

# A fractional model of viscoelasticity

The viscoelastic model problem is described as a Volterra integral equation of the second kind. A choice of *power law* type kernel yield the **fractional order model of viscoelasticity**. We solve the fractional order model by using spatial finite element method and *Crank-Nicolson* type finite difference schemes in time.

## Model Problem

Let  $\Omega \subset \mathbb{R}^d$  be a spatial domain for  $d = 2, 3$  and  $[0, T]$  be a time interval for  $T > 0$ . We consider the following model problem such that

$$\begin{aligned} \rho \dot{\mathbf{w}}(t) - \nabla \cdot {}_0 I_t^{1-\alpha} (\underline{\mathbf{D}} \underline{\boldsymbol{\varepsilon}}(\mathbf{w}(t))) &= \mathbf{f}(t), & \text{on } (0, T] \times \Omega, \\ {}_0 I_t^{1-\alpha} (\underline{\mathbf{D}} \underline{\boldsymbol{\varepsilon}}(\mathbf{w}(t))) \cdot \mathbf{n} &= \mathbf{g}_N(t), & \text{on } [0, T] \times \Gamma_N, \\ \mathbf{w}(t) &= \mathbf{0}, & \text{on } [0, T] \times \Gamma_D, \\ \mathbf{w}(0) &= \mathbf{w}_0, & \text{on } \Omega, \end{aligned}$$

where  $\mathbf{w}$  is velocity,  ${}_0 I_t^{1-\alpha}$  is fractional integral of order  $1 - \alpha$  for  $\alpha \in (0, 1)$ ,  $\underline{\mathbf{D}}$  is a symmetric positive definite fourth order tensor,  $\underline{\boldsymbol{\varepsilon}}$  is the Cauchy infinitesimal tensor,  $\mathbf{f}$  is the external body force,  $\mathbf{n}$  is an outward unit normal vector,  $\mathbf{g}_N$  is the traction,  $\Gamma_N$  is Neumann boundary and  $\Gamma_D$  is Dirichlet boundary. We assume that  $\Gamma_D$  and  $\Gamma_N$  are disjoint with  $\partial\Omega = \Gamma_D \cup \Gamma_N$  and  $\Gamma_D$  is of positive measured.

## Weak Formulation

Let  $\mathbf{V}$  be a test space such that  $\mathbf{V} = \{\mathbf{v} \in [H^1(\Omega)]^d \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$ . Integration by parts yields the following weak formulation.

Find a mapping  $\mathbf{w} : [0, T] \mapsto \mathbf{V}$  such that satisfies

$$\begin{aligned} (\rho \dot{\mathbf{w}}(t), \mathbf{v}) + a({}_0 I_t^{1-\alpha} \mathbf{w}(t), \mathbf{v}) &= F(t; \mathbf{v}), \quad \forall t \in (0, T], \\ a(\mathbf{w}(0), \mathbf{v}) &= a(\mathbf{w}_0, \mathbf{v}), \end{aligned}$$

for any  $\mathbf{v} \in \mathbf{V}$  where  $(\cdot, \cdot)$  is  $L_2$  inner product over  $\Omega$  ( $(\cdot, \cdot)_A$  is  $L_2$  inner product over  $A \subset \Omega$ ),  $a(\cdot, \cdot)$  and  $F$  are defined by

$$a(\mathbf{v}, \mathbf{w}) = \int_{\Omega} \underline{\mathbf{D}} \underline{\boldsymbol{\varepsilon}}(\mathbf{v}) : \underline{\boldsymbol{\varepsilon}}(\mathbf{w}) d\Omega \text{ and } F(t; \mathbf{v}) = (\mathbf{f}(t), \mathbf{v}) + (\mathbf{g}_N(t), \mathbf{v})_{\Gamma_N}.$$

## Fully Discrete Formulation

Let us consider the finite element space. Suppose that we have conformal meshes with respect to  $\Omega$  and let  $\mathbf{V}^h$  be the Lagrange finite element space of polynomial of degree  $k$  in  $\mathbf{V}$ . Then we define the time step size and introduce time discretisation.

Let  $\Delta t = T/N$  for some  $N \in \mathbb{N}$ . Define  $t_n = n\Delta t$  for  $n = 0, \dots, N$  and denote our fully discrete solution by  $\mathbf{W}_h^n$  for  $n = 0, \dots, N$ . For numerical approximations of fractional integration, we use linear interpolation technique such that

$${}_0I_{t_n}^{1-\alpha} \mathbf{w}(t) = \frac{\Delta t^{1-\alpha}}{\Gamma(3-\alpha)} \sum_{i=0}^n B_{n,i} \mathbf{w}(t_i) + \mathcal{O}(\Delta t^2) = \mathbf{q}_n(\mathbf{w}) + \mathcal{O}(\Delta t^2),$$

where  $\Gamma$  is the Gamma function and

$$B_{n,i} = \begin{cases} n^{1-\alpha}(2-\alpha-n) + (n-1)^{2-\alpha}, & i = 0, \\ (n-i-1)^{2-\alpha} + (n-i+1)^{2-\alpha} - 2(n-i)^{2-\alpha}, & i = 1, \dots, n-1, \\ 1, & i = n. \end{cases}$$

Consequently, use of Crank-Nicolson method and numerical integration leads us to obtain a fully discrete formulation as follows:

Find  $\mathbf{W}_h^n \in \mathbf{V}^h$  for  $n = 0, \dots, N$  such that satisfying for any  $\mathbf{v} \in \mathbf{V}^h$ ,

$$\left( \rho \frac{\mathbf{W}_h^{n+1} - \mathbf{W}_h^n}{\Delta t}, \mathbf{v} \right) + a \left( \frac{\mathbf{q}_{n+1}(\mathbf{W}_h) + \mathbf{q}_n(\mathbf{W}_h)}{2}, \mathbf{v} \right) = \frac{1}{2} (F(t_{n+1}; \mathbf{v}) + F(t_n; \mathbf{v})),$$

$$a(\mathbf{W}_h^0, \mathbf{v}) = a(\mathbf{w}_0, \mathbf{v}).$$

## Stability and Error Analysis

Our numerical scheme has well-posedness as well as convergence. *A priori* error estimates give us suboptimal and optimal convergence rates. For more details, see this paper and author's thesis (the link will appear here soon).

## Numerical Experiments

Let  $\Omega$  be the unit square,  $\Gamma_D = \partial\Omega$ ,  $\rho = 1$  and  $\alpha = 1/2$  for simplicity.

We consider two examples; one is of  $H^2$  regular in time and the other is of  $H^3$ . The former example does not satisfy the condition of second order schemes hence it fails to give convergence of the second order. But it could show advantage of the second order method rather than first order methods. On the other hand, the latter is sufficiently smooth with respect to time. We are able to observe optimal error rates.

1. Set

$$\mathbf{w}(x, y, t) = (t + t^{1.5}) \begin{bmatrix} \sin(\pi x) \sin(\pi y) \\ xy(1-x)(1-y) \end{bmatrix}.$$

Then  $\mathbf{w} \in C^2(0, T; [C^\infty(\Omega)]^2) \cap W_1^2(0, T; [C^\infty(\Omega)]^2)$  with homogeneous Dirichlet boundary condition.

2. Set

$$\mathbf{w}(x, y, t) = t^{2.5} \begin{bmatrix} \sin(\pi x) \sin(\pi y) \\ xy(1-x)(1-y) \end{bmatrix}.$$

According to the fully discrete formulation, code implementation has been carried out in **FEniCS**.

Numerical solutions of Example 1 can be obtained by running **homo\_nonsmooth\_cg.py**. We can specify mesh sizes with respect to the space as well as time. Once you run it,  $H^1$  norm errors and  $L_2$  norm errors are stored in *txt* files. In this manner, **homo\_smooth\_cg\_modi.py** deals with numerical approximations for Example 2.

- Run **main.sh** to get numerical solutions and their numerical errors in *txt* files.
- Run **CG\_graphic\_frac.py** to illustrate graphs of error convergence rates for linear and quadratic polynomial bases when  $h \approx \Delta t$ .