done on the system, $\mathrm{d}W$, and caused by the heat input to the system $\mathrm{d}Q$. The variation of the internal energy density of a particle can be driven either by mechanical $\mathrm{d}W$ or by thermal $\mathrm{d}Q$ work done on it, both of which are recoverable. The heat energy density,

$$\mathrm{d}Q = T\,\mathrm{d}S\;,$$

where S is the entropy density, and the mechanical energy density,

$$\mathrm{d}W = \sigma_{ij}^R\,\mathrm{d}\varepsilon_{ij}\;,$$

where σ_{ij}^R is the recoverable (*i.e.* elastic) stress in the body. Thus, Eq. (26) reads:

$$\mathrm{d}U = \sigma_{ij}^R \, \mathrm{d}\varepsilon_{ij} + T \, \mathrm{d}S \tag{27}$$

Writing (27) in rate form by using $U = \rho_0 u$,

$$\rho_0 \dot{u} = \sigma_{ij}^R \dot{\varepsilon_{ij}} + T\dot{S} \tag{28}$$

If we assume that ε_{ij} and T are state variables and that obtaining them is the goal of the calculations then, the entropy and stress in the body are functions of the independent variables in the system, ε_{ij} (or indirectly u_i) and T [6].

$$\mathrm{d}\sigma_{ij} = \left.\frac{\partial \sigma_{ij}^R}{\partial \varepsilon_{kl}}\right|_T \mathrm{d}\varepsilon_{kl} + \left.\frac{\partial \sigma_{ij}^R}{\partial T}\right|_{\varepsilon_{ij}} \mathrm{d}T$$

$$\mathrm{d}S = \left.\frac{\partial S}{\partial \varepsilon_{ij}}\right|_T \mathrm{d}\varepsilon_{ij} + \left.\frac{\partial S}{\partial T}\right|_{\varepsilon_{ij}} \mathrm{d}T \tag{29}$$

The four derivatives obtained above have a physical significance. The term $\left.\frac{\partial \sigma_{ij}^R}{\partial \varepsilon_{ij}}\right|_T$ represents the stresses caused due to strains, *i.e.* stiffness of the material and $\left.\frac{\partial \sigma_{ij}^R}{\partial T}\right|_{\varepsilon_{ij}}$ represents the stresses caused by changes in temperature known as the thermal pressure. The term $\left.\frac{\partial S}{\partial \varepsilon_{ij}}\right|_T$ is called the heat of deformation. It represents the heat developed in a body as a result of an isothermal deformation. The term $\left.\frac{\partial S}{\partial T}\right|_{\varepsilon_{ij}}$ represents the change on entropy caused by direct heating. This quantity times the temperature is the heat capacity per unit volume at constant strain and can be measured in a body at constant strain by varying the temperature and measuring the heat:

$$\mathrm{d}Q = C_\varepsilon \, \mathrm{d}T$$

$$T \, \mathrm{d}S = T \left.\frac{\partial S}{\partial T}\right|_{\varepsilon_{ij}} \mathrm{d}T.$$

Thus,

$$C_\varepsilon = T \left.\frac{\partial S}{\partial T}\right|_{\varepsilon_{ij}}$$

The specific heat capacity of a material is a property which is more commonly recorded. The specific heat capacity at constant strain, c_ε is related to the heat capacity per unit volume by the relation:

$$C_\varepsilon = \rho_0 c_\varepsilon \ .$$

If we want to find the coefficients,

$$C_{ijkl} = \left.\frac{\partial \sigma_{ij}^R}{\partial \varepsilon_{kl}}\right|_T \ ,$$

$$p_{ij} = \left.\frac{\partial \sigma_{ij}^R}{\partial T}\right|_{\varepsilon_{ij}} \ ,$$

for a isotropic body, therefore:

$$C_{ijkl} = \lambda \delta_{ij}\delta_{kl} + \mu \delta_{ik}\delta_{jl} + \mu \delta_{il}\delta_{jk}$$

$$p_{ij} = p\delta_{ij} \ ,$$

we need to perform a test on constant temperature where we variate the strains and measure the stresses (rotational viscometer) and we need to perform another experiment by clamping the sample and holding the strains fixed and variate the temperature so that the matter tries to expand and creates a pressure. Although it is conceptually simple to understand, the thermal pressure, p, is not a commonly recorded material property.

A similar material property which is used often is the coefficient of thermal expansion, α. They can be related as follows:

$$
\begin{aligned}
\frac{\partial \sigma^R_{ij}}{\partial T} &= \frac{\partial \sigma^R_{ij}}{\partial \varepsilon_{kl}} \frac{\partial \varepsilon_{kl}}{\partial T} \\
&= C_{ijkl}\ \alpha \delta_{kl} \\
&= \left(\lambda \delta_{ij}\delta_{kl} + \mu \delta_{ik}\delta_{jl} + \mu \delta_{il}\delta_{jk}\right) \alpha \delta_{kl} \\
&= (3\lambda + 2\mu)\, \alpha \delta_{ij} \\
&= 2\mu\alpha\delta_{ij} \qquad \text{For small strains, we consider that there is no change on volume and thus, } \lambda = 0 \\
&= G_e \alpha \ . \qquad (30)
\end{aligned}
$$

The heat of deformation can be obtained indirectly by a deformation calorimeter [1] where the temperature is kept constant by supplying heat to the system during deformation. However, the measurement is really difficult and moreover unnecessary. To be able to see this we introduce the HELMHOLTZ free energy density as:

$$\Psi = U - TS \ . \qquad (31)$$

Differentiating, we get:

$$\mathrm{d}\Psi = \mathrm{d}U - S\,\mathrm{d}T - T\,\mathrm{d}S \ . \qquad (32)$$

Combining with Eq. (27):

$$\mathrm{d}\Psi = \sigma^R_{ij}\,\mathrm{d}\varepsilon_{ij} - S\,\mathrm{d}T \ . \qquad (33)$$

Since ε_{ij} and T are the only variables needed for acquiring the state of the system,

$$\frac{\partial \Psi}{\partial \varepsilon_{ij}} = \sigma^R_{ij} \ \text{ and } \ \frac{\partial \Psi}{\partial T} = -S \qquad (34)$$

In order for Ψ to be a perfect differential: $\dfrac{\partial^2 \Psi}{\partial T \partial \varepsilon_{ij}} = \dfrac{\partial^2 \Psi}{\partial \varepsilon_{ij} \partial T}$. Hence:

$$\left.\frac{\partial \sigma^R_{ij}}{\partial T}\right|_{\varepsilon_{ij}} = -\left.\frac{\partial S}{\partial \varepsilon_{ij}}\right|_{T} \qquad (35)$$

This relation, thus, implies that the coefficients of the thermal stress tensor are the same as the coefficients of the heat of deformation. Finally, from Eq. (28) and Eq. (29):

$$\dot{S} = \frac{\mathrm{d}S}{\mathrm{d}t} = -G_e \alpha \delta_{ij} \dot{\varepsilon_{ij}} + \frac{C_\varepsilon}{T} \dot{T} \ , \qquad (36)$$

the constitutive relation for the internal energy reads:

$$\rho_0 \dot{u} = \sigma^R_{ij} \dot{\varepsilon_{ij}} - T G_e \alpha \dot{\varepsilon_{ii}} + C_\varepsilon \dot{T} \qquad (37)$$

3.3 Heat flux relation

For heat flux we can use the FOURIER law for isotropic materials in the material framework:

$$Q_r = -\kappa \delta_{rj} \frac{\partial T}{\partial X_j} \ . \qquad (38)$$

4 Variational form

After having obtained the constitutive relations for stress, internal energy and the heat flux, the Eq. (18) reads:

$$\int_{B_0} \left(\sigma^R_{ij} \dot{\varepsilon_{ij}} - T G_e \alpha \dot{\varepsilon_{ii}} + C_\varepsilon \dot{T} - \kappa \delta_{rj} \frac{\partial^2 T}{\partial X_r \partial X_j} - P_{ri} \frac{\partial v_i}{\partial X_r} \right) dV_0 = 0 \ , \qquad (39)$$

where κ is assumed to be uniform (*i.e.* constant in X_i). We can contract with appropriate test functions and employ an integration by parts to reach an integral equation with only first-order differentials:

$$\int_{B_0}\left(\sigma^R_{ij}\dot{\varepsilon_{ij}}\delta T - TG_e\alpha\dot{\varepsilon_{ii}}\delta T + C_\varepsilon\dot{T}\delta T + \kappa\delta_{rj}\frac{\partial T}{\partial X_j}\frac{\partial \delta T}{\partial X_r} - P_{ri}\frac{\partial v_i}{\partial X_r}\delta T\right)dV_0 - \oint_{\partial B_0}\kappa\delta_{rj}\frac{\partial T}{\partial X_r}\delta T N_r dA_0 = 0. \tag{40}$$

The strain and temperature rates can be discretised in time as follows:

$$\begin{aligned}\dot{\varepsilon_{ij}} &= \frac{1}{2\Delta t}((u_i - u_i^0)_{,j} + (u_j - u_j^0)_{,i}))\ ,\\ \dot{T} &= \frac{T - T^0}{\Delta t}\end{aligned} \tag{41}$$

We assume that the strains are small in magnitude as compared to the original lengths. The strain tensor can, thus, be defined in terms of the displacements as:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right)$$

Also, $F_{jr} \approx \delta_{jr}$ and $J \approx 1$. Thus, $P_{ji} = \sigma_{ji}$.
Also note here that the specific heat capacity at constant strain need not be equal to the specific heat capacity at constant volume (commonly used). But they are equal in this as we assume no change in volume, due to small strains.

The form for the balance of energy has two different types of stress terms: σ^R_{ij} and σ_{ij}, which are the recoverable and total stress respectively. The total stress includes contributions from the elastic (recoverable) part and the parts which are dependent on the fractional derivatives of strain:

$$\begin{aligned}\sigma_{ij} &= \sigma^R_{ij} + \sigma^{\text{frac}}_{ij}\ ,\\ \sigma^R_{ij} &= C_{ijkl}\varepsilon_{kl} = \lambda\delta_{ij}\varepsilon_{kk} + 2\mu\varepsilon_{ij}\ .\end{aligned} \tag{42}$$

As explained further in the section on the experimental setup, only the shear is measured, thus, $\lambda = 0$ and the above equation reads:

$$\sigma^R_{ij} = G_e\varepsilon_{ij} \tag{43}$$

$$\sigma^{\text{frac}}_{ij} = \frac{1}{(1+\frac{\tau_0^\alpha}{h^\alpha})}\left(-\tau_0^\alpha\sum_{m=1}^{t/h}\frac{1}{h^\alpha}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\ \Gamma|_{(\alpha-m+1)}}\ \sigma_{ij}|_{(t-mh)} + G_e\tau_0^\alpha\frac{d^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\ \alpha}} + G_0\tau_0^\beta\frac{d^\beta\varepsilon_{ij}}{\mathrm{d}t^{\ \beta}}\right)\ , \tag{44}$$

where:

$$\frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^\alpha} = \frac{1}{h^\alpha}\sum_{m=0}^{t/h}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\ \Gamma|_{(\alpha-m+1)}}\ \varepsilon_{ij}|_{(t-mh)}\ ,\quad \frac{\mathrm{d}^\beta\varepsilon_{ij}}{\mathrm{d}t^\beta} = \frac{1}{h^\beta}\sum_{m=1}^{t/h}(-1)^m\frac{\Gamma|_{(\beta+1)}}{m!\ \Gamma|_{(\beta-m+1)}}\ \varepsilon_{ij}|_{(t-mh)}\ , \tag{45}$$

so that the discretised form responsible for the temperature distribution reads:

$$\begin{aligned}&\int_{B_0}\left(\sigma^R_{ij}\frac{1}{2\Delta t}((u_{i,j} - u^0_{i,j}) + (u_{j,i} - u^0_{j,i}))\delta T - TG_e\alpha\frac{1}{\Delta t}(u_{i,i} - u^0_{i,i})\delta T + C_\varepsilon\frac{T - T^0}{\Delta t}\delta T\right.\\ &\left.+\kappa\delta_{rj}\frac{\partial T}{\partial X_j}\frac{\partial\delta T}{\partial X_r} - (\sigma^R_{ri} + \sigma^{\text{frac}}_{ri})\frac{1}{\Delta t}\left(\frac{\partial u_i}{\partial X_r} - \frac{\partial u_i^0}{\partial X_r}\right)\delta T\right)dV_0 - \oint_{\partial B_0}\kappa\delta_{rj}\frac{\partial T}{\partial X_r}\delta T N_r dA_0 = 0\ .\end{aligned} \tag{46}$$

The thermal and elastic effects on the system are coupled. Hence, the final variational form to be solved using FEniCS [5] will have two trial functions, which need to satisfy Eq. (14) and Eq. (4) simultaneously, for the independent variables viz. displacements (δu_i) and temperature (δT).

We need to make up a problem, *i.e.* set the boundaries. A box out of the material with known material parameters is clamped on bottom $\partial\Omega^1_D$, sheared on top $\partial\Omega^2_D$ with a given velocity function (τ is a parametrized time) and is free (tracted with $t_i = 0$ force) on rest of the boundaries $\partial\Omega_N$. For the temperature distribution, we assume a ROBIN boundary condition:

$$
\begin{aligned}
v_i = 0 \ , \ \forall x \in \partial\Omega_D^1 \\
v_i = \hat{v}_i = \begin{pmatrix} \sin(\pi\tau) \\ 0 \\ 0 \end{pmatrix} \ , \ \forall x \in \partial\Omega_D^2 \\
t_i = n_j \sigma_{ji} \ , \ \forall x \in \partial\Omega_N \\
-\kappa\delta_{rj}\frac{\partial T}{\partial X_j}N_r = Q_r N_r = h(T - T_{\text{ambient}}) \ , \ \forall x \in \partial B_0
\end{aligned} \tag{47}
$$

where, h is the heat transfer coefficient of the surface. Furthermore the right and left sides are periodic, which is actually the outer shell of the viscometer sample mapped onto two dimensional space. Now by setting $\delta u_i\big|_{\mathrm{d}\Omega_D} = 0$ and $t_i = 0$ (free surfaces are traction-free) the final variational form and a list of all the other relations read:

$$
\begin{aligned}
&\int_{\mathcal{B}_0}\left(\frac{1}{\mathrm{d}t\ \mathrm{d}t}\rho_0(u_i - 2u_i^0 + u_i^{00})\delta u_i + \left((\sigma_{ji}^R + \sigma_{ji}^{\text{frac}})\frac{\partial\delta u_j}{\partial X^r}\right)\right)\mathrm{d}V_0 + \\
&\int_{B_0}\left(\sigma_{ij}^R\frac{1}{2\Delta t}((u_{i,j} - u_{i,j}^0) + (u_{j,i} - u_{j,i}^0))\delta T - TG_e\alpha\delta_{ij}\frac{1}{2\Delta t}((u_{i,j} - u_{i,j}^0) + (u_{j,i} - u_{j,i}^0))\delta T + C_\varepsilon\frac{T - T^0}{\Delta t}\delta T + \right. \\
&\left. + \kappa\delta_{rj}\frac{\partial T}{\partial X_j}\frac{\partial \delta T}{\partial X_r} - (\sigma_{ri}^R + \sigma_{ri}^{\text{frac}})\frac{1}{\Delta t}\left(\frac{\partial u_i}{\partial X_r} - \frac{\partial u_i^0}{\partial X_r}\right)\delta T\right)dV_0 + \oint_{\partial B_0} h(T - T_{\text{ambient}})dA_0 = 0 \ ,
\end{aligned} \tag{48}
$$

$$
x_i = u_i + X_i \ , \ F_{jr} = \frac{\partial x_j}{\partial X_r} = \frac{\partial u_j}{\partial X_r} + \delta_{jr} \ , \tag{49}
$$

$$
\varepsilon_{ij} = \frac{1}{2}\Big(\frac{\partial u_i}{\partial X_j}\frac{\mathrm{d}X_j}{\mathrm{d}x_j} + \frac{\partial u_j}{\partial X_i}\frac{\mathrm{d}X_i}{\mathrm{d}x_i}\Big) \ , \ \frac{\mathrm{d}^\alpha\varepsilon_{ij}}{\mathrm{d}t^\alpha} = \frac{1}{h^\alpha}\sum_{m=0}^{t/h}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\ \Gamma|_{(\alpha-m+1)}}\ \varepsilon_{ij}|_{(t-mh)} \ , \tag{50}
$$

$$
\sigma_{ij} = \sigma_{ij}^R + \sigma_{ij}^{\text{frac}} \ , \ \sigma_{ij}^R = G_e\varepsilon_{ij} \ , \tag{51}
$$

$$
\sigma_{ij}^{\text{frac}} = \frac{1}{(1 + \frac{\tau_0^\alpha}{h^\alpha})}\Bigg(- \tau_0^\alpha\sum_{m=1}^{t/h}\frac{1}{h^\alpha}(-1)^m\frac{\Gamma|_{(\alpha+1)}}{m!\ \Gamma|_{(\alpha-m+1)}}\ \sigma_{ij}|_{(t-mh)} + G_e\tau_0^\alpha\frac{d^\alpha\varepsilon_{ij}}{\mathrm{d}t^{\,\alpha}} + G_0\tau_0^\beta\frac{d^\beta\varepsilon_{ij}}{\mathrm{d}t^{\,\beta}}\Bigg) \ . \tag{52}
$$

Let us recall that the limits of the summation in the fractional derivative of the stress and strain are different because the first term in the expansion of the fractional derivative of stress is included in the LHS.

5 Implementation

The combined form of momentum and energy balance is solved for the displacement and temperature fields using the open source finite element package FEniCS. The packages NumPy and SciPy are also used in some calculations like the evaluation of the gamma function needed for the fractional rates. The results of the simulation are visualised using ParaVIEW and matplotlib.

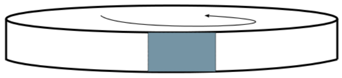

Figure 3: Schematic for the material being tested

The outer shell of the sample, a portion of which is marked in the Figure 3, is the region under consideration here. A 2 dimensional rectangular mesh is used to represent this region. A periodic boundary condition is imposed on the mesh. Using this, the outer shell can be simulated using the a 2D rectangular mesh.

The use of the fractional derivatives in the constitutive relation requires the storage of the all stress and strain

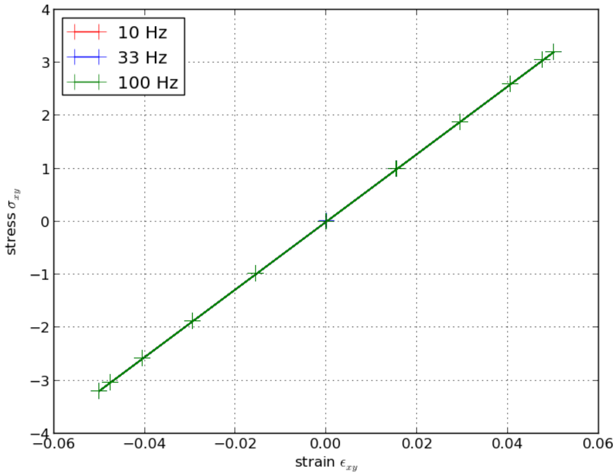

Figure 4: Shear stress v/s Shear strain for $\tau_0 = 0$

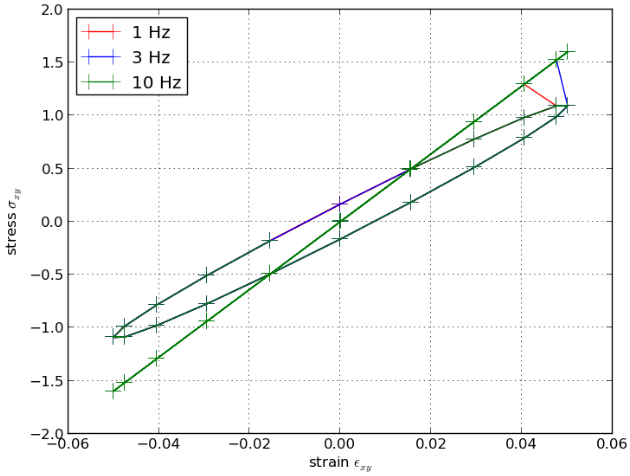

Figure 5: Shear stress v/s Shear strain for $\alpha = 0$ and $\beta = 1$

components each element of the body. This was initially implemented using matrices. However, the history matrices keep growing in size, which makes the solution process very memory-intensive.

An alternative implementation was then tried using methods from the TimeSeries class of the DOLFIN package. In this technique the values of stress and stress throughout the mesh at each time slice are written out to a binary file which is then read in the calculations for the further time slices. The use of the TimeSeries class, thus, reduces the time and memory required for the simulation.

6 Results and Discussions

The python code for the finite element calculations (given below) was run by varying different model parameters, particularly τ_0, α and β. The dependence of the mechanical response (shear stress) on the input (shear strain) was studied for different frequencies of oscillation of the upper plate of the viscometer. The plots of the variation of shear stress with shear strain for various values of τ_0, α and β are shown below.

Figure 4 shows the mechanical response for the case of $\tau_0 = 0$. Since $\tau = 0$, the values of α and β are not important. The plot is linear as expected. Thus, the model reduces to a linear elastic one for this value of τ. We also, observe, no dependence on frequency of oscillation of the upper plate.

For the case of $\tau_0 = 0.0001$, $\alpha = 0$ and $\beta = 1$, the constitutive equation includes a dependence on strain and strain rates, as shown in Figure 6. Thus, the model represents a viscoelastic material. The plot shows hysteresis as expected from the model, due to the viscous effects. A frequency dependence is also observed due to the viscous effects. The elastic effects are also observed to become more significant with an increase in frequency.

However, in Figure 5 where the mechanical response for the case of $\alpha = 0$ and $\beta = 2$ and $\tau_0 = 0.0001$ is shown, the plot is also linear. This is in fact, as expected. Thus, the model reduces to a linear elastic one for these values of α and β. We also, observe, no dependence on frequency of oscillation of the upper plate.

Figure 7 shows the mechanical response for the case of $\alpha = 0$, $\beta = 1.1$ and $\tau_0 = 0.0001$. The plot is expected to be similar in nature to Figure 6. This is observed as expected. However, the plot shows that the mechanical behaviour in this case is closer to the elastic case, *i.e.* the size of the hysteresis loop as well as the frequency dependence is reduced.

Figure 8 shows the mechanical response for the case of $\alpha = 0$ and $\beta = 0.5$. The value of τ_0 is set to 0.0001. This plot shows greater viscous effects. We observe that the elastic portion is much smaller in magnitude compared to the hysteresis loop. One unusual feature observed here is that the stress and strain have opposite signs in hysteresis region. This appears quite unrealistic. However, the constitutive relation is capable of presenting

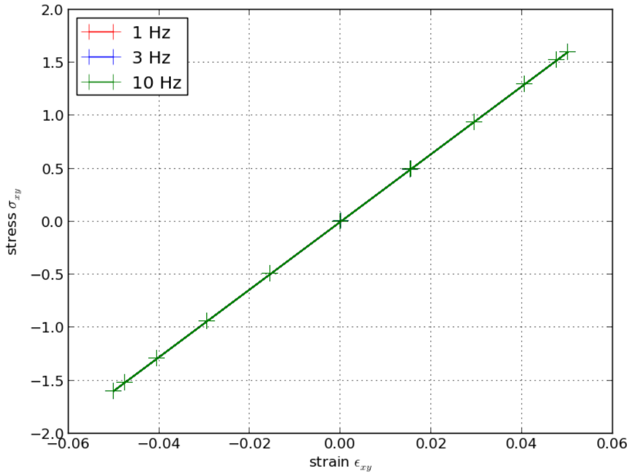

Figure 6: Shear stress v/s Shear strain for $\alpha = 0$, $\beta = 2$ and $\tau_0 = 0.0001$

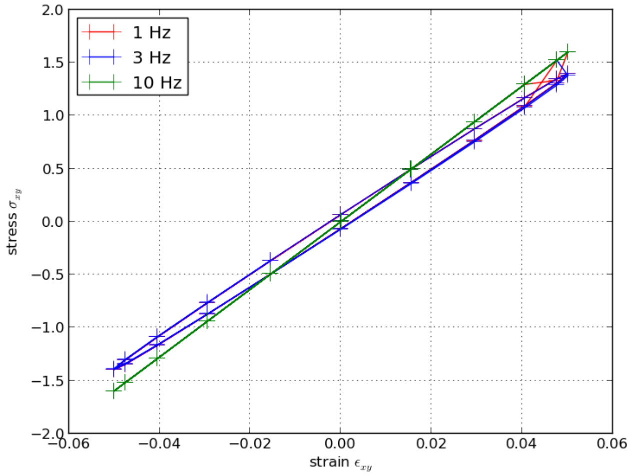

Figure 7: Shear stress v/s Shear strain for $\alpha = 0$, $\beta = 1.1$ and $\tau_0 = 0.0001$

such unexpected features, though it is not clear if there exist such materials. This result also came out for a composite of a commercial grade of polycarbonate (Makrolon® M2200) with 0.5% multi-wall carbon nanotubes (Baytubes®)[4], where the coefficients are acquired by an inverse analysis using the Eq. (20) and Eq. (21). It is of interest if the material Makrolon does indeed show such a behaviour or if the non-uniqueness of the inverse solution have provided these results. In Figure 9 the hysteresis curve for the Makrolon generated with the coefficients from [4], *i.e.* $\alpha = 0.994$ and $\beta = 0.804$, can be seen.

Figure 10 shows the mechanical response for the case of $\alpha = 0$ and $\beta = 0.5$ and $\tau_0 = 0.00001$. The plot is similar in nature to Figures 8 and 9, as expected. Here, the effect of τ_0 on the viscous effects can be observed. The magnitude of the shear stress in the hysteresis loop is similar to that in the elastic region (linear variation). This indicates that the viscous effects reduce with the decrease in τ_0. The plots also show the convergence towards a steady state of stress, *i.e.* the paths of successive loops come closer to each other and finally repeat after a few cycles.

An assumption made in the calculations is that the material properties do not change with temperature. This assumption is correct if the body remains almost isothermal. This is indeed observed in the temperature

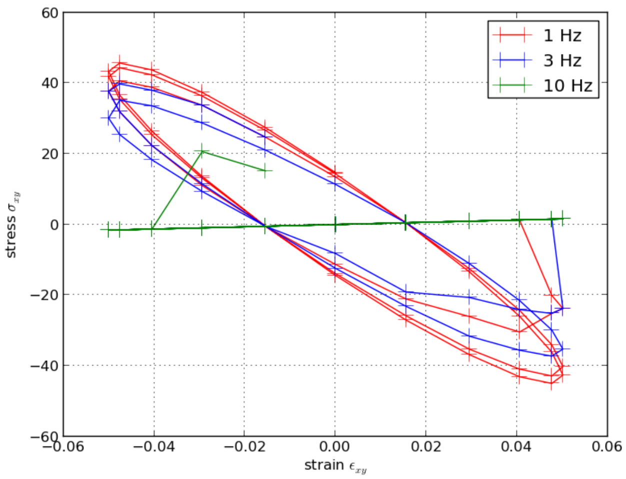

Figure 8: Shear stress v/s Shear strain for $\alpha = 0$, $\beta = 0.5$ and $\tau_0 = 0.0001$

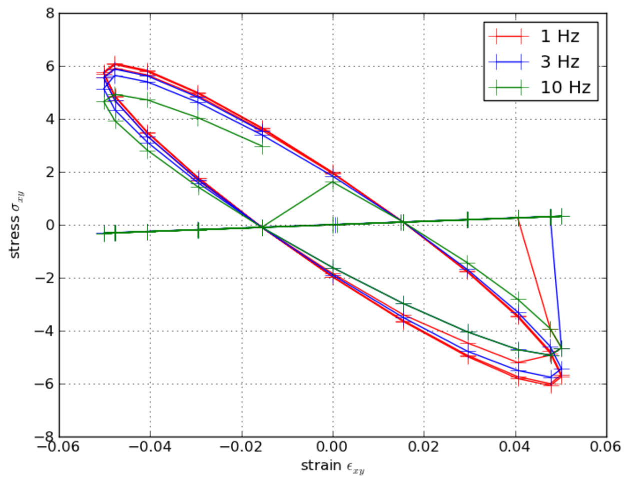

Figure 9: Shear stress v/s Shear strain for $\alpha = 0.994$, $\beta = 0.804$ and $\tau_0 = 0.0001$

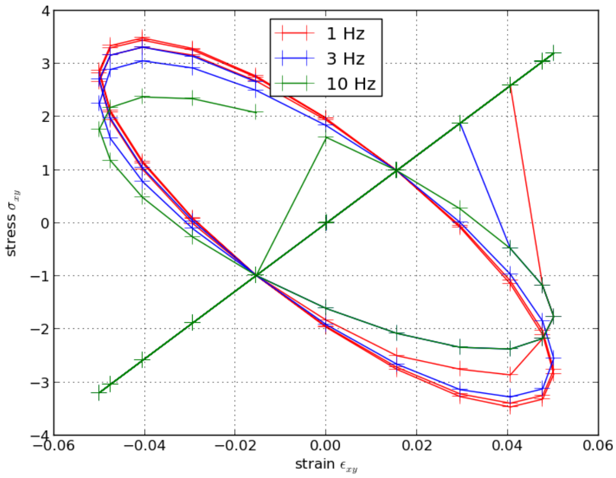

Figure 10: Shear stress v/s Shear strain
for $\alpha = 0$, $\beta = 0.5$ and $\tau_0 = 0.00001$

variations in all the above cases. The initial temperature of the sample is taken as being equal to the ambient temperature. The temperature is seen to change during the process by less than 1K. Although small, heat dissipation is also observed. This can be explained by the fact that the test sample is very thin and the heat dissipation can at a very high rate, leading to near isothermal conditions. Thus, the isothermal assumption is valid.

7 Summary

The modelling and numerically solving of deformation and temperature via history dependent viscoelastic model is discussed. The used fractional Zener model is shown to be capable of being used for the simulation of the mechanical response of viscoelastic materials. Moreover its dependency on the whole history seems promising especially for polymer type of amorph materials. Further investigations into the determination of material properties of the material to be simulated as well as in the expression for the internal energy stored in the body are, however, required.

References

[1] I. J. Duvdevani, J. A. Biesenberger, and C. G. Gogos.
Thermomechanical studies of polymer deformation part i: A new deformation calorimeter.
Polymer Engineering & Science, 9(4):250–254, 1969.

[2] Wilhelm Flgge.
Tensor Analysis and Continuum Mechanics.
Springer-Verlag, 1972.

[3] Chr. Friedrich.
Mechanical stress relaxation in polymers: fractional integral model versus fractional differential model.
Journal of Non-Newtonian Fluid Mechanics, 46(23):307 – 314, 1993.

[4] Ulrich Handge, Rico Zeiler, Dirk Dijkstra, Helmut Meyer, and Volker Altstdt.
On the determination of elastic properties of composites of polycarbonate and multi-wall carbon nanotubes in the melt.
Rheologica Acta, 50:503–518, 2011.
10.1007/s00397-011-0558-x.

[5] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al.
Automated Solution of Differential Equations by the Finite Element Method.
Springer, 2012.

[6] J. F. Nye and R. B. Lindsay.
Physical properties of crystals: Their representation by tensors and matrices.
Physics Today, 10(12):26–26, 1957.

[7] Igor Podlubny.
Fractional differential equations : an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications.
London : Academic Press, 1999.

8 Code

```
"""Coupled deformation and temperature computation for a fractional time rate
viscoelastic model"""

__author__ = "B. Emek Abali (abali@tu-berlin.de) & Aditya Desai"
__date__ = "2012-07-13"
__copyright__ = "Copyright (C) 2012 B. Emek Abali"
__license__  = "GNU LGPL Version 3 or later"
# Copyright (C) 2012 B. Emek Abali
#
# This code is free software: you can redistribute it and/or modify
# it under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This code is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU Lesser General Public License for more details.
#
# For the GNU Lesser General Public License see <http://www.gnu.org/licenses/>.
#
#-------------------------------------------------------------
#---run like: python file_name.py
#---from shell after installing FEniCS and SciPy (matplotlib)
#---www.fenicsproject.org
#---www.scipy.org
#--------------------------------------------------------------
from dolfin import *
from scipy.special import gamma
from scipy.misc import factorial
import numpy as np
import scipy

#for faster assembly and solving
parameters ["form_compiler"]["cpp_optimize"]= True

class iterate(NonlinearProblem):
	def __init__ (self,a,L,bc,exter_B):
		NonlinearProblem.__init__(self)
		self.L = L
		self.a = a
		self.bc = bc
		self.exter= exter_B
	def F(self,b,x) :
		assemble(self.L,tensor=b,exterior_facet_domains=self.exter)
		for condition in self.bc : condition.apply(b,x)
	def J(self,A,x) :
		assemble(self.a,tensor=A, exterior_facet_domains=self.exter)
		for condition in  self.bc : condition.apply(A)
```

```
# https://answers.launchpad.net/dolfin/+question/196105
stress_history = TimeSeries("stressHist")
strain_history = TimeSeries("strainHist")

# https://answers.launchpad.net/dolfin/+question/147406
stress_history.parameters["clear_on_write"] = False
strain_history.parameters["clear_on_write"] = False

xlength =0.120 #[m]
ylength =0.005 #[m]
mesh = Rectangle(0,0,xlength,ylength,20,5)

DispSpace = VectorFunctionSpace(mesh,'CG',1)
TempSpace = FunctionSpace(mesh,'CG',1)
TensorSpace = TensorFunctionSpace(mesh,'CG',1)
Space = MixedFunctionSpace([DispSpace,TempSpace])

delMF = TestFunction(Space)
dMF = TrialFunction(Space)

# Separate test functions
(delu,delT) = split(delMF)

# Mixed definitions of function w.r.t. time steps n,n-1,n-2
MFref = Function(Space)
MF = Function(Space)
MF0 = Function(Space)
MF00 = Function(Space)

# Displacement and temperature w.r.t. time steps n,n-1,n-2
(u,T) = split(MF)
(u0,T0) = split(MF0)
(u00,T00) = split(MF00)

Tref = 293.15


class initialConditions(Expression):
        def __init__(self,Tini):
                self.Tini = Tini

        def eval(self,vec,x) :
                vec[0] = 0.0
                vec[1] = 0.0
                vec[2] = self.Tini
                #vec[2] = 0.0

        def value_shape(self):
                return (3, )

print 'initializing,time'
t_start=0.0
t_bc = 1.0
t_end = 10.0
dt = 0.1
t = t_start

#setting initial conditions
iniObj = initialConditions(Tref)
MFref.interpolate(iniObj)
MF.assign(MFref)
MF0.assign(MF)
```

```
#Defining boundary conditions

Dimensions = mesh.topology().dim()
left = compile_subdomains('x[0] == 0.0')
right = compile_subdomains('x[0]== xlength')
right.xlength = xlength

bottom = compile_subdomains('x[1]== 0.0')
top = compile_subdomains('x[1]== ylength')
top.ylength = ylength

neumann_domains = MeshFunction("uint",mesh,Dimensions-1)
neumann_domains.set_all(0)

#shear with amplitude 0.5mm to right on top
shear = Expression(('0.0005*sin(2.*pi*f*time)','0.0'),f=0,time=0)

bc1 = DirichletBC(DispSpace,shear,top)
bc2 = DirichletBC(DispSpace,(0.0,0.0),bottom)

class PeriodicBoundary(SubDomain):
        def inside(self, x, on_boundary):
                #left side is x[0]=-xlength/2
                return x[0] < +1.e-5 and x[0] > -1.e-5 and on_boundary
        def map(self, y, x):
                #this maps right side x[0]=xlength to the left
                x[0] = y[0]-xlength
                x[1] = y[1]

bc = [bc1,bc2,PeriodicBC(DispSpace, PeriodicBoundary())]

print 'initializing,space'
# material coeffs

#Setting the material properties
beta1= 0.994
beta2= 0.190
alpha=0.#beta1
beta=1.1#beta1-beta2
G0= 2.133e5
tau0=0.0036
Ge=6.393
rho0=1200. #[kg/m3]
C_eps= 1.70
kappa=0.173
h = 2.0
pres= 65.0e-6*Ge

# index notation
i,j,k,l,r = indices(5)
delta= Identity(2)

F = as_tensor(delta[i,j]+ u[i].dx(j) ,[i,j])

epsilon= as_tensor(1.0/2.0*(u[i].dx(j)*inv(F)[j,k]+u[k].dx(j)*inv(F)[j,i]),[i,k])
epsilon0= as_tensor(1.0/2.0*(u0[i].dx(j)*inv(F)[j,k]+u0[k].dx(j)*inv(F)[j,i]),[i,k])

def fractional(index,arg,t,h,power):
        m=index+1
        if Dimensions==3: temp=as_tensor([[0.,0.,0.],[0.,0.,0.],[0.,0.,0.]])
        if Dimensions==2: temp=as_tensor([[0.,0.],[0.,0.]])
        while m <= t/h:
            hist=int(t/h-m)
```

```
            tensor=Function(TensorSpace)
            arg.retrieve(tensor.vector(), hist)
            temp += as_tensor((-1.)**m*gamma(power+1.)/gamma(power-m+1.)/factorial(m)*\
                    tensor[i,j]/(h**power),[i,j])
            m += 1
        return temp

S_r = as_tensor(Ge*epsilon[i,j],[i,j])
S_frac = as_tensor( 1.0/(1.0+(tau0/dt)**alpha)*( \
        - tau0**alpha*fractional(1,stress_history,t,dt,alpha)[i,j] \
        + Ge*(tau0**alpha*fractional(0,strain_history,t,dt,alpha)[i,j]) \
        + G0*tau0**beta*fractional(0,strain_history,t,dt,beta)[i,j]), [i,j])


J= det(F)

f= Constant((0.0,0.0))

form = 1.0/(dt**2.)*rho0*(u[i]-2.*u0[i]+u00[i])*delu[i]*dx \
    + (S_r[j,i]/(1.0+(tau0/dt)**alpha)+S_frac[j,i])*inv(F)[j,r]*J*delu[i].dx(r)*dx \
    - rho0*f[i]*delu[i]*dx \
    + S_r[i,j]*1.0/dt*(epsilon[i,j]-epsilon0[i,j])*delT*dx \
    + C_eps*(T-T0)/dt*delT*dx \
    + kappa*delta[r,j]*T.dx(j)*delT.dx(r)*dx \
    + T*pres*1./dt*(u-u0)[k].dx(k)*delT*dx \
    - (S_r[r,i]/(1.0+(tau0/dt)**alpha)+S_frac[r,i])*1./dt*(u-u0)[i].dx(r)*delT*dx \
    + h*(T-Tref)*delT*ds


gain = derivative(form,MF,dMF)

# set the output file
file_u = File ('viscoelasticitydeformations.pvd')
file_T = File ('viscoelasticitytemperatures.pvd')

def stress(u,tau0,stress_history,strain_history,t):
    F = as_tensor(delta[i,j]+ u[i].dx(j) ,[i,j])
    epsilon= as_tensor(1.0/2.0*(u[i].dx(j)*inv(F)[j,k]+u[k].dx(j)*inv(F)[j,i]),[i,k])
    return as_tensor(1.0/(1.0+(tau0/dt)**alpha)*(- \
     tau0**alpha*fractional(1,stress_history,t,dt,alpha)[i,j] + \
     Ge*(epsilon[i,j]+tau0**alpha*fractional(0,strain_history,t,dt,alpha)[i,j])+ \
       G0*tau0**beta*fractional(0,strain_history,t,dt,beta)[i,j]), [i,j])

def strain(u,tau0):
        F = as_tensor(delta[i,j]+ u[i].dx(j) ,[i,j])
        return as_tensor(1.0/2.0*(u[i].dx(j)*inv(F)[j,k]+u[k].dx(j)*inv(F)[j,i]),[i,k])

#Plotting stress v/s strain curves
import matplotlib.pyplot as pylab

pylab.ion()
pylab.clf()
pylab.cla()

pylab.xlabel('strain $\epsilon_{xy}$')
pylab.ylabel('stress $\sigma_{xy}$')

pylab.grid(True)

def compute(freq,tau):
        t=t_start
        stresses = []
        strains = []
```

```
        temp_array= []
        time= []
        MFref.interpolate(iniObj)
        MF.assign(MFref)
        MF0.assign(MF)
        dt = 0.05/freq

        while t< 3./freq :
                shear.time = t
                shear.f = freq # if t<t_bc else 0.0
                print 'time: ', t
                print 'time_step : ' , dt
                tau0 = tau
                problem = iterate(gain,form,bc,neumann_domains)
                solver = NewtonSolver('lu') # for non symmetric case
                solver.parameters['convergence_criterion'] = 'incremental'
                solver.parameters['relative_tolerance'] = 1.0e-2
                solver.solve(problem,MF.vector())
                print ' write out, assign '
                file_u <<(MF.split()[0],t)
                file_T <<(MF.split()[1],t)
                MF00.assign(MF0)
                MF0.assign(MF)
                u,T = MF.split()
                info('stress history: ')
                tic()
                actualStress = \
                project(stress(u,tau,stress_history,strain_history,t),TensorSpace)
                stress_history.store(actualStress.vector(),t)
                print toc(), 'seconds'
                info('strain history: ')
                tic()
                actualStrain = project(strain(u,tau),TensorSpace)
                strain_history.store(actualStrain.vector(),t)
                print toc(), 'seconds'
                P=(xlength,ylength/2.)
                sigma12 = actualStress(P)[1]
                epsilon12 = actualStrain(P)[1]
                stresses.append(sigma12)
                strains.append(epsilon12)
                t = t + dt

        return stresses, strains



stresses, strains = compute(1.,0.00001)
#np.save('10Hz0tau_stress.npy', stresses)
#np.save('10Hz0tau_strain.npy', strains)
#stresses=np.load('3Hz0tau_stress.npy')
#strains=np.load('3Hz0tau_strain.npy')
pylab.plot(strains,stresses,color='red',marker='+',markersize=13,label='1 Hz')

pylab.draw()

stresses, strains = compute(3.,0.00001)
#np.save('33Hz0tau_stress.npy', stresses)
#np.save('33Hz0tau_strain.npy', strains)
#stresses=np.load('33Hz0tau_stress.npy')
#strains=np.load('33Hz0tau_strain.npy')
pylab.plot(strains,stresses,color='blue',marker='+',markersize=13,label='3 Hz')
```

```
pylab.draw()

stresses, strains = compute(10.,0.00001)
#np.save('100Hz0tau_stress.npy', stresses)
#np.save('100Hz0tau_strain.npy', strains)
#stresses=np.load('100Hz0tau_stress.npy')
#strains=np.load('100Hz0tau_strain.npy')
pylab.plot(strains,stresses,color='green',marker='+',markersize=13,label='10 Hz')

pylab.draw()

pylab.legend(loc='best')

pylab.savefig("varfreq_0alpha1_1beta000001tau.png")
#-------------------------------------
```