Commun Nonlinear Sci Numer Simulat 000 (2019) 104876

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns

Research paper

A 3D OpenFOAM based finite volume solver for incompressible Oldroyd-B model with infinity relaxation time

Jing Tian[a,*], Mingchao Cai[b], Zhendong Luo[c], Yuncheng You[d], Goong Chen[e]

[a] Department of Mathematics, Towson University Towson, MD 21252-0001, United States
[b] Department of Mathematics, Morgan State University Baltimore, MD 21251, United States
[c] School of Mathematics and Physics, North China Electric Power University Beijing, China
[d] Department of Mathematics and Statistics, University of South Florida Tampa, FL 33620, United States
[e] Department of Mathematics and Institute for Quantum Science and Engineering, Texas A&M University College Station, TX 77843-3368, United States

ARTICLE INFO

Article history:
Received 15 November 2018
Revised 28 May 2019
Accepted 9 June 2019
Available online xxx

ABSTRACT

In this paper, we develop a Finite Volume solver for a 3D incompressible Oldroyd-B model with infinity relaxation time. The Finite Volume solver is implemented by using a leading open-source computational mechanics software OpenFOAM. We have imposed the divergence free condition as a constraint on the momentum equation to derive a pressure equation and a predictor-corrector procedure is applied when solving the velocity field. Both stability analysis and numerical experiments are given to show the robustness and accuracy of our algorithm. Two concrete examples on a cubical domain and a dumbbell are computed and illustrated.

1. Introduction

Viscoelastic states can be viewed as an intermediate state between fluid and solid. Viscoelastic fluid flows exhibit both elastic behavior and fluid properties [2,5,7,9]. Typically, the flow motion of viscoelastic fluids is described by using the Oldroyd-B model [7–9,11,14,18]. This model can be regarded as an extension of the Upper Convected Maxwell model and is also equivalent to a fluid filled with elastic bead and spring dumbbells [18]. A general form of a viscoelastic model for an Oldroyd-B fluid may be given as follows:

$$\begin{cases} \rho \boldsymbol{u}_t + \rho \boldsymbol{u}\cdot\nabla\boldsymbol{u} - 2\eta_s \mathrm{div}\boldsymbol{D} = -\nabla p + \mathrm{div}(\boldsymbol{\tau}), \\ \boldsymbol{\tau}_t + \boldsymbol{u}\cdot\nabla\boldsymbol{\tau} = \boldsymbol{\tau}\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\boldsymbol{\tau} + \frac{1}{\lambda}(2\eta_p\boldsymbol{D} - \boldsymbol{\tau}), \\ \mathrm{div}\ \boldsymbol{u} = 0. \end{cases} \tag{1.1}$$

Here, ρ is the density, $\boldsymbol{u}$ the velocity, p the pressure, η_s the solvent viscosity, $\boldsymbol{D}$ denotes the rate of deformation tensor, η_p the polymer viscosity, λ the relaxation time, and $\boldsymbol{\tau} = \boldsymbol{F}\boldsymbol{F}^T$ with $\boldsymbol{F}$ being the deformation tensor. In (1.1), by taking $\rho = 1$ and letting the relaxation time λ go to infinity, we obtain:

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \mathrm{div}\left(\boldsymbol{F}\boldsymbol{F}^T\right), \\ \boldsymbol{F}_t - (\nabla\boldsymbol{u})\boldsymbol{F} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{F} = 0, \\ \mathrm{div}\ \boldsymbol{u} = 0. \end{cases} \tag{1.2}$$

* Corresponding author.
E-mail address: jtian@towson.edu (J. Tian).

https://doi.org/10.1016/j.cnsns.2019.104876

In the above system of equations, all the variables are defined on $\Omega \times (0, T)$ with $\Omega \subset \mathbb{R}^3$ and $T > 0$, $\boldsymbol{u} = (u, v, w)$ is the unknown fluid velocity, and $\nu = 1/Re$ with Re being the Reynolds number. Associated with the PDE system, there are also the boundary and initial conditions:

$$\begin{cases} \boldsymbol{u}(x, y, z, t) = \boldsymbol{u}_0(x, y, z, t), \ \boldsymbol{F}(x, y, z, t) = \boldsymbol{F}_0(x, y, z, t), & (x, y, z, t) \in \partial\Omega \times (0, T), \\ \boldsymbol{u}(x, y, z, 0) = \boldsymbol{u}^0(x, y, z), \ \boldsymbol{F}(x, y, z, 0) = \boldsymbol{F}^0(x, y, z), & (x, y, z) \in \Omega. \end{cases} \tag{1.3}$$

Here $\boldsymbol{u}_0 = (u_0, v_0, w_0)$ and $\boldsymbol{u}^0 = (u^0, v^0, w^0)$. $\boldsymbol{F}^0(x, y, z)$, $u^0(x, y, z)$, $v^0(x, y, z)$, $w^0(x, y, z)$, and $\boldsymbol{F}_0(x, y, z, t)$, $u_0(x, y, z, t)$, $v_0(x, y, z, t)$, $w_0(x, y, z, t)$ are all given sufficiently smooth functions. In addition, we have:

$$\det\left(\boldsymbol{F}^0(x, y, z)\right) = 1,$$

and

$$\operatorname{div}\left(u^0, v^0, w^0\right) = \operatorname{div}(u_0, v_0, w_0) = 0.$$

The incompressible viscoelastic fluid system (1.2) is one of the basic macroscopic models for viscoelastic flows. It describes many important physical models, for example, some complicated hydrodynamic and rheological behaviors of fluids and the electromagnetic behaviors of the materials [2,7,9,16]. This system contains a linear momentum equation (force balance law), the incompressibility constraint, and a microscopic transport equation specifying the special transport of the elastic tensor $\boldsymbol{F}$ [10]. The presence of the viscosity term $\nu\Delta\boldsymbol{u}$ prevents the system from possessing the scaling invariant properties. Moreover, any distortion of microstructures, patterns or configurations in the dynamical flow involves the deformation tensor $\boldsymbol{F}$ [10]. Theoretical analysis of the incompressible viscoelastic fluid (1.1) is intricate because the system is nonlinear and contains not only the velocity vector field and the pressure field but also the deformation tensor matrix. Here we briefly mention some recent progress on the theoretical analysis aspect. In [18], by using a perturbation argument, Zhao and Du studied the local well-posedness of the system (1.2). Lin and Zhang [12] proved that the local solutions of (1.2) can be extended globally and the global solutions decay exponentially fast as time goes to infinity provided that the initial data are sufficiently close to the equilibrium state. In [5], Chen and Zhang presented results for the global existence of small solutions of (1.2).

Although some theoretical advances as mentioned above have been made, a thorough understanding of this model is far from complete. When the initial and boundary conditions are non-smooth functions or if the domain is irregular, the assumptions made in theoretical analysis are often not satisfied, and one has to resort to numerical investigations. Nevertheless, developing numerical solution methods for the system (1.2) is a challenging task, *particularly when the spatial dimension is three*. We comment here that there are already plenty of studies for the viscoelastic fluid flow model, but most of these works are either based on the formulation (1.1) [4] or for the 2D case of system (1.2) [13]. The difficulties in numerics for solving (1.2) are: on the one hand, numerically solving (1.2) is more difficult than solving system (1.1) because the larger the relaxation time is, the more instabilities the system may evolve to have; on the other hand, as the spatial dimension increases, so do the complexities of coding and computing cost, *significantly*. In this paper, we develop a *3D solver* for system (1.2).

The prevalent numerical methods for (1.2) are based on finite element or finite difference discretizations. The presence of the nonlinear term in both the momentum equation and the elastic transport equation then engenders the use of nonlinear iterations. Consequently, one often has to use fixed point theorem to prove the existence, uniqueness, and convergence of the numerical solutions. Moreover, the stability and error analysis of these methods rely on the discrete inf-sup condition, which may not hold for some commonly used Finite Elements [1] and Finite Differences. Most existing numerical studies seem to be based on mixed Finite Element Methods. However, constructing inf-sup stable Finite Elements for them are complicated and numerical tests are usually for 2D cases only.

In this paper, we choose to use the Finite Volume method (FVM) to solve the system (1.2). We first rewrite the system into a conservative form, then we discretize the conservative system in a coupled way, in which the nonlinear terms and their discretized approximations are treated as *flux terms*. By doing this, no nonlinear iterations are needed. In addition, we need to take into account of the hyperbolic character of the constitutive equation. In the Finite Element community, people usually introduce stabilizations to deal with the numerical instabilities for (1.2). In our FVM, we apply an *upwind technique* for the convective terms, whose implementation is quite standard and natural. Furthermore, under some assumptions on the time step and the Reynolds number (if the time step is small enough and the Reynolds number is small), we prove that our FVM is *locally stable*. The analysis is based on the discrete Gronwall inequality and some estimates on flux terms under the L^∞- norm.

Our paper is motivated by the FVM developed in the widely acclaimed, open-source computational mechanics software OpenFOAM [3,6,15]. OpenFOAM is a free, open-source software used by thousands of people worldwide in both academic and industrial communities [3]. There are various advantages of this software, chief among them are the following:

(i) It is open-source. All the codes are open to the users to adapt and customize for their special purposes. (In contrast, commercial softwares do not allow access to their source codes.)
(ii) It has user-friendly syntax for Partial Differential Equations. The codes are written in an object-oriented $C{+}{+}$ language, which is a general-purpose advanced programming language. It offers a unified point of view to a collection of numerical methods or a simplified interface for various Partial Differential Equations.

(iii) It has wide range of applications and models ready to use. In particular, its library of numerical packages is comprehensive, containing linear elasticity, CFD with turbulence modeling, combustion, multiphase flows, etcetera.
(iv) It is easy to automate the model and the solution procedure. By pushing buttons of selection, one can easily set up a fully automated solution based on the platform.
(v) The *preprocessing* and *postprocessing* tools are user-friendly. The *preprocessing* mesh generations are based on *blockMesh* or *snappyHexMesh*, which can deal with unstructured meshes. For the postprocessing, one can work with *ParaView*, or one can use other third-party commercial products such as *Gambit, EnSight*, etcetera.

We have successfully used OpenFOAM to compute the 3D model problem (1.2) in this paper and totally appreciated the many advantages offered by OpenFOAM as described above.

We came back and asked ourselves the important question: *"Can theoretical error estimates be analyzed, based on the FVM numerical scheme as afforded by OpenFOAM?"*

The answer is affirmative. Our study here has demonstrated that theoretical error analysis and practical numerical computation (based on an open-source software) can be successfully fused together in the problem solving of this concrete viscoelastic flow model, even for a relatively complex (dumbell-shaped domain with cavity) geometry, without adding an excessive extra burden of coding work.

The rest of our paper is organized as follows. In Section 2, we introduce an FVM-OpenFOAM scheme for the system (1.2). In Section 3, we conduct stability analysis of our scheme. In Section 4, we test the stability of our algorithm by offering two concrete computed examples. Concluding remarks are given the in the Section 5. We provide essential parts of our OpenFOAM codes at the end of the paper.

2. Finite-volume discretization

To present our FVM for discretizing (1.2), we first rewrite it into a conservative form and then discuss how to approximate each equation in the system. Because of the incompressible condition div $\boldsymbol{u} = 0$, we first cast the system (1.2) into the following conservative form:

$$\boldsymbol{u}_t + \mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{u}) - \mathrm{div}(\nu\nabla\boldsymbol{u} - p\mathbf{I}) = \mathrm{div}\left(\boldsymbol{F}\boldsymbol{F}^T\right), \tag{2.1}$$

$$\boldsymbol{F}_t + \mathrm{div}(\boldsymbol{u}\boldsymbol{F}) = (\nabla\boldsymbol{u})\boldsymbol{F}, \tag{2.2}$$

$$\mathrm{div}\ \boldsymbol{u} = 0. \tag{2.3}$$

Again, the boundary and initial conditions are provided by (1.3). Among the three commonly used methods for spatial discretization [17]: Finite Element, Finite Difference and Finite Volume methods, the FVM is advantageous in that it is flux conservative and therefore mimics physical laws. In this work, we adopt the Finite Volume method to solve the system numerically.

The 3D domain Ω is subdivided by using cells. Each cell is a cube called "control volume", with volume V. As shown Fig. 1, we have four control volumes (with two of them centered at N and P), and we denote S_f as the surface area. To show the Finite Volume discretization of Eq. (2.2), we integrate (2.2) over a control volume V:

$$\int_V \boldsymbol{F}_t dV + \int_V \mathrm{div}(\boldsymbol{u}\boldsymbol{F})dV = \int_V (\nabla\boldsymbol{u})\boldsymbol{F}dV. \tag{2.4}$$

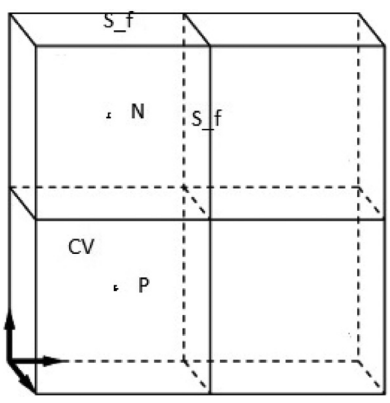

Fig. 1. An illustration of the control volume.

Then, we approximate (2.4) term by term. The first term is:

$$\int_V \boldsymbol{F}_t dV \approx V\frac{\boldsymbol{F}_P^{n+1}-\boldsymbol{F}_P^n}{\Delta t}.$$

Here and thereafter, we denote Δt as the time step. In the above equation, $\boldsymbol{F}_P^{n+1}$ and $\boldsymbol{F}_P^n$ represent, respectively, the value of $\boldsymbol{F}$ at the cell center P at the $(n+1)$-th and n-th time steps. For the second term on the left hand side of (2.4), we apply the divergence theorem and obtain:

$$\int_V \mathrm{div}(\boldsymbol{u}\boldsymbol{F})dV = \int_S d\boldsymbol{S}\cdot(\boldsymbol{u}\boldsymbol{F}) \approx \sum_f \boldsymbol{S}_f\cdot\boldsymbol{u}_f\boldsymbol{F}_f = \sum_f \boldsymbol{\phi}\boldsymbol{F}_f,$$

where $\boldsymbol{S}_f$ denotes the face area vector of the face f. So $\boldsymbol{S}_f = S_f\boldsymbol{n}_f$, where S_f and $\boldsymbol{n}_f$ are the surface area and the unit outward normal vector of the face f, respectively. We denote $\boldsymbol{\phi} = \boldsymbol{S}_f\cdot\boldsymbol{u}_f$ and apply the upwind scheme to obtain $\boldsymbol{F}_f$, i.e.,

$$\boldsymbol{F}_f = \begin{cases} \boldsymbol{F}_P & \text{for} \quad \boldsymbol{\phi}\geq 0, \\ \boldsymbol{F}_N & \text{for} \quad \boldsymbol{\phi}\leq 0. \end{cases}$$

The term on the right hand side of (2.4) is approximated as follows:

$$\int_V (\nabla\boldsymbol{u})\boldsymbol{F}dV \approx \sum_f \left(S_f\boldsymbol{n}_f\cdot\boldsymbol{u}_f\right)\boldsymbol{F}.$$

Here, $\boldsymbol{u}_f = 0.5(\boldsymbol{u}_P+\boldsymbol{u}_N)$.

Consequently, the discretization form of equation $\boldsymbol{F}_t + \mathrm{div}(\boldsymbol{u}\boldsymbol{F}) = (\nabla\boldsymbol{u})\boldsymbol{F}$ is:

$$V\frac{\boldsymbol{F}_P^{n+1}-\boldsymbol{F}_P^n}{\Delta t} + \sum_f \boldsymbol{\phi}^n\boldsymbol{F}_f^n = \sum_f \left(S_f\boldsymbol{n}_f\cdot\boldsymbol{u}_f^n\right)\boldsymbol{F}^n. \tag{2.5}$$

Next, we will illustrate how to discretize Eq. (2.1). By integrating every term over a control volume V, we have:

$$\int_V \boldsymbol{u}_t dV + \int_V \mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{u})dV + \int_V \nabla p dV = \int_V \nu\Delta\boldsymbol{u}dV + \int_V \mathrm{div}\left(\boldsymbol{F}\boldsymbol{F}^T\right)dV.$$

Similarly, we approximate the above equation term by term:

$$\begin{aligned} \int_V \boldsymbol{u}_t dV &\approx V\frac{\boldsymbol{u}_P^{n+1}-\boldsymbol{u}_P^n}{\Delta t}, \\ \int_V \mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{u})dV &\approx \int_V \mathrm{div}\left(\boldsymbol{u}^n\otimes\boldsymbol{u}^{n+1}\right)dV = \int_S d\boldsymbol{S}\cdot\left(\boldsymbol{u}^n\otimes\boldsymbol{u}^{n+1}\right) \\ &= \sum_f \boldsymbol{S}_f\cdot\boldsymbol{u}_f^n\boldsymbol{u}_f^{n+1} = \sum_f \boldsymbol{\phi}^n\boldsymbol{u}_f^{n+1}, \end{aligned} \tag{2.6}$$

where $\boldsymbol{u}_f$ is approximated by using the upwind scheme:

$$\boldsymbol{u}_f = \begin{cases} \boldsymbol{u}_P & \text{for} \quad \boldsymbol{\phi}^n\geq 0, \\ \boldsymbol{u}_N & \text{for} \quad \boldsymbol{\phi}^n\leq 0. \end{cases}$$

We comment here that in approximating $\mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{u})$, an explicit technique is applied, i.e. $\mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{u}) \approx \mathrm{div}\left(\boldsymbol{u}^n\otimes\boldsymbol{u}^{n+1}\right)$. We also have:

$$\int_V \nabla p dV \approx \sum_f \boldsymbol{S}_f p_f = \sum_f S_f\boldsymbol{n}_f p_f,$$

where $p_f = 0.5(P_P+P_N)$. The Laplacian term is discretized as follows:

$$\int_V \mathrm{div}(\nu\nabla\boldsymbol{u})dV = \int_S d\boldsymbol{S}\cdot(\nu\nabla\boldsymbol{u}) \approx \sum_f \nu\boldsymbol{S}_f\cdot(\nabla\boldsymbol{u})_f = \sum_f \nu S_f\frac{\boldsymbol{u}_N^{n+1}-\boldsymbol{u}_P^{n+1}}{|\boldsymbol{d}|},$$

where $\boldsymbol{d}$ is the length vector pointing from the cell center P to the cell center N; For the term containing $\boldsymbol{F}$, we approximate it by:

$$\int_V \mathrm{div}\left(\boldsymbol{F}\boldsymbol{F}^T\right)dV = \int_S d\boldsymbol{S}\cdot\left(\boldsymbol{F}_f\boldsymbol{F}_f^T\right) \approx \sum_f S_f\boldsymbol{n}_f\cdot\left(\boldsymbol{F}_f\boldsymbol{F}_f^T\right),$$

where $\boldsymbol{F}_f = 0.5(\boldsymbol{F}_P+\boldsymbol{F}_N)$.

Therefore, the approximation of (2.1) reads follows:

$$V\frac{\boldsymbol{u}_P^{n+1}-\boldsymbol{u}_P^n}{\Delta t}+\sum_f \boldsymbol{\phi}^n\boldsymbol{u}_f^{n+1}+\sum_f S_f\boldsymbol{n}_f p_f^n=\sum_f \nu S_f\frac{\boldsymbol{u}_N^{n+1}-\boldsymbol{u}_P^{n+1}}{|\boldsymbol{d}|}+\sum_f S_f\boldsymbol{n}_f\cdot\left(\boldsymbol{F}_f\boldsymbol{F}_f^T\right)^n. \tag{2.7}$$

There is no pressure equation, but we can impose the divergence free condition as a constraint on the momentum equation in order to derive a pressure equation. To see this more clearly, we first rewrite (2.1) as:

$$\boldsymbol{u}_t+\mathrm{div}(\boldsymbol{u}\otimes\boldsymbol{u})-\nu\Delta\boldsymbol{u}-\mathrm{div}\left(\boldsymbol{F}\boldsymbol{F}^T\right)=\nabla p.$$

Then, we discretizate just as how we obtain (2.7), but keeping the pressure gradient in its original form. Denoting the diagonal matrix on the left hand side by $\boldsymbol{A}$ (the coefficients corresponding to terms of $\boldsymbol{u}_P$) and the off-diagonal matrix on the left hand side by $\boldsymbol{H}(\boldsymbol{u})$ (the coefficients corresponding to terms of $\boldsymbol{u}_N$), we have $\boldsymbol{A}\cdot\boldsymbol{u}-\boldsymbol{H}(\boldsymbol{u})=-\nabla p$. Taking the divergence on both sides, and applying div $\boldsymbol{u}=0$, we derive an equation for p:

$$\mathrm{div}\left(\boldsymbol{A}^{-1}\cdot\nabla p\right)=\mathrm{div}\left(\boldsymbol{A}^{-1}\cdot\boldsymbol{H}(\boldsymbol{u})\right).$$

Integrating over the control volume V and applying discrete approximations on faces, we obtain the following discretized equation for p:

$$\sum_f \boldsymbol{A}^{-1}S_f\frac{p_N^{n+1}-p_P^{n+1}}{|\boldsymbol{d}|}=\sum_f S_f\boldsymbol{n}_f\left(\boldsymbol{A}_f^{-1}\boldsymbol{H}_f(\boldsymbol{u})\right)^n. \tag{2.8}$$

In summary, the discretized form of the system (2.1)–(2.3) reads as follows:

$$V\frac{\boldsymbol{u}_P^{n+1}-\boldsymbol{u}_P^n}{\Delta t}+\sum_f \boldsymbol{\phi}^n\boldsymbol{u}_f^{n+1}+\sum_f S_f\boldsymbol{n}_f p_f^n=\sum_f \nu S_f\frac{\boldsymbol{u}_N^{n+1}-\boldsymbol{u}_P^{n+1}}{|\boldsymbol{d}|}+\sum_f S_f\boldsymbol{n}_f\cdot\left(\boldsymbol{F}_f\boldsymbol{F}_f^T\right)^n, \tag{2.9}$$

$$V\frac{\boldsymbol{F}_P^{n+1}-\boldsymbol{F}_P^n}{\Delta t}+\sum_f \boldsymbol{\phi}^n\boldsymbol{F}_f^n=\sum_f\left(S_f\boldsymbol{n}_f\cdot\boldsymbol{u}_f^n\right)\boldsymbol{F}^n, \tag{2.10}$$

$$\sum_f \boldsymbol{A}^{-1}S_f\frac{p_N^{n+1}-p_P^{n+1}}{|\boldsymbol{d}|}=\sum_f S_f\boldsymbol{n}_f\left(\boldsymbol{A}_f^{-1}\boldsymbol{H}_f(\boldsymbol{u})\right)^n. \tag{2.11}$$

A predictor-corrector procedure is employed to solve this discretized system. The solution steps of the procedure are:

Step 1: using the initial condition and the momentum Eq. (2.9) to solve for $\boldsymbol{u}$; the obtained solution is denoted as a *predictor* $\boldsymbol{u}_1$.

Step 2: using $\boldsymbol{u}_1$ to supplement the right hand side of (2.11), and applying the pressure Eq. (2.11) to solve for p. Then, using the p to correct the velocity field by solving the momentum Eq. (2.9) again. The new velocity field is denoted as a corrector $\boldsymbol{u}_2$. By this way, $\boldsymbol{u}_2$ satisfies the divergence free condition.

Step 3: using the $\boldsymbol{u}_2$ to solve the tensor Eq. (2.10) for $\boldsymbol{F}$.

Step 4: repeating Steps 1–3 for the next time step. End.

3. Stability of the scheme

In this section, we focus on the stability analysis of our numerical scheme. To study the stability of our FVM, we will treat it as a finite difference scheme (based on a Finite Volume technique). For simplicity, we only present the details for the first component of each equation in (2.9)–(2.11). First, we see that the equation corresponding to the first component of (2.9) is:

$$\begin{aligned}
V\frac{u_{i,j,k}^{n+1}-u_{i,j,k}^n}{\Delta t}=&\ \nu S_x\frac{u_{i+1,j,k}^{n+1}-2u_{i,j,k}^{n+1}+u_{i-1,j,k}^{n+1}}{\Delta x}+\nu S_y\frac{u_{i,j+1,k}^{n+1}-2u_{i,j,k}^{n+1}+u_{i,j-1,k}^{n+1}}{\Delta y}\\
&+\nu S_z\frac{u_{i,j,k+1}^{n+1}-2u_{i,j,k}^{n+1}+u_{i,j,k-1}^{n+1}}{\Delta z}-\left[(\phi^n)_u^+\left(u_{i,j,k}^{n+1}\right)_x^-+(\phi^n)_u^-\left(u_{i,j,k}^{n+1}\right)_x^+\right]\\
&-\left[(\phi^n)_v^+\left(u_{i,j,k}^{n+1}\right)_y^-+(\phi^n)_v^-\left(u_{i,j,k}^{n+1}\right)_y^+\right]-\left[(\phi^n)_w^+\left(u_{i,j,k}^{n+1}\right)_z^-+(\phi^n)_w^-\left(u_{i,j,k}^{n+1}\right)_z^+\right]\\
&-\frac{S_x}{2}\left(p_{i+1,j,k}^n-p_{i-1,j,k}^n\right)+\frac{S_x}{2}\left[(F_{1,1}^2+F_{1,2}^2+F_{1,3}^2)_{i+1,j,k}^n-\left(F_{1,1}^2+F_{1,2}^2+F_{1,3}^2\right)_{i-1,j,k}^n\right]\\
&+\frac{S_y}{2}\left[(F_{1,1}F_{2,1}+F_{1,2}F_{2,2}+F_{1,3}F_{2,3})_{i,j+1,k}^n-(F_{1,1}F_{2,1}+F_{1,2}F_{2,2}+F_{1,3}F_{2,3})_{i,j-1,k}^n\right]\\
&+\frac{S_z}{2}\left[(F_{1,1}F_{3,1}+F_{1,2}F_{3,2}+F_{1,3}F_{3,3})_{i,j,k+1}^n-(F_{1,1}F_{3,1}+F_{1,2}F_{3,2}+F_{1,3}F_{3,3})_{i,j,k-1}^n\right].
\end{aligned} \tag{3.1}$$

Here, $V=\Delta x\Delta y\Delta z$, $S_x=\Delta y\Delta z$, $S_y=\Delta x\Delta z$, $S_z=\Delta x\Delta y$, $(\phi^n)_u,(\phi^n)_v,(\phi^n)_w$ are the surface values of u^n, v^n, w^n respectively, $(\phi^n)^+=\max\{\phi^n,0\}$, $(\phi^n)^-=\min\{\phi^n,0\}$ and $(u^n_{i,j,k})^+_x=u^n_{i+1,j,k}-u^n_{i,j,k}$, $(u^n_{i,j,k})^-_x=u^n_{i,j,k}-u^n_{i-1,j,k}$, $(u^n_{i,j,k})^+_y=u^n_{i,j+1,k}-u^n_{i,j,k}$, $(u^n_{i,j,k})^-_y=u_{i,j,k}-u_{i,j-1,k}$, $(u^n_{i,j,k})^+_z=u_{i,j,k+1}-u_{i,j,k}$, $(u^n_{i,j,k})^-_z=u^n_{i,j,k}-u^n_{i,j,k-1}$.

Second, the equation corresponding to the first component of (2.10) is:

$$\begin{aligned}V\frac{F^{n+1}_{1,1,i,j,k}-F^n_{1,1,i,j,k}}{\Delta t}=&-\left[(\phi^n)^+_u\left(F^n_{1,1,i,j,k}\right)^-_x+(\phi^n)^-_u\left(F^n_{1,1,i,j,k}\right)^+_x\right]\\&-\left[(\phi^n)^+_v\left(F^n_{1,1,i,j,k}\right)^-_y+(\phi^n)^-_v\left(F^n_{1,1,i,j,k}\right)^+_y\right]-\left[(\phi^n)^+_w\left(F^n_{1,1,i,j,k}\right)^-_z+(\phi^n)^-_w\left(F^n_{1,1,i,j,k}\right)^+_z\right]\\&+\frac{S_x}{2}F^n_{1,1,i,j,k}\left(u^n_{i+1,j,k}-u^n_{i-1,j,k}\right)+\frac{S_y}{2}F^n_{2,1,i,j,k}\left(u^n_{i,j+1,k}-u^n_{i,j-1,k}\right)+\frac{S_z}{2}F^n_{3,1,i,j,k}\left(u^n_{i,j,k+1}-u^n_{i,j,k-1}\right).\end{aligned}\tag{3.2}$$

Third, the equation corresponding to the first component of (2.11) is:

$$\begin{aligned}&\boldsymbol{A}^{-1}S_x\frac{p^{n+1}_{i+1,j,k}-2p^{n+1}_{i,j,k}+p^{n+1}_{i-1,j,k}}{\Delta x}+\boldsymbol{A}^{-1}S_y\frac{p^{n+1}_{i,j+1,k}-2p^{n+1}_{i,j,k}+p^{n+1}_{i,j-1,k}}{\Delta y}\\&+\boldsymbol{A}^{-1}S_z\frac{p^{n+1}_{i,j,k+1}-2p^{n+1}_{i,j,k}-p^{n+1}_{i,j,k-1}}{\Delta z}=\sum_f S_f\boldsymbol{n}_f\left(\boldsymbol{A}_f^{-1}\boldsymbol{H}_f\right)^n.\end{aligned}\tag{3.3}$$

Simplifying (3.1)–(3.3) and by using Taylor's expansion at a reference point (x_i, y_j, z_k, t_n), we obtain:

$$\begin{aligned}\left(\frac{\partial u}{\partial t}\right)^n_{i,j,k}&=\frac{u^{n+1}_{i,j,k}-u^n_{i,j,k}}{\Delta t}+O(\Delta t),\\\left(\frac{\partial u}{\partial x}\right)^n_{i,j,k}&=\frac{u^n_{i+1,j,k}-u^n_{i-1,j,k}}{2\Delta x}+O\left(\Delta x^2\right),\\\left(\frac{\partial u}{\partial y}\right)^n_{i,j,k}&=\frac{u^n_{i,j+1,k}-u^n_{i,j-1,k}}{2\Delta y}+O\left(\Delta y^2\right),\\\left(\frac{\partial u}{\partial z}\right)^n_{i,j,k}&=\frac{u^n_{i,j,k+1}-u^n_{i,j,k-1}}{2\Delta z}+O\left(\Delta z^2\right),\\\left(\frac{\partial p}{\partial x}\right)^n_{i,j,k}&=\frac{p^n_{i+1,j,k}-p^n_{i-1,j,k}}{2\Delta x}+O\left(\Delta x^2\right),\\\left(\frac{\partial F_{1,1}}{\partial t}\right)^n_{i,j,k}&=\frac{F^{n+1}_{1,1,i,j,k}-F^n_{1,1,i,j,k}}{\Delta t}+O(\Delta t),\\\left(\frac{\partial^2 u}{\partial x^2}\right)^n_{x,j,k}&=\frac{u^{n+1}_{i+1,j,k}-2u^{n+1}_{i,j,k}+u^{n+1}_{i-1,j,k}}{\Delta x^2}+O\left(\Delta x^2\right),\\\left(\frac{\partial^2 u}{\partial y^2}\right)^n_{i,j,k}&=\frac{u^{n+1}_{i,j+1,k}-2u^{n+1}_{i,j,k}+u^{n+1}_{i,j-1,k}}{\Delta y^2}+O\left(\Delta y^2\right),\\\left(\frac{\partial^2 u}{\partial z^2}\right)^n_{i,j,k}&=\frac{u^{n+1}_{i,j,k+1}-2u^{n+1}_{i,j,k}+u^{n+1}_{i,j,k-1}}{\Delta z^2}+O\left(\Delta z^2\right),\\\left(\frac{\partial^2 p}{\partial x^2}\right)^n_{i,j,k}&=\frac{p^{n+1}_{i+1,j,k}-2p^{n+1}_{i,j,k}+p^{n+1}_{i-1,j,k}}{\Delta x^2}+O\left(\Delta x^2\right),\\\left(\frac{\partial^2 p}{\partial y^2}\right)^n_{i,j,k}&=\frac{p^{n+1}_{i,j+1,k}-2p^{n+1}_{i,j,k}+p^{n+1}_{i,j-1,k}}{\Delta y^2}+O\left(\Delta y^2\right),\\\left(\frac{\partial^2 p}{\partial z^2}\right)^n_{i,j,k}&=\frac{p^{n+1}_{i,j,k+1}-2p^{n+1}_{i,j,k}+p^{n+1}_{i,j,k-1}}{2\Delta z}+O\left(\Delta z^2\right),\\\left(\frac{\partial\left(F^2_{1,1}+F^2_{1,2}+F^2_{1,3}\right)}{\partial x}\right)^n_{i,j,k}&=\frac{\left(F^2_{1,1}+F^2_{1,2}+F^2_{1,3}\right)^n_{i+1,j,k}-\left(F^2_{1,1}+F^2_{1,2}+F^2_{1,3}\right)^n_{i-1,j,k}}{2\Delta x}+O\left(\Delta x^2\right),\end{aligned}$$

$$\left(\frac{\partial\left(F_{1,1}F_{2,1}+F_{1,2}F_{2,2}+F_{1,3}F_{2,3}\right)}{\partial y}\right)_{i,j,k}^{n}=\frac{\left(F_{1,1}F_{2,1}+F_{1,2}F_{2,2}+F_{1,3}F_{2,3}\right)_{i,j+1,k}^{n}}{2\Delta y}$$
$$-\frac{\left(F_{1,1}F_{2,1}+F_{1,2}F_{2,2}+F_{1,3}F_{2,3}\right)_{i,j-1,k}^{n}}{2\Delta y}+O\left(\Delta y^{2}\right),$$
$$\left(\frac{\partial\left(F_{1,1}F_{3,1}+F_{1,2}F_{3,2}+F_{1,3}F_{3,3}\right)}{\partial z}\right)_{i,j,k}^{n}=\frac{\left(F_{1,1}F_{3,1}+F_{1,2}F_{3,2}+F_{1,3}F_{3,3}\right)_{i,j,k+1}^{n}}{2\Delta z}$$
$$-\frac{\left(F_{1,1}F_{3,1}+F_{1,2}F_{3,2}+F_{1,3}F_{3,3}\right)_{i,j,k-1}^{n}}{2\Delta z}+O\left(\Delta z^{2}\right).$$

For the other two components of (2.9)–(2.11), similar results can be obtained by using the same kind of arguments. The upwind scheme will result in an $O(\Delta x, \Delta y, \Delta z)$ or $O(\Delta t, \Delta x, \Delta y, \Delta z)$ order accuracy. Therefore, the following error estimates can be obtained:

$$\begin{aligned}&\left|p\left(x_i,y_j,z_k,t_n\right)-p_{i,j,k}^{n}\right|+\left|u\left(x_i,y_j,z_k,t_n\right)-u_{i,j,k}^{n}\right|+\left|v\left(x_i,y_j,z_k,t_n\right)-v_{i,j,k}^{n}\right|\\&\quad+\left|w\left(x_i,y_j,z_k,t_n\right)-w_{i,j,k}^{n}\right|+\left|F_{l,m}\left(x_i,y_j,z_k,t_n\right)-F_{l,m,i,j,k}^{n}\right|\\&\quad=O(\Delta t,\Delta x,\Delta y,\Delta z),\quad 1\le n\le N,1\le i\le I,1\le j\le J,1\le k\le K,m,l=1,2,3,\end{aligned}\tag{3.4}$$

where $I=\frac{1}{\Delta x}\max\limits_{(x_1,y,z),(x_2,y,z)\in\Omega}|x_1-x_2|$, $J=\frac{1}{\Delta y}\max\limits_{(x,y_1,z),(x,y_2,z)\in\Omega}|y_1-y_2|$, $K=\frac{1}{\Delta z}\max\limits_{(x,y,z_1),(x,y,z_2)\in\Omega}|z_1-z_2|$.

In order to prove the stability of the FVM scheme, it is necessary to introduce the following discrete Gronwall lemma.

Lemma 3.1. *(Gronwall's Lemma) Let $\lambda>0$, if $\{a_n\}$, $\{b_n\}$, and $\{c_n\}$ are three positive sequences, and $\{c_n\}$ is monotone, that satisfy:*

$$a_n+b_n\le c_n+\lambda\sum_{i=0}^{n-1}a_i,\quad a_0+b_0\le c_0,$$

then

$$a_n+b_n\le c_n\exp(n\lambda),\ \ n=0,1,2,\cdots.$$

Theorem 3.1. *Under the conditions $\Delta t\cdot(|u|+|v|+|w|)\le\nu$ and $\Delta t\nu\le\min\{\Delta x,\Delta y,\Delta z\}$, the FVM scheme (2.9)-(2.11) is locally stable.*

Proof. If it holds that $\Delta t|u|/\Delta x\le 1/3$, $\Delta t|v|/\Delta y\le 1/3$ and $\Delta t|w|/\Delta z\le 1/3$ as well as $\Delta t(|u|+|v|+|w|)\le\nu$, which implies $\Delta t(\|u\|_\infty+\|v\|_\infty+\|w\|_\infty)\le\nu$, by (3.2), then we have:

$$\begin{aligned}\left|F_{1,1,i,j,k}^{n+1}\right|&\le\left(1-\frac{\Delta t|u|}{\Delta x}-\frac{\Delta t|v|}{\Delta y}-\frac{\Delta t|w|}{\Delta z}\right)\left|F_{1,1,i,j,k}^{n}\right|\\&\quad+\frac{\Delta t|u|}{\Delta x}\left|F_{1,1,i+1,j,k}^{n}\right|+\frac{\Delta t|v|}{\Delta y}\left|F_{1,1,i,j+1,k}^{n}\right|+\frac{\Delta t|w|}{\Delta z}\left|F_{1,1,i,j,k+1}^{n}\right|\\&\quad+\frac{\Delta t}{2\Delta x}\left(\left|u_{i+1,j,k}^{n}\right|+\left|u_{i-1,j,k}^{n}\right|\right)\left|F_{1,1,i,j,k}^{n}\right|\\&\quad+\frac{\Delta t}{2\Delta y}\left(\left|u_{i,j+1,k}^{n}\right|+\left|u_{i,j-1,k}^{n}\right|\right)\left|F_{2,1,i,j,k}^{n}\right|\\&\quad+\frac{\Delta t}{2\Delta z}\left(\left|u_{i,j,k+1}^{n}\right|+\left|u_{i,j,k-1}^{n}\right|\right)\left|F_{3,1,i,j,k}^{n}\right|\\&\le\left(1+\frac{\Delta t}{\Delta x}\|u\|_\infty\right)\|F_{1,1}^{n}\|_\infty+\frac{\Delta t}{\Delta y}\|u\|_\infty\|F_{2,1}^{n}\|_\infty+\frac{\Delta t}{\Delta z}\|u\|_\infty\|F_{3,1}^{n}\|_\infty,\end{aligned}\tag{3.5}$$

where $\|\cdot\|_\infty$ is the $L^\infty(\Omega)$ norm. Similarly, we have:

$$|F_{2,1,i,j,k}^{n}|\le\left(1+\frac{\Delta t}{\Delta y}\|v\|_\infty\right)\|F_{2,1}^{n}\|_\infty+\frac{\Delta t}{\Delta x}\|v\|_\infty\|F_{1,1}^{n}\|_\infty+\frac{\Delta t}{\Delta z}\|w\|_\infty\|F_{3,1}\|_\infty,\tag{3.6}$$

$$|F_{3,1,i,j,k}^{n}|\le\left(1+\frac{\Delta t}{\Delta z}\|w\|_\infty\right)\|F_{3,1}^{n}\|_\infty+\frac{\Delta t}{\Delta x}\|w\|_\infty\|F_{1,1}^{n}\|_\infty+\frac{\Delta t}{\Delta y}\|w\|_\infty\|F_{2,1}\|_\infty.\tag{3.7}$$

Thus, from the assumption, we obtain:

$$\|F_{1,1}^{n+1}\|_\infty\le\left(1+\frac{\nu}{\Delta x}\right)\|F_{1,1}^{n}\|_\infty,n=0,1,2,\cdots,N.\tag{3.8}$$

Summing (3.8) from 0 to n, we obtain:

$$\|F_{1,1}^{n+1}\|_\infty \le \|F_{1,1}^0\|_\infty + \frac{\nu}{\Delta x}\sum_{j=0}^{n}\|F_{1,1}^j\|_\infty, n = 0, 1, 2, \cdots, N. \tag{3.9}$$

Applying the discrete Gronwall Lemma to (3.9), we obtain:

$$\|F_{1,1}^{n+1}\|_\infty \le \|F_{1,1}^0\|_\infty \exp\left(\frac{n\nu}{\Delta x}\right), n = 0, 1, 2, \ldots, N. \tag{3.10}$$

Therefore, the series $\{F_{1,1}^{n+1}\}$ is locally stable provided that the time interval $[0, T]$ is finite. Following a similar procedure, one can prove that the series of the other components of $\boldsymbol{F}$ are also locally stable. In summary, we conclude that $\{\boldsymbol{F}^{n+1}\}$ is locally stable.

Similarly to (3.5), we can obtain:

$$\begin{aligned}\|u^{n+1}\|_\infty \le\ & \|u^n\|_\infty + \frac{\Delta t}{\Delta x}\left(\|F_{1,1}^n\|_\infty^2 + \|F_{1,2}^n\|_\infty^2 + \|F_{1,3}^n\|_\infty^2\right)\\ &+ \frac{\Delta t}{\Delta y}\left(\|F_{1,1}^n\|_\infty\|F_{2,1}^n\|_\infty + \|F_{1,2}^n\|_\infty\|F_{2,2}^n\|_\infty + \|F_{1,2}^n\|_\infty\|F_{2,3}^n\|_\infty\right)\\ &+ \frac{\Delta t}{\Delta z}\left(\|F_{1,1}^n\|_\infty\|F_{3,1}^n\|_\infty + \|F_{1,2}^n\|_\infty\|F_{3,2}^n\|_\infty + \|F_{1,3}^n\|_\infty\|F_{3,3}^n\|_\infty\right).\end{aligned} \tag{3.11}$$

Denoting by $\tau = \min\{\Delta x, \Delta y, \Delta z\}$, we have:

$$\|u^{n+1}\|_\infty \le \|u^0\|_\infty + \frac{9n\Delta t}{\tau}\|F^0\|_\infty^2 \exp\left(\frac{2n\nu}{\tau}\right), n = 0, 1, 2, \cdots, N. \tag{3.12}$$

This shows that the series $\{u^{n+1}\}$ is locally stable if the time interval $[0, T]$ is finite. Following a similar procedure, one can prove that the series of the other components of $\boldsymbol{u}$ are also locally stable, i.e., $\{\boldsymbol{u}^{n+1}\}$ is locally stable. Thus, we conclude that the scheme (2.9)–(2.11) is locally stable. □

Remark 1. Our Finite Volume scheme here is only first-order accurate in space because we employ a first-order upwind technique. Higher order spatial accuracy can be obtained by changing the first-order upwind scheme to a second-order upwind scheme. For example, $\left(u_{i,j,k}^n\right)_x^- = \left(3u_{i,j,k}^n - 4u_{i-1,j,k}^n + u_{i-2,j,k}^n\right)/2\Delta x$ and $\left(u_{i,j,k}^n\right)_x^+ = \left(-u_{i+2,j,k}^n + 4u_{i+1,j,k}^n - 3u_{i,j,k}^n\right)/2\Delta x$.

4. Numerical experiments

To verify the stability and the robustness of our Finite Volume solver, we test two numerical examples for the 3D incompressible viscoelastic fluid system in this section. In both experiments, the time interval is $[0, 1.5]s$; $\nu = 0.01$; the boundary condition for $\boldsymbol{F}$ is set to be $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{n}} = 0$; the Dirichlet boundary condition for $\boldsymbol{u}$ is $\boldsymbol{u} = \boldsymbol{0}$. For notation, we will denote the interior of Ω as $\Omega^\wedge$ and the boundary of Ω as $\partial\Omega$.

(i) As the first example, the computational domain Ω is a cube of size $0.1mm \times 0.1mm \times 0.1mm$, divided into 8000 cells. We set $\Delta x = \Delta y = \Delta z = 0.005mm$. The initial condition for $\boldsymbol{F}$ is:

$$\boldsymbol{F} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.125 \end{bmatrix}. \tag{4.1}$$

The initial condition for $\boldsymbol{u}$ is

$$\boldsymbol{u} = \begin{cases} \boldsymbol{f} & (x, y, z) \in \Omega^\wedge, \\ \boldsymbol{0} & (x, y, z) \in \partial\Omega, \end{cases}$$

$$\boldsymbol{f} = \left(10e^{-8000\left((y-0.055)^2+(z-0.045)^2\right)}, 10e^{-8000\left((x-0.055)^2+(z-0.045)^2\right)}, 10e^{-8000\left((x-0.055)^2+(y-0.055)^2\right)}\right)$$

Although the initial and boundary conditions have explicit expressions, no exact solution is available because of the complicated interaction mechanism of the coupled system. In order to check the stability of our scheme, we conduct two experiments by using two different time step sizes. The first one is $\Delta t = 0.0005s$ while the second is $\Delta t = 0.00025s$. Numerical solutions using different time step sizes for $\boldsymbol{u}$, $\boldsymbol{F}$ and p are provided and compared for Figs. 2–10. A video animation available for online readers can be found in Fig. 11, see also: https://www.dropbox.com/s/jd0mc4f78edkge4/pcube.ogv?dl=0. From the numerical results, we observe that those obtained from using different time step sizes are almost the same, particularly, for p and $\boldsymbol{F}$. The results show that our numerical solutions are convergent, accurate and, therefore, support that our Finite Volume scheme is stable. Moreover, we observe that numerical solutions decay as time evolves. This reflects the energy decay property of the incompressible viscoelastic system. From the results, we see that if a smaller time step size is used, the numerical solution of velocity decays faster, which is perhaps caused by the numerical dissipation of the scheme.

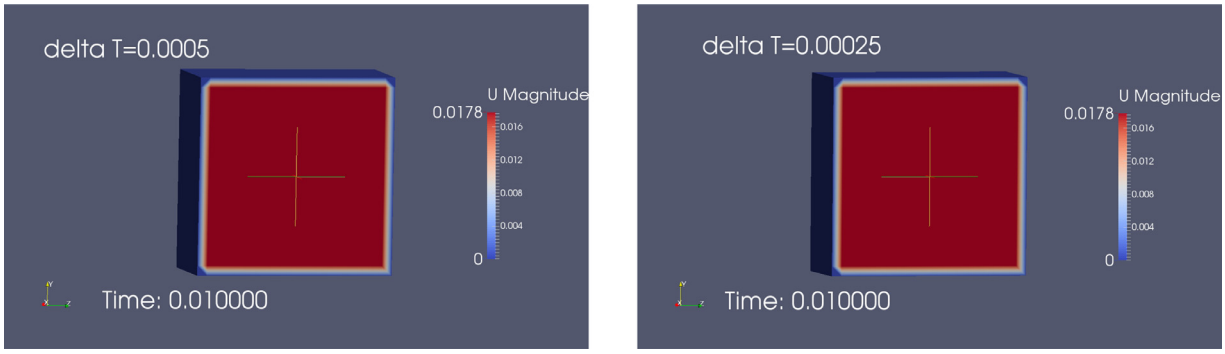

Fig. 2. Comparisons of the profiles of $|\boldsymbol{u}|$ at $t = 0.01$ using different time step sizes.

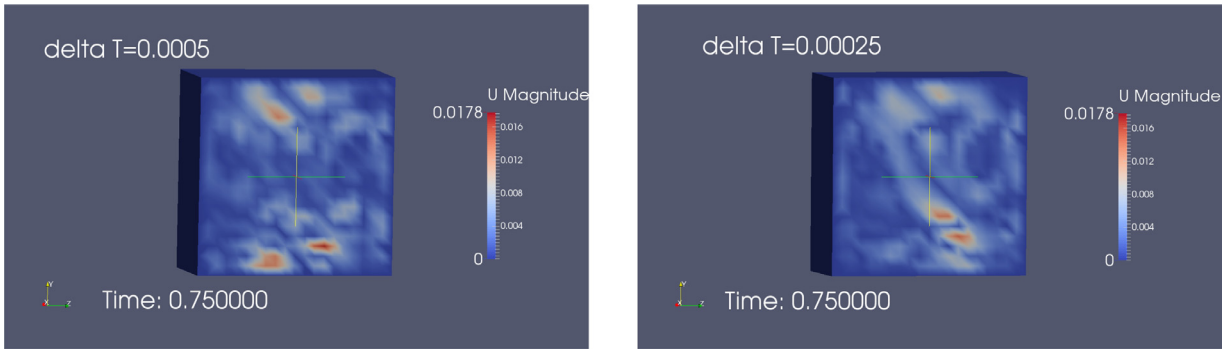

Fig. 3. Comparisons of the profiles of $|\boldsymbol{u}|$ at $t = 0.75$ using different time step sizes.

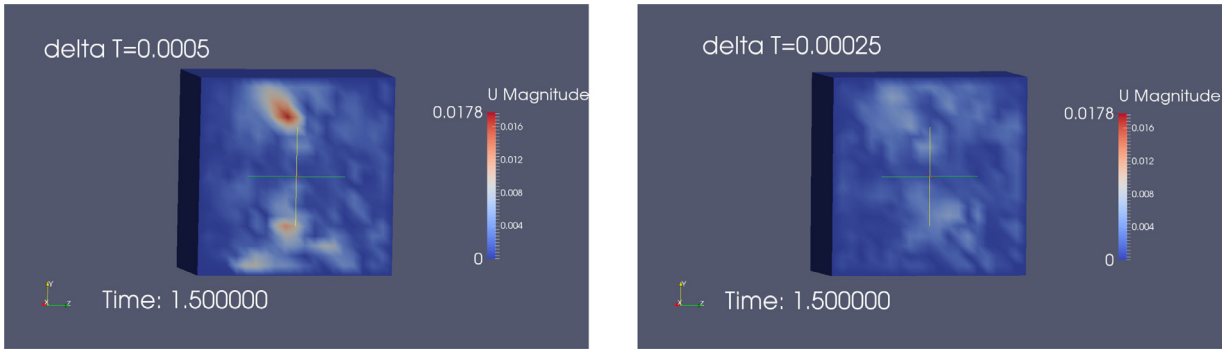

Fig. 4. Comparisons of the profiles of $|\boldsymbol{u}|$ at $t = 1.5$ using different time stepping sizes.

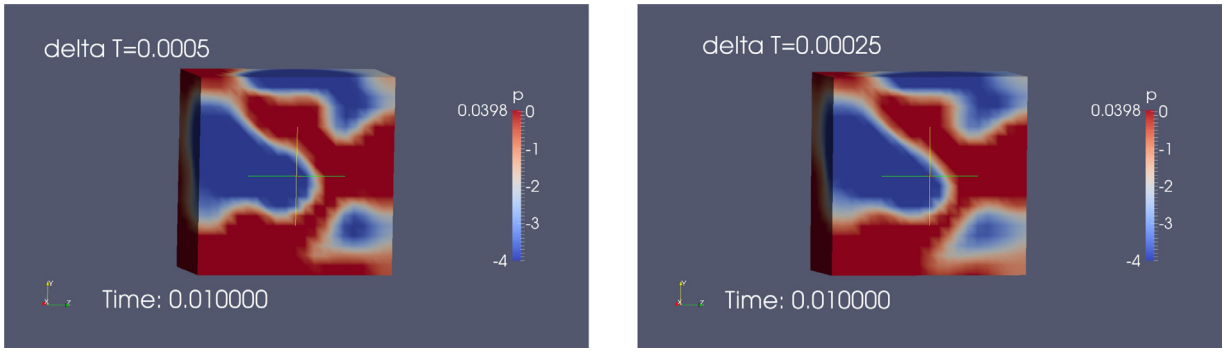

Fig. 5. Comparisons of the profiles of p at $t = 0.01$ using different time stepping sizes.

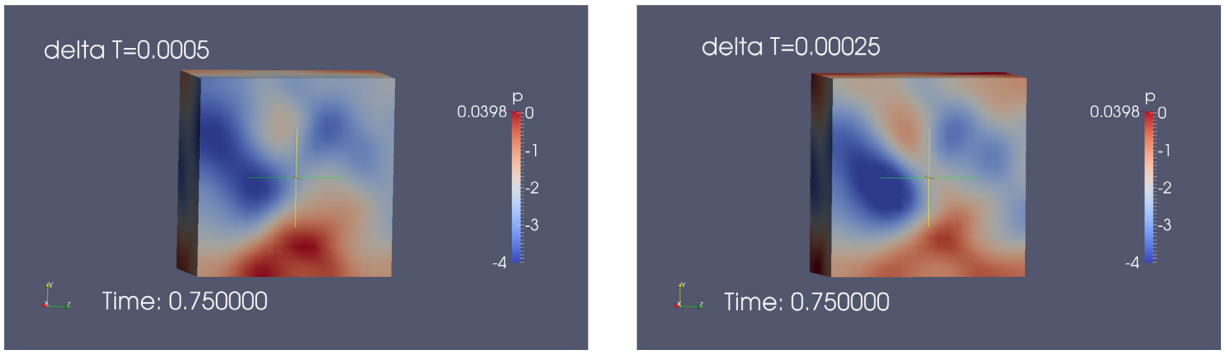

Fig. 6. Comparisons of the profiles of p at $t = 0.75$ using different time stepping sizes.

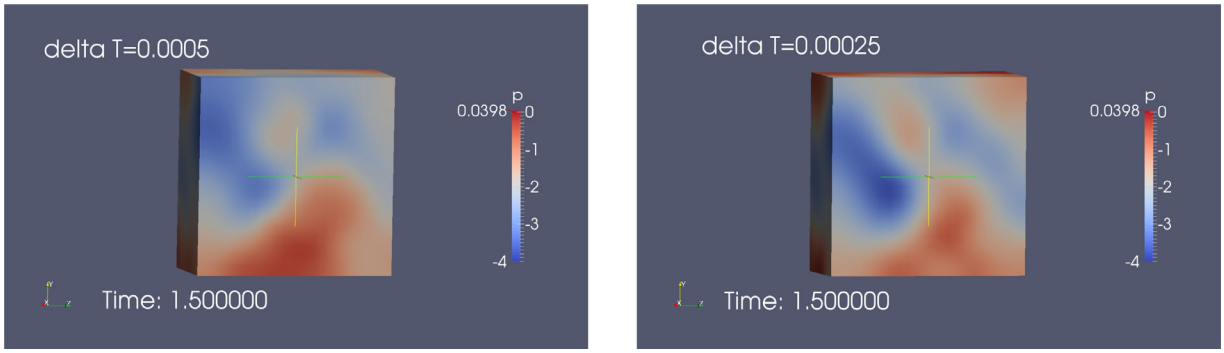

Fig. 7. Comparisons of the profiles of p at $t = 1.5$ using different time stepping sizes.

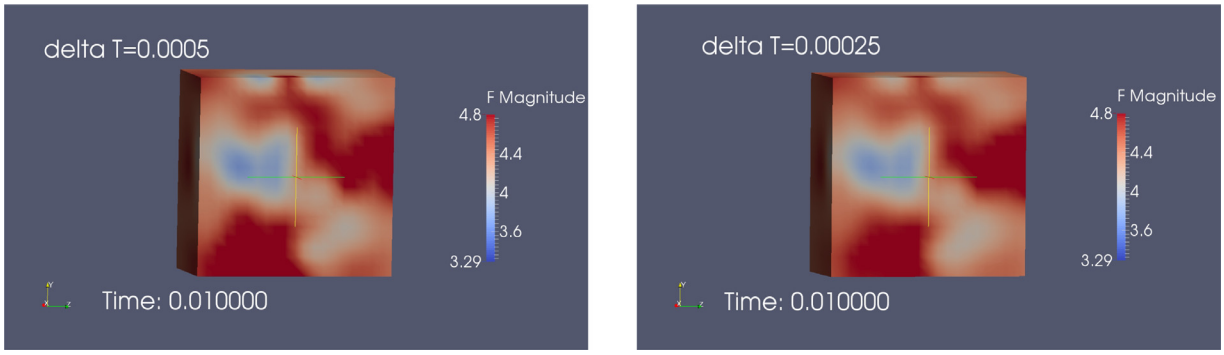

Fig. 8. Comparisons of the profiles of $|\boldsymbol{F}|$ at $t = 0.01$ using different time stepping sizes.

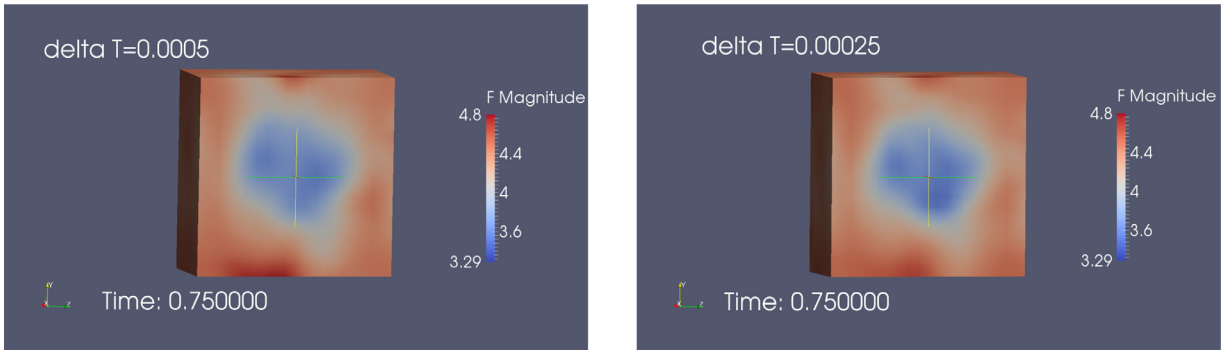

Fig. 9. Comparisons of the profiles of $|\boldsymbol{F}|$ at $t = 0.75$ using different time stepping sizes.

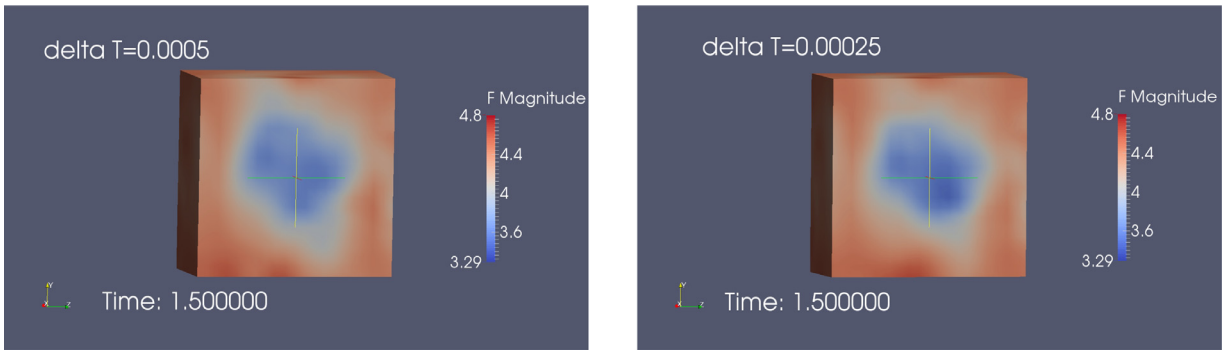

Fig. 10. Comparisons of the profiles of $|\boldsymbol{F}|$ at $t = 1.5$ using different time stepping sizes.

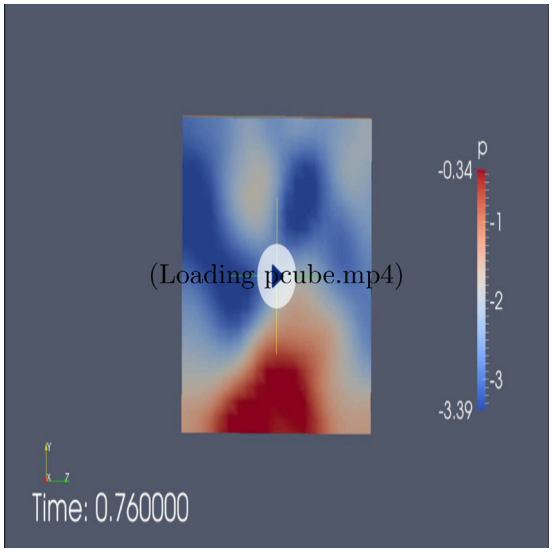

Fig. 11. Video animation for the numerical example whose domain is a cube.

Table 1
Table of numerical errors in L_2 norm. The last column is the final approximation of the order of accuracy.

Δt	$\lvert p_{\Delta t} - p_{\Delta t/2}\rvert$	$\frac{\lvert p_{\Delta t} - p_{\Delta t/2}\rvert}{\lvert p_{\Delta t/2} - p_{\Delta t/4}\rvert}$	$log_2 \frac{\lvert p_{\Delta t} - p_{\Delta t/2}\rvert}{\lvert p_{\Delta t/2} - p_{\Delta t/4}\rvert}$
0.0005	0.08674	2.43105	1.2816
0.00025	0.03568	2.30491	1.20471
0.000125	0.01548		

To validate the order of accuracy in (3.4), we calculate the L_2 norm of the numerical errors by using different time step sizes in Table 1. We choose the scalar variable p to calculate the numerical errors. From the results in Table 1, we see clearly that the order of the time errors is around 1, which verifies the estimate (3.4).

In Fig. 12, we apply log-log plot to Δt versus the numerical errors. From the plot, we see clearly that the slope of numerical errors is approximately 1. Therefore, the time accuracy is of order 1.

In Fig. 13, we apply log-log plot to Δh (here $\Delta h = \Delta x, \Delta y, \Delta z$) versus the numerical errors. From the plot, we see clearly that the slope of numerical errors is approximately equal 1. Therefore, the order of spatial accuracy is 1.

(ii) In the second example, we show that our Finite Volume solver can be applied to the incompressible Oldroyd-B model with complex geometry. In this example, the computational domain Ω is a dumbbell formed by connecting a large ball with a small ball through a handle. The radius of the small ball is $0.7mm$; the radius of the large ball is $1.4mm$. There is a small spherical cavity inside the large ball with radius $0.35mm$. The center of the large ball, small ball and ball cavity are located at $(0.21, -0.01, -2.25)$, $(0.21, -0.01, 2.25)$, $(0.21, -0.01, -2.6)$, respectively. The radius of the (cylindrically shaped) handle is $0.18mm$. The mesh is generated by using the "SnappyHexMesh" utility in OpenFOAM. The initial condition of $\boldsymbol{F}$ is again

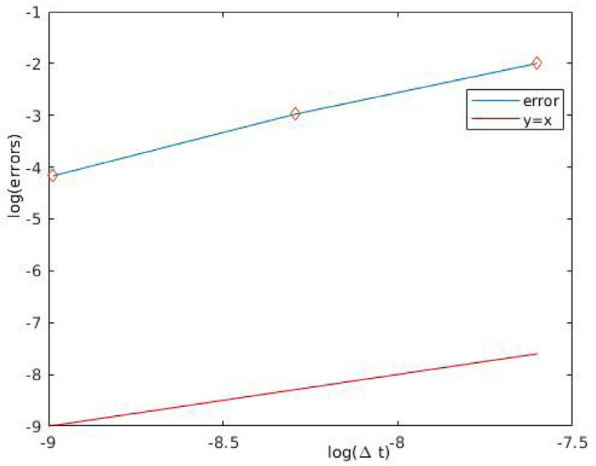

Fig. 12. The loglog plot of Δt versus the numerical errors. The slope of the error plot is (nearly) equal to 1.

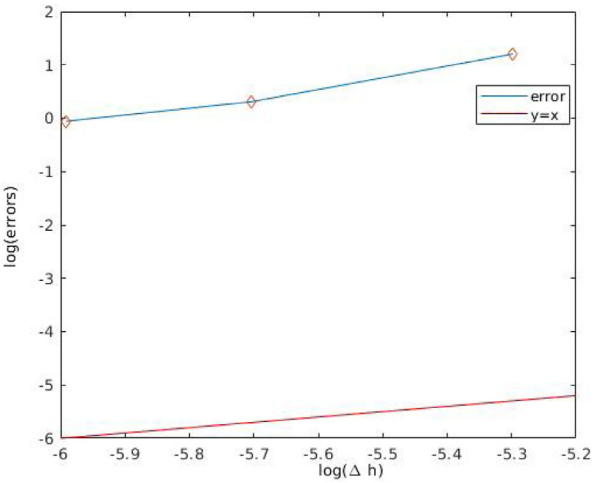

Fig. 13. The loglog plot of the Δh versus the numerical errors. The slope of the error plot is close to 1.

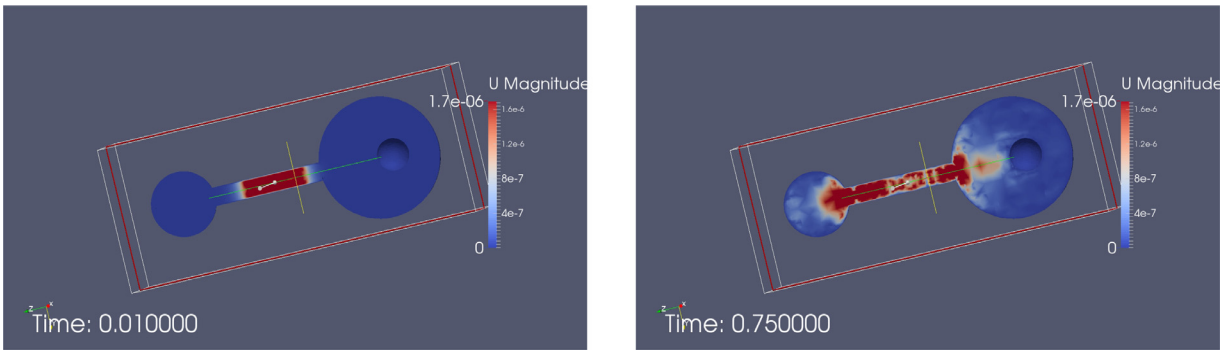

Fig. 14. The profiles of $|\boldsymbol{u}|$ at $t = 0.01$ (left) and at $t = 0.75$ (right).

defined by (4.1). The initial condition for $\boldsymbol{u}$ is:

$$\boldsymbol{u} = \begin{cases} \left(80e^{-8000\left(y^2+z^2\right)}, 80e^{-8000\left(x^2+z^2\right)}, 80e^{-8000\left(x^2+y^2\right)}\right), & x^2+y^2+z^2 < 0.025, \\ 0, & \text{Otherwise.} \end{cases}$$

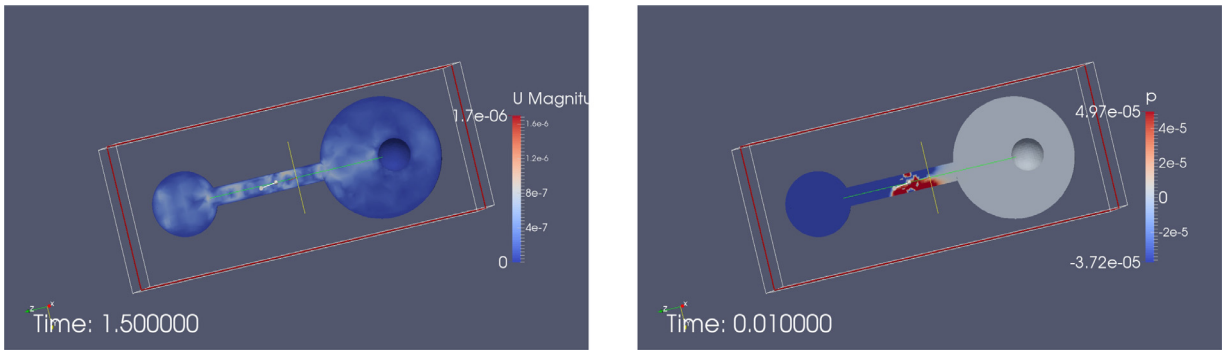

Fig. 15. The profiles of $|\boldsymbol{u}|$ at $t = 1.5$ (left) and the profile of p at $t = 0.01$ (right).

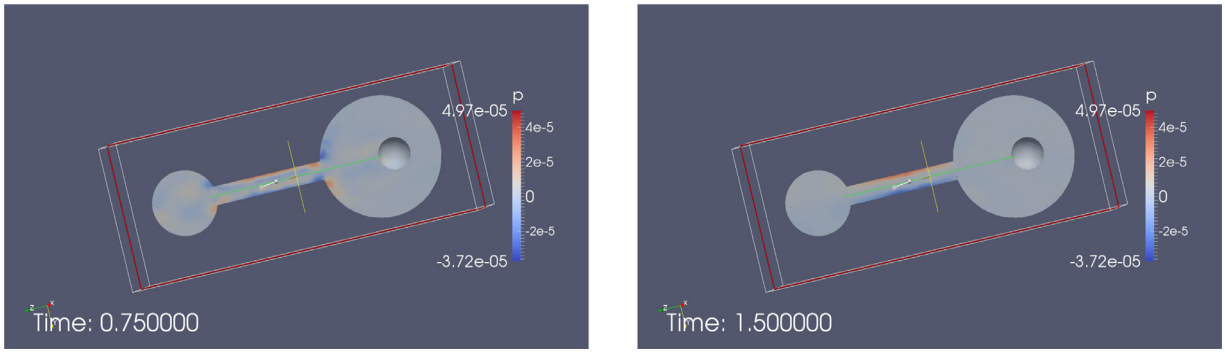

Fig. 16. The profiles of p at $t = 0.75$ (left) and at $t = 1.5$ (right).

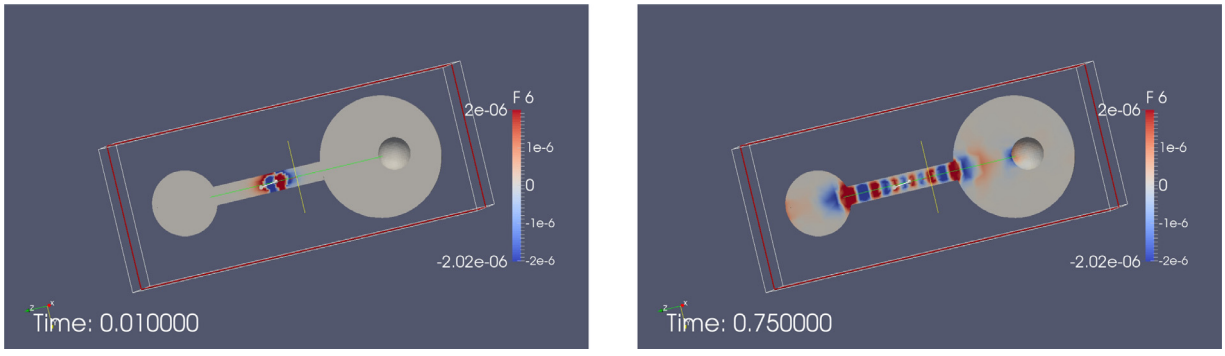

Fig. 17. The profiles of F_{23} at $t = 0.01$ (left) and at $t = 0.75$ (right).

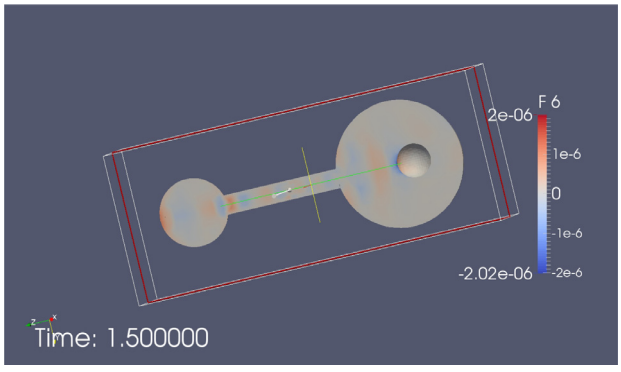

Fig. 18. The profile of F_{23} at $t = 1.5$.

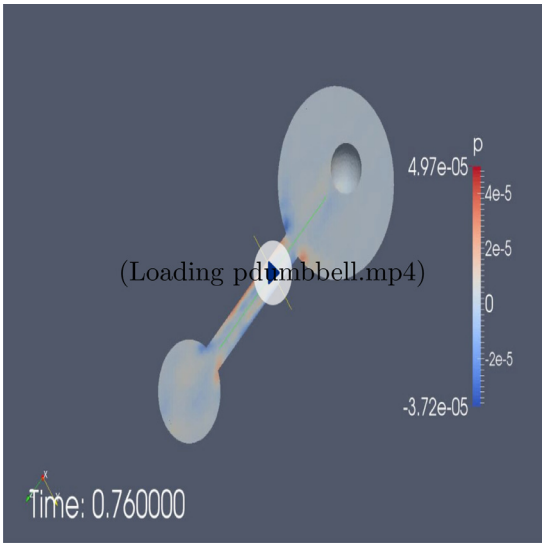

Fig. 19. An video animation of the dumbbell experiment.

The time step size is set to be $\Delta t = 0.0005s$. Snapshots for $\boldsymbol{u}$, $\boldsymbol{F}$ and p are given in Figs. 14–18. The mesh and a video animations available online can be found in Fig. 19, see also: https://www.dropbox.com/s/7ndtg2yyrhx3act/pdumbbell.ogv?dl=0.

From the numerical results, again, we observe that all numerical solutions decay with time. Such results reflect this energy decay property of the incompressible viscoelastic flow model.

5. Concluding remarks

Most of the PDE numerical analysts work on the Finite Element Method (FEM). There appears to be a shortage of numerical analysis work for the FVM. Nevertheless, many contemporary commercial computational mechanics software packages are based on FVM, including the open-source software OpenFOAM. Thus, stability analysis of the kind studied in our work here fills in some requisite knowledge for FVM in this regard, especially due to the fact that our numerical scheme is exactly the one being used in the OpenFOAM codes.

There are many other similar models in electromagnetism, heat conduction, elasticity, etc., whose FVM schemes' stability, error and convergence need to be analyzed, whether such schemes are based on OpenFOAM or other commercial packages. We hope that our work here can also provide some head-start for those models.

Acknowledgement

M. Cai's work is supported in part by the NIH BUILD grant (ASCEND pilot project) through UL1GM118973, the NSF HBCU-UP Research Initiation Award #HRD-1700328 and the NSF HBCU-UP Excellence in Research Award #DMS-1831950. Z. Luo's work is supported by the National Science Foundation of China grant #11671106. G. Chen's work is supported in part by Qatar National Research Fund grant NPRP #9-166-1-031. He also thanks Texas A&M University's High Performance Research Computing center's generous allocation of supercomputing hours.

Appendix

OpenFOAM codes

```
/
*---------------------------------------------------------------------------
------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2012 OpenFOAM
Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------
---------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify
it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but
WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public
License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Application
    newnseFoam

\*---------------------------------------------------------------------------
-------*/

#include "fvCFD.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"

    #include "createTime.H"
    #include "createMesh.H"
    #include "createFields.H"
    #include "initContinuityErrs.H"
```

```
Info<< "\nStarting time loop\n" << endl;

while (runTime.loop())
{
    Info<< "Time = " << runTime.timeName() << nl << endl;

    #include "readPISOControls.H"
    #include "CourantNo.H"

    fvVectorMatrix UEqn
    (
        fvm::ddt(U)
      + fvm::div(phi, U)
      - fvm::laplacian(nu, U)
      -fvc::div(F & F.T())
    );
    solve(fvm::ddt(F) + fvm::div(phi,F) - (fvc::grad(U)&F));
    solve(UEqn == -fvc::grad(p));
    for (int corr=0; corr<nCorr; corr++)
    {
        volScalarField rAU(1.0/UEqn.A());

        volVectorField HbyA("HbyA", U);
        HbyA = rAU*UEqn.H();
        surfaceScalarField phiHbyA
        (
            "phiHbyA",
```

```
        (fvc::interpolate(HbyA) & mesh.Sf())
      + fvc::ddtPhiCorr(rAU, U, phi)
    );

    adjustPhi(phiHbyA, U, p);

    for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
    {
        fvScalarMatrix pEqn
        (
            fvm::laplacian(rAU, p) == fvc::div(phiHbyA)
        );

        pEqn.setReference(pRefCell, pRefValue);
        pEqn.solve();

        if (nonOrth == nNonOrthCorr)
        {
            phi = phiHbyA - pEqn.flux();


                }
            }

            #include "continuityErrs.H"

            U = HbyA - rAU*fvc::grad(p);
            U.correctBoundaryConditions();
        }

        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }

    Info<< "End\n" << endl;

    return 0;
}


//
************************************************************************
*** //
```

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cnsns.2019.104876

References

[1] Baranger J, Sandri D. Finite element approximation of viscoelastic fluid flow: existence of approximate solutions and error bounds. Numerische Mathematik 1992;63:13–27.
[2] Bird RB, Armstrong RC, Hassager O. Dynamics of polymeric liquids. vol. 1: fluid mechanics. New York, NY: John Wiley and Sons Inc.; 1987.
[3] Chen G, Xiong Q, Morris PJ, Paterson EG, Sergeev A, Wang Y. OpenFOAM for computational fluid dynamics. Notices Am Math Soc 2014;61:354–63.
[4] Chen XY, Marschall H, Schäfer M, Bothe D. A comparison of stabilisation approaches for finite-volume simulation of viscoelastic fluid flow. Int J Comut Fluid Dyn 2013;27:229–50.
[5] Chen YM, Zhang P. The global existence of small solutions to the incompressible viscoelastic fluid system in 2 and 3 space dimensions. Commun Partial Diff Eq 2006;31:1793–810.
[6] Greenshields CJ. Openfoam user guide. OpenFOAM Foundation Ltd, version 2015;3.
[7] D Gennes P, Prost J. The physics of liquid crystals. publisher not identified; 1995.
[8] Jamil M, Fetecau C, Imran M. Unsteady helical flows of oldroyd-b fluids. Commun Nonlinear Sci Numer Simul 2011;16:1378–86.
[9] Larson RG. The structure and rheology of complex fluids. New York: Oxford University Press; 1999.
[10] Lei Z, Liu C, Zhou Y. Global solutions for incompressible viscoelastic fluids. Arch Ration Mech Anal 2008;188:371–98.
[11] Li C, Zheng L, Zhang Y, Ma L, Zhang X. Helical flows of a heated generalized oldroyd-b fluid subject to a time-dependent shear stress in porous medium. Commun Nonlinear Sci Numer Simul 2012;17:5026–41.
[12] Lin FH, Zhang P. On the initial-boundary value problem of the incompressible viscoelastic fluid system. Commun Pure Appl Math 2008;61:539–58.
[13] Lukáčová-Medvidová M, Mizerová H, She B. Error analysis of finite element and finite volume methods for some viscoelastic fluids. J Num Math 2016;24:105–23.
[14] Nazar M, Sultan Q, Athar M, Kamran M. Unsteady longitudinal flow of a generalized oldroyd-b fluid in cylindrical domains. Commun Nonlinear Sci Numer Simul 2011;16:2737–44.
[15] OpenFOAM user manual. http://OpenFOAM.com/.
[16] Sideris TC, Thomases B. Global existence for three-dimensional incompressible isotropic elastodynamics via the incompressible limit. Commun Pure Appl Math 2005;58:750–88.
[17] Versteeg HK, Malalasekera W. An introduction to computational fluid dynamics: the finite volume method. Pearson Education; 2007.
[18] Zhao W, Du D. Local well-posedness of lower regularity solutions for the incompressible viscoelastic fluid system. Sci China Math 2010;53:1521–30.