Journal of Non-Newtonian Fluid Mechanics 239 (2017) 85–104

Contents lists available at ScienceDirect

Journal of Non-Newtonian Fluid Mechanics

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

Stabilization of an open-source finite-volume solver for viscoelastic fluid flows

F. Pimenta, M.A. Alves*

CEFT, Departamento de Engenharia Química, Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:
Received 25 June 2016
Revised 27 October 2016
Accepted 4 December 2016
Available online 7 December 2016

Keywords:
OpenFOAM®
Finite-volume method
Viscoelastic fluid flow
Stabilization
4:1 contraction

ABSTRACT

In this work, we modify the viscoelastic solver available in the OpenFOAM® toolbox (Favero et al., 2010), in order to improve its stability for differential-type constitutive equations. The Oldroyd-B constitutive equation is solved using the log-conformation approach and the high-resolution schemes used to discretize the convective terms are handled with a component-wise and deferred correction approach. The pressure-velocity coupling is ensured using the SIMPLEC algorithm, and a new stress-velocity coupling term is also introduced. We demonstrate that the new solver is second-order accurate, both in space and time, by assessing the performance in problems with known analytical solution and using Richardson's extrapolation. The solver is further tested on the 4:1 planar contraction benchmark problem using an Oldroyd-B fluid ($\beta = 1/9$) at low Reynolds number flow conditions ($Re = 0.01$), considering a wide range of Deborah numbers, $0 \leq De \leq 12$. A good agreement with reference works is observed at low *De,* as well as with an in-house viscoelastic flow solver. At higher *De*, the vortex dynamics is essentially controlled by the singularity region in the re-entrant corner of the contraction, revealing a significant dependence of the numerical results on the mesh resolution. The corner vortex dynamics is also analyzed, from the flow startup at several *De*, providing new accurate data on the transient behavior of this problem. In summary, this work provides a robust open-source solver for viscoelastic flows, as well as new data on an old problem, which has still open questions and challenges.

1. Introduction

The flow of viscoelastic fluids has scientific and industrial relevance, and has been a matter of intensive research over the last decades. Currently, the available constitutive models arising from different theories can reasonably predict some exotic behaviors of viscoelastic fluids, such as the rod climbing or the die swell effects [1]. However, some experimental observations are still hardly predicted by the actual models and a number of open problems exist in this area of research [2,3]. Far from being only a field of academic interest, computational rheology is also important for the industry working with such complex fluids, since, for example, a significant waste reduction could be obtained if the elastic phenomena involved were better understood and predicted [3,4].

Due to the high complexity of the constitutive equations used to model viscoelastic fluids, numerical methods have been the workhorse to simulate real-life problems, among which finite-difference, finite-element and finite-volume methods should be emphasized. These three methods start from the same set of closed form partial differential equations, but their discretization, both in space and time, is handled differently in each approach, leading to different advantages/drawbacks regarding accuracy, stability, computational cost and versatility to handle different cases. While it is outside the scope of this work to present a comparison between the three numerical methods, it is worth to mention that the finite-volume method is especially well suited for general CFD (Computational Fluid Dynamics) problems due to its intrinsic conservative properties. An additional feature to be taken into account, mostly within the industrial context, is the availability of a software package that can be used with reliability to perform the numerical simulation. Although there are numerous options, both commercial and open-source, to handle Newtonian and also non-Newtonian inelastic fluids, the range of options is more limited for viscoelastic fluid flow simulations. Open-source packages are particularly attractive due to being cost-free for the user and usually allowing the customization of the available source code.

The OpenFOAM® toolbox is an open-source finite-volume solver, which can handle general unstructured polyhedral meshes and perform parallel computations. A general viscoelastic solver

* Corresponding author.
E-mail addresses: fpimenta@fe.up.pt (F. Pimenta), mmalves@fe.up.pt (M.A. Alves).

http://dx.doi.org/10.1016/j.jnnfm.2016.12.002

has already been developed in OpenFOAM® [5], which has been used in several works, including single-phase [6–9] and two-phase [10,11] flow studies. Habla et al. [12] modified this solver in order to strengthen the coupling between velocity and polymeric stresses, through a semi-implicit handling of the constitutive equation. The robustness of the solver was further increased with the implementation of the log-conformation approach [13], allowing to reach higher Weissenberg numbers (*Wi*) without loss of positive definiteness of the conformation tensor. Nevertheless, to the best of our knowledge, none of these recent solver improvements has been made available to the public.

In this work, we further extend the previous efforts [5,12,13] directed to increase the robustness and accuracy of the viscoelastic solver in OpenFOAM®. We implement the log-conformation methodology in a similar approach as described in Habla et al. [13] and in Afonso et al. [14], but further modifications are introduced that can increase the stability and also the computational speed. Those modifications are mainly related with the pressure-velocity and stress-velocity coupling, the discretization of convective terms and with the sparse matrix solver used to compute the solution of the linear system of equations.

The remainder of this paper is organized as follows: the governing equations and the numerical method are described in Section 2, along with the modifications introduced in the original solver of OpenFOAM®. In Section 3, we first assess the spatial and temporal order of convergence of the modified solver, after which the 4:1 planar contraction benchmark is addressed. The results for this benchmark problem are compared against reference works, as well as with an in-house solver that has been thoroughly validated in the past [14–17]. Finally, Section 4 presents the main conclusions of this work. In addition to describe and provide access to the source code of a robust open-source viscoelastic flow solver (available for download on GitHub: https://github.com/fppimenta/rheoTool), this work also presents new insights and accurate results in the 4:1 planar contraction benchmark flow problem.

2. Governing equations and numerical method

2.1. Overview of the original viscoelasticFluidFoam solver [5]

The OpenFOAM® viscoelastic solver originally developed by Favero et al. [5], named *viscoelasticFluidFoam*, can be found in the extend version of OpenFOAM®. Several constitutive models are available at runtime, including multi-mode modelling. Focusing on the Oldroyd-B model [18], which is used in the present work, the following set of equations is solved [5]:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \nabla\cdot\left[(\eta_s+\eta_p)\nabla\mathbf{u}\right] = -\nabla p - \nabla\cdot\left(\eta_p\nabla\mathbf{u}\right) + \nabla\cdot\boldsymbol{\tau} + \mathbf{f} \tag{2}$$

$$\boldsymbol{\tau} + \lambda\left(\frac{\partial \boldsymbol{\tau}}{\partial t} + \mathbf{u}\cdot\nabla\boldsymbol{\tau} - \boldsymbol{\tau}\cdot\nabla\mathbf{u} - \nabla\mathbf{u}^{\mathrm{T}}\cdot\boldsymbol{\tau}\right) = \eta_p\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right) \tag{3}$$

where $\mathbf{u}$ is the velocity vector, t is the time, p is the pressure, $\boldsymbol{\tau}$ is the polymeric contribution to the extra-stress tensor, $\mathbf{f}$ represents any external forcing per unit volume, ρ is the fluid density, η_s is the solvent viscosity, η_p is the polymeric viscosity and λ is the relaxation time.

The continuity equation is represented by Eq. (1), while Eq. (2) describes the momentum balance. The extra-stress tensor ($\boldsymbol{\tau}$') was split in polymeric ($\boldsymbol{\tau}$) and solvent ($\boldsymbol{\tau}_s$) contributions, such that $\boldsymbol{\tau}' = \boldsymbol{\tau} + \boldsymbol{\tau}_s$, with $\boldsymbol{\tau}_s = \eta_s(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}})$. After mathematical manipulation, the divergence of $\boldsymbol{\tau}_s$ appears in Eq. (2) as a diffusive term ($\nabla\cdot\eta_s\nabla\mathbf{u}$) to be discretized implicitly, while the divergence of $\boldsymbol{\tau}$ is discretized explicitly. Furthermore, a stabilizing diffusive term ($\nabla\cdot\eta_p\nabla\mathbf{u}$) is added to both sides of Eq. (2) [19], one discretized implicitly (lhs) and the other one explicitly (rhs), which in steady-state exactly cancel each other – a technique known as both-sides diffusion (BSD).

The constitutive equation for the polymeric component of the extra-stress tensor is described by Eq. (3). For the Oldroyd-B constitutive model, a solvent viscosity ratio can be defined, $\beta = \eta_s/(\eta_p+\eta_s) = \eta_s/\eta_0$, also known as retardation ratio. When $\beta = 0$, the Upper-Convected Maxwell (UCM) model is recovered.

In the original *viscoelasticFluidFoam* solver [5], the governing equations are solved sequentially in a segregated manner, where the momentum equation (Eq. 2) is solved first, followed by the continuity (pressure) equation (Eq. 1) and finally the constitutive equation (Eq. 3). The pressure-velocity coupling is ensured using the PISO (Pressure-Implicit Split Operator) algorithm [20].

In the following sections, we describe the main modifications introduced in the original *viscoelasticFluidFoam* solver [5], which can increase its stability and computational speed, while keeping second-order accuracy, both in time and space.

2.2. The log-conformation approach

The numerical difficulties arising at high Weissenberg number flows are well known in the literature and they were commonly associated with a loss of resolution of discretization methods to solve the exponential growth of stresses at critical points [21]. A common indicator of such undesirable situation is the loss of positive definiteness of the conformation tensor [21], which can ultimately lead to numerical divergence. Several solutions were proposed to remedy this problem and most of them consist in a change of variable in the constitutive equation [21–24]. The log-conformation approach proposed by Fattal and Kupferman [21,24] became such a popular methodology, based on the reformulation of the constitutive equation in terms of the logarithm of the conformation tensor, which naturally keeps the conformation tensor positive definite and linearizes the stress field in regions of exponential growth, leading to enhanced numerical stability [24].

Habla et al. [13] were among the first to document the implementation of the log-conformation approach in OpenFOAM®, as aforementioned. In the present work, we follow essentially this implementation, which is briefly described next. For details on the mathematics behind each step, we refer the reader to the seminal works of Fattal and Kupferman [21,24].

The relation between the conformation tensor ($\mathbf{A}$) and the polymeric extra-stress tensor for the Oldroyd-B model is given by

$$\boldsymbol{\tau} = \frac{\eta_p}{\lambda}(\mathbf{A} - \mathbf{I}) \tag{4}$$

Since $\mathbf{A}$ is positive definite, it can be diagonalized in the form

$$\mathbf{A} = \mathbf{R}\,\boldsymbol{\Lambda}\,\mathbf{R}^{\mathrm{T}} \tag{5}$$

where the columns of the orthogonal tensor $\mathbf{R}$ are the eigenvectors of $\mathbf{A}$ and the diagonal matrix $\boldsymbol{\Lambda}$ contains the corresponding eigenvalues.

Instead of solving the constitutive equation in $\mathbf{A}$, Fattal and Kupferman [21,24] reformulated Eq. (3) in terms of the natural logarithm of $\mathbf{A}$,

$$\boldsymbol{\Theta} = \ln(\mathbf{A}) = \mathbf{R}\ln(\boldsymbol{\Lambda})\mathbf{R}^{\mathrm{T}} \tag{6}$$

leading to the following evolution equation for $\boldsymbol{\Theta}$

$$\frac{\partial \boldsymbol{\Theta}}{\partial t} + \mathbf{u}\cdot\nabla\boldsymbol{\Theta} - (\boldsymbol{\Omega}\boldsymbol{\Theta} - \boldsymbol{\Theta}\boldsymbol{\Omega}) - 2\mathbf{B} = \mathbf{R}\left[\frac{1}{\lambda}\left(\boldsymbol{\Lambda}^{-1} - \mathbf{I}\right)\right]\mathbf{R}^{\mathrm{T}} \tag{7}$$