Viscoelastic Fluid Simulation with OpenFOAM®

Marc Arnau Roca and Alejandro Fernández Alburquerque
Universitat Politècnica de Catalunya. Enginyeria Física.

(Advisor: Laureano Ramírez de la Piscina)

(Dated: June 18, 2020)

Since non-Newtonian fluids were discovered, their numerous interesting behaviours and potential uses have been investigated. Particularly, the frequency-dependent properties of wormlike micellar solutions have risen a good deal of interest among the scientific community. In this report, we have attempted to simulate those properties using the Giesekus model for viscoelastic fluids and, more precisely, we have tried to obtain a velocity profile with resonant behaviour inside a cylinder cavity. The computational environment chosen to perform the simulations is OpenFOAM, an open-source CFD program with great flexibility and capabilities. The results confirm a resonant response to oscillating boundary conditions, although perfect stationary-like velocity profiles were not obtained.

I. INTRODUCTION

Fluid dynamics and rheology have become two of the main areas of research of modern physics due to their high complexity and wide array of possible applications. Moreover, current computers are capable of performing high-demanding simulations, what has encouraged the emergence of CFD (Computational Fluid Dynamics) software. In this context, OpenFOAM has differentiated itself as an open source alternative with a highly-modifiable environment that allows the user to adapt the program to its needs.

The first goal of our project was to recreate the experiment performed by Casanellas Vilageliu [1] with OpenFOAM, in order to observe how viscoelastic fluids have a different behaviour than Newtonian ones when specific conditions are imposed. The set-up of the experiment was formed by a closed cylindrical pipe with two covers with freedom of movement. One of them was attached to a mechanical system which was able to provide an oscillatory motion (see Fig. 1). When this cavity was filled with fluid and the oscillatory movement was enabled, both covers moved jointly. In that experiment, the velocity profile of a non-Newtonian viscoelastic fluid inside the pipe became standing-wave shaped at certain frequencies, what differs from the Poiseuille profile of a Newtonian fluid like water. The simulations done in this project are aimed to find this particular behaviour.

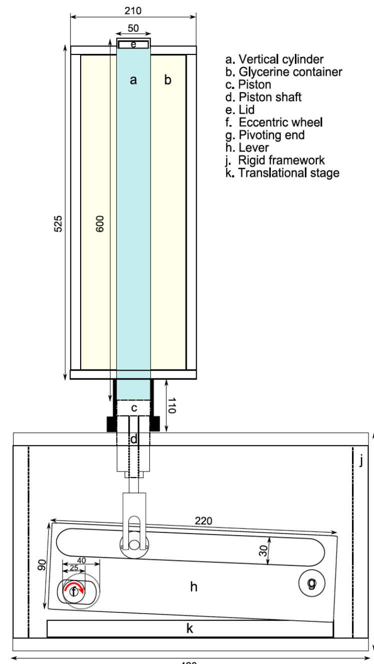


Figure 1: Front view of the experimental set-up (image reproduced from [1]). The dimensions are in millimetres.

II. SOFTWARE INSTALLATION

OpenFOAM is an open source project and, as such, it intends to be accessible for as many people as possible —therefore, its installation process is not specifically tailored for any operating system (OS)—. This means that the end user has to invest more time and effort into the installation process. Additionally, ParaView® (the visualization tool used by OpenFOAM) and rheoTool™ (a third-party toolbox that adds viscoelastic fluid flows solvers) have their own installation requirements.

The first decision we had to make was on which OS the software was going to be installed. The OS we chose was Ubuntu® (versions used were 18.04 and 19.10) running on a virtual machine (VirtualBox® and Parallels®), as it is one of the most popular distributions of Linux-based operating systems and has a well established community. Furthermore, there is an official installation package prepared for this OS with a pre-compiled version of OpenFOAM and ParaView [2], which eases the installation process considerably.

The only downside of the aforementioned package is that it doesn't include the latest version of ParaView and it is not immediately updated when a new Ubuntu version is released. For that reason, we also attempted to install OpenFOAM and ParaView from source code on the latest release of Ubuntu (20.04), which did not have a pre-compiled package available at that time. This attempt was unsuccessful due to an incompatibility of ParaView with the OpenGL® version used in virtual machines. The reason behind is that the current development state of virtualization software doesn't allow to take full advantage of graphics hardware. Another alternative was to install OpenFOAM using a Docker® environment (a virtualization tool that allows software compiled in Linux to be run in other platforms), but since we were already habituated to the aforementioned virtual machines, we discarded the idea for simplicity.

Once OpenFOAM v7 was running correctly, the last step was to install rheoTool. This time the only installation possible was from source code (the toolbox only provides a step-by-step guide for Debian-based systems [3]). After some unsuccessful attempts and the corresponding troubleshooting (the installPetsc installation file had to be modified to download the latest release of PETSc™ instead of a legacy one), we were finally able to correctly install it.

III. THEORETICAL OVERVIEW

The non-Newtonian fluid chosen consists of a solvent and a polymer (solute). This polymer has hydrophobic and hydrophylic parts which will lead to the formation of micelles. The complex fluids that fulfill those conditions are called wormlike micellar solutions. The constitutive model used to approach their behaviour is the Giesekus one, a variation of Oldroyd-B [4]. Whereas it is not the objective of this study to deepen in the theoretical basis of the Giesekus model, it is convenient to review its origins and equations.

The Oldroyd-B model is an extension of the Maxwell model for viscoelasticity which includes two different viscosities, one for the solute, typically a polymer, and the other for the solvent. However, this model was not accurate enough since it failed to predict nonlinear features observed in wormlike micellar solutions. The addition of a parameter α which regulates the first quadratic term in shear stress allows to recreate a more exact drag condition on the polymer molecules. The equations of the Giesekus model are:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_p \tag{1a}$$

$$\boldsymbol{\tau}_s = \eta_s \boldsymbol{\gamma}_{(1)} \tag{1b}$$

$$\boldsymbol{\tau}_p + \lambda \boldsymbol{\tau}_{p(1)} - \alpha \frac{\lambda}{\eta_p} \{\boldsymbol{\tau}_p \cdot \boldsymbol{\tau}_p\} = \eta_p \boldsymbol{\gamma}_{(1)} \tag{1c}$$

The term $\boldsymbol{\tau}$ refers to the stress tensor, which is composed of a solvent and a polymeric contribution with an associated viscosity (η_s and η_p, respectively). The rate-of-strain tensor is represented by γ, while λ is the relaxation time of the fluid, the time needed by the polymeric solution to re-assemble after a breakage.

IV. SIMULATION RESULTS AND DISCUSSION

In order to recreate the set-up conditions as similar to those of Fig. 1 as possible, we designed a cylindrical mesh —characterized in Table I— using the mesh generation functionality of OpenFOAM.

Table I: Parameters of the cylinder. XY are the transversal directions and Z is the longitudinal one.

Radius (Cylinder)	Edge (Inner Box)	Length	No. Voxels (XY)	No. Voxels (Z)
0.025 m	0.016 m	0.6 m	22	20

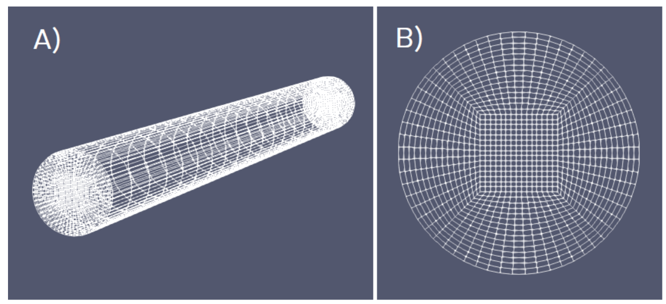


Figure 2: A: Generated mesh 3D overview. B: Section of the cylindrical mesh.

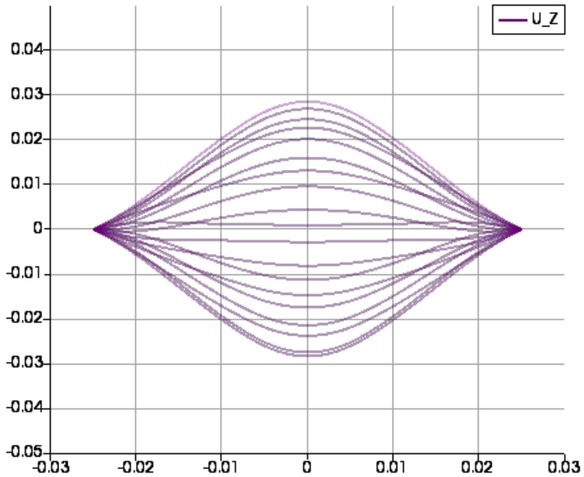
The value of the Giesekus model parameters employed are displayed in Table II. They were calculated experimentally by Casanellas Vilageliu to correctly characterize the wormlike solution used.

Table II: Parameters of the micellar solution.

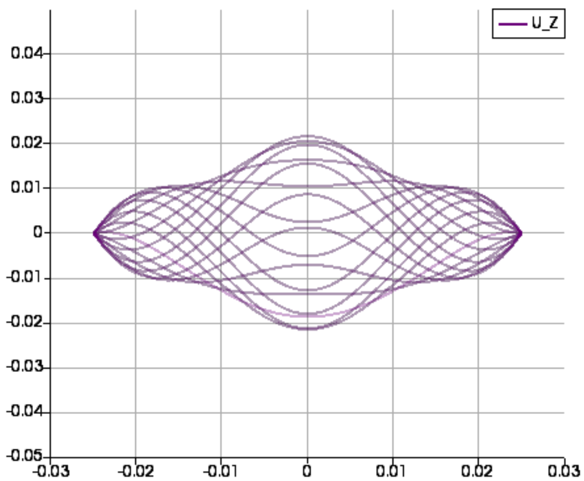
$\rho\ (kg/m^3)$	$\lambda\ (s)$	α	$\eta_s\ (kg/m \cdot s)$	$\eta_p\ (kg/m \cdot s)$
1050	1.9	0.85	0	64

The oscillating covers were recreated by applying a sine function to the inlet/outlet velocity boundary conditions of the top and bottom mesh faces. Therefore, the fluid velocity at both ends of the cylinder had the same time-varying module, direction and sense, emulating the jointly oscillating motion of the experimental set-up. The frequencies and amplitudes chosen for the sinusoidal boundary conditions of the simulations were the ones that showed resonant behaviour in the experiments [1].

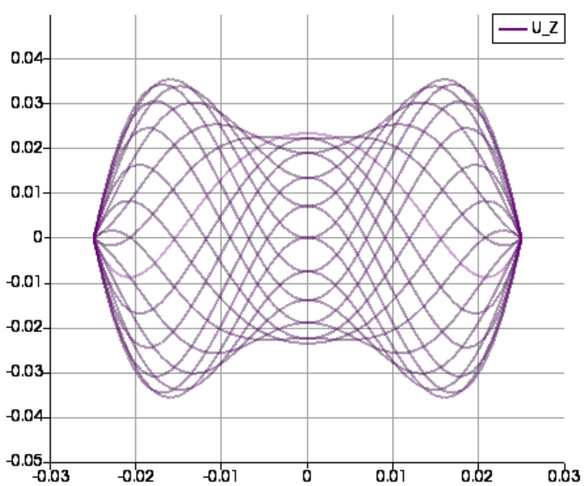
The rheoFoam solver module of rheoTool was used to simulate the Giesekus fluid dynamics with two different approaches. The results presented in this report were obtained with a simulation time of 3.5 seconds and a time step of 10^{-4} seconds (these numbers were tuned empirically until the expected resonant behaviour was obtained with enough accuracy). The second approach was aimed to obtain exactly 8 full periods of simulation time while having the same amount of time steps as before (by adjusting the time step for each frequency). As no noticeable difference between the results of both methods was observed, only the first approach has been included here.



(a) $f = 3.50$ Hz, $z_o = 0.5$ mm

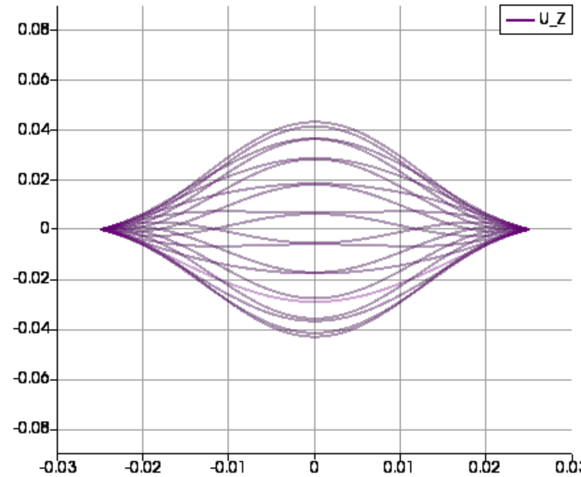


(b) $f = 5.90$ Hz, $z_o = 0.2$ mm

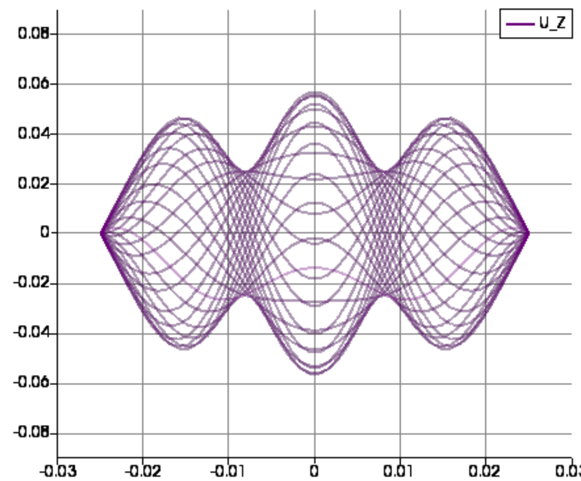


(c) $f = 7.49$ Hz, $z_o = 0.5$ mm

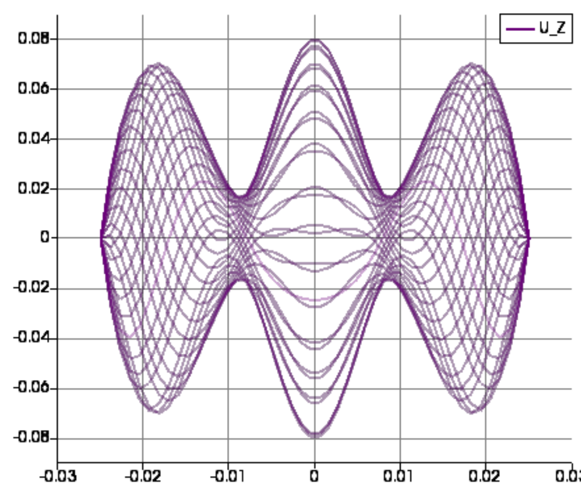
Figure 3: Velocity profile of a Giesekus fluid flow for the first 3 resonant frequencies.



(a) $f = 2.26$ Hz, $z_o = 1$ mm



(b) $f = 3.69$ Hz, $z_o = 1$ mm



(c) $f = 6.44$ Hz, $z_o = 1$ mm

Figure 4: Velocity profile of a diluted Giesekus fluid flow for its first 3 resonant frequencies.

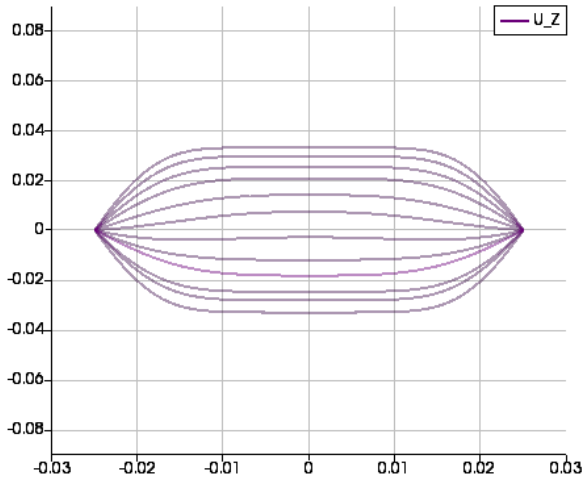


Figure 5: Velocity profile of a Newtonian fluid flow for $f = 3.69$ Hz and $z_o = 1$ mm

The results obtained can be seen in Fig. 3 (only the last period of the simulations is displayed). In Fig. 3a, the velocity profile behaves more or less as the one of a Newtonian fluid. In Fig. 3b and 3c, the velocity in the Z direction has a different sense depending on the radial coordinate, thus verifying the elastic behaviour nature of the fluid. Although the results were really interesting, a clear stationary-like velocity profile inside the pipe was not found. According to Giesekus (the author of the article in which the model that we are using was developed) [5], the model should work fine for *"moderately concentrated solutions"* of polymer. Since in Ref [1] a more dilute solution was characterized too, we performed another simulation to see if a clearer stationary velocity profile could be obtained. The new parameters used are shown in Table III and the results of this simulation are found in Fig. 4.

Table III: Parameters of the dilute micellar solution, obtained experimentally.

$\rho\ (kg/m^3)$	$\lambda\ (s)$	α	$\eta_s\ (kg/m \cdot s)$	$\eta_p\ (kg/m \cdot s)$
1050	0.3	0.85	0	3

For the lowest frequency (Fig. 4a), a Newtonian-like velocity profile is observed again, although the elastic response is already evident. However, in Fig. 4b the results are more interesting than the ones obtained for the non-dilute solution, since an almost stationary velocity profile was attained. It can be clearly seen that, when the center has a positive velocity sense, the peripheral sections have a negative sense and vice versa, depending on time. Even though the velocity profile of Fig. 4c is remarkable too, the actual dynamical evolution of it [6] does not have a stationary behaviour as well defined as the former case.

To further understand the importance of the result obtained in Fig. 4b, a Newtonian fluid resembling water was simulated with the same frequency and amplitude conditions. As it is seen in Fig. 5, the behaviour differs completely from the result of the non-Newtonian fluid.

The cross section velocity profiles inside the cylinder were also obtained when performing the simulations. The mesh generated had an inner rectangular prism as it can be seen in Fig. 2B, so this led to a slight symmetry breaking (spikes) along the edges of such prism (Fig. 6A). Another cylindrical mesh was designed with a true cylindrical symmetry (Fig. 6B), however, when the simulations were performed, the much smaller size of the inner center cells caused a numerical discontinuity and instability. Consequently, the algorithms of the solvers crashed and were not able to finish the simulations.

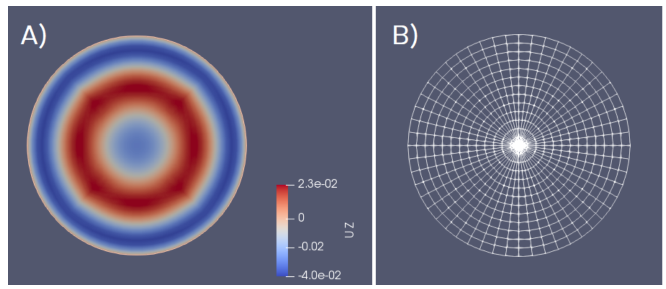


Figure 6: A: Cross section of the dilute solution at $f = 6.44$ Hz, $z_o = 1$ mm. B: Section of the alternate cylindrical mesh that did not work.

V. CONCLUSIONS

The outcome of this project is quite satisfactory, as the simulations performed for a dilute solution show a very similar pattern compared to the experimental results of Casanellas Vilageliu. Furthermore, the comparison with the behaviour of a Newtonian fluid helps to better understand the frequency-related properties of Giesekus fluids.

Even so, we expected to obtain better-defined stationary waves corresponding to a truly resonating phenomenon. The reason behind such results might be threefold. On one hand, the cylindrical mesh used in our simulations does not have azimuthal symmetry in the center, which may cause enough distortion to prevent the visualization of perfect standing waves. On the other hand, the parameters used to characterize a Giesekus fluid are experimental and prone to errors. And, finally, the equations used by rheoTool to model Giesekus fluids might not be fully accurate, though this topic has not been thoroughly analyzed.

An extra amount of work has been put into sharing all the progress achieved to facilitate further research around this subject. To that end, a GitHub® repository has been created where complementary material such as all OpenFOAM code files used can be found [6], as well as additional graphs and animations of all the cases mentioned along this report.

[1] Laura Casanellas Vilageliu. Oscillatory pipe flow of wormlike micellar solutions. *PhD Thesis, Universitat de Barcelona*, 2013.

[2] The OpenFOAM Foundation. OpenFOAM v7: Patch Releases. `https://openfoam.org/news/v7-patch/` [Accessed: May 2020].

[3] Francisco Pimenta. Toolbox to simulate GNF and viscoelastic fluid flows in OpenFOAM. `https://github.com/fppimenta/rheoTool` [Accessed: May 2020].

[4] Alexander N Morozov and Wim van Saarloos. An introductory essay on subcritical instabilities and the transition to turbulence in visco-elastic parallel shear flows. *Physics Reports*, 447(3-6):112–143, 2007.

[5] H. Giesekus. A simple constitutive equations for polymer fluids based on the concept of deformation-dependent tensorial mobility. *Journal of Non-Newtonian Fluid Mechanics*, 11:69–109, 1982.

[6] Alejandro Fernández and Marc Arnau. Viscoelastic (Giesekus) and Newtonian fluid simulation with OpenFOAM and rheoTool. Resolution for a cylindrical mesh with inlet/outlet boundary conditions and no gravity. 2020. `https://github.com/Alejandro-FA/pipeFlow`.