Published in final edited form as:
J Comput Phys. 2016 October 15; 323: 219–242. doi:10.1016/j.jcp.2016.06.038.

Fractional modeling of viscoelasticity in 3D cerebral arteries and aneurysms

Yue Yu[a], **Paris Perdikaris**[b], and **George Em Karniadakis**[c]

[a]Department of Mathematics, Lehigh University, Bethlehem, PA 18015, USA

[b]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[c]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

Abstract

We develop efficient numerical methods for fractional order PDEs, and employ them to investigate viscoelastic constitutive laws for arterial wall mechanics. Recent simulations using one-dimensional models [1] have indicated that fractional order models may offer a more powerful alternative for modeling the arterial wall response, exhibiting reduced sensitivity to parametric uncertainties compared with the integer-calculus-based models. Here, we study three-dimensional (3D) fractional PDEs that naturally model the continuous relaxation properties of soft tissue, and for the first time employ them to simulate flow structure interactions for patient-specific brain aneurysms. To deal with the high memory requirements and in order to accelerate the numerical evaluation of hereditary integrals, we employ a fast convolution method [2] that reduces the memory cost to $O(\log(N))$ and the computational complexity to $O(N\log(N))$. Furthermore, we combine the fast convolution with high-order backward differentiation to achieve third-order time integration accuracy. We confirm that in 3D viscoelastic simulations, the integer order models strongly depends on the relaxation parameters, while the fractional order models are less sensitive. As an application to *long-time* simulations in complex geometries, we also apply the method to modeling fluid–structure interaction of a 3D patient-specific *compliant* cerebral artery with an aneurysm. Taken together, our findings demonstrate that fractional calculus can be employed effectively in modeling complex behavior of materials in realistic 3D time-dependent problems if properly designed efficient algorithms are employed to overcome the extra memory requirements and computational complexity associated with the non-local character of fractional derivatives.

Keywords

Fast convolution method; Fractional differential equations; Fluid–structure interaction; Spectral element method; Patient-specific vasculature; Brain aneurysm

Correspondence to: Yue Yu.

1. Introduction

The arterial wall is a heterogeneous soft tissue with complex biomechanical properties, which may vary over space and the temporal states of deformation. Typically, integer-order differential equations are used to model the stress-strain relations for arterial walls [3–5]. A family of integer-order models can be derived using Fung's theory of quasilinear viscoelasticity [3], and one of its simplest examples is the Kelvin–Zener (Standard Linear Solid or SLS) model [6]. The SLS model accounts for creep, hysteresis and stress relaxation phenomena, and it can be described as a parallel combination of a spring with a spring and a dashpot in series. In the one-dimensional (1D) SLS model, the axial stress $\sigma(t)$ is related to strain $\varepsilon(t)$ as:

$$\sigma(t)+\tau_\sigma\frac{d\sigma(t)}{dt}=E\left(\varepsilon(t)+\tau_\varepsilon\frac{d\varepsilon(t)}{dt}\right), \quad (1.1)$$

where E is the Young modulus of the wall, and τ_σ, τ_ε are the relaxation parameters. Here we note that when $\tau_\sigma = \tau_\varepsilon$, the purely elastic model is retrieved. In [4,5], healthy central systemic arteries in the human were simulated with 1D integer-order viscoelastic models, and compared with measured data. However, although the integer-order model can capture the time-dependent response of the arterial wall, in [7] it was reported that this model's viscoelastic response is very sensitive to the relaxation parameters. In arterial simulations, these relaxation parameters vary significantly among patients and anatomic locations, which makes the estimation of these parameters a challenging task and hinders the use of the integer order model. Therefore, in [7–10] fractional time derivatives [11–13] are introduced to describe the layered structure of arterial walls and capture the continuous relaxation spectrum of soft tissue. In this new model, the arteries' non-local behavior in time is highlighted, i.e., the current behavior of the arterial wall has a strong dependence on its deformation history. To be specific, in the 1D case the integer-order SLS model (1.1) has been generalized by employing the so-called *spring-pot element*, which interpolates between the purely viscous and elastic behaviors, and can be interpreted as a parallel combination of a spring with a spring in series with a spring-pot. The constitutive law becomes

$$\sigma(t)+\tau_\sigma^\alpha\frac{d^\alpha\sigma(t)}{dt^\alpha}=E\left(\varepsilon(t)+\tau_\varepsilon^\alpha\frac{d^\alpha\varepsilon(t)}{dt^\alpha}\right). \quad (1.2)$$

Here we use the Riemann–Liouville definition for the fractional derivative $\frac{d^\alpha f(t)}{dt^\alpha}$ of order α $(0 \le \alpha < 1)$

$$ {}_a^{RL}\mathbf{D}_t^\alpha f(t)=\frac{1}{\Gamma(1-\alpha)}\frac{d}{dt}\int_a^t(t-\upsilon)^{-\alpha}f(\upsilon)d\upsilon, \quad (1.3)$$

which is a non-local operator that depends on the history of function $f(t)$. The Γ function is defined as $\Gamma(r)=\int_0^\infty e^{-t}t^{r-1}dt$. To investigate the interaction of this 1D model with blood flow, one-dimensional FSI arterial models were studied numerically in [1], and the simulation results indicated that the fractional order models are less sensitive to the material parameters employed compared with the integer-order-based models.

The 1D model is relatively simple but it fails to provide shear stress distributions, which strongly depend on the viscoelastic properties of the arterial wall. More importantly, it cannot represent the patient-specific geometries which are clinically important in the diagnosis and treatment of several vascular diseases [14]. Therefore, 3D studies are required. To this end, 3D fluid–structure interaction simulations have served as an effective tool in studying hemodynamics [15–24]. However, to the authors' best knowledge, no work has been done on the 3D study of fractional-order models for arterial walls.

To numerically discretize the fractional derivatives, the Grüwald–Letnikov formula [12] has been widely employed. However, upon integration to N time steps, these algorithms consume at least $O(N^2)$ operations and $O(N)$ memory due to the history effect of fractional derivatives, which hinders the application of these methods in large-scale 3D numerical simulations. To approximate the time-fractional derivatives and hence to obtain the solution of fractional-order PDEs, various algorithms have been proposed in the last 15 years. In [25,26], the decaying of the convolution kernel was exploited and the use of nested meshes was proposed, which requires $O(\log(N))$ memory. Alternatively, instead of truncating the convolution kernel, in [27] the time-fractional derivative is discretized by means of a fractional backward differentiation formula which truncates the Taylor expansion. Exploiting the Laplace transform for the convolution kernel in [28], an algorithm was proposed based on the quadrature rule. This algorithm requires $O(m)$ storage where m is the number of quadrature points. A similar approach was considered in the fast convolution method [2,29], where the authors considered a contour integral representation of the convolution kernel, with the resulted algorithm requiring $O(\log(N))$ memory. To achieve a higher-order accuracy, the fast methods was further extended into the Runge–Kutta method, by calculating multiple increments at each time step [30–32].

In this paper, we introduce fractional-order viscoelasticity in complicated arterial geometries, such as patient-specific aneurysms, and for the first time apply it to 3D fluid–structure interactions simulations. To overcome the excessive memory and computational requirements, we develop and implement efficient and accurate numerical schemes for solving fractional-order PDEs by combining the spectral element method [33–35] for spatial discretization with the *multi-patch fast convolution method* of [2,29] for the time-fractional derivatives. The fast convolution method of [2,29] achieves at most second-order accuracy in time regardless of the time integration algorithm being used, because of the linear interpolation employed. In [31] the convolution integration was approximated by the Runge–Kutta method hence achieving higher-order convergence rates. Here, we make use of the derivative estimations calculated in the high-order BDF methods, and propose a *multi-step fast convolution method* by constructing high-order interpolation. This method does not degrade the time convergence rate while estimating the convolution integrations, but instead achieves high-order accuracy.

The paper is organized as follows. In section 2 we first describe the governing fractional-order differential equations (section 2.1) and two discretization methods for the fractional-order derivative in section 2.2. In section 2.2.2, we derive a new fast convolution method and demonstrate its numerical performance in two numerical examples in sections 3.1–3.2 highlighting the convergence rate in time and its computational efficiency. In section 3.3 we apply the fractional-order arterial wall model in fluid–structure interaction simulations for a straight idealized arterial geometry, and investigate the sensitivities of the wall model on its relaxation parameters as well as on the fractional order. In 3.4 we employ a fractional-order model to study the *long-term* deformation of a patient-specific cerebral aneurysm sac. Finally, we summarize in section 4 the main findings of this study.

2. Mathematical formulation

In this section, we consider the solid problem in Lagrangian coordinates $\mathbf{X}$, and derive the formulation for the displacement field $\boldsymbol{\eta}(\mathbf{X}, t)$ in a three-dimensional object occupying the domain $\Omega_s \subset \mathbf{R}^3$ with boundary $\partial\Omega_s = \partial\Omega_{sD} \cup \partial\Omega_{sN}$. Here Dirichlet type boundary conditions are prescribed on $\partial\Omega_{sD}$, Neumann type boundary conditions (traction) on $\partial\Omega_{sN}$, and Ω_s stands for the initial configuration of the object.

2.1. Fractional-order model

The 1D fractional-order model described in (1.2) can be generalized into the 3D case:

$$\mathbf{S}(t)+\tau_\sigma^\alpha\frac{\partial^\alpha\mathbf{S}(t)}{\partial t^\alpha}=\mathbf{S}_0(t)+\tau_\varepsilon^\alpha\frac{\partial^\alpha\mathbf{S}_0(t)}{\partial t^\alpha}, \qquad (2.1)$$

where $\mathbf{S}$ is the second Piola–Kirchhoff stress tensor, α is the fractional order, and τ_σ, τ_ε are the relaxation parameters. Here we focus on the linear elastic model with infinitesimal deformations for simplicity, and the same methodology can also be applied on nonlinear materials if we take $\mathbf{S}_0$ as the stress tensor for hyperelastic materials and employ a Newton–Raphson procedure at each time step (more details can be found in [35]). The stress tensor $\mathbf{S}_0$ is then given by

$$\mathbf{S}_0(t)=\lambda\mathrm{tr}(\mathbf{E}(t))\mathbf{I}+2\mu\mathbf{E}(t),$$

where $\mathbf{E}(t)=\frac{1}{2}\left(\left(\frac{\partial\boldsymbol{\eta}}{\partial\mathbf{X}}\right)^T+\left(\frac{\partial\boldsymbol{\eta}}{\partial\mathbf{X}}\right)\right)$ is the strain tensor, and μ, λ are the shear and Lame modulus, respectively. The fractional-order model (2.1) now becomes

$$\mathbf{S}(t)+\tau_\sigma^\alpha\frac{\partial^\alpha\mathbf{S}(t)}{\partial t^\alpha}=\lambda\left(\mathrm{tr}\left(\mathbf{E}(t)+\tau_\varepsilon^\alpha\frac{\partial^\alpha\mathbf{E}(t)}{\partial t^\alpha}\right)\right)\mathbf{I}+2\mu\left(\mathbf{E}(t)+\tau_\varepsilon^\alpha\frac{\partial^\alpha\mathbf{E}(t)}{\partial t^\alpha}\right). \qquad (2.2)$$

Here we assume that both the stress tensor $\mathbf{S}(t)$ and the strain tensor $\mathbf{E}(t)$ are continuously differentiable, and the object is at a static equilibrium when $t \le 0$, i.e., $\mathbf{S}(t)$ and $\mathbf{E}(t)$ vanish before $t = 0$. Then we can set the lower limit of the fractional derivative (1.3) to be $a = 0$. Consequently, as α approaches 1, the model (2.1) converges to the integer-order SLS model

$$\mathbf{S}(t)+\tau_\sigma\frac{\partial\mathbf{S}(t)}{\partial t}=\mathbf{S}_0(t)+\tau_\varepsilon\frac{\partial\mathbf{S}_0(t)}{\partial t}.$$

Next we derive a formulation for computing the displacement field $\boldsymbol{\eta}(\mathbf{X})$. The equation describing the deformation can be expressed as follows:

$$\rho_s\frac{\partial^2\boldsymbol{\eta}}{\partial t^2}-\mathrm{div}(\mathbf{S})=\rho_s\mathbf{f},\ \ \mathrm{in}\,\Omega_s, \qquad (2.3)$$

with the external traction load $\mathbf{Sn}_s = \mathbf{T}$ applied at $\partial\Omega_{sN}$ and the static initial deformation $\boldsymbol{\eta}(\mathbf{X}, t = 0) = 0$ in Ω_s. In the above equations, $\mathbf{n}_s$, $\mathbf{f}$ and ρ_s are the normal vector pointing outward from the structure domain, the body load on the structure, and the structure density, respectively. With test functions $\mathbf{v}$, the deformation of this object is governed by the following weak formulation:

$$\int_{\Omega_s}\rho_s\frac{\partial^2\boldsymbol{\eta}}{\partial t^2}\mathbf{v}d\Omega_s+\int_{\Omega_s}\mathbf{S}:\nabla\mathbf{v}d\Omega_s-\int_{\partial\Omega_{sN}}\mathbf{T}\cdot\mathbf{v}d\Gamma-\int_{\Omega_s}\mathbf{f}\cdot\mathbf{v}d\Omega_s=0,\forall\mathbf{v}\in\mathbf{U}_0, \qquad (2.4)$$

where the displacement field $\boldsymbol{\eta}(\mathbf{X}, t) \in \mathbf{U}(t) = \{\mathbf{w}(\mathbf{X}, t) \in [H^1(\Omega_s)]^3|\mathbf{w}(\mathbf{X}, t) = \boldsymbol{\eta}_D(\mathbf{X}, t)$ on $\partial\Omega_{sD}\}$, and $\mathbf{U}_0 = \{\mathbf{w}(\mathbf{X}) \in [H^1(\Omega_s)]^3|\mathbf{w}(\mathbf{X}) = \mathbf{0}$ on $\partial\Omega_{sD}\}$.

With the above weak formulation, the structure problem is then solved by employing a continuous Galerkin formulation with Jacobi polynomials as trial and test functions. All models are implemented in our parallel spectral element method code, StressNEKTAR [34,35], which can deal with different types of 3D elements in space, including hexahedrons, prisms, tetrahedrons and pyramids. To be specific, we employ the high-order hierarchical shape functions as defined in [35] to discretize in space. On each element we expand the displacement $\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3)$ and the test function $\mathbf{v} = (v_1, v_2, v_3)$:

$$\eta_i(\mathbf{X})=\sum_{p=1}^{N_e}\hat{\eta}_{ip}\phi_p(\mathbf{X}),\qquad v_i(\mathbf{X})=\sum_{p=1}^{N_e}\hat{v}_{ip}\phi_p(\mathbf{X}),\qquad i=1,2,3, \qquad (2.5)$$

where N_e is the total number of shape functions for displacement on this element, $\phi_p(\mathbf{X})$ are the shape functions, and $\hat{\eta}_{ip}$, $\hat{v}_{ip}$ are the expansion coefficients. We then have the local vector of unknown expansion coefficients for displacement on each element

$$\begin{gathered}[(\mathbf{U}_1)^T\cdots,(\mathbf{U}_p)^T,(\mathbf{U}_{p+1})^T,\cdots,(\mathbf{U}_{N_e})^T]^T,\\ \text{where}\,\mathbf{U}_p=[\hat{\eta}_{1p},\hat{\eta}_{2p},\hat{\eta}_{3p}]\end{gathered} \qquad (2.6)$$

with length $3N_e$. Here the coefficient vector is organized in a such a way that the three displacement components corresponding to a particular mode are adjacent, so as to have smaller bandwidth for the coefficient matrix [34]. Since the interior modes are coupled with the boundary modes of the same element only, while assembling the local vector (2.6) into the global vector **U**, we arrange the modes following the "vertex-edge-face-interior" sequence as in [33]. With expression (2.5), one can then also obtain the global vector of the external forces **F**, the mass matrix **M**, and the stiffness matrix **K** for stress tensor $\mathbf{S}_0$. On each element Ω_e, the local vector of the external forces is a vector of length $3N_e$, with components

$$\hat{F}_{ip}=\int_{\partial(\Omega_e)_N}(T_e)_i\phi_p d\Gamma_e+\int_{\Omega_e}(f_e)_i\phi_p d\Omega_e,\ i=1,2,3, \qquad (2.7)$$

where $(T_e)_i$ and $(f_e)_i$ are the components of the external traction and body forces applied on this element, respectively. Based on (2.7), the global vector of the external forces **F** is then organized in the same ordering of indices as **U**. Correspondingly, on each element we have a $3N_e \times 3N_e$ local mass matrix, which is organized as a matrix of $N_e \times N_e$ blocks, with each block being a 3 × 3 submatrix:

$$\begin{bmatrix}(\mathbf{M}_e)_{11} & \cdots & (\mathbf{M}_e)_{1N_e}\\ \vdots & \ddots & \vdots\\ (\mathbf{M}_e)_{N_e1} & \cdots & (\mathbf{M}_e)_{N_eN_e}\end{bmatrix},\ \text{where}\,(\mathbf{M}_e)_{pq}=\left(\int_{\Omega_e}\rho_s\phi_p\phi_q d\Omega_e\right)\mathbf{I}. \qquad (2.8)$$

Here **I** is the 3 × 3 identity matrix. Similarly, on each element we can organize the $3N_e \times 3N_e$ stiffness matrix for the stress tensor $\mathbf{S}_0$ into $N_e \times N_e$ blocks, with each 3 × 3 submatrix being

$$\int_{\Omega_e}\begin{bmatrix}(\lambda+\mu)\frac{\partial\phi_p}{\partial x}\frac{\partial\phi_q}{\partial x}+\beta & \lambda\frac{\partial\phi_p}{\partial x}\frac{\partial\phi_q}{\partial y}+\mu\frac{\partial\phi_p}{\partial y}\frac{\partial\phi_q}{\partial x} & \lambda\frac{\partial\phi_p}{\partial x}\frac{\partial\phi_q}{\partial z}+\mu\frac{\partial\phi_p}{\partial z}\frac{\partial\phi_q}{\partial x}\\ \lambda\frac{\partial\phi_p}{\partial y}\frac{\partial\phi_q}{\partial x}+\mu\frac{\partial\phi_p}{\partial x}\frac{\partial\phi_q}{\partial y} & (\lambda+\mu)\frac{\partial\phi_p}{\partial y}\frac{\partial\phi_q}{\partial y}+\beta & \lambda\frac{\partial\phi_p}{\partial y}\frac{\partial\phi_q}{\partial z}+\mu\frac{\partial\phi_p}{\partial z}\frac{\partial\phi_q}{\partial y}\\ \lambda\frac{\partial\phi_p}{\partial z}\frac{\partial\phi_q}{\partial x}+\mu\frac{\partial\phi_p}{\partial x}\frac{\partial\phi_q}{\partial z} & \lambda\frac{\partial\phi_p}{\partial z}\frac{\partial\phi_q}{\partial y}+\mu\frac{\partial\phi_p}{\partial y}\frac{\partial\phi_q}{\partial z} & (\lambda+\mu)\frac{\partial\phi_p}{\partial z}\frac{\partial\phi_q}{\partial z}+\beta\end{bmatrix}d\Omega_e, \qquad (2.9)$$

where $\beta=\mu\left(\frac{\partial\phi_p}{\partial x}\frac{\partial\phi_q}{\partial x}+\frac{\partial\phi_p}{\partial y}\frac{\partial\phi_q}{\partial y}+\frac{\partial\phi_p}{\partial z}\frac{\partial\phi_q}{\partial z}\right)$. With the local mass matrix (2.8) and the local stiffness matrix (2.9), one can then assemble the global mass matrix **M** and the global stiffness matrix **K**, respectively. For more details, we refer the interested readers to [36].

To discretize the second order time derivative $\frac{\partial^2 \eta}{\partial t^2}$, two types of methods are implemented in StressNEKTAR: the Newmark scheme and a three-step backward differentiation formula (BDF) [37]. They can be defined by different expressions for the displacement and velocity at the next time step. Let n denote the index of the time step, in the *Newmark scheme*, based on the current displacement $\boldsymbol{\eta}^n$, previous time step displacement $\boldsymbol{\eta}^{n-1}$, velocity $\dot{\boldsymbol{\eta}}^{n-1}$ and acceleration $\ddot{\boldsymbol{\eta}}^{n-1}$, we can approximate the acceleration and velocity at the n-th step as:

$$\dot{\boldsymbol{\eta}}^n = \dot{\boldsymbol{\eta}}^{n-1} + \Delta t[(1-\theta_1)\ddot{\boldsymbol{\eta}}^{n-1} + \theta_1 \ddot{\boldsymbol{\eta}}^n], \quad \text{(2.10a)}$$

$$\ddot{\boldsymbol{\eta}}^n = \frac{1}{\Delta t^2 \theta_2}\left(\boldsymbol{\eta}^n - \boldsymbol{\eta}^{n-1} - \Delta t \dot{\boldsymbol{\eta}}^{n-1} - \frac{\Delta t^2}{2}(1-2\theta_2)\ddot{\boldsymbol{\eta}}^{n-1}\right), \quad \text{(2.10b)}$$

where we have used "·" to denote the temporal derivatives and Δt is the time step size. Depending on the requirements of accuracy and stability, different values can be set to the parameters (θ_1, θ_2). In the following, we use the parameters (θ_1, θ_2) = (0.5, 0.25).

In the *three-step BDF scheme*, the acceleration and velocity approximations are also expressed based on the current information ($\boldsymbol{\eta}^n$, $\dot{\boldsymbol{\eta}}^n$) and previous time steps results ($\boldsymbol{\eta}^{n-i}$, $\dot{\boldsymbol{\eta}}^{n-i}$) as:

$$\dot{\boldsymbol{\eta}}^n = A_1 \boldsymbol{\eta}^n + B_1 \boldsymbol{\eta}^{n-1} + C_1 \boldsymbol{\eta}^{n-2} + D_1 \boldsymbol{\eta}^{n-3}, \quad \text{(2.11a)}$$

$$\ddot{\boldsymbol{\eta}}^n = A_2 \dot{\boldsymbol{\eta}}^n + B_2 \dot{\boldsymbol{\eta}}^{n-1} + C_2 \dot{\boldsymbol{\eta}}^{n-2} + D_2 \dot{\boldsymbol{\eta}}^{n-3}. \quad \text{(2.11b)}$$

Here A_i, B_i, C_i and D_i (i = 1, 2) are related following the rule

$$A_1 = \frac{\theta_1}{\Delta t}, \ B_1 = \frac{2.5 - 3\theta_1}{\Delta t}, \ C_1 = \frac{3\theta_1 - 4}{\Delta t}, \ D_1 = \frac{1.5 - \theta_1}{\Delta t}, \quad \text{(2.12a)}$$

$$A_2 = \frac{\theta_2}{\Delta t}, \ B_2 = \frac{2.5 - 3\theta_2}{\Delta t}, \ C_2 = \frac{3\theta_2 - 4}{\Delta t}, \ D_2 = \frac{1.5 - \theta_2}{\Delta t}, \quad \text{(2.12b)}$$

and (θ_1, θ_2) are also decided based on considerations of stability and dissipation. When taking (θ_1, θ_2) = (0, 0), the three-step BDF scheme reaches its highest accuracy, i.e., the third-order convergence in time. In the numerical examples we will also test the case with (θ_1, θ_2) = (5/3, 5/3) because of its enhanced stability [37]. This pair of parameters corresponds to a three-step second-order scheme.

2.2. Discretization methods for fractional derivative

2.2.1. Grüwald–Letnikov formula—While considering the temporal discretization for fractional derivative, based on the assumption that **S**(*t*) and **E**(*t*) are both continuously differentiable, the Riemann–Liouville fractional derivative (1.3) is equivalent to the Grüwald–Letnikov fractional derivative [12]

$$
{}_{a}^{G}\mathbf{D}_{t}^{\alpha}\, f(t)=\lim_{\Delta t\to 0^{+}}\frac{1}{\Delta t^{\alpha}}\sum_{r=0}^{(t-a)/\Delta t}(-1)^{r}\begin{pmatrix}\alpha\\ r\end{pmatrix}f(t-r\Delta t). \qquad (2.13)
$$

This fact enables a popular way of discretizing the fractional derivative, i.e., the Grüwald–Letnikov formula which is of first order accuracy in time:

$$
{}_{a}^{RL}\mathbf{D}_{t}^{\alpha}\, f(t)=\lim_{\Delta t\to 0^{+}}\frac{1}{\Delta t^{\alpha}}\sum_{r=0}^{(t-a)/\Delta t}GL_{r}^{\alpha}\, f(t-r\Delta t) \qquad (2.14)
$$

with

$$
GL_{0}^{\alpha}=1, \quad GL_{r}^{\alpha}=\frac{r-\alpha-1}{r}GL_{r-1}^{\alpha}. \qquad (2.15)
$$

Substituting (2.14) into (2.1), at the *n*-th time step we obtain that

$$
\mathbf{S}^{n}+\frac{\tau_{\sigma}^{\alpha}}{\Delta t^{\alpha}}\left(\sum_{r=0}^{n}GL_{r}^{\alpha}\mathbf{S}^{n-r}\right)=\mathbf{S}_{0}^{n}+\frac{\tau_{\varepsilon}^{\alpha}}{\Delta t^{\alpha}}\left(\sum_{r=0}^{n}GL_{r}^{\alpha}\mathbf{S}_{0}^{n-r}\right) \qquad (2.16)
$$

which yields

$$
\mathbf{S}^{n}=\frac{\Delta t^{-\alpha}}{1+\tau_{\sigma}^{\alpha}\Delta t^{-\alpha}}\sum_{r=1}^{n}GL_{r}^{\alpha}(\tau_{\varepsilon}^{\alpha}\mathbf{S}_{0}^{n-r}-\tau_{\sigma}^{\alpha}\mathbf{S}^{n-r})+\frac{1+\tau_{\varepsilon}^{\alpha}\Delta t^{-\alpha}}{1+\tau_{\sigma}^{\alpha}\Delta t^{-\alpha}}\mathbf{S}_{0}^{n}. \qquad (2.17)
$$

Therefore, the weak formulation (2.4) can be rewritten as

$$\begin{aligned}&\int\limits_{\Omega_s}\rho_s\frac{\partial^2\boldsymbol{\eta}^n}{\partial t^2}\mathbf{v}d\Omega_s+\frac{1+\tau_\varepsilon^\alpha\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\int\limits_{\Omega_s}\mathbf{S}_0^n:\nabla\mathbf{v}d\Omega_s\\&+\frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\sum_{r=1}^{n}GL_r^\alpha\left(\tau_\varepsilon^\alpha\int\limits_{\Omega_s}\mathbf{S}_0^{n-r}:\nabla\mathbf{v}d\Omega_s-\tau_\sigma^\alpha\int\limits_{\Omega_s}\mathbf{S}^{n-r}:\nabla\mathbf{v}d\Omega_s\right)\\&=\int\limits_{\partial\Omega_{sN}}\mathbf{T}^n\cdot\mathbf{v}d\Gamma+\int\limits_{\Omega_s}\mathbf{f}^n\cdot\mathbf{v}d\Omega_s.\end{aligned}\qquad(2.18)$$

One can now approximate the acceleration $\frac{\partial^2\boldsymbol{\eta}}{\partial t^2}$ at the *n*-th time step, with either the Newmark scheme (2.10) or the three-step backward differentiation formula (BDF) (2.11). Let *n* denote the index of the time step, in the *Newmark scheme* we obtain the temporal discretized formulation:

$$\begin{aligned}&\int\limits_{\Omega_s}\rho_s\frac{\boldsymbol{\eta}^n}{\Delta t^2\theta_2}\mathbf{v}d\Omega_s+\frac{1+\tau_\varepsilon^\alpha\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\int\limits_{\Omega_s}\mathbf{S}_0^n:\nabla\mathbf{v}d\Omega_s\\&=\int\limits_{\partial\Omega_{sN}}\mathbf{T}^n\cdot\mathbf{v}d\Gamma+\int\limits_{\Omega_s}\mathbf{f}^n\cdot\mathbf{v}d\Omega_s+\int\limits_{\Omega_s}\frac{\rho_s}{\Delta t^2\theta_2}\left(\boldsymbol{\eta}^{n-1}+\Delta t\dot{\boldsymbol{\eta}}^{n-1}+\frac{\Delta t^2}{2}(1-2\theta_2)\ddot{\boldsymbol{\eta}}^{n-1}\right)\mathbf{v}d\Omega_s\\&-\frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\sum_{r=1}^{n}GL_r^\alpha\left(\tau_\varepsilon^\alpha\int\limits_{\Omega_s}\mathbf{S}_0^{n-r}:\nabla\mathbf{v}d\Omega_s-\tau_\sigma^\alpha\int\limits_{\Omega}\mathbf{S}^{n-r}:\nabla\mathbf{v}d\Omega_s\right).\end{aligned}\qquad(2.19)$$

We then obtain a linear system for the coefficient vector **U**:

$$\left(\frac{1}{\Delta t^2\theta_2}\mathbf{M}+\frac{1+\tau_\varepsilon^\alpha\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\mathbf{K}\right)\mathbf{U}^n=\mathbf{F}^n+\mathbf{G}^n\qquad(2.20)$$

where the internal load vector $\mathbf{G}^n$ is also organized in the same ordering of indices as **U**, with the element-wise component as

$$\begin{aligned}&\hat{G}_{ip}^n=\int\limits_{\Omega_e}\frac{\rho_s}{\Delta t^2\theta_2}\left(\eta_i^{n-1}+\Delta t\dot{\eta}_i^{n-1}+\frac{\Delta t^2}{2}(1-2\theta_2)\ddot{\eta}_i^{n-1}\right)\phi_p d\Omega_e\\&-\frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\sum_{r=1}^{n}GL_r^\alpha\left(\tau_\varepsilon^\alpha\int\limits_{\Omega_e}\mathbf{S}_0^{n-r}:\nabla\phi_p d\Omega_e-\tau_\sigma^\alpha\int\limits_{\Omega_e}\mathbf{S}^{n-r}:\nabla\phi_p d\Omega_e\right)_i,\quad i=1,2,3.\end{aligned}$$

Similarly, in the *three-step BDF scheme* we obtain the temporal discretized formulation:

$$\begin{aligned}&\int\limits_{\Omega_s}\rho_s A_1A_2\boldsymbol{\eta}^n\mathbf{v}d\Omega_s+\frac{1+\tau_\varepsilon^\alpha\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\int\limits_{\Omega_s}\mathbf{S}_0^n:\nabla\mathbf{v}d\Omega_s\\&=\int\limits_{\partial\Omega_{sN}}\mathbf{T}^n\cdot\mathbf{v}d\Gamma+\int\limits_{\Omega_s}\mathbf{f}^n\cdot\mathbf{v}d\Omega_s\\&-\int\limits_{\Omega_s}\rho_s(A_2(B_1\boldsymbol{\eta}^{n-1}+C_1\boldsymbol{\eta}^{n-2}+D_1\boldsymbol{\eta}^{n-3})+(B_2\dot{\boldsymbol{\eta}}^{n-1}+C_2\dot{\boldsymbol{\eta}}^{n-2}+D_2\dot{\boldsymbol{\eta}}^{n-3}))\mathbf{v}d\Omega_s\\&-\frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\sum_{r=1}^{n}GL_r^\alpha\left(\tau_\varepsilon^\alpha\int\limits_{\Omega_s}\mathbf{S}_0^{n-r}:\nabla\mathbf{v}d\Omega_s-\tau_\sigma^\alpha\int\limits_{\Omega_s}\mathbf{S}^{n-r}:\nabla\mathbf{v}d\Omega_s\right),\end{aligned}\qquad(2.21)$$

where A_i, B_i, C_i and D_i ($i = 1, 2$) are defined as in (2.12). It can therefore also be formulated into the vector form:

$$\left(A_1 A_2 \mathbf{M} + \frac{1+\tau_\varepsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \mathbf{K}\right) \mathbf{U}^n = \mathbf{F}^n + \mathbf{G}^n. \tag{2.22}$$

The internal load vector components are

$$\begin{aligned}\hat{G}_{ip}^n = & -\int_{\Omega_e} \rho_s \left(A_2\left(B_1 \eta_i^{n-1} + C_1 \eta_i^{n-2} + D_1 \eta_i^{n-3}\right) + \left(B_2 \dot{\eta}_i^{n-1} + C_2 \dot{\eta}_i^{n-2} + D_2 \dot{\eta}_i^{n-3}\right)\right) \phi_p d\Omega_e \\ & - \frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \sum_{r=1}^{n} GL_r^\alpha \left(\tau_\varepsilon^\alpha \int_{\Omega_e} \mathbf{S}_0^{n-r} : \nabla \phi_p d\Omega_e - \tau_\sigma^\alpha \int_{\Omega_e} \mathbf{S}^{n-r} : \nabla \phi_p d\Omega_e\right)_i, \quad i=1,2,3.\end{aligned} \tag{2.23}$$

In each time step, we can then calculate and store the memory load as

$$\begin{aligned}\widehat{HIS}_{ip}^n = & \frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \left(\tau_\varepsilon^\alpha \int_{\Omega_e} \mathbf{S}_0^n : \nabla \phi_p d\Omega_e - \tau_\sigma^\alpha \int_{\Omega_e} \mathbf{S}^n : \nabla \phi_p d\Omega_e\right)_i \\ = & \frac{\tau_\varepsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \left(\int_{\Omega_e} \mathbf{S}_0^n : \nabla \phi_p d\Omega_e\right)_i - \frac{\tau_\sigma^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \left(\int_{\Omega_e} \mathbf{S}^n : \nabla \phi_p d\Omega_e\right)_i \\ = & \frac{\tau_\varepsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \left(\int_{\Omega_e} \mathbf{S}_0^n : \nabla \phi_p d\Omega_e\right)_i - \frac{\tau_\sigma^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \left(\frac{1+\tau_\varepsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \left(\int_{\Omega_e} \mathbf{S}_0^n : \nabla \phi_p d\Omega_e\right)_i \right. \\ & \left. + \frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \sum_{r=1}^{n} GL_r^\alpha \left(\tau_\varepsilon^\alpha \int_{\Omega_e} \mathbf{S}_0^{n-r} : \nabla \phi_p d\Omega_e - \tau_\sigma^\alpha \int_{\Omega_e} \mathbf{S}^{n-r} : \nabla \phi_p d\Omega_e\right)_i\right) \\ = & \left(\frac{\tau_\varepsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} - \frac{\tau_\sigma^\alpha \Delta t^{-\alpha}\left(1+\tau_\varepsilon^\alpha \Delta t^{-\alpha}\right)}{\left(1+\tau_\sigma^\alpha \Delta t^{-\alpha}\right)^2}\right) \left(\int_{\Omega_e} \mathbf{S}_0^n : \nabla \phi_p d\Omega_e\right)_i \\ & - \frac{\tau_\sigma^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \sum_{r=1}^{n} GL_r^\alpha \widehat{HIS}_{ip}^{n-r}, \quad i=1,2,3.\end{aligned} \tag{2.24}$$

Assembling the above element-wise components into the global vector, it yields the recursive formulation for

$$\mathbf{HIS}^n = \left(\frac{\tau_\varepsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} - \frac{\tau_\sigma^\alpha \Delta t^{-\alpha}\left(1-\tau_\varepsilon^\alpha \Delta t^{-\alpha}\right)}{\left(1+\tau_\sigma^\alpha \Delta t^{-\alpha}\right)^2}\right) \mathbf{K}\mathbf{U}^n - \frac{\tau_\sigma^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \sum_{r=1}^{n} GL_r^\alpha \mathbf{HIS}^{n-r}. \tag{2.25}$$

Starting from the static equilibrium, we solve the system with Newmark scheme (2.20) for the first two steps and with BDF scheme (2.22) for the rest steps, following the algorithm:

- Set

$$\mathbf{U}^0 = 0,\ \dot{\mathbf{U}}^0 = 0,\ \ddot{\mathbf{U}}^0 = 0,\ \mathbf{HIS}^0 = 0$$

for the n-th time step, **do**

1. When $n \le 2$, solve for $\mathbf{U}^n$ with the Newmark scheme

$$\left(\frac{1}{\Delta t^2\theta_2}\mathbf{M}+\frac{1+\tau_\varepsilon^\alpha\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\mathbf{K}\right)\mathbf{U}^n=\mathbf{F}^n-\sum_{r=1}^{n}GL_r^\alpha\mathbf{HIS}^{n-r}+\frac{\mathbf{M}}{\Delta t^2\theta_2}\left(\mathbf{U}^{n-1}+\Delta t\dot{\mathbf{U}}^{n-1}+\frac{\Delta t^2}{2}(1-2\theta_2)\ddot{\mathbf{U}}^{n-1}\right);$$

(2.26)

else, solve with the three-step BDF scheme

$$\left(A_1A_2\mathbf{M}+\tfrac{1+\tau_\varepsilon^\alpha\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha\Delta t^{-\alpha}}\mathbf{K}\right)\mathbf{U}^n$$
$$=\mathbf{F}^n-\sum_{r=1}^{n}GL_r^\alpha\mathbf{HIS}^{n-r}-(A_2(B_1\mathbf{U}^{n-1}+C_1\mathbf{U}^{n-2}+D_1\mathbf{U}^{n-3})+(B_2\dot{\mathbf{U}}^{n-1}+C_2\dot{\mathbf{U}}^{n-2}+D_2\dot{\mathbf{U}}^{n-3}))\mathbf{M}.$$

(2.27)

2. Calculate and store the memory load $\mathbf{HIS}^n$ following (2.25).

3. Approximate $\dot{\mathbf{U}}^n$ and $\ddot{\mathbf{U}}^n$ following the Newmark scheme (2.10) or the three-step BDF scheme (2.11).

- Go to time step $n+1$.

Because the Grüwald–Letnikov formula (2.14) introduces an $O(\Delta t)$ error, the above Grüwald–Letnikov algorithm should be of first accuracy in time. Regarding the efficiency, at the n-th time step, for evaluating and storing the accumulated term in memory load HIS^n it requires $O(n)$ operations. With $N = T/\Delta t$ numbers of total time steps, this Grüwald–Letnikov formula consumes $O(N^2)$ operations and $O(N)$ memory in evaluating the accumulated right-hand-side from deformation history.

2.2.2. Fast convolution method—The Grüwald–Letnikov formula is limited by its first-order accuracy and the computational bottleneck of $O(N^2)$ operations and $O(N)$ memory in evaluating **HIS**, especially in long-term time integration as in the FSI problems of interest here. To improve both the accuracy as well as the efficiency, we now convert equation (2.1) to the convolution form. Applying the Laplace transform $\mathscr{L}$ to (2.1), we have

$$(1+\tau_\sigma^\alpha s^\alpha)\mathscr{L}(\mathbf{S})(s)=(1+\tau_\varepsilon^\alpha s^\alpha)\mathscr{L}(\mathbf{S}_0)(s), \quad (2.28)$$

which yields

$$\begin{aligned}\mathscr{L}(\mathbf{S})(s)=&\left(\tfrac{1+\tau_\varepsilon^\alpha s^\alpha}{1+\tau_\sigma^\alpha s^\alpha}\right)\mathscr{L}(\mathbf{S}_0)(s)\\=&\left(\left(\tfrac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha+\tfrac{1}{\tau_\sigma^\alpha}\left(1-\left(\tfrac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha\right)\tfrac{1}{\tau_\sigma{}^{-\alpha}+s^\alpha}\right)\mathscr{L}(\mathbf{S}_0)(s).\end{aligned} \tag{2.29}$$

Then taking the inverse Laplace transform, we get the relation between **S**(*t*) and **S**$_0$(*t*)

$$\mathbf{S}(t)=\left(\frac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha\mathbf{S}_0(t)+\frac{1}{\tau_\sigma^\alpha}\left(1-\left(\frac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha\right)\int_0^t I(t-r)\mathbf{S}_0(r)dr, \tag{2.30}$$

where

$$I(t):=t^{\alpha-1}E_{\alpha,\alpha}\left(-\left(\frac{t}{\tau_\sigma}\right)^\alpha\right),\quad \mathscr{L}(I)(s)=\frac{1}{\tau_\sigma^{-\alpha}+s^\alpha},$$

and $E_{\beta_1,\beta_2}(t)$ is the two parameter Mittag–Leffler function [12], which can be seen as the fractional generalization of the exponential. By substituting the relaxation form of constitutive law (2.30) and discretizing in space, we obtain the discretized form

$$\mathbf{M}\ddot{\mathbf{U}}^n+\frac{\tau_\varepsilon^\alpha}{\tau_\varepsilon^\alpha}\mathbf{K}\mathbf{U}^n=\frac{1}{\tau_\sigma^\alpha}\left(\left(\frac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha-1\right)\mathbf{K}\int_0^{t^n}I(t^n-r)\mathbf{U}(r)dr+\mathbf{F}^n. \tag{2.31}$$

The Mittag–Leffler function exhibits a much faster decay than the exponential function for small time *t*. On the other hand, when $t\rightarrow\infty$, this function has a heavy tail and its decay is only algebraic. Therefore, although we may directly calculate the Mittag–Leffler function to compute the convolution integral $\int_0^t I(t-r)\mathbf{U}(r)dr$ in (2.31), such heavy tail behavior would prohibit us from alleviating the computational complexity.

The methods introduced in [2,29] provide an effective alternative to evaluate such convolution integrals. As illustrated in Fig. 1, the idea is to decompose the integral range $\int_0^{t^n}$ with base 2:

$$\int_0^{n\Delta t}=\int_0^{2^{m_1}\Delta t}+\int_{2^{m_1}\Delta t}^{(2^{m_1}+2^{m_2})\Delta t}+\ldots+\int_{(n-1)\Delta t}^{n\Delta t} \tag{2.32}$$

where $m_1 = [log_2(n)]$, $m_2 = [log_2(n - 2^{m_1})]$, etc. For example, when $n = 7$, the integral $\int_0^{7\Delta t}I(7\Delta t-r)\mathbf{U}(r)dr$ can be divided into 3 parts

$$\int_{0}^{7\Delta t} I(7\Delta t-r)\mathbf{U}(r)dr=\int_{0}^{4\Delta t} I(7\Delta t-r)\mathbf{U}(r)dr+\int_{4\Delta t}^{6\Delta t} I(7\Delta t-r)\mathbf{U}(r)dr+\int_{6\Delta t}^{7\Delta t} I(7\Delta t-r)\mathbf{U}(r)dr;$$

when $n = 8$, we have 4 parts

$$\int_{0}^{8\Delta t} I(8\Delta t-r)\mathbf{U}(r)dr=\int_{0}^{4\Delta t} I(8\Delta t-r)\mathbf{U}(r)dr+\int_{4\Delta t}^{6\Delta t} I(8\Delta t-r)\mathbf{U}(r)dr+\int_{6\Delta t}^{7\Delta t} I(8\Delta t-r)\mathbf{U}(r)dr\int_{7\Delta t}^{8\Delta t} I(8\Delta t-r)\mathbf{U}(r)dr.$$

To approximate the l-th part of the convolution integral, we apply the trapezoidal rule to a parametrization of the contour integral for the inverse Laplace transform:

$$I(t)=\frac{1}{2\pi i}\int_{\Gamma_l}\mathscr{L}(I)(\gamma)e^{t\gamma}d\gamma \approx \sum_{j=-L}^{L}\omega_j^{(l)}\mathscr{L}(I)(\gamma_j^{(l)})e^{t\gamma_j^{(l)}}, \qquad (2.33)$$

where the complex contour Γ_l is chosen to be the Talbot contour [38,39], and $\omega_j^{(l)}$ and $\gamma_j^{(l)}$ are the weights and quadrature points on each Talbot contour:

$$\omega_j^{(l)}=-\frac{j}{4\pi(L+1)i}\frac{d\varpi_l}{d\kappa}(\kappa_j), \quad \gamma_j^{(l)}=\varpi_l(\kappa_j), \quad \kappa_j=\frac{j\pi}{L+1}.$$

The function $\bar{\omega}_l(\kappa)$ is chosen such that the singularities of $\mathscr{L}(I)(s)$ lie in the contour [29]. Note that the Talbot contours can vary step by step if an adaptive algorithm [2] is employed. However, in the current method, we generate the quadrature points for all $O(\log N)$ numbers of Talbot contours before the first time step (pre-processing cost only). While computing the last part of the convolution integral in (2.32), we improve the linear approximation proposed in [29] by making a full use of the velocity $\dot{\mathbf{U}}^{n-1}$ and acceleration $\ddot{\mathbf{U}}^{n-1}$ information approximated following the Newmark scheme (2.10) or the three-step BDF scheme (2.11):

$$\begin{aligned}\int_{(n-1)\Delta t}^{n\Delta t} I(n\Delta t-r)\mathbf{U}(r)dr \approx \int_{(n-1)\Delta t}^{n\Delta t} I(n\Delta t-r)dr\mathbf{U}^{n-1}+\int_{(n-1)\Delta t}^{n\Delta t}(r-(n-1)\Delta t)I(n\Delta t-r)dr\dot{\mathbf{U}}^{n-1}+\int_{(n-1)\Delta t}^{n\Delta t}\tfrac{(r-(n-1)\Delta t)^2}{2}I(n\Delta t-r)dr\\ =H_1\mathbf{U}^{n-1}+H_2\dot{\mathbf{U}}^{n-1}+H_3\ddot{\mathbf{U}}^{n-1}\end{aligned}$$

(2.34)

which introduces an $O(\Delta t^4)$ error. Here the integrals can be calculated a priori as the inverse Laplace transforms

$$
\begin{aligned}
H_1 &= \int_{(n-1)\Delta t}^{n\Delta t} I(n\Delta t-r)dr=\int_0^{\Delta t} I(r)dr\\
&\approx \sum_{j=-L}^{L} \omega_j^{(0)} \mathscr{L}\left(\int_0^{\Delta t} I(r)dr\right)(\gamma_j^{(0)})e^{t\gamma_j^{(0)}}\\
&= \sum_{j=-L}^{L} \omega_j^{(0)} \frac{1}{\gamma_j^{(0)}\left(\tau_\sigma^{-\alpha}+\left(\gamma_j^{(0)}\right)^{\alpha}\right)}e^{t\gamma_j^{(0)}}\\
H_2 &= \int_{(n-1)\Delta t}^{n\Delta t} (r-(n-1)\Delta t)I(n\Delta t-r)dr=\int_0^{\Delta t} rI(\Delta t-r)dr\\
&\approx \sum_{j=-L}^{L} \omega_j^{(0)} \mathscr{L}\left(\int_0^{\Delta t} rI(\Delta t-r)dr\right)(\gamma_j^{(0)})e^{t\gamma_j^{(0)}}\\
&= \sum_{j=-L}^{L} \omega_j^{(0)} \frac{1}{\left(\gamma_j^{(0)}\right)^2\left(\tau_\sigma^{-\alpha}+\left(\gamma_j^{(0)}\right)^{\alpha}\right)}e^{t\gamma_j^{(0)}}\\
H_3 &= \int_{(n-1)\Delta t}^{n\Delta t} \frac{(r-(n-1)\Delta t)^2}{2}I(n\Delta t-r)dr=\frac{1}{2}\int_0^{\Delta t} r^2I(\Delta t-r)dr\\
&\approx \frac{1}{2}\sum_{j=-L}^{L} \omega_j^{(0)} \mathscr{L}\left(\int_0^{\Delta t} r^2I(\Delta t-r)dr\right)(\gamma_j^{(0)})e^{t\gamma_j^{(0)}}\\
&=\frac{1}{2}\sum_{j=-L}^{L} \omega_j^{(0)} \frac{1}{\left(\gamma_j^{(0)}\right)^3\left(\tau_\sigma^{-\alpha}+\left(\gamma_j^{(0)}\right)^{\alpha}\right)}e^{t\gamma_j^{(0)}}
\end{aligned}
$$

and the weights as well as the quadrature points correspond to the Talbot contour chosen for $t = \Delta t$. For the rest of the convolution integrals $\int_a^b I(n\Delta t-r)\mathbf{U}(r)dr$ in (2.32), we can approximate them on the corresponding Talbot contour Γ_l

$$
\begin{aligned}
\int_a^b I(n\Delta t-r)\mathbf{U}(r)dr &= \int_a^b \frac{1}{2\pi i}\int_{\Gamma_l} \frac{1}{\tau_\sigma^{-\alpha}+\gamma^{\alpha}}e^{(n\Delta t-r)\gamma}d\gamma\mathbf{U}(r)dr\\
&=\frac{1}{2\pi i}\int_{\Gamma_l} \frac{1}{\tau_\sigma^{-\alpha}+\gamma^{\alpha}}e^{(n\Delta t-b)\gamma}\int_a^b e^{(b-r)\gamma}\mathbf{U}(r)\,drd\gamma\\
&\approx \sum_{j=-L}^{L} \omega_j^{(l)} \frac{1}{\tau_\sigma^{-\alpha}+\left(\gamma_j^{(l)}\right)^{\alpha}}e^{(n\Delta t-b)\gamma_j^{(l)}}\int_a^b e^{(b-r)\gamma_j^{(l)}}\mathbf{U}(r)dr,
\end{aligned}
\qquad (2.35)
$$

where the inner integral $\int_a^b e^{(b-r)\gamma_j^{(l)}}\mathbf{U}(r)dr$ can be updated recursively by solving

$$
\dot{y}\;[b,a,\gamma_j^{(l)}](r)=\gamma_j^{(l)}y[b,a,\gamma_j^{(l)}](r)+\mathbf{U}(r), \quad y(a)=0. \qquad (2.36)
$$

In each time step, to integrate for $y[b,a,\gamma_j^{(l)}](r)$ we update $y^n \approx y[b,a,\gamma_j^{(l)}](n\Delta t)$ recursively after obtaining the approximated displacement $\mathbf{U}^n$, velocity $\dot{\mathbf{U}}^n$ and acceleration $\ddot{\mathbf{U}}^n$:

$$\begin{aligned} y^n=&e^{\gamma_j^{(l)}\Delta t}y^{n-1}+\int_{(n-1)\Delta t}^{n\Delta t} e^{\gamma_j^{(l)}(n\Delta t-r)}\mathbf{U}(r)dr\\ \approx& e^{\gamma_j^{(l)}\Delta t}y^{n-1}+\int_0^{\Delta t} e^{\gamma_j^{(l)}(\Delta t-r)}\left(\mathbf{U}^n-r\dot{\mathbf{U}}^n+\tfrac{r^2}{2}\ddot{\mathbf{U}}^n\right)dr\\ =&e^{\gamma_j^{(l)}\Delta t}y^{n-1}+\frac{e^{\gamma_j^{(l)}\Delta t}-1}{\gamma_j^{(l)}}\mathbf{U}^n-\frac{e^{\gamma_j^{(l)}\Delta t}(\gamma_j^{(l)}\Delta t-1)+1}{(\gamma_j^{(l)})^2}\dot{\mathbf{U}}^n+\frac{e^{\gamma_j^{(l)}\Delta t}((\gamma_j^{(l)}\Delta t)^2-2\gamma_j^{(l)}\Delta t+2)-2}{2(\gamma_j^{(l)})^3}\ddot{\mathbf{U}}^n. \end{aligned} \tag{2.37}$$

Therefore, time derivatives obtained from the Newmark scheme (2.10) or the three-step BDF scheme (2.11), improve the piecewise linear interpolant used in [29] which introduces an $O(\Delta t^2)$ error. Higher-order interpolation of $\mathbf{U}(r)$ in (2.37) is more effective in long-term computation because it only gives $O(\Delta t^4)$ error at each time step which yields an $O(\Delta t^3)$ error accumulatively. This accuracy could be further improved by using higher-order BDF schemes. Since the Talbot contours are fixed, the computational cost in our algorithm for approximating the convolution integrals (2.32) is:

- $O(N \log N)$ multiplications,
- $O(\log N)$ evaluations of the Laplace transform $\mathcal{L}(I)(s)$,
- $O(\log N)$ active memory.

where N is the total number of time-history steps.

Combining (2.34) and (2.35), we can rewrite (2.38)

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{U}}^n+\tfrac{\tau_\varepsilon^\alpha}{\tau_\varepsilon^\alpha}\mathbf{K}\mathbf{U}^n=&\mathbf{F}^n+\tfrac{1}{\tau_\sigma^\alpha}\left(\left(\tfrac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha-1\right)\mathbf{K}\int_0^{t^n} I(t^n-r)\mathbf{U}(r)dr\\ =&\mathbf{F}^n+\tfrac{1}{\tau_\sigma^\alpha}\left(\left(\tfrac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha-1\right)\mathbf{K}\left(H_1\mathbf{U}^{n-1}+H_2\dot{\mathbf{U}}^{n-1}+H_3\ddot{\mathbf{U}}^{n-1}+\sum_l\left(\sum_{j=-L}^{L}\omega_j^{(l)}\frac{1}{\tau_\sigma^{-\alpha}+\left(\gamma_j^{(l)}\right)^\alpha}e^{((n-1)\Delta t-b)\gamma_j^{(l)}}y[b,a,\gamma_j^{(l)}]((n-1)\Delta t)\right)\right) \end{aligned} \tag{2.38}$$

which yields the following fast convolution algorithm:

- Set

$$\mathbf{U}^0=0,\ \dot{\mathbf{U}}^0=0,\ \ddot{\mathbf{U}}^0=0$$

- **for** the n-th time step, **do**

  1. When $n \le 2$, solve for $\mathbf{U}^n$ with the Newmark scheme

$$\left(\frac{1}{\Delta t^2\theta_2}\mathbf{M}+\frac{\tau_\varepsilon^\alpha}{\tau_\varepsilon^\alpha}\mathbf{K}\right)\mathbf{U}^n=\mathbf{F}^n+\frac{\mathbf{M}}{\Delta t^2\theta_2}\left(\mathbf{U}^{n-1}+\Delta t\dot{\mathbf{U}}^{n-1}+\frac{\Delta t^2}{2}(1-2\theta_2)\ddot{\mathbf{U}}^{n-1}\right)$$
$$+\frac{1}{\tau_\sigma^\alpha}\left(\left(\frac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha-1\right)\mathbf{K}\left(H_1\mathbf{U}^{n-1}+H_2\dot{\mathbf{U}}^{n-1}+H_3\ddot{\mathbf{U}}^{n-1}\right.$$
$$\left.+\sum_l\left(\sum_{j=-L}^{L}\omega_j^{(l)}\frac{1}{\tau_\sigma^{-\alpha}+\left(\gamma_j^{(l)}\right)^\alpha}e^{((n-1)\Delta t-b)\gamma_j^{(l)}}y[b,a,\gamma_j^{(l)}]((n-1)\Delta t)\right)\right);$$

(2.39)

else, solve with the three-step BDF scheme

$$\left(A_1A_2\mathbf{M}+\frac{\tau_\varepsilon^\alpha}{\tau_\varepsilon^\alpha}\mathbf{K}\right)\mathbf{U}^n=\mathbf{F}^n-(A_2(B_1\mathbf{U}^{n-1}+C_1\mathbf{U}^{n-2}+D_1\mathbf{U}^{n-3})+(B_2\dot{\mathbf{U}}^{n-1}+C_2\dot{\mathbf{U}}^{n-2}+D_2\dot{\mathbf{U}}^{n-3}))\mathbf{M}$$
$$+\frac{1}{\tau_\sigma^\alpha}\left(\left(\frac{\tau_\varepsilon}{\tau_\sigma}\right)^\alpha-1\right)\mathbf{K}\left(H_1\mathbf{U}^{n-1}+H_2\dot{\mathbf{U}}^{n-1}+H_3\ddot{\mathbf{U}}^{n-1}\right.$$
$$\left.+\sum_l\left(\sum_{j=-L}^{L}\omega_j^{(l)}\frac{1}{\tau_\sigma^{-\alpha}+\left(\gamma_j^{(l)}\right)^\alpha}e^{((n-1)\Delta t-b)\gamma_j^{(l)}}y[b,a,\gamma_j^{(l)}]((n-1)\Delta t)\right)\right).$$

(2.40)

2. Approximate $\dot{\mathbf{U}}^n$ and $\ddot{\mathbf{U}}^n$ following the Newmark scheme (2.10) or the three-step BDF scheme (2.11).

3. Update the inner integral approximations y^n following (2.37).

- Go to time step n +1.

The estimated integrals in (2.34) and (2.35) introduce an $O(\Delta t^4)$ error each, which accumulatively give an $O(\Delta t^3)$ error from the fast convolution method. Therefore, while using the fast convolution method, the fractional derivative estimation would only bring an error which is smaller than the errors from the Newmark scheme (2.10) and the three-step BDF scheme (2.11). Regarding the efficiency, with $N = T/\Delta t$ numbers of total time steps, this resultant algorithm requires $O(N \log N)$ operations and $O(\log N)$ memory.

2.3. Fluid–structure interaction

In the arterial simulations, we use the *partitioned* fluid–structure interaction approach to study the blood flow and arterial wall physics numerically. We employ the fractional order model for the structure part in order to investigate the performance of the viscoelastic models in simulating the arterial wall motion. At the fluid structure interface, the *Neumann* boundary condition

$$\mathbf{S}\mathbf{n}_s=-[-p\mathbf{I}+\rho_f\varsigma(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)]\mathbf{n}_f,\ \text{on}\sum(t) \quad (2.41)$$

is applied to ensure the continuity of normal stresses, where p, $\mathbf{u}$, ς, ρ_f and $\mathbf{n}_f$ are the fluid pressure, velocity, viscosity, density and the normal vector of fluid subdomain pointing outward on the interface, respectively. The fluid part is assumed to be incompressible and Newtonian, hence, it can be described by the incompressible Navier–Stokes equation in the arbitrary Lagrangian–Eulerian (ALE) framework,

$$\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}-\mathbf{w})\cdot\nabla\mathbf{u}=-\frac{\nabla p}{\rho_f}+\varsigma\nabla^2\mathbf{u},\ \text{in}\,\Omega_f(t), \quad (2.42a)$$

$$\nabla\cdot\mathbf{u}=0,\ \text{in}\,\Omega_f(t), \quad (2.42b)$$

combined with the initial condition

$$\mathbf{u}(\mathbf{x},t=0)=\mathbf{u}_0(\mathbf{x}),\ \text{in}\,\Omega_f(0), \quad (2.43)$$

and the Dirichlet boundary conditions on the interface, i.e.,

$$\mathbf{u}=\frac{\partial \boldsymbol{\eta}}{\partial t},\ \text{on}\sum(t), \quad (2.44)$$

which enforces the continuity of velocities. Here, **w** stands for the mesh velocity. On the fluid side the parallel Navier–Stokes solver NEKTAR [33] is employed, which is based on unstructured spectral elements (polymorphic shapes) and corresponding Jacobi polynomial expansions as trail and test basis in the mixed Galerkin formulation.

When the mass ratio between the structure and the fluid parts is around one as in our case, the partitioned fluid–structure solver suffers from the so-called added-mass effect [40,41], which causes divergence of the subiterations. To resolve the added-mass effect which appears in the hemodynamic FSI simulations, we adopt the *fictitious mass* method with optimal coefficients suggested by the analysis. In these simulations, subiterations are employed on each time step to guarantee continuity at the interface. For more details of our fluid–structure interaction solver, we refer the interested readers to [35].

3. Numerical results

3.1. First convergence test

In this section, we investigate the time accuracy of the Grüwald–Letnikov formula as well as the fast convolution method. We first consider the deformation of a cubic solid occupying the domain $0 \le X \le 1$, $0 \le Y \le 1$ and $0 \le Z \le 1$ discretized by eight hexahedral elements as illustrated in Fig. 2. For the test example, we use two fabricated problems with parameters as listed in Table 1, then compare the simulation results against analytical fabricated solutions. For the material properties, we use Young's modulus E and the Poisson ratio ν, which are related to the material constants λ and μ in (2.2) by

$$\mu=\frac{E}{2(1+\nu)}, \quad \lambda=\frac{\nu E}{(1+\nu)(1-2\nu)}. \quad (3.1)$$

In cases FS1 and FS2, the face $X = 0$ is clamped, and a traction force field $\mathbf{T} = (T_1, T_2, T_3)$ is applied on the rest of the faces, which is given by

$$T_1=n_x(\lambda+2\mu)t^{3+\alpha}, \quad T_2=n_y\lambda t^{3+\alpha}, \quad T_3=n_z\lambda t^{3+\alpha}, \quad (3.2)$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the outward-pointing unit vector normal to the surface. A body force field $\mathbf{f} = (f_1, f_2, f_3)$ is applied on the subject with components:

$$f_1=x((3+\alpha)(2+\alpha)t^{1+\alpha}+\tau_\sigma^\alpha\Gamma(4+\alpha)t), \quad f_2=f_3=0. \quad (3.3)$$

When $t = 0$, the displacement and velocity fields vanish. This problem has the following analytical solution for the displacement of the object:

$$\eta_1=x\left(t^{3+\alpha}+\frac{\tau_\sigma^\alpha\Gamma(4+\alpha)}{6}t^3\right), \eta_2=\eta_3=0. \quad (3.4)$$

To study the convergence behavior at time $T = 0.4$, we vary the time step size Δt from 0.00015625 to 0.02, as in the left plot of Fig. 3. To compare the computational errors with respect to the exact solution, for Δt we calculate the errors in the relative energy norm. Here the relative energy norm is defined as

$$\|\boldsymbol{\eta}\|_{RE}=\frac{\|\boldsymbol{\eta}_a-\boldsymbol{\eta}_c\|_E}{\|\boldsymbol{\eta}_a\|_E}, \quad (3.5)$$

where

$$\|\mathbf{a}\|_E:=\mu\int_\Omega\varepsilon(\mathbf{a}):\varepsilon(\mathbf{a})d\Omega+\lambda\int_\Omega\operatorname{div}(\mathbf{a})\operatorname{div}(\mathbf{a})d\Omega,$$

$\boldsymbol{\eta}_a$ is the analytic solution for displacement, and $\boldsymbol{\eta}_c$ is the result from computation. Here all the simulations are performed with polynomial order 4 elements. In the left plot of Fig. 3 we plot these errors in logarithmic scale as a function of time step size, which is also in logarithmic scale. Red solid lines show the results from the Grüwald–Letnikov formula, while the results plotted in blue dashed lines are obtained with the fast convolution method, with (θ_1, θ_2) = (5/3, 5/3), which corresponds to a second-order three-step BDF scheme. Different symbols show the results from different cases. One could observe that for both methods, the numerical errors decrease almost along a straight line as the time step size decreases, indicating a polynomial order convergence rate in time. To be specific, a second-order convergence rate in time is achieved for the proposed fast convolution method, while the Grüwald–Letnikov formula is of first-order convergence.

In the right plot of Fig. 3, we compare the efficiency of the Grüwald–Letnikov formula with that of the fast convolution method for time step numbers $N = T/\Delta t$ from 1280 to 40960, by showing the CPU time required for evaluating the memory effects at step N. To be specific, for the Grüwald–Letnikov formula, we show the CPU time in calculating the **HIS** load in (2.25), while for the fast convolution method we calculate the CPU time for approximating the convolution integral $\int_0^t I(t-r)\mathbf{U}(r)dr$. The results are plotted in linear scale as a function of the total number of times, which is in logarithmic scale. The red solid line shows the results from the Grüwald–Letnikov formula while the blue dashed line gives the results from the fast convolution method. We observe that the CPU time required by the fast convolution method is increasing linearly with the increase of time steps, which indicates an $O(\log(N))$ computational cost for calculating the memory effect at each time step. On the other hand, the CPU time for the Grüwald–Letnikov formula is of $O(N)$ and therefore it grows much faster as shown by the red line. Both accuracy and efficiency for the two methods are therefore validated: The Grüwald–Letnikov formula is of first order convergence in time and requires $O(N)$ operations in calculating the memory effect at each time step. The fast convolution method converges with a second-order accuracy in time, and it requires only $O(\log(N))$ operations for evaluating the memory effect at each time step.

To further investigate the high accuracy of the fast convolution method, we now employ a *third-order* BDF scheme in the case “FS2”. Here we vary the time step size Δt from 0.0125 to 0.2, and integrate until time $T = 2.0$ for each Δt. The errors in logarithmic scale are plotted in Fig. 4 as a function of time step size, which is also in logarithmic scale. One can see that the fast convolution method with third-order BDF scheme achieves a third-order convergence in time, which verifies our error analysis in Section 2.2.2.

3.2. Second convergence test

We now consider a simple but realistic example for a two-dimensional structure which has been employed in [42], to show that the model accurately captures the mechanical response and to investigate the accuracy of the fast convolution method when $\tau_\varepsilon \neq 0$. The geometry and mesh are illustrated in Fig. 5: we consider an object with length 1.5 m in the x and y direction, and with a thickness 0.3 m in the z direction. The face $x = 0$ is clamped, and on $x = 1.5$ a traction force field is applied following the negative y direction:

$$T_1 = T_3 = 0,\ T_2 = -1. \quad (3.6)$$

Here we employ a 3D code to obtain a 2D solution and hence we impose on $z = 0$ and $z = -0.3$ periodic boundary conditions, i.e., $\boldsymbol{\eta}|_{z=0} = \boldsymbol{\eta}|_{z=-0.3}$. We assume no volume load **f** for this object, and the model parameters are taken as

$$E = 2\,\mathrm{Mpa}, \quad \nu = 0.3, \quad \rho_s = 7000\,\mathrm{kg/m}^3.$$

To compare the viscoelastic behavior of the material, three sets of fractional parameters are considered:

$$\begin{aligned}&(1)\ \alpha = 0.33, \quad \tau_\varepsilon = 2.0, \quad \tau_\sigma = 0.25;\\&(2)\ \alpha = 0.5. \quad \tau_\varepsilon = 1.0, \quad \tau_\sigma = 0.25;\\&(3)\ \alpha = 1.0, \quad \tau_\varepsilon = 2.0, \quad \tau_\sigma = 0.25.\end{aligned}$$

All tests are performed with the fast convolution method using polynomial order 4 elements, and this structure problem is simulated from $t = 0$ to $t = 20$. In the right plot of Fig. 6, we show the vertical displacement for the case with the first set of fractional parameter at time 20. The computed vertical displacements at the point $(x, y, z) = (1.5, 1.5, 0.0)$ for different fractional parameters are shown in the left plot of Fig. 6, from which one could notice that smaller fractional order α generally yields less damping, which is consistent with the claims in [42].

To study the convergence behavior in time, in Fig. 7 we present the errors for displacement at the point $(x, y, z) = (1.5, 1.5, 0.0)$ when taking the first and the second sets of fractional parameters, as a function of the time step size Δt. Due to the lack of an explicit solution, the numerical solutions are compared with the results obtained from polynomial order 6 elements and finer time step size $\Delta t = 0.0001$. Both the errors and the time step sizes are in the logarithmic scale. We see that the results from the fast convolution method with second-order BDF scheme are approximately on a straight line with the time step size decreasing. To be specific, a second-order convergence rate in time is obtained when $\alpha = 0.33$, and when $\alpha = 0.5$ we can see that the method converges slightly faster than second order in time.

3.3. Fluid–structure interaction: blood flow in a compliant artery

We now consider a straight tube with pulsatile inflow as an idealization of arterial flow-structure interactions, and investigate the performance of fractional order wall models with different parameterizations. To further assess the need for 3D simulations, the simulation results are compared with those obtained from 1D models in [1]. The fluid subdomain is taken as a cylinder with radius 1 mm and length 4 mm, and the structure subdomain is the thin-layered arterial wall with a thickness 0.105 mm, which covers the fluid subdomain uniformly. The computational mesh for both subdomains is illustrated in Fig. 8. In all tests we take the fluid density as $\rho_f = 1060$ kg/m^3, and the structure density $\rho_s = 1.2\rho_f$. On the fluid solver, a Womersley velocity profile with 8 Fourier coefficients is imposed at the inlet boundary, while at the outlet the pressure $\bar{p}$ is flow-dependent and is determined from the flow rate $Q(t)$, given the resistant R, and the capacitance C as [43]

$$\overline{p}(t)+RC\frac{d\overline{p}(t)}{dt}=Q(t)R. \quad (3.7)$$

Here we take $R = 2.8 \times 10^3$ dyn · s/cm^5 and $C = 3.4 \times 10^{-7}$ cm^5/dyn for all tests in this section, and the arterial outlet is assumed to be undeformed when the outlet pressure $\bar{p}(t)$ reaches its lowest point at $t = 0.0151$ s. At the inlet we clamp the arterial wall, while a no-traction load is applied at the external surface as well as at the outlet. All the other physical and numerical parameters are summarized in Table 2, and all results shown here are generated using spectral elements of polynomial order 3.

3.3.1. Comparison between different models—We have considered a purely elastic model for the arterial wall, and in Fig. 9 we show the distribution of structural displacements at four time instants: $t = 2T$, $t=2\frac{T}{3}$, $t=2\frac{4T}{9}$ and $t=2\frac{5T}{9}$, where T is a cardiac cycle. Similarly as in [1], four viscoelastic models are studied for the arterial wall. In Table 3 we summarize the viscoelastic parameters employed in these models, and we compare the results against simulation results from the pure elastic model. In Fig. 10 we present the displacement distributions from the pure elastic wall model at the systolic peak $t=2\frac{T}{3}$, and compare with the results from the integer-order/fractional-order viscoelastic wall models. We observe that at the systolic peak, the radial deformations in all models reach the maximum. On the other hand, in the viscoelastic models the viscous dissipation decreases the deformation to different extent: the FOV-SLS2 case gives results which are the closest to that from the pure elastic model, and in cases SLS2 and FOV-SLS1 the new models greatly decrease the predicted deformation. To see the temporal changes of arterial wall deformation more clearly, we present in Fig. 11 the time traces for displacement at a sample point (1, 0, 2), with the pressure-radial displacement hysteresis loop shown in Fig. 12. Note that in the left plot of Fig. 11, the black solid lines mark four time instants in the cardiac cycle, which correspond to the time instants for the displacement magnitudes shown in Fig. 9. We see that although the elastic response dominates in all models, the varying viscoelastic parameters result a significant change in displacement magnitudes. Among all models, the SLS2 and FOV-SLS1 models produce relatively stiffer responses, and therefore predict much smaller arterial wall deformations. Moreover, it is shown in Fig. 12 that the viscoelastic dissipation can be clearly observed from SLS1 model, but not that much from the SLS2 model. This finding highlights that the integer order SLS model is very sensitive to the relaxation parameters. On the other hand, although the fractional order model appears to be overall less sensitive to the parameters, an increased sensitivity is observed when we switch from 1D to 3D simulations. Comparing the two fractional order model cases FOV-SLS1 and FOV-SLS2, the maximum radial displacement varies by 50%, which is opposite to the observations in [1], where these two models predicted very close results. This discrepancy motivates the investigation presented in the next section that aims to identify whether the increased sensitivity is due to the relaxation parameters τ_ε and τ_σ, or the fractional order α.

3.3.2. Sensitivity to viscoelastic coefficients—To further evaluate the sensitivity of the 3D viscoelastic arterial wall models, we study the behavior of fractional order arterial wall models by varying the viscoelastic parameters α, τ_ε and τ_σ. We first vary the fractional order α between $0 \le \alpha \le 1$ while keeping the same relaxation times of SLS1 model: τ_ε = 0.050 s, τ_ε = 0.025 s. The displacement time traces at the middle of the tube are compared in Fig. 13, for α = 0.0, 0.2, 0.4 and 1.0. Note that for α = 0 the pure elastic model is recovered, while α = 1.0 yields the solution of the integer-order SLS1 model. Although the displacement time trace results in Fig. 13 do not vary much, from the pressure-radial displacement hysteresis loop in Fig. 14 we can see that when α increases the viscous dissipation becomes more important. This is consistent with the discussions in [1], except that comparing with the 1D cases, varying the fractional order α in the 3D cases changes more in both the displacement traces and the hysteresis loops. Next, now we investigate how sensitive the solution is to the choice of relaxation parameters τ_ε and τ_σ. In Figs. 15–16, we present results for fixed α = 0.2 with various relaxation time parameters, and the solutions for α = 1.0 are displayed in Figs. 17–18. When α = 1.0, a more noticeable viscous response can be observed, and the arterial wall deformation is more sensitive to the relaxation spectrum used. This means that when the fractional order α is relatively small, the effect of the fractional order SLS model (2.1) has low sensitivity on the relaxation parameters. Therefore, it overcomes the uncertainty for using the integer order SLS model in simulating arterial networks, where α is typically less than 0.3, since we can only hope to get accurate estimates of τ_ε and τ_σ at limited anatomic locations. In fact, we have also numerically verified that as the fractional order α approaches 1, the simulation results from the fractional-order model converges to the results from the integer order SLS model, and therefore increasingly more sensitive to the parameters (also shown in [1]).

3.3.3. Comparison between 1D and 3D simulations—The above findings reveal differences between 1D and 3D simulations, i.e., the 3D solutions seem more sensitive to the fractional order α. Now we compare the 1D and 3D computed solutions in terms of radial displacement and the pressure waveform at the middle point of the artery (with the coordinate (1, 0, 2)). In Figs. 19–20 we present the results from the pure elastic model and one of the fractional-order viscoelastic models. Generally, the 1D models provide good prediction on the trend of radial displacement/pressure time traces for this simple tube-shape geometry. However, when accurate predictions are needed, the 1D simulations tend to underestimate the arterial wall deformations by as large as 20%. Therefore, to get more reliable predictions and to enable the possibility of simulating on complicated geometries, the 3D models are recommended.

3.4. Fluid–structure interaction: patient-specific aneurysm

The authors in [7–10,44] have demonstrated that fractional orders would be able to properly predict the different mechanical behavior of the smooth muscle cell activities. Since the smooth muscle decrease plays a crucial role in the change of cerebral aneurysm walls [45–48], in this section we investigate the fractional order models on a patient-specific aneurysm geometry [24,43]. This aneurysm is located at the eye level, in the cavernous segment of the right internal carotid artery. We perform our simulations on the meshes as shown in Fig. 21, where the fluid subdomain has finer elements near the vessel wall to resolve the boundary layers, and the solid subdomain is composed of two layers of prismatic elements on the vessel wall. A Womersley velocity profile with 8 harmonics is imposed at the inlet boundary of the fluid solver, and a zero Neumann boundary condition for velocity is employed at the outlet. The outlet pressure $\bar{p}$ is determined by (3.7), with proper resistant and capacitance coefficients. The arterial outlet is assumed to be undeformed when the outlet pressure $\bar{p}(t)$ reaches its lowest point at $t = 0.0571$ s. We start the FSI simulations from the resultant flow field of rigid wall simulations on the fluid subdomain (where we set $t = 0$ as the starting time of FSI). The structure is considered undeformed at $t = 0$. To integrate in time, we employed the second-order splitting scheme for the fluid solver. On the structure side, the fast convolution method introduced in 2.2.2 is used, with fictitious mass method [24,43] applied to stabilize the FSI partitioned procedure. All the simulations are done with the physical and numerical parameters summarized in Table 4.

3.4.1. Comparison between different models—To compare the computed evolution of the aneurysm deformation from different models, we first display the displacement time traces of three points on the interface. The locations of these points are shown in Fig. 22. The computed inlet pressure and the time traces for displacements at these three points are shown in Fig. 23, for the pure elastic model, the FOV-SLS1 and the FOV-SLS2 models. Similar to what we have observed in section 3.3, the FOV-SLS1 model predicts a much smaller deformation compared with that from the other two models, which results in a larger blood pressure drop in each cardiac cycle. A minor phase shift could be observed while comparing the location of the maximum displacement for each model: while the maximum displacements were achieved after the systolic peak in all cases, the case using the FOV-SLS1 model reaches its maximum deformation firstly, then the case using the elastic arterial wall model, and finally the case with the FOV-SLS2 model. To investigate the displacement distribution on the whole aneurysm, we take four time instants in a cardiac cycle: $\frac{T}{7}, \frac{2T}{7}, \frac{3T}{7}, \frac{5T}{7}$ (as marked by black solid lines in the inlet pressure plot of Fig. 23), and display their corresponding displacement amplitudes from the pure elastic, FOV-SLS1 and FOV-SLS2 arterial wall models in Figs. 24, 25 and 26, respectively. It can be observed that although the displacement magnitudes vary for different arterial wall models, the displacement magnitude distribution pattern does not change much.

4. Summary

In this paper, we have developed an accurate and efficient numerical method for the 3D fractional-order standard linear solid (SLS) model (2.1) by combining a fast convolution method with the spectral element discretization based on a general Jacobi polynomial basis that can be used to generate 3D polymorphic high-order elements. To sustain the high-order accuracy of the backward differentiation formula (BDF), we designed a high-order fast convolution method and verified its third-order convergence rate in time using the third-order three-step BDF method on problems with fabricated analytical solutions. The $N\log(N)$ operation count was also verified.

In numerical tests, we investigated the ability of fractional-order model to capture the viscoelastic behavior of arterial walls under fluid–structure interactions. On an idealized artery geometry, we compared several models with the same elasticity parameters and different viscoelastic parameters. This comparison study indicates that for the integer-order SLS models, the viscous behavior strongly depends on the relaxation parameters τ_ϵ and τ_σ, while the fractional order models are less sensitive. Especially when the fractional order $\alpha \to 0$, the viscous response is less pronounced, and the fractional model is tuned by a single parameter, the fractional order α. This finding is overall consistent with what has been observed for 1D models [1], although model parametrization appears to be relatively more sensitive in the 3D setting. As an example of *long-term* large-scale simulations, we have also investigated the performance of this fractional aneurysm model on a complex patient-specific geometry.

In ongoing work, we are now investigating the proper fractional parameters specifically for the aneurysm sac using patient-specific data. Since the fractional order is related with the loss of smooth muscle cells, this aneurysm-specific model as well as the numerical tools developed here may help to understand the structural integrity of the aneurysm wall, and therefore contribute in studying the subclasses of unruptured aneurysm walls [49,50].

Acknowledgments

We would like to acknowledge support by the Lehigh FRG faculty research grant, the Simons collaboration grant, and the computational resources of CCV at Brown University and the DOE INCITE program. We also acknowledge support from the NSF grant 1620434, the NIH grant 1U01HL116323-01 and the ARO/MURI grant W911NF-15-1-0562.

References

1. Perdikaris P, Karniadakis GE. Fractional-order viscoelasticity in one-dimensional blood flow models. Ann Biomed Eng. 2014; 42(5):1012–1023. [PubMed: 24414838]
2. López-Fernández M, Lubich C, Schädle A. Adaptive, fast, and oblivious convolution in evolution equations with memory. SIAM J Sci Comput. 2008; 30(2):1015–1037.
3. Fung, Y. Biomechanics: Mechanical Properties of Living Tissues. Springer-Verlag; 1993.
4. Alastruey J, Khir AW, Matthys KS, Segers P, Sherwin SJ, Verdonck PR, Parker KH, Peiró J. Pulse wave propagation in a model human arterial network: assessment of 1-d visco-elastic simulations against in vitro measurements. J Biomech. 2011; 44(12):2250–2258. [PubMed: 21724188]
5. Reymond P, Bohraus Y, Perren F, Lazeyras F, Stergiopulos N. Validation of a patient-specific one-dimensional model of the systemic arterial tree. Am J Physiol Heart Circ Physiol. 2011; 301(3):H1173–H1182. [PubMed: 21622820]
6. Eringen, AC. Mechanics of Continua. Robert E. Krieger Publishing Co; Huntington, NY: 1980. p. 606
7. Doehring TC, Freed AD, Carew EO, Vesely I. Fractional order viscoelasticity of the aortic valve cusp: an alternative to quasilinear viscoelasticity. ASME J Biomech Eng. 2005; 127(4):700–708.
8. Craiem DO, Armentano RL. A fractional derivative model to describe arterial viscoelasticity. Biorheology. 2007; 44(4):251–263. [PubMed: 18094449]
9. Craiem DO, Rojo FJ, Atienza JM, Guinea GV, Armentano RL. Fractional calculus applied to model arterial viscoelasticity. Lat Am Appl Res. 2008; 38(2):141–145.
10. Craiem DO, Rojo FJ, Atienza JM, Armentano RL, Guinea GV. Fractional-order viscoelasticity applied to describe uniaxial stress relaxation of human arteries. Phys Med Biol. 2008; 53(17): 4543. [PubMed: 18677037]
11. Mainardi, F. Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models. World Scientific; 2010.
12. Podlubny, I. Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications. Vol. 198. Academic Press; 1998. Fractional Differential Equations: An Introduction to Fractional Derivatives.
13. Magin, RL. Fractional Calculus in Bioengineering. Begell House Publishers Inc; Redding, CT: 2006.

14. Steinman DA, Hoi Y, et al. Variability of CFD solutions for pressure and flow in a giant aneurysm: the SBC2012 CFD challenge. J Biomech Eng. 2013; 135(2):542–554. (authors are ordered alphabetically by first author within the group).
15. Torii R, Oshima M, Kobayashi T, Takagi K, Tezduyar TE. Computer modeling of cardiovascular fluid–structure interactions with the deforming-spatial-domain/stabilized space–time formulation. Comput Methods Appl Mech Eng. 2006; 195(13–16):1885–1895.
16. Tezduyar TE, Takizawa K, Brummer T, Chen PR. Space–time fluid–structure interaction modeling of patient-specific cerebral aneurysms. Int J Numer Methods Biomed Eng. 2011; 27(11):1665–1710.
17. Torii R, Oshima M, Kobayashi T, Takagi K, Tezduyar TE. Influencing factors in image-based fluid–structure interaction computation of cerebral aneurysms. Int J Numer Methods Fluids. 2011; 65(1–3):324–340.
18. Järvinen E, Råback P, Lyly M, Salenius JP. A method for partitioned fluid–structure interaction computation of flow in arteries. Med Eng Phys. 2008; 30(7):917–923. [PubMed: 18243762]
19. Crosetto P, Reymond P, Deparis S, Kontaxakis D, Stergiopulos N, Quarteroni A. Fluid–structure interaction simulation of aortic blood flow. Comput Fluids. 2011; 43(1):46–57.
20. Crosetto P, Deparis S, Fourestey G, Quarteroni A. Parallel algorithms for fluid–structure interaction problems in hemodynamics. SIAM J Sci Comput. 2011; 33(4):1598–1622.
21. Nobile F, Vergara C. An effective fluid–structure interaction formulation for vascular dynamics by generalized Robin conditions. SIAM J Sci Comput. 2008; 30(2):731–763.
22. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen JG. A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. Comput Mech. 2010; 46(1): 3–16.
23. Hsu MC, Bazilevs Y. Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulation. Finite Elem Anal Des. 2011; 47(6):593–599.
24. Yu Y, Baek H, Karniadakis GE. Generalized fictitious methods for fluidâstructure interactions: analysis and simulations. J Comput Phys. 2013; 245:317–346.
25. Ford NJ, Simpson AC. The numerical solution of fractional differential equations: speed versus accuracy. Numer Algorithms. 2001; 26(4):333–346.
26. Deng W. Short memory principle and a predictor–corrector approach for fractional differential equations. J Comput Appl Math. 2007; 206(1):174–188.
27. Aceto L, Magherini C, Novati P. On the construction and properties of m-step methods for FDEs. SIAM J Sci Comput. 2015; 37(2):A653–A675.
28. Li JR. A fast time stepping method for evaluating fractional integrals. SIAM J Sci Comput. 2010; 31(6):4696–4714.
29. Lubich C, Schädle A. Fast convolution for nonreflecting boundary conditions. SIAM J Sci Comput. 2002; 24(1):161–182.
30. Baffet, DH., Hesthaven, JS. Tech rep. Society for Industrial and Applied Mathematics; 2015. A Laplace transform based kernel reduction scheme for fractional differential equations.
31. Schädle A, López-Fernández M, Lubich C. Fast and oblivious convolution quadrature. SIAM J Sci Comput. 2006; 28(2):421–438.
32. López-Fernández M, Palencia C, Schädle A. A spectral order method for inverting sectorial Laplace transforms. SIAM J Numer Anal. 2006; 44(3):1332–1350.
33. Karniadakis, GE., Sherwin, SJ. Numerical Mathematics and Scientific Computation. 2. Oxford University Press; Oxford: 2005. Spectral/hp Element Methods for Computational Fluid Dynamics.
34. Dong S, Yosibash Z. A parallel spectral element method for dynamic three-dimensional nonlinear elasticity problems. Comput Struct. 2009; 87(1–2):59–72.
35. Yu Y, Baek H, Bittencourt ML, Karniadakis GE. Mixed spectral/hp element formulation for nonlinear elasticity. Comput Methods Appl Mech Eng. 2012; 213–216(0):42–57.
36. Yu Y, Bittencourt ML, Karniadakis GE. A semi-local spectral/hp element solver for linear elasticity problems. Int J Numer Methods Eng. 2014; 100(5):347–373.
37. Dong S. BDF-like methods for nonlinear dynamic analysis. J Comput Phys. 2010; 229(8):3019–3045.

38. Talbot A. The accurate numerical inversion of Laplace transforms. IMA J Appl Math. 1979; 23(1): 97–120.

39. Rizzardi M. A modification of talbot's method for the simultaneous approximation of several values of the inverse Laplace transform. ACM Trans Math Software (TOMS). 1995; 21(4):347–371.

40. Idelsohn SR, Del Pin F, Rossi R, Oñate E. Fluid–structure interaction problems with strong added-mass effect. Int J Numer Methods Eng. 2009; 80(10):1261–1294.

41. Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. Comput Methods Appl Mech Eng. 2005; 194(42–44):4506–4527.

42. Larsson S, Saedpanah F. The continuous Galerkin method for an integro-differential equation modeling dynamic fractional order viscoelasticity. IMA J Numer Anal. 2010; 30(4):964–986.

43. Baek H, Karniadakis GE. A convergence study of a new partitioned fluid–structure interaction algorithm based on fictitious mass and damping. J Comput Phys. 2012; 231(2):629–652.

44. Craiem D, Magin RL. Fractional order models of viscoelasticity as an alternative in the analysis of red blood cell (RBC) membrane mechanics. Phys Biol. 2010; 7(1):013001.

45. Robertson, AM., Hill, MR., Li, D. Modeling of Physiological Flows. Springer; 2012. Structurally motivated damage models for arterial walls. Theory and application; p. 143-185.

46. Bank AJ, Kaiser DR, Rajala S, Cheng A. In vivo human brachial artery elastic mechanics effects of smooth muscle relaxation. Circulation. 1999; 100(1):41–47. [PubMed: 10393679]

47. Toth M, Nadasy GL, Nyary I, Kerényi T, Orosz Ms, Molnarka G, Monos E. Sterically inhomogenous viscoelastic behavior of human saccular cerebral aneurysms. J Vasc Res. 1998; 35(5):345–355. [PubMed: 9789115]

48. Sadasivan C, Fiorella DJ, Woo HH, Lieber BB. Physical factors effecting cerebral aneurysm pathophysiology. Ann Biomed Eng. 2013; 41(7):1347–1365. [PubMed: 23549899]

49. Cebral JR, Duan X, Chung BJ, Putman C, Aziz K, Robertson AM. Wall mechanical properties and hemodynamics of unruptured intracranial aneurysms. Am J Neuroradiol. 2015; 36(9):1695–1703. [PubMed: 26228891]

50. Robertson AM, Duan X, Aziz KM, Hill MR, Watkins SC, Cebral JR. Diversity in the strength and structure of unruptured cerebral aneurysms. Ann Biomed Eng. 2015:1–14. [PubMed: 25527321]

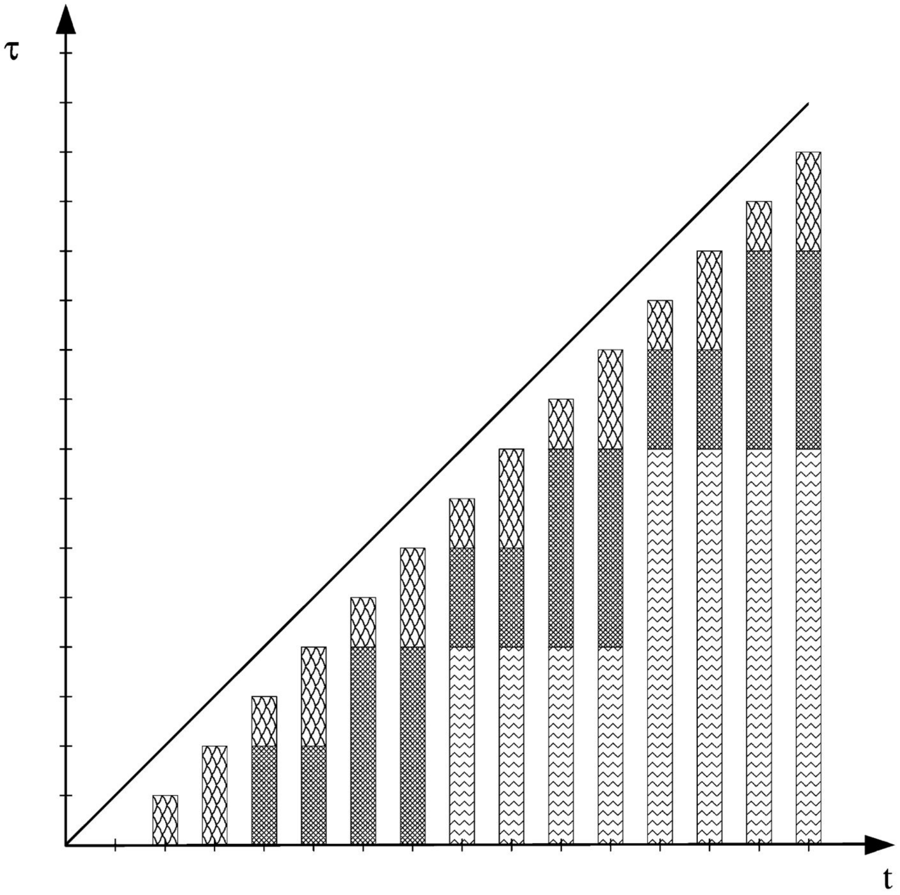

Fig. 1.

Decomposition for integral ranges (*a, b*) in the fast convolution method of [29].

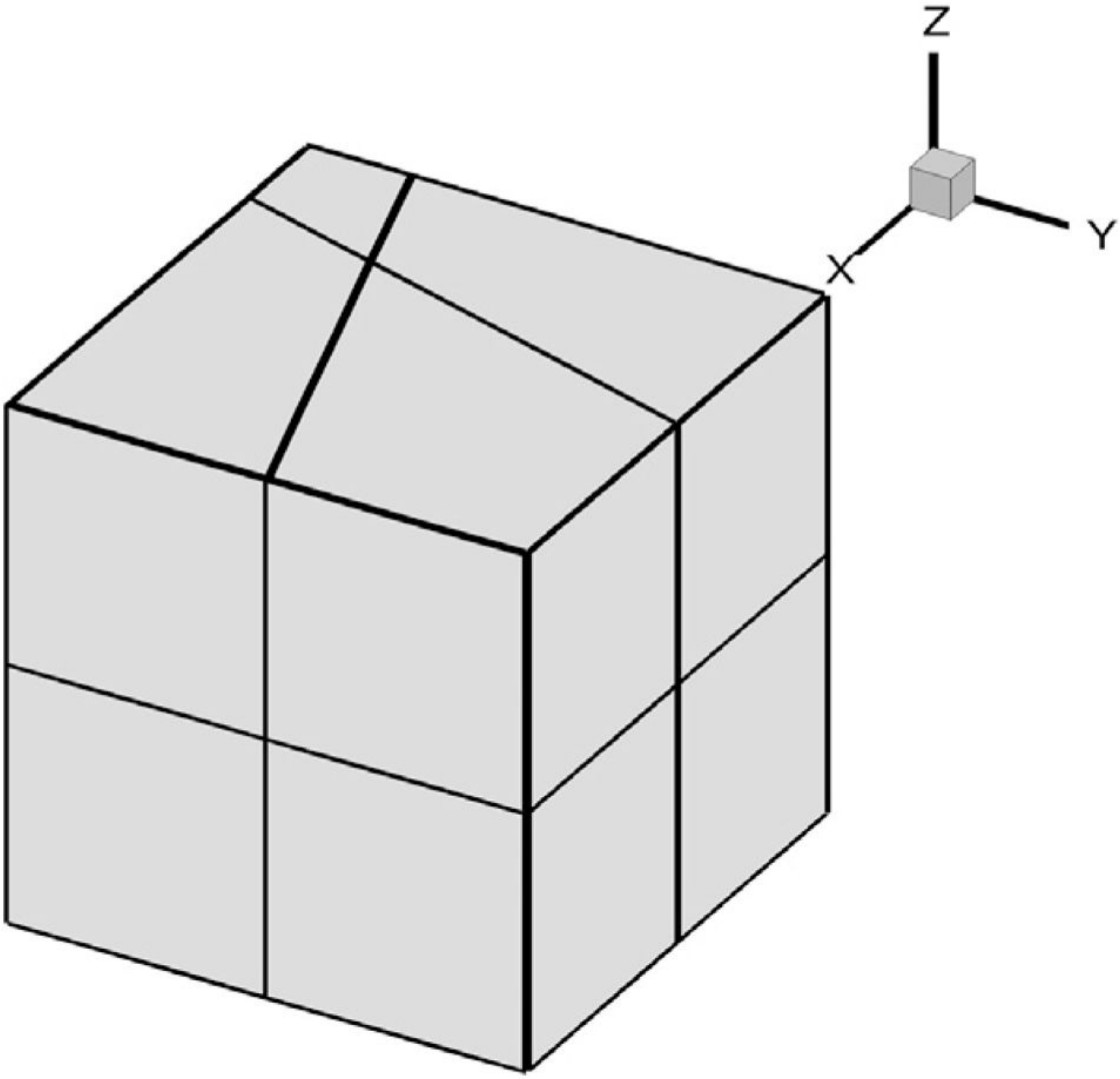

Fig. 2.
Mesh for the first convergence test.

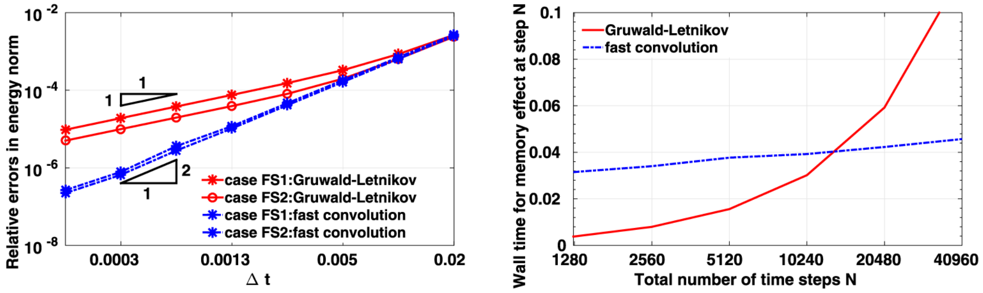

Fig. 3.

Results from the first convergence test. Left: errors as a function of the time step size. Right: CPU time (Sec) in evaluating the memory effect at the last time step. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

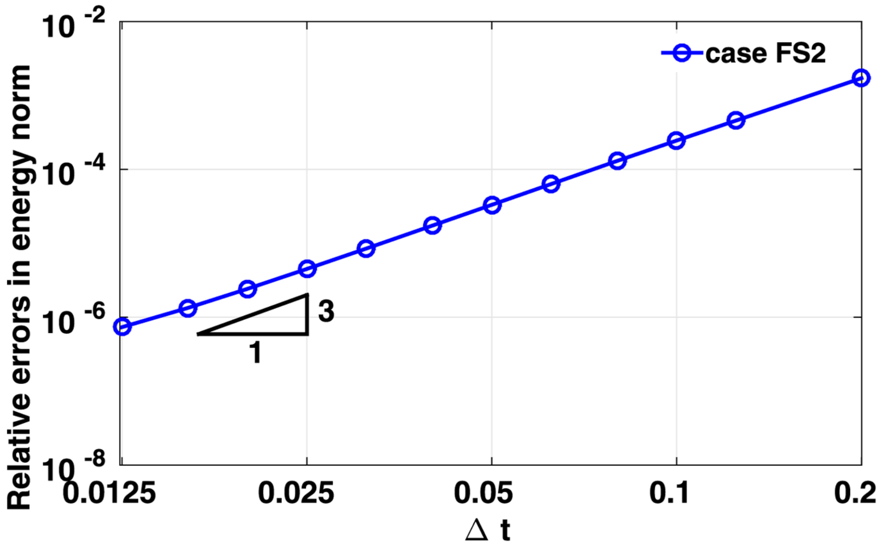

Fig. 4.

Errors as a function of the time step sizes, from the first convergence test while using the third-order three-step BDF scheme.

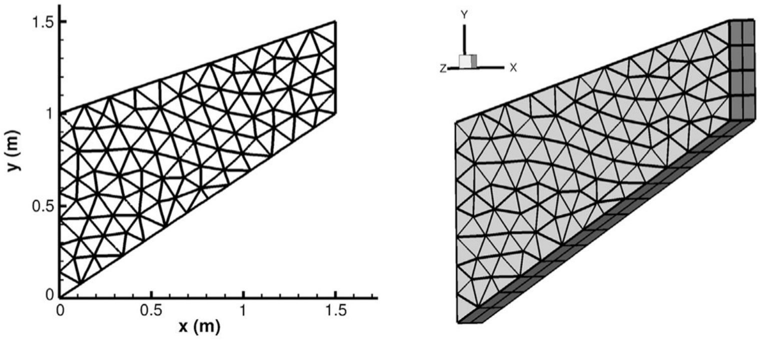

Fig. 5.

Mesh for the second convergence test.

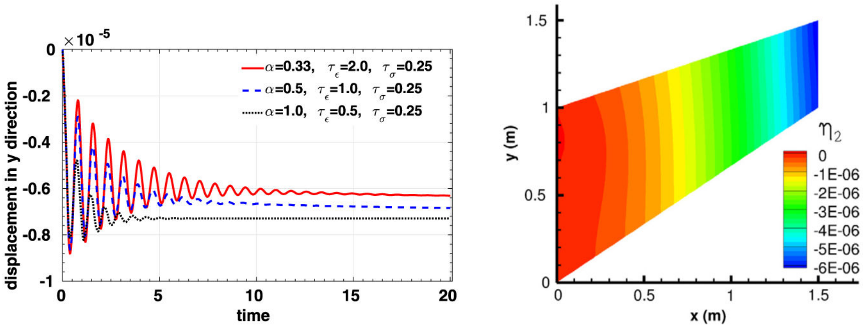

Fig. 6.

Simulation results for the second convergence test. Left: time traces of the displacement at the right end of the object (x =1.5). Right: displacement distribution in y direction when α = 0.33, τ_ϵ = 2.0, τ_σ = 0.25 and time = 20 (for interpretation of the colors in this figure, the reader is referred to the web version of this article).

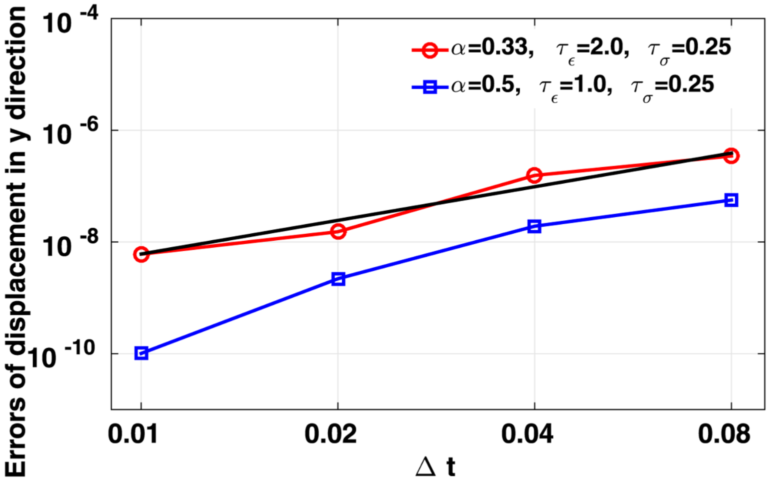

Fig. 7.

Results from the second convergence test: errors of the fast convolution method as a function of the time step size. The reference solution is obtained from a higher resolution simulation.

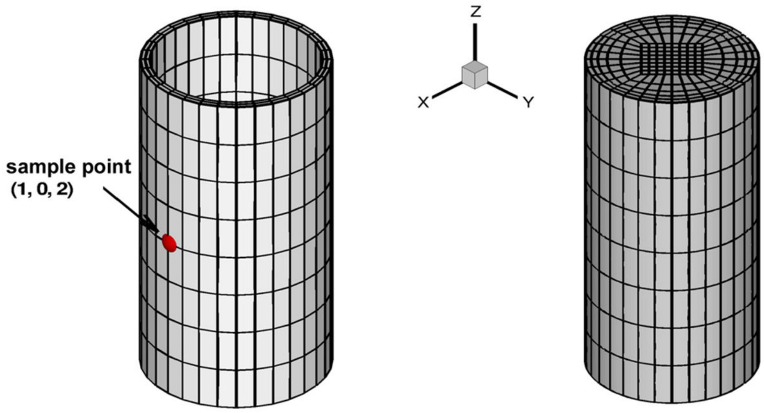

Fig. 8.

Meshes for blood flow in an artery. Left: structure mesh; right: fluid mesh.

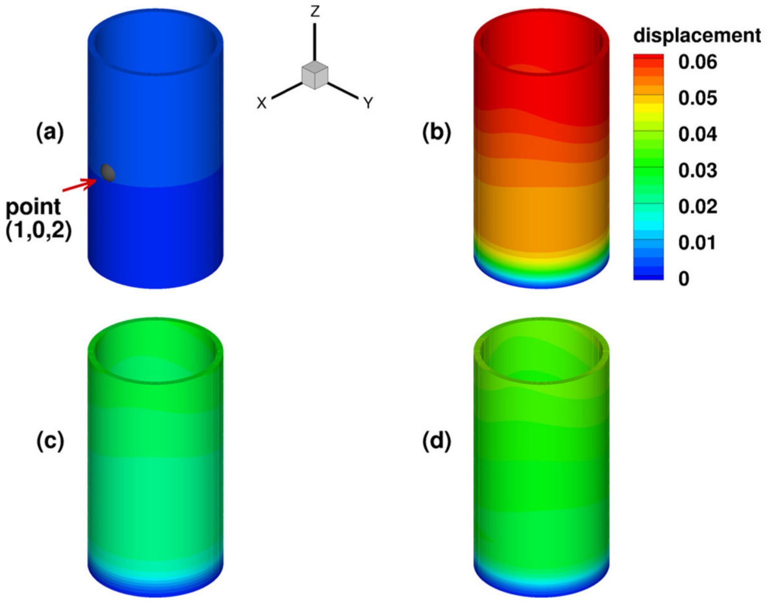

Fig. 9.

Displacements (in millimeters) for the pure elastic model in the idealized artery simulations: at $t = 2T$ (upper left), $t=2\frac{T}{3}$ (upper right), $t=2\frac{4T}{9}$ (lower left), $t=2\frac{5T}{9}$ (lower right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

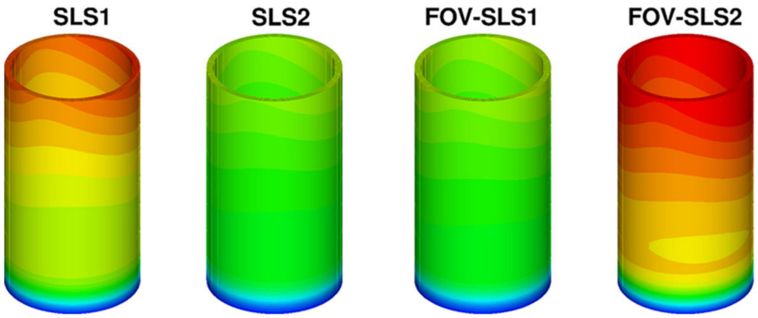

Fig. 10.

Displacements (in millimeters) for the viscoelastic models in the idealized artery simulations at systolic peak $t=2\frac{T}{3}$. From left to right are obtained with: model SLS1; model SLS2; model FOV-SLS1 and model FOV-SLS2. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

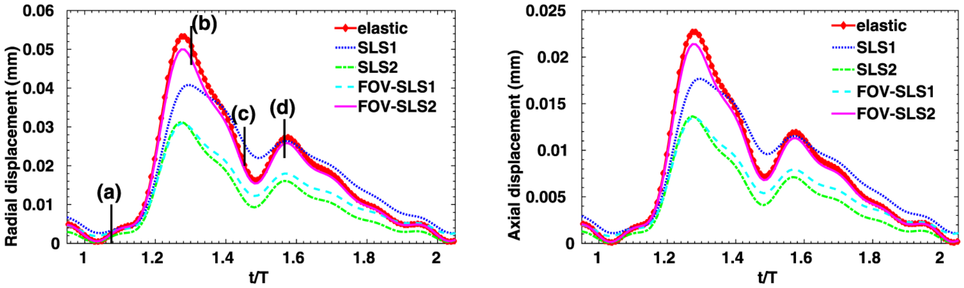

Fig. 11.

Time traces at point (1, 0, 2) in the idealized artery simulations, from varying viscoelastic models. Left: radial displacement; right: axial displacement.

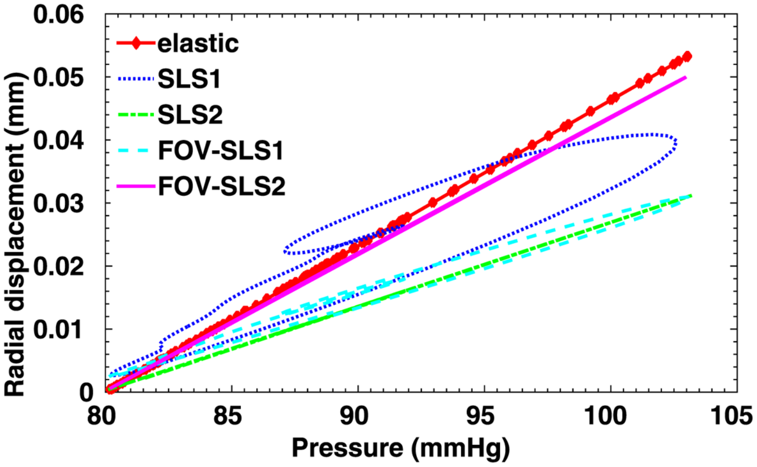

Fig. 12.

Pressure-radial displacement hysteresis loop at point (1, 0, 2) in the idealized artery simulations, from varying viscoelastic models.

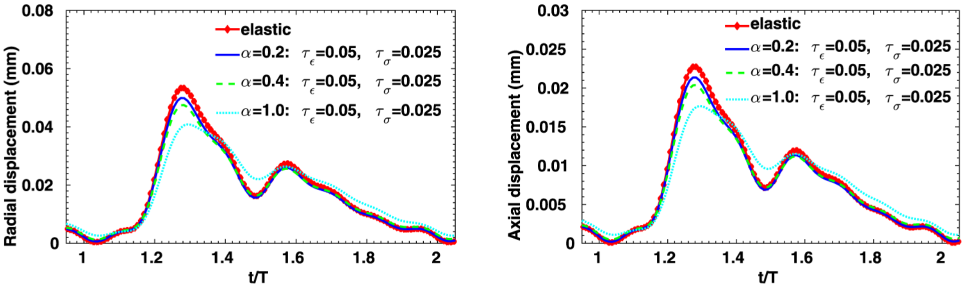

Fig. 13.

Time traces at point (1, 0, 2) in the idealized artery simulations, from fixed $\tau_\varepsilon = 0.05$, $\tau_\sigma = 0.025$ and varying α. Left: radial displacement; right: axial displacement.

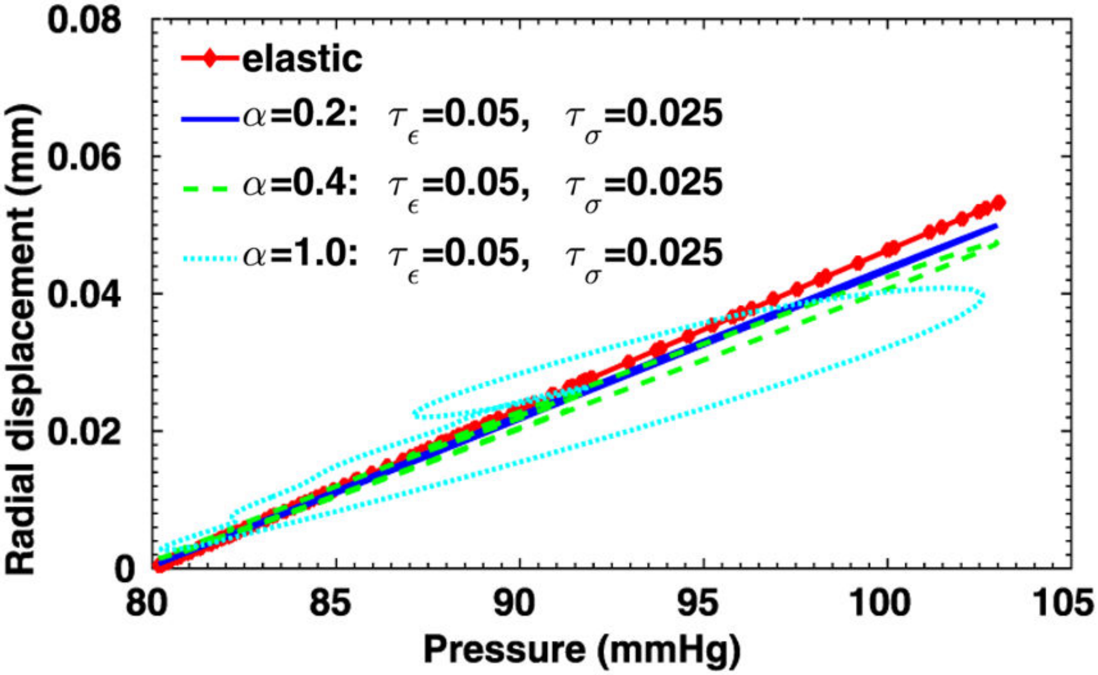

Fig. 14.

Pressure-radial displacement hysteresis loops at point (1, 0, 2) in the idealized artery simulations, from fixed $\tau_\varepsilon = 0.05$, $\tau_\sigma = 0.025$ and varying α.

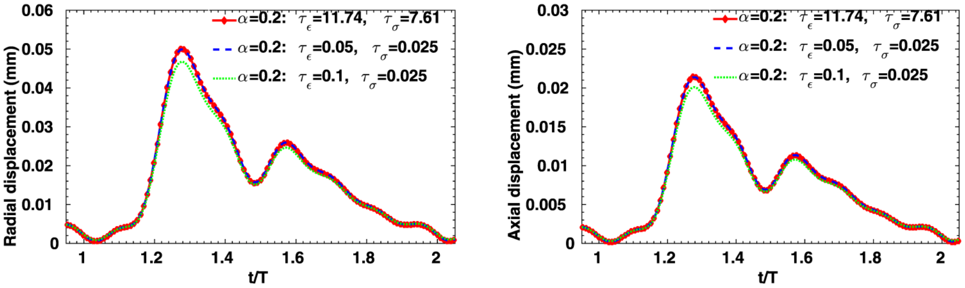

Fig. 15.

Time traces at point (1, 0, 2) in the idealized artery simulations, from fixed $\alpha = 0.2$ and varying relaxation parameters. Left: radial displacement; right: axial displacement.

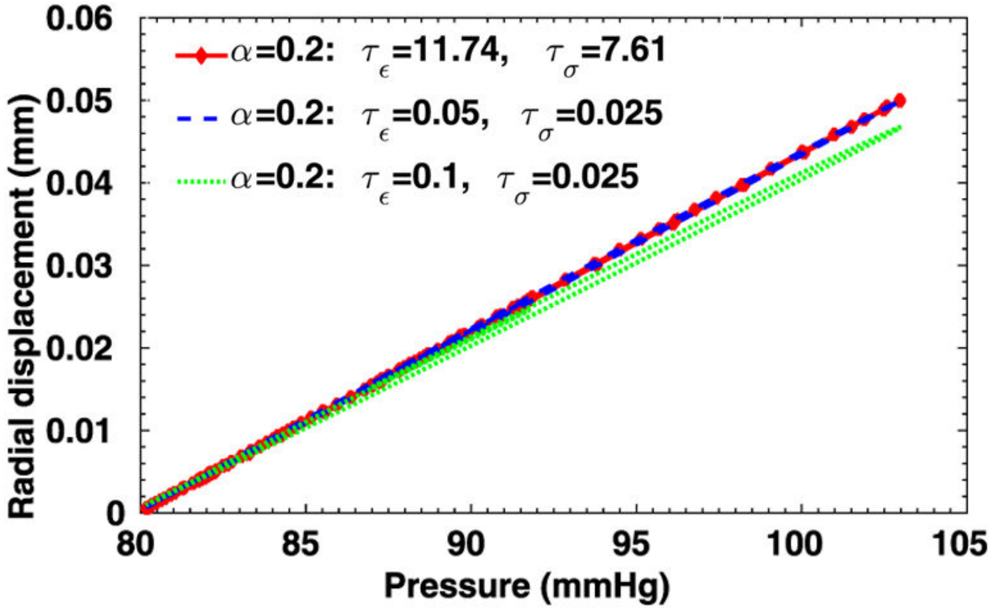

Fig. 16.

Pressure-radial displacement hysteresis loops at point (1, 0, 2) in the idealized artery simulations, from fixed α = 0.2 and varying relaxation parameters.

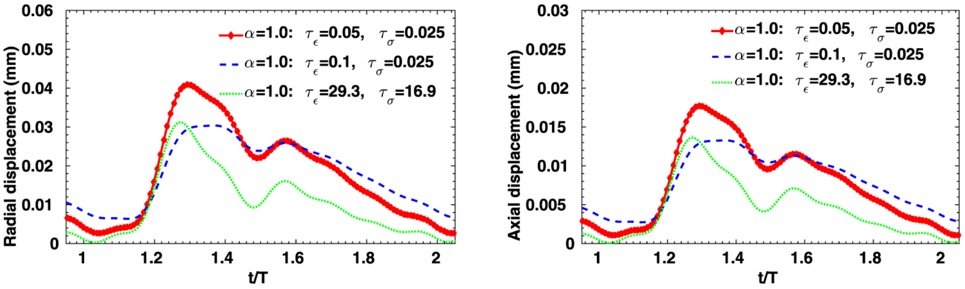

Fig. 17.

Time traces at point (1, 0, 2) in the idealized artery simulations, from fixed α = 1.0 and varying relaxation parameters. Left: radial displacement; right: axial displacement.

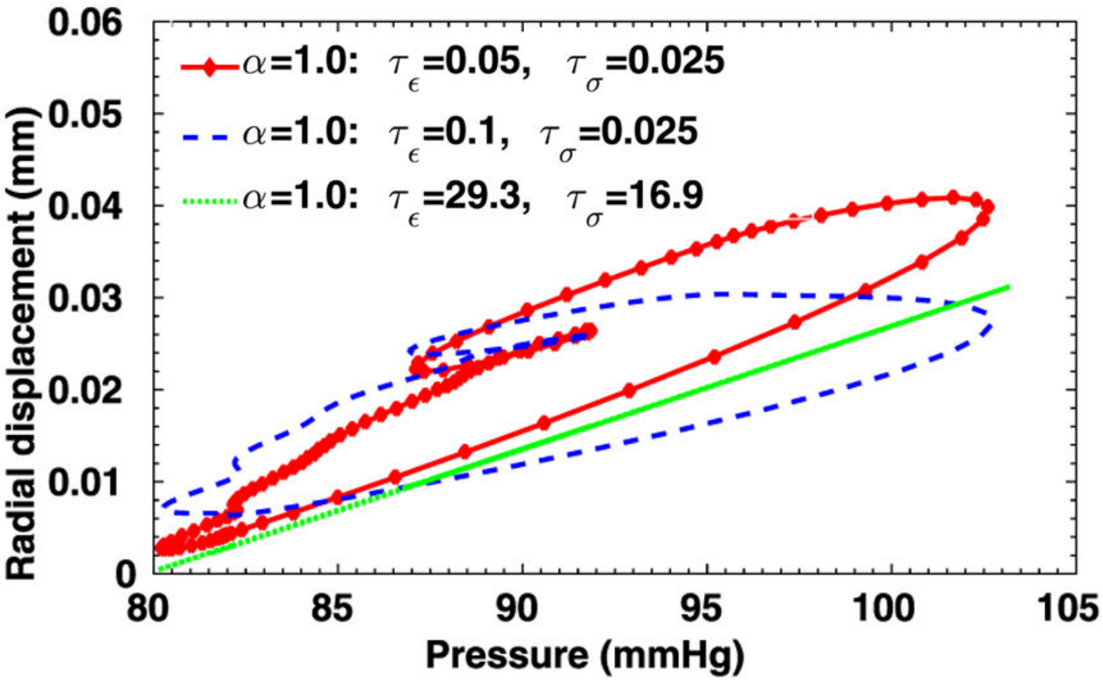

Fig. 18.

Pressure-radial displacement hysteresis loops at point (1, 0, 2) in the idealized artery simulations, from fixed α = 1.0 and varying relaxation parameters.

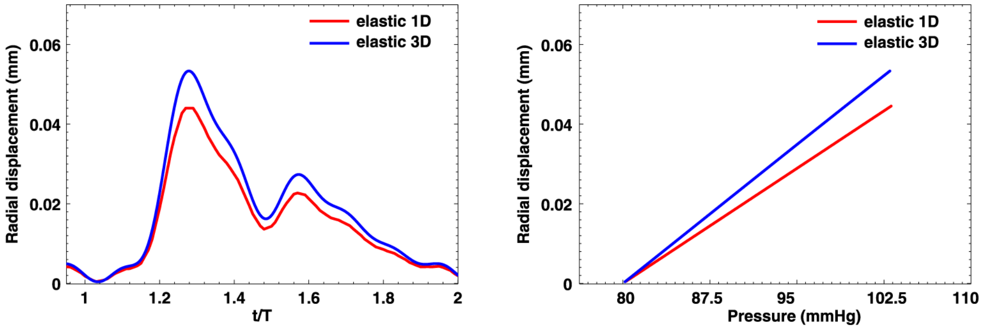

Fig. 19.

1D–3D comparison at point (1, 0, 2) in the idealized artery simulations, from pure elastic model. Left: radial displacement time trace; Right: pressure-radial displacement hysteresis loop. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

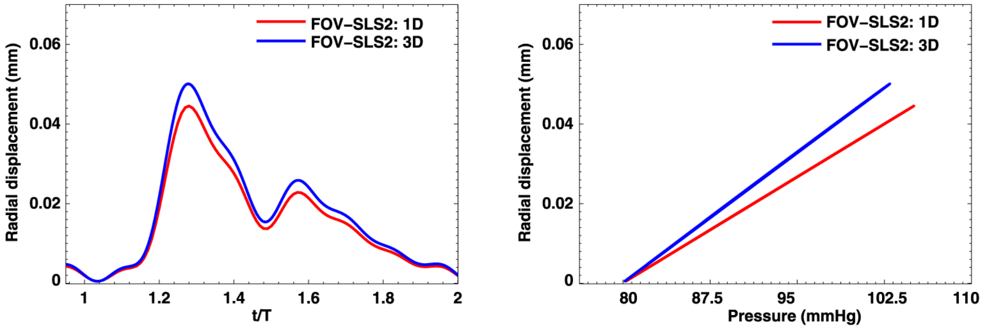

Fig. 20.

1D–3D comparison at point (1, 0, 2) in the idealized artery simulations, from FOV-SLS2 model. Left: radial displacement time trace; Right: pressure-radial displacement hysteresis loop. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

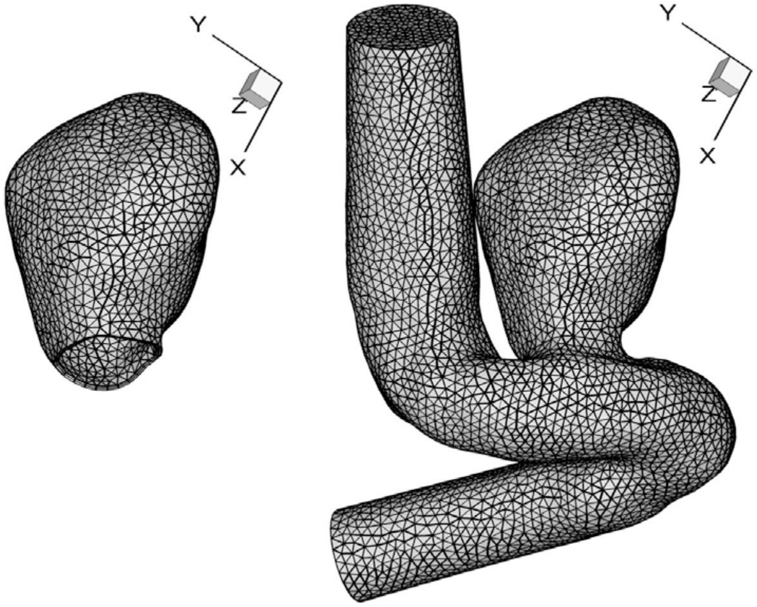

Fig. 21.

Meshes for patient-specific aneurysm model. Left: structure mesh (with 5628 prismatic elements); right: fluid mesh (with 58939 tetrahedral elements).

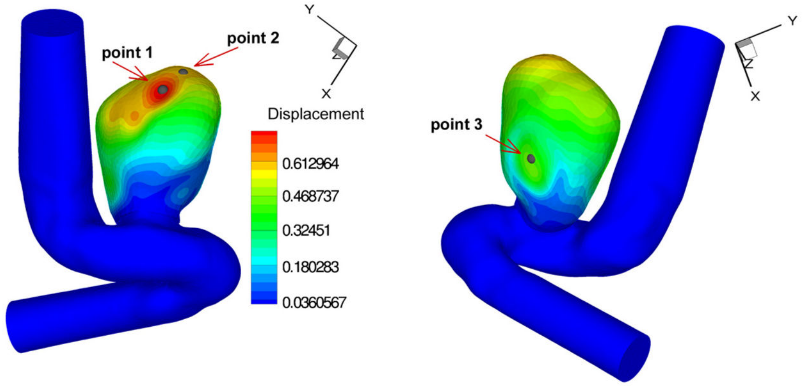

Fig. 22.

Positions of the history points in patient-specific aneurysm test I at $t = 3T/7$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

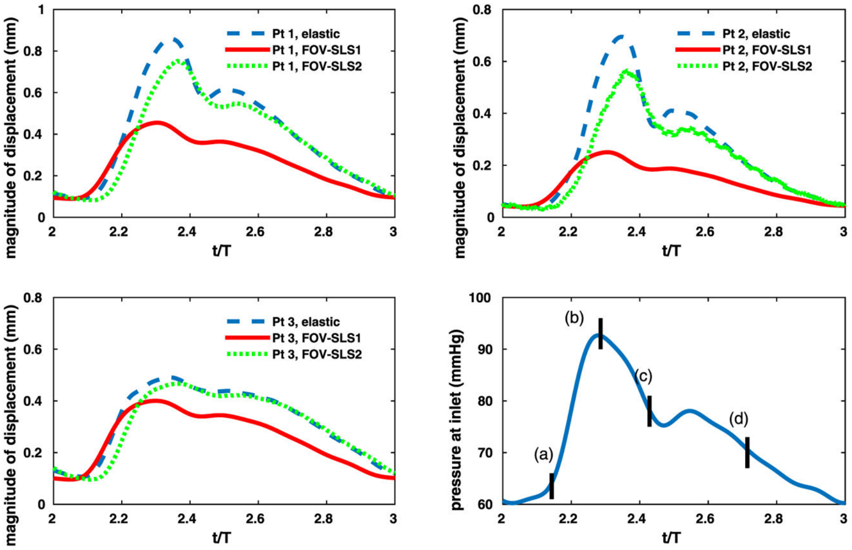

Fig. 23.

Time traces in patient-specific aneurysm with viscoelastic arterial wall models. Upper left, upper right, bottom left: displacement profiles for corresponding points; bottom right: pressure imposed at the inlet.

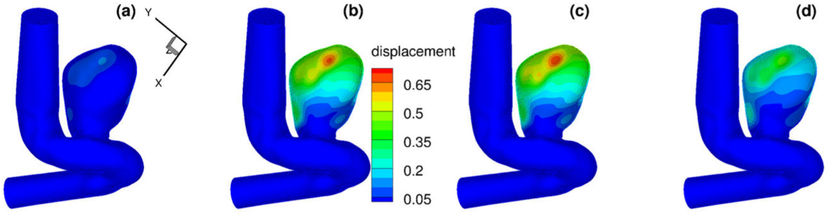

Fig. 24.

Displacements (in mm) for patient-specific aneurysm with pure elastic arterial wall model, at $\frac{T}{7}$ (upper left), $\frac{2T}{7}$ (upper right), $\frac{3T}{7}$ (bottom left) and $\frac{5T}{7}$ (bottom right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

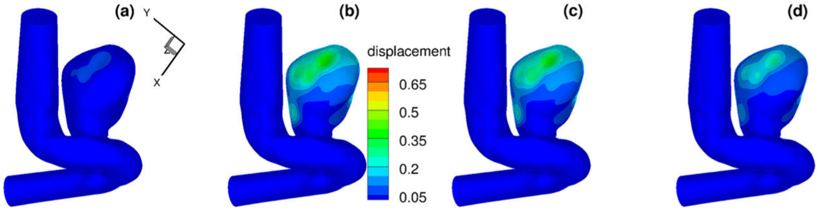

Fig. 25.

Displacements (in mm) for patient-specific aneurysm with FOV-SLS1 model, at $\frac{T}{7}$ (upper left), $\frac{2T}{7}$ (upper right), $\frac{3T}{7}$ (bottom left) and $\frac{5T}{7}$ (bottom right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

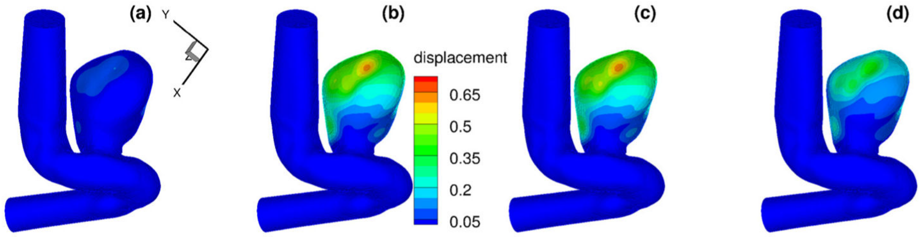

Fig. 26.

Displacements (in mm) for patient-specific aneurysm with FOV-SLS2 model, at $\frac{T}{7}$ (upper left), $\frac{2T}{7}$ (upper right), $\frac{3T}{7}$ (bottom left) and $\frac{5T}{7}$ (bottom right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

Table 1

t-convergence tests for fractional-order SLS model: parameters for cases.

	α	τ_e	τ_σ	E	ν	ρ_s
FS1	0.25	0	10.5	100	0.3	1000
FS2	0.75	0	10.5	100	0.3	1000

Table 2

Parameters used in the idealized artery model simulations.

Parameter	Value
Young’s Modulus 0.3	Mpa
Poisson ratio	0.3
Blood density	1.06 g/cm^3
Wall thickness	0.0105 cm
Inlet radius	0.1 cm
Kinematic viscosity	3.8×10^{-2} cm^2/s
Average flow rate	20 ml/min
Time step size	9.4×10^{-5} s
Polynomial order	3
Fictitious mass coefficient	2.16

Table 3

Parameters employed in fractional-order SLS models for the idealized artery simulations.

	α	τ_e (s)	τ_σ (s)
SLS1	N/A	0.05	0.025
SLS2	N/A	29.3	16.9
FOV-SLS1	0.29	1.84	0.076
FOV-SLS2	0.20	11.74	7.61

Table 4

Parameters used in the patient-specific aneurysm simulations with fractional order models.

Parameter	Value
Young’s Modulus	3 Mpa
Poisson ratio	0.3
Blood density	1.06 g/cm^3
Arterial wall density	2.12 g/cm^3
Wall thickness	0.01 cm
Inlet radius	0.224 cm
Kinematic viscosity	3.8×10^{-2} cm^2/s
Womersley Number	4.2
Average flow rate	70 ml/min
Time step size	6.3×10^{-4} s
Mass damping parameter	0 s^{-1}
Stiffness damping parameter	0 s
Resistant parameter	7.844×10^{2} dyn · s/cm^5
Capacitance parameter	7.285×10^{-5} cm^5/dyn
Polynomial order	2
Fictitious mass coefficient	2.08