VISCOELASTIC FLUID SIMULATION WITH OPENFOAM

Revised 30/5/2019

Miquel Negre and Pau López
Universitat Politècnica de Catalunya

Abstract: This report examines the solution of a problem concerning the dynamics of a viscoelastic fluid. For this purpose we use the OpenFoam software in conjunction to the Rheotool extension. The experiment, in which the fluid is contained in a cylinder with oscillating caps, showed a behaviour in agreement with the theoretical expectations. Finally, we conduct a convergence test to certify the validity of the discretization in the simulation.

Keywords: Physics, viscoelastic, Giesekus, OpenFoam

I. INTRODUCTION

Complex fluids exhibit non-newtonian rheological properties. Their study is interesting for many areas of industry research. The flow of sun-cream, ketchup sauce or concrete are governed by non-newtonian effects. It is also the case for biological fluids such as saliva and blood which are interesting from the perspective of biophysics. Another example of successful application is in the design of more efficient bullet-proof vests. The fluid model we deal with is of a viscoelastic fluid, a complex fluid which shows elastic as well as viscose properties.

II. OBJECTIVES

The main purpose of our project was to familiarize with the intricate complex fluid simulator OpenFOAM and the extension RheoTool, which is an open-source toolbox necessary to deal with viscoelastic fluids. In order to do so, we have simulated an actual experiment to study the properties that arise from a non-Newtonian behaviour. Since we aim to study a viscoelastic fluid, the viscosity will not be constant but dependent on the time scale. Our experiment analyzes the response of the fluid reacting to the harmonic movement of two parallel infinite cylindrical plates.

We run the simulation with the two first resonant frequencies of our model, which are obtained theoretically using next formula, and a frequency in between (22Hz). Parameters used in simulations are based on an actual experiment broadly explained in a Phd thesis cited in references **[1]**.

$$\omega_0 = \frac{\eta_P \pi^2 (2n+1)^2}{2a^2\rho}\left(1+\frac{\eta_P \pi^2 (2n+1)^2 \lambda}{a^2\rho}\right)^{-\frac{1}{2}}$$

Parameters used:

$$\eta_S = 0\,\frac{kg}{m\cdot s} \quad ; \quad \eta_P = 64\frac{kg}{m\cdot s}$$

$$\rho = 1050\frac{kg}{m^3} \;\; ; \; \lambda = 1.9\,s \;\; ; \; a = 0.025\,m$$

Where λ is the single relaxation time. With ηp being the rate independent viscosity and ηs the constant viscosity. We can also note that ρ corresponds to fluid's density and a to the cylinder's radius. For *n=0* we obtain the first harmonic ($\omega \simeq 11.25\,Hz$) and for *n=1* the second one ($\omega \simeq 33.76\,Hz$). The third angular frequency studied will be between harmonics ($\omega = 22\ Hz$).

III. MODEL

Our work is based on a theoretical model which was named after an Hungarian physicist: the Giesekus model. **[2]**

$$\tau = \tau_S + \tau_P$$

$$\tau_S = \eta_s \gamma(1)$$

$$\tau_S + \lambda\tau_{P(1)} - \frac{\alpha\lambda}{\eta_P}\cdot\{\tau_P\cdot\tau_P\} = \eta_P\gamma(1)$$

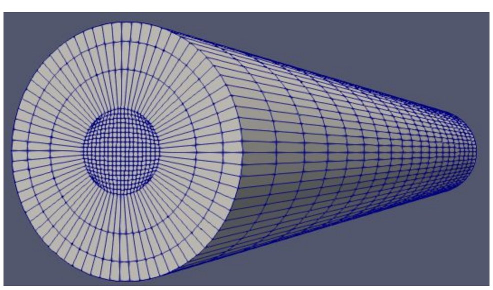
Were τ is the stress tensor and α a non-linear term coefficient, that we'll fix to 0.85 [**1**]. These constitutive equations are solved alongside with the Navier-Stokes equations, which are the pillars of fluid dynamics. OpenFoam, in conjunction with a Rheotool solver specialised in viscoelastic fluids *rheoFoam*, allowed us to solve the model and simulate our experiment without having to deal with the mathematical intricacies. We had to properly set the boundary conditions and the geometry of the problem so that OpenFoam yielded the results. Equations solved by our model are not exactly the ones exposed above but a logarithmic approximation (*GieskusLog* constitutive equations of Rheotool). The different simulations have been carried out on an Ubuntu virtual machine (VirtualBox).

IV. SETUP

The simulation consists in the fluid response to the sinusoidal movement of two cylindrical plates. The geometry, boundary conditions and model parameters employed are detailed henceforth.

Geometry:

We generate a quarter of a cylinder of 30 cm in length and 2.5 cm in radius and do so by using the *blockMesh* function of OpenFoam over a *blockMeshDict* where this geometry is conveniently defined. To generate the full cylinder a *mirrorMesh* applet, that as its name indicates generates a mirrored grid in the chosen direction, is employed for both *x* and *y* axis (cylinder's axial direction corresponds to *z* coordinate). The circular sides of the cylinder will be the oscillating feature of the problem.



With the purpose of implementing the model, we need to discretize the geometry. We will work with two different meshes in order to assure that the finesse of our grid does not interfere with the final result. However, the election of the divisions is by no means trivial and has to be done in accordance to the convergence analysis that we later expose. Setting them too low might we insufficient to account for the complexity of the internal state of the fluid during the simulation, with this oversimplification potentially leading to misleading results. On the other hand, using excessive divisions comes at the price of high computational time-scales, something we have experienced first-hand by running everlasting simulations that had to be eventually dismissed. It is important to notice that, as can be observed in last figure, mesh discretization is not equidistant but more dense in cylinder's center. This is not casual, because that is the zone where model can be reproduced with more accurately (less affected by the artificial conditions imposed in plates). In order to simplify the mesh study only radial divisions will be varied, that is the direction that we are really interested in. Angular and axial divisions will be kept constant with values of 18 and 20 respectively.

Boundary conditions

Experiment requires of an harmonic movement of plates (assumed as cosinusoidal) with amplitude A (that in order to avoid non-linearities will remain on a low value ~0.0001 m). We can derive the initial velocity of the fluid as the derivative of first expression. It can be trivially seen that the velocity will have a cyclic behaviour. To reproduce this on simulations, we have imposed an harmonic change of the fluid's velocity (U) in the axis direction in the inlet as well as in the outlet of the cylinder using OpenFoam's function *uniformFixedValue* type *sine*.

$$x(t) = -A\cos(\omega t)$$

$$U(t) = A\,\omega\,\sin(\omega t)$$

Elsewhere, that is, anywhere in the cylinder with the exception of the circular caps, we set the velocity to be zero at the initial time.

V. CONVERGENCE ANALYSIS

Time convergence:

Time step (*k)* plays an important role when determining the convergence of a simulation. We define *k* as the lapse of time that OpenFoam's lets pass between two resolutions of the model. As happened previously with problem's mesh huge time-steps guide to misleading solutions while tiny ones require of more computational power of what we dispose. The first step taken to ensure that a model works as expected is to verify that for a simulation time long enough it remains on a steady-state, even for the less-demanding conditions (in our case this will imply the largest *k* possible and the simplest mesh (with 12 radial divisions)). Simulations for the maximum time-step (*k*) recommended by OpenFoam determined that for all three studied frequencies steady-state is reached between 2 and 2.5 s. In order to ascertain whether a simulation converges or not we choose a point of problem's grid and analyze its velocity over time. The chosen point is (0,0,0.15) which corresponds to cylinder's axial center. In first graph on next page, can be observed how for two different *k* (concretely for *k=0.0005* and *k=0.00025*, chosen after multiple unsatisfactory convergence studies with larger values of time-step) each frequency converges on a steady-velocity. This gathering is not perfect there exist a certain discrepancy between different *k*'s steady-velocities in fact, as ω increases this difference also spreads. However, this difference will be considered negligible so the study of velocities behaviour can be continued with our actual tools. As a matter of fact, results obtained match our work's initial expectancies with a maximum absolute divergence of 0.0005 m/s for 2nd harmonic. As a consequence of these results *k=0.0005* will be used in our following

Mesh convergence:

Proceedings in the study of mesh convergence are the very same of the ones done for time convergence, with the exception that instead of evaluating time-steps we will be interested in the solutions found for different grids. Unfortunately, unlike previous case, no such a good convergence has been found within our computational/time-processing capacity range. Results found for grids with 12 and 20 radial divisions, finest grid computed, are exposed in next page's second plot in order to prove how geometries resolution conditions our final result. Even though those imperfections and assumptions final velocity profiles results achieved are satisfying.

Velocity profile analysis:

Velocity profiles for the different angular frequencies studied are shown at the end of page 4. These representations are done for a 100 point division between points (0,-0.025,0.15) and (0,0.025,0.15) with a *k=0.0005* and a mesh with 20 radial divisons. Representation of both resonant frequencies match theoretical results correctly, while the 22Hz figure displays a transient state between both resonances.

VI. CONCLUSIONS

We want to stress the complications that creating this simulation entails. Not only did we have to install all the dependencies, which constituted quite a conundrum per se, but we also had to input the model details, set the simulation parameters and extract the resulting information. Finally, we fulfilled our expectations of conducting simulations for different meshes, time-steps and frequencies. Then had to entrust the pertinence of the results to the convergence tests, which resulted satisfactorily. Besides, we observed correspondence with the theoretical results when resonant frequencies were used.

ACKNOWLEDGEMENTS

To Mr. Laureano Ramírez de la Piscina for his valuable remarks during our project.

REFERENCES

[1] Laura Casanellas. "*Oscillatory pipe flow of wormlike micellar solutions"*, PhD Thesis, Physics Dept., UB.
[2] Alexander Morozov and Saverio E. Spagnolie. Book: "*Introduction to Complex Fluids*"
[3] "*OpenFoam user guide version 6*" (10-06-2018)..
[4] "*Rheotool user guide*" (4-04-2019).
[5] Jordi Casacuberta, "*Laminar through a circular pipe*"

Graphical Results:

All the data of the plots shown in this section has been obtained with OpenFoam's function *singleGraph*. This function generates a *xy* file for a specific number of points specified in *singleGraph* file (located inside the system folder). Once acquired the data the *xy* files are converted to .xlsx archives. Then, thanks *xlsxread* function data is imported to Matlab in order to be represented afterwards. The fact that the data we were really interested in (Uz) could not be represented directly from Ubuntu's *gnuplot* nor from *ParaView* hindered our task and reduced our initial plotting expectancies. However, the method found, although time-consuming and tedious, proved to work properly.