Code Verification of Non-Newtonian Fluid Solvers for Single- and Two-Phase Laminar Flows

Lovato, S.L.; Toxopeus, Serge Leon; Settels, Just W. ; Keetels, G.H.; Vaz, Guilherme

DOI
10.1115/1.4050131

Publication date
2021

Document Version
Final published version

Published in
Journal of Verification, Validation and Uncertainty Quantification

Citation (APA)
Lovato, S. L., Toxopeus, S. L., Settels, J. W., Keetels, G. H., & Vaz, G. (2021). Code Verification of Non-Newtonian Fluid Solvers for Single- and Two-Phase Laminar Flows. *Journal of Verification, Validation and Uncertainty Quantification*, *6*(2), Article 021002. https://doi.org/10.1115/1.4050131

Stefano Lovato
Department of Maritime and Transportation Technology,
Offshore and Dredging, Delft University of Technology,
P.O. Box 5,
Delft 2600AA, The Netherlands
e-mail: s.l.lovato@tudelft.nl

Serge L. Toxopeus
MARIN,
P.O. Box 28,
Wageningen 6700AA, The Netherlands
e-mail: s.l.toxopeus@marin.nl

Just W. Settels
MARIN,
P.O. Box 28,
Wageningen 6700AA, The Netherlands
e-mail: j.settels@marin.nl

Geert H. Keetels
Department of Maritime and Transportation Technology,
Offshore and Dredging, Delft University of Technology,
P.O. Box 5,
Delft 2600AA, The Netherlands
e-mail: g.h.keetels@tudelft.nl

Guilherme Vaz
WavEC—Offshore Renewables,
Lisbon 1350-352, Portugal
e-mail: guilherme.vaz@wavec.org

Code Verification of Non-Newtonian Fluid Solvers for Single- and Two-Phase Laminar Flows

The presence of complex fluids in nature and industrial applications combined with the rapid growth of computer power over the past decades has led to an increasing number of numerical studies of non-Newtonian flows. In most cases, non-Newtonian models can be implemented in existing Newtonian solvers by relatively simple modifications of the viscosity. However, due to the scarcity of analytical solutions for non-Newtonian fluid flows and the widespread use of regularization methods, performing rigorous code verification is a challenging task. The method of manufactured solutions (MMS) is a powerful tool to generate analytical solutions for code verification. In this article, we present and discuss the results of three verification exercises based on MMS: (i) steady single-phase flow; (ii) unsteady two-phase flow with a smooth interface; (iii) unsteady two-phase flow with a free surface. The first and second exercises showed that rigorous verification of non-Newtonian fluid solvers is possible both on single- and two-phase flows. The third exercise revealed that "spurious velocities" typical of free-surface calculations with the Volume-of-Fluid model lead to "spurious viscosities" in the non-Newtonian fluid. The procedure is illustrated herein on a second-order finite volume flow solver, using the regularized Herschel-Bulkley fluid model as an example. The same methodology is however applicable to any flow solver and to all the rheological models falling under the class of generalized Newtonian fluid models. [DOI: 10.1115/1.4050131]

1 Introduction

Non-Newtonian fluids are frequently encountered in nature and in industrial applications. Examples of non-Newtonian fluids are blood in capillaries, volcanic lava, paints, cosmetics, drilling muds [1]. The ability of numerical methods to study complex flows that cannot be modeled analytically and the rapid growth of computational power over the past decades have led to an increasing number of numerical studies of non-Newtonian fluid flows. Evidently, carrying out such numerical studies requires computational fluid dynamics (CFD) software equipped with non-Newtonian fluid models. These models can be rather complex, sometimes requiring solving additional partial differential equations. However, a broad class of non-Newtonian models called generalized Newtonian fluid (GNF) models can be implemented in existing Newtonian fluid solvers by simply modifying the viscosity of the fluid (examples of GNF models are given, e.g., in Refs. [1] and [2]). The governing equations for these non-Newtonian fluids can thus be solved using the same solution methods adopted for Newtonian fluids. This article focuses on these types of non-Newtonian models.

It is important that the implementation of new features is followed by code verification [3,4] to ensure that the code is free of mistakes and numerical algorithm deficiencies. The most rigorous code verification exercise is the order-of-accuracy test, which consists of demonstrating that the rate of convergence of the discretization error tends to the theoretical/expected order of accuracy with grid/time-step refinement.

However, due to the scarcity of analytical solutions for non-Newtonian fluid flow, performing code verification may be a challenging task. While some analytical solutions, such as Poiseuille flow (see e.g., Refs. [5] and [6]), are actually available, these solutions are often rather simple and thus they exercise only few terms of the governing equations. Furthermore, non-Newtonian models are sometimes modified by regularization methods [7], making analytical solutions inadequate for code verification purposes.

The possibility to generate analytical solutions that can be used for code verification of non-Newtonian fluid solvers is offered by the method of manufactured solutions (MMS) [4,8]. The method consists of adding source terms to the right-hand side of the governing equations in such a way that a previously chosen (manufactured) solution is the exact solution of the modified equations. Examples of code verification of Newtonian fluid solvers based on MMS can be found in Refs. [9–15] and in the references therein. On the other hand, in the context of non-Newtonian fluids, formal code verification studies based on MMS have started to appear only recently in the literature [16–19].

In this article, we present a code verification study based on MMS of a CFD code for generalized Newtonian fluids. The present work differs from the previous studies in a number of ways. First, the code verification procedure and results are discussed more in detail. The grid/time convergence properties of the viscosity are also analyzed, which helps to build confidence in the correctness of the implemented rheological models. Second, the extrapolated error for cell size/time-step zero is examined instead of assuming that it is zero, which is especially helpful when the expected order of convergence is unknown [15]. Last, code verification is performed for both single- and two-phase flows, including a test case with a free surface. Two-phase flows are modeled with the Volume-of-Fluid method in this study.

[1]Corresponding author.
Manuscript received September 15, 2020; final manuscript received February 1, 2021; published online March 15, 2021. Assoc. Editor: Luis Eca.

The procedure is illustrated herein on three test cases of increasing complexity using a finite volume CFD code and Herschel-Bulkley fluids as an example. Nonetheless, the same methodology is applicable to any CFD technique and it can be easily extended to all rheological models falling under the class of GNF models, which are the most common types of non-Newtonian models.

The article is organized as follows. Section 2 illustrates the Herschel-Bulkley constitutive equation and its regularization. Section 3 presents the governing equations involved in the verification while Sec. 4 outlines the verification procedure. Section 5 gives an overview of the flow solver used in this article. Section 6 to 8 present and discuss results of the three test cases, respectively. The main conclusions are summarized in Sec. 9.

2 The Constitutive Equation for Herschel-Bulkley Fluids

Any fluid displaying flow characteristics that significantly deviate from the Newtonian constitutive equation is classified as non-Newtonian. The present work focuses on a subclass of non-Newtonian models, the GNF models, for which the deviatoric part of the stress tensor can be written in the form

$$\bar{\bar{\tau}} = 2\,\mu(\dot{\gamma})\,\bar{\bar{D}} \tag{1}$$

where $\bar{\bar{D}} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ is the rate of strain tensor, $\mathbf{u} = (u_x, u_y, u_z)$ is the velocity vector in Cartesian coordinates, $\dot{\gamma} = 2\sqrt{D_{ij}D_{ij}/2}$ is the shear rate and $\mu(\dot{\gamma})$ is the so-called apparent viscosity. For Newtonian fluids, $\mu(\dot{\gamma})$ is a constant equal to the molecular viscosity, whereas for Herschel-Bulkley fluids the viscosity reads

$$\begin{cases} \mu = \dfrac{\tau_0 + k\ \dot{\gamma}^n}{\dot{\gamma}}\,, & \tau_0 \le \tau \\ \mu = \infty\,, & \tau < \tau_0 \end{cases} \tag{2}$$

where τ_0 is the yield stress, $\tau = \sqrt{\tau_{ij}\tau_{ij}/2}$ is the second invariant of $\bar{\bar{\tau}}$, n is the flow index and k is the consistency parameter, which has dimensions of a viscosity when $n = 1$. The infinite viscosity means that the fluid does not deform ($\bar{\bar{D}} = 0$) when the stress level is below the yield stress.

The issue associated with the infinite viscosity in Eq. (2) is typical of viscoplastic models and it can be avoided using regularization methods [7,20]. These methods consist in approximating the nondifferentiable constitutive equation with a smoother and differentiable equation that is valid in the whole domain, regardless of the level of shear stress. Over the years, several regularization methods have been proposed. For this work we used the regularization proposed by Souza Mendes and Dutra [21]. Thus, the ideal Herschel-Bulkley model given by Eq. (2) is replaced by the regularized variant

$$\mu = \frac{\tau_0 + k\ \dot{\gamma}^n}{\dot{\gamma}}(1 - e^{-m\dot{\gamma}}) \tag{3}$$

where m is the regularization parameter that controls the level of approximation of the regularized model. In the limit of $m \to \infty$, Eq. (3) tends to Eq. (2). The effect of the regularization on the shear stress and apparent viscosity for the Herschel-Bulkley model is illustrated in Fig. 1 as a function of the shear rate $\dot{\gamma}$. Throughout the article, the term "apparent" will be omitted.

3 Governing Equations

The isothermal flow of a GNF fluid is governed by the continuity and the momentum equations:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}) = 0 \tag{4}$$

$$\frac{\partial(\rho\boldsymbol{u})}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}\,\boldsymbol{u}) = \nabla\cdot\bar{\bar{\tau}} - \nabla p + \rho\boldsymbol{g} \tag{5}$$

with p being the pressure, ρ the fluid density and $\boldsymbol{g} = (0, 0, g)$ the acceleration of gravity vector. The stress tensor $\bar{\bar{\tau}}$ is given by Eq. (1). If the flow is assumed incompressible, the continuity equation reduces to the requirement of a solenoidal velocity, i.e., $\nabla\cdot\boldsymbol{u} = 0$.

For multiphase flows, the flow of two immiscible fluids is modeled with the Volume-of-Fluid (VoF) method introduced by Hirt and Nichols [22], which considers a single continuum fluid having density ρ and viscosity μ defined as

$$\rho = \rho_1(1-c) + \rho_2 c\,, \quad \mu = \mu_1(1-c) + \mu_2 c \tag{6}$$

where c is the volume fraction, which is equal to 0 and 1 in the fluid region occupied by fluid 1 and 2, respectively. Note that, if fluid 1 is an Herschel-Bulkley fluid, the viscosity μ_1 is not simply a constant but rather a function of the shear rate by virtue of Eq. (3). The problem is closed by solving, in addition to the continuity and the momentum equations, the transport equation for c

$$\frac{\partial c}{\partial t} + \boldsymbol{u}\cdot\nabla c = 0 \tag{7}$$

which stems from the assumption that c of each particle remains constant and moves with the fluid. Note that the continuity equation Eq. (4) reduces again to $\nabla\cdot\boldsymbol{u} = 0$ by virtue of Eq. (7).

4 Verification Procedure

4.1 The Method of Manufactured Solutions. The MMS is a powerful tool to generate analytical solutions for code verification purposes. A complete description of the MMS can be found, for example, in Refs. [4] and [8]. In brief, an arbitrary (manufactured) solution is chosen and substituted in the governing equations; the remaining terms are then considered as source terms. In other words, the manufactured solution is the solution of a new set of equations that differs from the original one by additional source terms. The latter can be obtained with the aid of computer algebra systems. For the present study, the expressions of the source terms were obtained with the free software Maxima [23]. The scripts used to generate the source terms are reported in Appendix C.

4.2 Discretization Error and Order of Accuracy. It is generally accepted to divide numerical errors in three components: the round-off error, the iterative error and the discretization error.[2] Code verification requires the evaluation of the discretization error, therefore the other two components must be reduced to negligible levels.

Round-off errors are due to the finite precision of computers and, for the calculations in this article, they can be safely neglected using double-precision number format. On the other hand, iterative errors arise from the use of iterative methods to solve the nonlinear system of equations. Their contribution can be neglected reducing the residuals to machine accuracy, although for practical applications less strict criteria are often sufficient. For the present work, we ensured that the convergence tolerances adopted for the residuals were sufficiently strict to avoid contamination of discretization errors by iterative errors. This was done by systematically reducing the convergence tolerance until numerical errors were no longer influenced by the choice of the convergence tolerance.[3] Therefore, in the remainder, it can be safely

[2]For periodic flows, there is also the contribution of statistical errors [24], which are out of the scope of this article.

[3]We also made sure that residuals did not stagnate.

assumed that the contributions of iterative and round-off errors are both negligible compared to the contribution of the discretization error reads.

Discretization errors stem from discretization of space and time derivatives in the governing equations. The standard approach is to assume that the discretization error $e(\phi)$ of any local or functional quantity ϕ follows a truncated power-series behavior [15,25]. Thus, for steady calculations, the discretization error reads

$$e(\phi_i) = \phi_i - \phi_{\text{exact}} = e_0 + \alpha\left(\frac{h_i}{h_1}\right)^p \tag{8}$$

where ϕ_{exact} is the exact solution of the mathematical model, ϕ_i is the computed solution on a grid having cell size h_i ($i = 1$ corresponds to the finest grid), α is a constant, e_0 is the extrapolated error for cell size zero ($h_i = 0$) and p is the observed order of grid convergence. The three constants e_0, α and p can be determined knowing the error on at least three grids. In this article, the three selected grids cover a refinement ratio of about 2.

In Eq. (8) it is implicitly assumed that the grid refinement is constant for the whole grid, allowing the use of one parameter (cell size in this case) as representative of the grid resolution. In other words, the grids are assumed to be geometrically similar. The consequences of using unsimilar grids in code verification are discussed in Ref. [15].

For unsteady calculations, the error is a function of both the grid size h_i and the time-step $\tau_i = \Delta t_i$

$$e(\phi_i) = \phi_i - \phi_{\text{exact}} = e_0 + \alpha_x\left(\frac{h_i}{h_1}\right)^{p_x} + \alpha_t\left(\frac{\tau_i}{\tau_1}\right)^{p_t} \tag{9}$$

where τ_1 is the smallest time-step. By writing the grid refinement as a function of the time-step or vice versa

$$\left(\frac{h_i}{h_1}\right) = \left(\frac{\tau_i}{\tau_1}\right)^{\frac{p_t}{p_x}} \quad \text{or} \quad \left(\frac{\tau_i}{\tau_1}\right) = \left(\frac{h_i}{h_1}\right)^{\frac{p_x}{p_t}} \tag{10}$$

the form of Eq. (8) is retrieved

$$e(\phi_i) = \phi_i - \phi_{\text{exact}} = e_0 + \alpha\left(\frac{\lambda_i}{\lambda_1}\right)^p \tag{11}$$

where

$$\frac{\lambda_i}{\lambda_1} = \frac{h_i}{h_1} = \left(\frac{\tau_i}{\tau_1}\right)^{\frac{p_t}{p_x}} \tag{12}$$

and e_0, α, and p have the same meaning as in Eq. (8). In this work, Eqs. (8) and (11) are used to estimate e_0, α and p for steady and unsteady simulations, respectively. The use of Eq. (9) is discussed in Ref. [24].

While errors are evaluated at every grid cell, the convergence properties of ϕ will be evaluated based on the L_1, L_2, and L_∞ error norms

$$L_q[e(\phi)] = \left(\frac{1}{N_{\text{cells}}}\sum_i^{N_{\text{cells}}} |\phi_i - \phi_{\text{exact}}|^q\right)^{\frac{1}{q}} \quad \text{with} \quad q = 1, 2 \tag{13}$$

$$L_\infty[e(\phi)] = \max|\phi_i - \phi_{\text{exact}}| \quad \text{for} \quad 1 < i < N_{\text{cells}} \tag{14}$$

with N_{cells} being the total number of grid cells.

The goal of code verification is to demonstrate the correctness of the code by showing that $e_0 = 0$ and that the observed rate of convergence p matches the theoretical order of accuracy. However, some remarks should be made:

- Correct application of Eqs. (8) and (11) requires sufficiently fine grids/time steps to have the numerical solution in the asymptotic range, i.e. in the range of space/time discretization such that the observed order p is nearly constant.
- e_0 includes the contribution of high order terms that were neglected in Eqs. (8) and (11). Thus, at best, we can expect e_0 to decrease with refinement (if the code is correct) but, in any case, it cannot drop below round-off and iterative errors [15].
- The theoretical order for velocity and pressure is unknown due to the nonlinearity of the system [26], therefore it is common practice to replace the theoretical order with the expected order of the discretization techniques adopted. For the present solver, we expect second-order accuracy for velocity, pressure and volume fraction, unless stated otherwise.
- In code verification one usually examines the grid/time convergence of only dependent variables. The dependent variables for the present work are velocity, pressure and volume fraction. However, as the implementation of GNF models is done by modifying the viscosity (which is a derived quantity and not a dependent variable) it is intuitive that the examination of the grid/time convergence properties of the viscosity also plays an important role for code verification of non-Newtonian fluid solvers. Unfortunately, neither the theoretical nor the expected order of accuracy of viscosity is known, thus the correctness of the viscosity will be verified by ensuring that discretization and extrapolated errors decrease with grid/time step refinement.

5 Flow Solver

The CFD code used for this work is ReFRESCO [27], a viscous-flow code currently being developed and verified for maritime purposes by the Maritime Research Institute of the Netherlands (MARIN) in collaboration with several nonprofit organizations around the world. The code solves multiphase

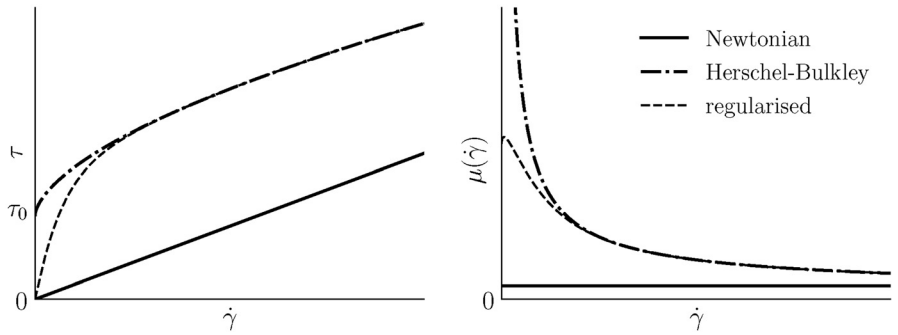

Fig. 1 Shear stress $\tau = \mu(\dot{\gamma})\dot{\gamma}$ and apparent viscosity as a function of the shear rate for Newtonian and Herschel-Bulkley models ($n < 1$). The dashed lines show the effect of the regularization on the Herschel-Bulkley model. The axes are in linear scales.

(unsteady) incompressible flows using momentum, continuity and volume-fraction transport equations. A number of other features such as turbulence and cavitation models are also included, however they are not considered for the present verification.

Equations are discretized in strong-conservation form with the finite volume method for unstructured meshes with cell-centred colocated variables. Spurious pressure oscillations arising from the colocated arrangment are prevented using the pressure-weighted interpolation method [28]. Mass conservation is ensured with a pressure-correction equation based on the SIMPLE algorithm [29]. The convective flux in the momentum equation is linearized with the Picard's method and it is discretized with the total variation diminishing (TVD) Harmonic scheme [30].

After each SIMPLE (outer) iteration, the computed velocity is used to evaluate first the shear rate $\dot{\gamma}$ and then the apparent viscosity $\mu(\dot{\gamma})$, by virtue of Eq. (3). The viscosity is then updated in the momentum equation for the next SIMPLE iteration.

For the unsteady two-phase flow simulations, time integration is performed implicitly with the second-order backward scheme (BDF2). At each implicit time-step, the linearized system for velocity and pressure is solved using the SIMPLE algorithm. After each outer iteration, the calculated velocity is then used to solve the volume-fraction equation and to calculate the apparent viscosity.

We anticipate that for case 3 (Sec. 8), where the chosen manufactured solution yields a sharp (discontinuous) interface between the fluids, the convective flux of the volume-fraction equation is discretized with an interface-capturing scheme [31], a blend of compressive and high-resolution interpolation schemes. This scheme prevents both the smearing of the interface due to the numerical diffusion and the appearance of spurious oscillations of the volume fraction. However, for case 2 (Sec. 7), where the interface between the fluids is not sharp, the interface-capturing scheme is not suitable. In this case the interface-capturing scheme would tend to sharpen the interface, causing large numerical errors in the volume fraction. For this reason, the convective flux of the volume-fraction equation for case 2 is discretized with the TVD Harmonic scheme.

Details about the above mentioned numerical techinques can be found in a number of textbooks (e.g., Ref. [32]), hence they are not further discussed.

In the following sections, ReFRESCO is used to illustrate three verification exercises that can be used to demonstrate the correct implementation of the Herschel-Bulkley model and, more in general, GNF models.

6 Case 1: Steady Single-Phase Flow

6.1 Test Case Set-Up. The first code verification exercise assesses the correctness of the implementation of the Herschel-Bulkley model for laminar single-phase flows using the manufactured solution from [10,15,33]

$$\begin{aligned} u_x(x,y) &= \sin(x^2+y^2)+\epsilon \\ u_y(x,y) &= \cos(x^2+y^2)+\epsilon \\ p(x,y) &= \sin(x^2+y^2)+\epsilon \end{aligned} \tag{15}$$

where $\epsilon = 0.001$ is a small constant added to avoid symmetry in the solution. Note that the manufactured solution is not divergence-free, therefore the term $-2/3(\nabla \cdot \boldsymbol{u})\,\delta_{ij}$ must be included in the stress tensor, with δ_{ij} being the Kronecker delta. With this manufactured solution, pressure is expected to be first-order accurate at the boundaries for the reasons explained in Ref. [15].

To obtain asymptotic grid convergence without excessive grid resolution, large viscosity gradients should be avoided. This is achieved (a) by selecting low values for both the yield stress and the regularization parameter (thus limiting the maximum viscosity) and (b) by moving the computational domain away from the axes origin. The latter stems from the fact that $\dot{\gamma}(0,0) = 0$ and so the viscosity variation is steeper around the origin (see also Fig. 1 to recognize the stronger variation of $\mu(\dot{\gamma})$ near $\dot{\gamma} = 0$). This results in a relatively small variation of the viscosity within the domain (Fig. 2), especially when compared with realistic flow conditions in which the viscosity in low-deformation regions can be thousands/millions times larger than the viscosity in high-deformation regions (e.g., in boundary-layer flows). We recall however that physical realism is not a requirement for code verification as the latter is purely a mathematical exercise [8].

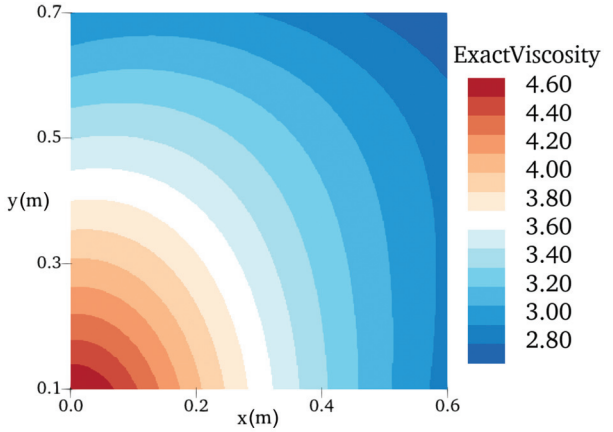

Fig. 2 Exact viscosity $\mu(x, y)$ (Pa·s) of the regularized Herschel-Bulkley model for case 1

On the other hand, a convenient choice of the fluid density should ensure a dominant contribution of the viscous term in the momentum equations. By doing so, there are more chances that anomalies in the viscous term will be detected from the convergence properties of velocity and pressure. With the current choice of fluid properties, convective and pressure terms in the momentum equations have about the same order of magnitude, whereas the diffusive term is about one order of magnitude larger[4] (Fig. 3).

The domain is discretized with 21 two-dimensional uniform Cartesian grids, the finest grid having 640×640 cells. Dirichlet conditions based on the manufactured velocity are applied to all boundaries together with Neumann conditions for pressure. Since no pressure boundaries are used, a reference pressure is imposed in one point using the exact pressure. Furthermore, pressure and viscosity are both linearly extrapolated to the boundaries using the gradient from the previous outer iteration. The parameters used for the calculations are given in Table 6 in Appendix B.

6.2 Results and Discussion. The L_2 and L_∞ error norms and the observed order of convergence p are reported in Table 1 as function of grid refinement.

All quantities appear to be in the asymptotic range for about $h_i/h_1 < 4$, where p is nearly constant. Velocity matches the expected second-order both for the L_2 and the L_∞ error norms, whereas the pressure coefficient C_p exhibits first-order accuracy in the L_∞ error norm. The largest errors were found near the boundaries, with the maximum error on the top left corner. This confirms that pressure is first-order accurate at the boundaries, as we expected. As a result of the first-order behavior at the boundaries and second-order on interior cells, the L_2 error norm of C_p converges with an order about 1.9. It is also remarkable that, with the chosen fluid properties and regularization parameter, the grid convergence properties of velocity and pressure with the Herschel-Bulkley model are very similar to those of the verification

[4]For a thorough code verification, different combinations of the fluid properties should be considered to give more relevance also to other terms in the governing equations. However, in this article, the focus is on the diffusive term.

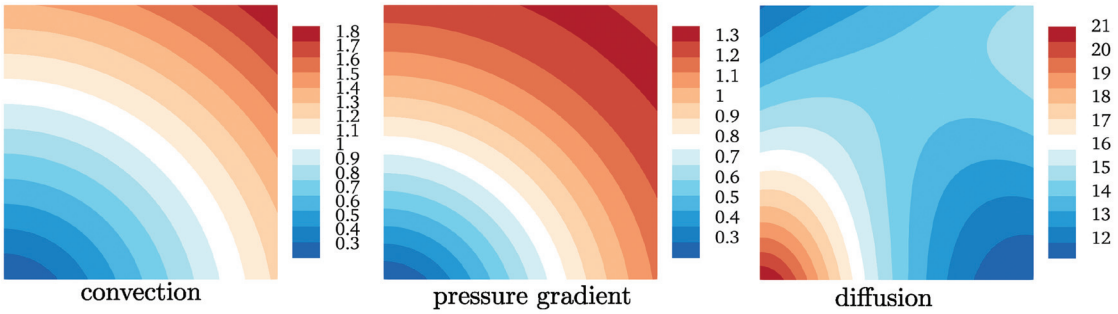

Fig. 3 Magnitude of convection, pressure and diffusion terms (N m^{-3}) corresponding to the manufactured solution of case 1

exercise in Ref. [15] with the same manufactured solution and Newtonian fluids.

The grid convergence properties of μ are very similar to C_p, with a rate of convergence about 1.9 for the L_2 norm and with first-order for the L_∞ norm. The maximum error was found on the top boundary, where also the maximum error of the tangential velocity derivative is located. It was observed that velocity derivatives converged with first-order accuracy at the boundaries.[5] Thus, it appears that the first-order accuracy of velocity derivatives leads to first-order accuracy of the viscosity, as a result of their dependency (Eq. (3)).

The examination of the extrapolated error for cell size zero e_0 is convenient when p does not match the theoretical (expected) order of grid convergence or when the asymptotic order of convergence is hard to determine [15]. For the present exercise, this is the case of viscosity and pressure. Nonetheless, e_0 decreases upon grid refinement for all quantities and for both norms (Fig. 4), providing convincing evidence about the correctness of the code.

Larger values of the yield stress and regularization parameter were also tested, and results were virtually identical to those reported in Table 1, except for marginally larger discretization errors for all quantities. Larger errors are in fact expected for larger yield stress and regularization parameters because of the increased viscosity gradient [35]. We mention, however, that increasing the regularization parameter to very large values, as normally required by practical applications, is neither necessary nor convenient for code verification. In fact, using large regularization parameters will likely cause stagnation of residuals[6] and consequent contamination of numerical errors by iterative errors. Moreover, even when residuals converge, asymptotic grid convergence may be difficult to achieve. In any case, it is shown in Appendix A that with the adopted choice of yield stress and regularization parameter the exercise is very sensitive to coding mistakes in the rheological model.

In conclusion, for all quantities, the observed orders matched the expected orders, and the extrapolated errors e_0 tend to zero upon grid/time-step refinement. Furthermore, the grid convergence properties of velocity and pressure are analogous to the verification with Newtonian fluids in Ref. [15]. The code can thus be considered verified for laminar single-phase flow of Herschel-Bulkley fluids.

7 Case 2: Unsteady Two-Phase Flow With a Continuous Interface

7.1 Test Case Set-Up. This exercise verifies that laminar flows of Herschel-Bulkley fluids are correctly solved also for two-phase flows. The manufactured solution is taken from Ref. [31] and it represents a sinusoidal wave on deep water from potential flow theory [36]

$$u_x(x,z,t) = \frac{g}{c}A\exp\left(\frac{gz}{c^2}\right)\cos\left(\frac{gx}{c^2}-\frac{gt}{c}\right) \tag{16}$$

$$u_z(x,z,t) = \frac{g}{c}A\exp\left(\frac{gz}{c^2}\right)\sin\left(\frac{gx}{c^2}-\frac{gt}{c}\right) \tag{17}$$

$$p(x,z,t) = \rho g(\zeta(x,z,t)-z)) \tag{18}$$

$$c(x,z,t) = \frac{1}{2}\left(1+\mathrm{erf}\left(b\left(z-\zeta(x,z,t)\right)\right)\right) \tag{19}$$

where ζ is the wave elevation

$$\zeta(x,z,t) = A\exp\left(\frac{gz}{c^2}\right)\cos\left(\frac{gx}{c^2}-\frac{gt}{c}\right) \tag{20}$$

The pressure is assumed to be zero at the free surface, i.e., at $z=\zeta$. We recall that the above solution describes the circular motion of fluid particles for $z \le \zeta$. However, for code verification purposes, the above solution is applied also for $z > \zeta$.

The parameter b in the error function erf in Eq. (19) determines how steeply the volume fraction (and fluid properties) varies around $z=\zeta$. The error function assumes values from 0 to 1 over a distance of about $2/b$. Thus, if $2/b$ is less than the cell size, the volume fraction is discontinuous at the discrete level. Verification with a discontinuous solution is more challenging because the asymptotic grid/time convergence is extremely hard to achieve, thus we postpone it to Sec. 8. For the present test case we considered $b=12$, which produces a smooth variation of the volume fraction from one fluid to the other (Fig. 5). Note that the viscosity of fluid 1 (right plot in Fig. 5) varies in space not only because of the volume fraction but also because fluid 1 is non-Newtonian.

Considering a smooth interface increases the chances of achieving asymptotic grid/time convergence with reasonable grid/time resolution. The drawback is that special treatments for discontinuities may not be fully exercised by the verification. Therefore, a successful outcome from this exercise can be seen as a necessary but not sufficient condition to ensure that the code works correctly for free-surface calculations with Herschel-Bulkley fluids. On the other hand, this test case well represents those applications where the interface between two liquids (e.g., between water and mud suspensions) is not as sharp as a gas–liquid interface. For such applications, one might avoid the use of interface-capturing schemes, thus allowing smearing of the interface by numerical diffusion. In light of the above considerations, liquid-like properties are assigned to both fluids to justify the use of a smooth interface between the fluids.

[5]This is because the velocity derivatives are computed with the Gauss's theorem, which is a second-order method but it reduces to first order at the boundary, even on Cartesian grids [34].

[6]Stagnation of residuals for large regularization parameters is a known issue for SIMPLE-type solvers [35]. The robustness of the code should be addressed by performing solution verification on benchmark/realistic flow problems.

Table 1 L_2 and L_∞ error norms and observerd order of convergence p for velocity, pressure coefficient and viscosity for case 1

h_i/h_1	$L_2[e(u)]$	p	$L_2[e(C_p)]$	p	$L_2[e(\mu)]$	p	$L_\infty[e(u)]$	p	$L_\infty[e(C_p)]$	p	$L_\infty[e(\mu)]$	p
16.00	1.23×10^{-5}	2.00	5.27×10^{-4}	1.86	8.06×10^{-5}	1.77	1.03×10^{-4}	1.99	4.71×10^{-2}	0.70	7.16×10^{-3}	0.98
13.33	8.55×10^{-6}	2.00	3.78×10^{-4}	1.78	5.76×10^{-5}	1.79	7.20×10^{-5}	1.99	3.99×10^{-2}	0.78	5.97×10^{-3}	0.98
11.23	6.06×10^{-6}	2.00	2.78×10^{-4}	1.81	4.19×10^{-5}	1.81	5.12×10^{-5}	1.99	3.39×10^{-2}	0.81	5.03×10^{-3}	0.98
9.28	4.14×10^{-6}	2.00	1.96×10^{-4}	1.88	2.93×10^{-5}	1.82	3.50×10^{-5}	1.98	2.84×10^{-2}	0.81	4.16×10^{-3}	0.99
8.00	3.08×10^{-6}	2.00	1.49×10^{-4}	1.77	2.23×10^{-5}	1.83	2.61×10^{-5}	1.98	2.47×10^{-2}	0.88	3.59×10^{-3}	0.99
6.53	2.05×10^{-6}	2.00	1.03×10^{-4}	1.77	1.52×10^{-5}	1.84	1.74×10^{-5}	1.98	2.03×10^{-2}	0.90	2.93×10^{-3}	0.99
5.47	1.44×10^{-6}	2.00	7.40×10^{-5}	1.83	1.09×10^{-5}	1.85	1.22×10^{-5}	1.99	1.71×10^{-2}	0.90	2.46×10^{-3}	0.99
4.60	1.02×10^{-6}	2.00	5.38×10^{-5}	1.83	7.88×10^{-6}	1.86	8.71×10^{-6}	1.99	1.45×10^{-2}	0.92	2.07×10^{-3}	0.99
4.00	7.69×10^{-7}	2.00	4.15×10^{-5}	1.84	6.04×10^{-6}	1.87	6.59×10^{-6}	1.99	1.27×10^{-2}	0.93	1.80×10^{-3}	0.99
3.23	5.02×10^{-7}	2.00	2.79×10^{-5}	1.84	4.03×10^{-6}	1.87	4.32×10^{-6}	1.98	1.03×10^{-2}	0.94	1.45×10^{-3}	1.00
2.71	3.54×10^{-7}	2.00	2.01×10^{-5}	1.84	2.89×10^{-6}	1.88	3.05×10^{-6}	1.98	8.66×10^{-3}	0.95	1.22×10^{-3}	1.00
2.28	2.49×10^{-7}	2.00	1.45×10^{-5}	1.85	2.07×10^{-6}	1.88	2.15×10^{-6}	1.98	7.29×10^{-3}	0.96	1.03×10^{-3}	1.00
2.00	1.92×10^{-7}	2.00	1.14×10^{-5}	1.85	1.62×10^{-6}	1.89	1.66×10^{-6}	1.98	6.42×10^{-3}	0.96	9.01×10^{-4}	1.00
1.61	1.24×10^{-7}	2.00	7.54×10^{-6}	1.86	1.07×10^{-6}	1.89	1.08×10^{-6}	1.98	5.17×10^{-3}	0.97	7.24×10^{-4}	1.00
1.35	8.80×10^{-8}	2.00	5.45×10^{-6}	1.86	7.68×10^{-7}	1.90	7.64×10^{-7}	1.99	4.36×10^{-3}	0.97	6.10×10^{-4}	1.00
1.14	6.21×10^{-8}	2.00	3.92×10^{-6}	1.87	5.51×10^{-7}	1.90	5.40×10^{-7}	1.99	3.67×10^{-3}	0.98	5.12×10^{-4}	1.00
1.00	4.81×10^{-8}	2.00	3.08×10^{-6}	1.87	4.31×10^{-7}	1.90	4.18×10^{-7}	1.99	3.23×10^{-3}	0.98	4.51×10^{-4}	1.00

The computational domain is a square with sides of 1.0 (m), discretized with 21 two-dimensional uniform Cartesian grids, the finest grid having 640×640 cells. Dirichlet boundary conditions based on the manufactured velocity and volume fraction are applied to all boundaries together with Neumann conditions for pressure. As for case 1, pressure and viscosity are both linearly extrapolated to the boundaries using their gradients from the previous outer iteration.

Calculations are initialized with the manufactured solution and are carried out for one wave period T. The time steps are chosen such that the CFL numbers are less than 1/6 to ensure that the BDF2 scheme is TVD [31,37], and the time-step for the finest grid is $\tau_1 = T/4800$. Time steps are refined using the same ratio as the grid refinement, i.e., $h_i/h_1 = \tau_i/\tau_1 = \lambda_i/\lambda_1$. The parameters used for the computations are summarized in Table 6 in Appendix B.

7.2 Results and Discussion. The order of convergence p and the L_1 and L_2 error norms are given in Table 2. It is immediately evident that errors in the velocity, pressure coefficient and volume fraction converge asymptotically with the expected second-order accuracy. The first goal of code verification is thus fulfilled for the dependent variables.

The convergence of the viscosity does not appear asymptotic, with rates of convergence ranging between 1 and 2. The largest viscosity errors were found on the side and bottom boundaries, where also the largest errors of velocity derivatives are located. This suggests that local grid refinement at the boundaries is needed to reduce such errors and to achieve asymptotic convergence for the viscosity without a significant increase of the computational costs. However, local refinement, which requires the use of unstructured/nonuniform grids, is out of the scope of this work and it should be investigated in future studies. Another option may be to improve the discretization of the gradients at the boundaries, but for solvers that use unstructured grid assumptions this option is not straightforward and therefore outside of the scope of the present study. At least, the present exercise proves its usefulness by demonstrating the expected convergence behavior for the dependent variables, while also detecting inconsistencies in the solution that went unnoticed in case 1.

In any case, the extrapolated error e_0 for cell size/time-step zero clearly decreases upon grid/time-step refinement for all quantities (Fig. 6). This, combined with the second-order accuracy of the dependent variables and with the results of case 1, provides compelling evidence that the code performs correctly also for two-phase laminar flows of Herschel-Bulkley fluids with a smooth interface.

8 Case 3: Unsteady Two-Phase Flow With a Free Surface

8.1 Test Case Set-Up. In the previous test case we have shown that by considering a smooth interface between the fluids rigorous code verification is possible also for two-phase flows. However, for applications in which the top fluid is a gas, one usually wants to keep the interface as sharp as possible to accurately

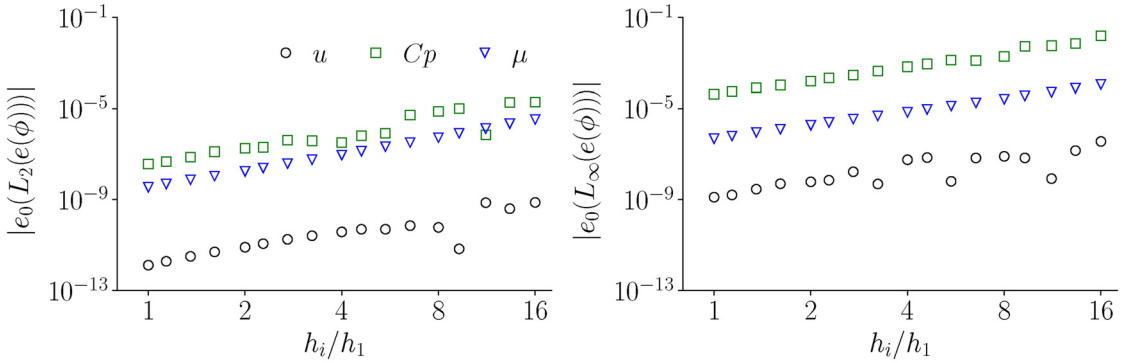

Fig. 4 Extrapolated error for cell size zero e_0 for the L_2 and L_∞ error norms as function of grid refinement for case 1

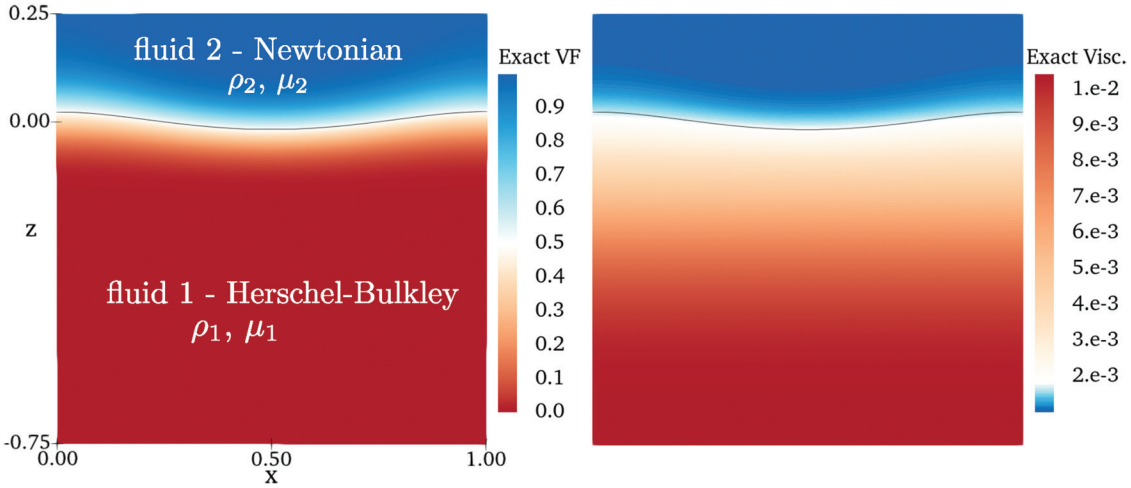

Fig. 5 Exact volume fraction (left) and viscosity (Pa·s) (right) for case 2 with steepening parameter $b = 12$. The black isoline corresponds to $c = 0.5$.

capture the free surface. It can be the case, for example, of mud slurries in an open channel or dip-coating processes [38].

In this case study we test the code on free-surface calculations by increasing the steepening parameter in the manufactured volume fraction (Eq. (19)) from 12 to 1200. Now the volume fraction varies between 0 and 1 over a distance about the cell size of our finest grid, hence the volume fraction is discontinuous at the discrete level, as shown in Fig. 7. Moreover, the convective fluxes of the volume-fraction equation are now discretized with an interface-capturing scheme, as was anticipated in Sec. 5. The grids and all the other parameters are the same as case 2, except for fluid 2, which now has the density and viscosity of air (see Table 6 in Appendix B).

8.2 Results and Discussion. The oscillatory convergence due to the presence of a discontinuity (free surface) is clearly noticeable from the convergence of the L_1 and L_2 error norms in Fig. 8, with discretization errors that sometimes increase upon grid/time-step refinement. For this reason we have estimated α, e_0 and p in the least-square sense by fitting Eq. (11) to the data on the 10 finest grids. The observed order of accuracy is thus indicated in Fig. 8 with p_{ls}.

A close look at the velocity errors in Fig. 9 reveals the presence of the well-known "spurious velocities" [31] for free-surface calculations with the VoF model. These are caused by the density-weighted interpolation scheme for pressure [28], which assumes that the free surface coincides with the cell boundaries. When the free surface is not aligned with the cells boundaries, an imbalance occurs in the discretization of the pressure gradient, which acts as a source/sink of momentum. Since $\rho\boldsymbol{u}$ is conserved across the free surface, errors in the velocity are amplified in the low-density fluid (Fig. 9). For this reason they are often called "spurious air velocities." This explains the oscillatory convergence of pressure and velocity, and it is reassuring that, overall, there is a clear decreasing trend as the grid/time-step is refined.

We remark that artifacts in the velocity have nothing to do with the use of non-Newtonian fluids. However, when non-Newtonian fluids are used, additional artifacts are generated. In fact, spurious velocities cause large errors in the shear rate $\dot{\gamma}$, as shown in the top panel of Fig. 10. In turn, such large errors in the shear rate produce "spurious viscosities" in the non-Newtonian fluid near the free surface (bottom panel in Fig. 10), hence explaining the oscillatory convergence of the viscosity.

On the other hand, the convergence of the volume fraction is monotonic and exhibits a rate of convergence p_{ls} for the L_1 error

Table 2 L_1 and L_2 error norms and observerd order of convergence p for case 2

λ_i/λ_1	$L_1[e(u)]$	p	$L_1[e(C_p)]$	p	$L_1[e(c)]$	p	$L_1[e(\mu)]$	p	$L_2[e(u)]$	p	$L_2[e(C_p)]$	p	$L_2[e(c)]$	p	$L_2[e(\mu)]$	p
16.00	1.12×10^{-3}	2.08	1.32×10^{-2}	2.15	4.18×10^{-3}	2.42	3.00×10^{-2}	2.00	1.70×10^{-3}	2.08	2.09×10^{-2}	2.18	8.41×10^{-3}	2.35	3.87×10^{-2}	1.92
13.33	7.73×10^{-4}	2.04	8.98×10^{-3}	2.08	2.66×10^{-3}	2.59	2.09×10^{-2}	2.00	1.16×10^{-3}	2.13	1.41×10^{-2}	2.17	5.44×10^{-3}	2.49	2.74×10^{-2}	1.92
11.23	5.44×10^{-4}	2.03	6.55×10^{-3}	2.05	1.79×10^{-3}	2.68	1.53×10^{-2}	2.00	8.12×10^{-4}	2.12	1.01×10^{-2}	2.14	3.71×10^{-3}	2.58	2.04×10^{-2}	1.90
9.28	3.70×10^{-4}	2.05	4.67×10^{-3}	2.04	1.16×10^{-3}	2.70	1.10×10^{-2}	2.00	5.46×10^{-4}	2.12	7.12×10^{-3}	2.12	2.45×10^{-3}	2.61	1.50×10^{-2}	1.88
8.00	2.74×10^{-4}	2.04	3.16×10^{-3}	2.05	6.99×10^{-4}	2.69	7.52×10^{-3}	2.00	4.02×10^{-4}	2.10	4.75×10^{-3}	2.13	1.50×10^{-3}	2.59	1.05×10^{-2}	1.86
6.53	1.82×10^{-4}	2.04	2.18×10^{-3}	2.04	4.34×10^{-4}	2.64	5.23×10^{-3}	2.00	2.65×10^{-4}	2.09	3.23×10^{-3}	2.12	9.49×10^{-4}	2.53	7.50×10^{-3}	1.85
5.47	1.28×10^{-4}	2.03	1.53×10^{-3}	2.03	2.80×10^{-4}	2.56	3.71×10^{-3}	1.99	1.84×10^{-4}	2.09	2.25×10^{-3}	2.09	6.22×10^{-4}	2.46	5.48×10^{-3}	1.85
4.60	9.03×10^{-5}	2.02	1.04×10^{-3}	2.03	1.74×10^{-4}	2.46	2.54×10^{-3}	1.98	1.29×10^{-4}	2.08	1.51×10^{-3}	2.08	3.94×10^{-4}	2.37	3.87×10^{-3}	1.84
4.00	6.81×10^{-5}	2.02	7.72×10^{-4}	2.03	1.22×10^{-4}	2.39	1.90×10^{-3}	1.98	9.70×10^{-5}	2.07	1.11×10^{-3}	2.08	2.80×10^{-4}	2.31	2.96×10^{-3}	1.84
3.23	4.44×10^{-5}	2.01	5.12×10^{-4}	2.03	7.63×10^{-5}	2.30	1.27×10^{-3}	1.97	6.28×10^{-5}	2.06	7.33×10^{-4}	2.08	1.77×10^{-4}	2.23	2.05×10^{-3}	1.84
2.71	3.13×10^{-5}	2.01	3.57×10^{-4}	2.02	5.12×10^{-5}	2.24	8.97×10^{-4}	1.97	4.39×10^{-5}	2.06	5.08×10^{-4}	2.06	1.20×10^{-4}	2.18	1.50×10^{-3}	1.84
2.28	2.20×10^{-5}	2.01	2.52×10^{-4}	2.02	3.52×10^{-5}	2.18	6.40×10^{-4}	1.97	3.08×10^{-5}	2.05	3.57×10^{-4}	2.05	8.33×10^{-5}	2.13	1.11×10^{-3}	1.81
2.00	1.70×10^{-5}	2.00	1.90×10^{-4}	2.02	2.60×10^{-5}	2.13	4.86×10^{-4}	1.97	2.37×10^{-5}	2.04	2.68×10^{-4}	2.05	6.19×10^{-5}	2.10	8.70×10^{-4}	1.78
1.61	1.10×10^{-5}	2.00	1.23×10^{-4}	2.02	1.66×10^{-5}	2.08	3.21×10^{-4}	1.96	1.52×10^{-5}	2.04	1.73×10^{-4}	2.04	3.98×10^{-5}	2.06	6.08×10^{-4}	1.72
1.35	7.77×10^{-6}	2.00	8.66×10^{-5}	2.02	1.15×10^{-5}	2.06	2.28×10^{-4}	1.94	1.07×10^{-5}	2.03	1.21×10^{-4}	2.03	2.78×10^{-5}	2.05	4.56×10^{-4}	1.65
1.14	5.48×10^{-6}	2.00	6.09×10^{-5}	2.01	8.08×10^{-6}	2.06	1.62×10^{-4}	1.92	7.56×10^{-6}	2.03	8.51×10^{-5}	2.03	1.95×10^{-5}	2.04	3.46×10^{-4}	1.56
1.00	4.24×10^{-6}	2.00	4.69×10^{-5}	2.01	6.20×10^{-6}	2.04	1.26×10^{-4}	1.92	5.84×10^{-6}	2.03	6.55×10^{-5}	2.02	1.50×10^{-5}	2.03	2.83×10^{-4}	1.50

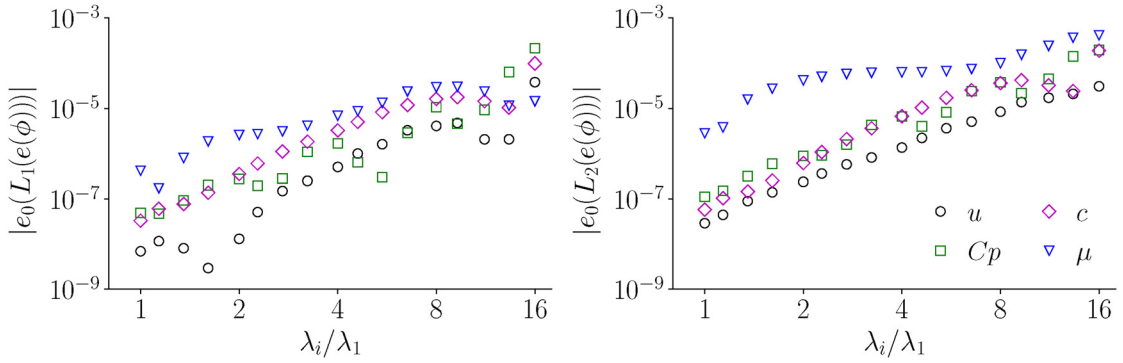

Fig. 6 Extrapolated error for grid size/time-step zero e_0 for the L_1 and L_2 error norms as a function of the refinement factor for case 2

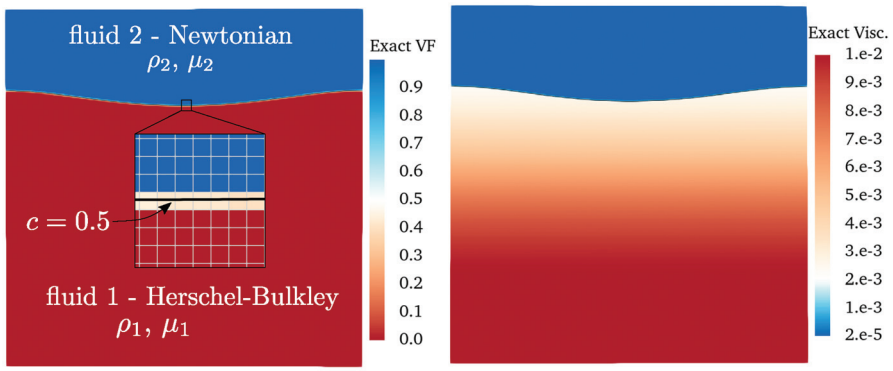

Fig. 7 Exact volume fraction (left) and viscosity (Pa·s) (right) for case 3 with steepening parameter $b = 1200$. The black isoline corresponds to $c = 0.5$.

norm close to 1.0, whereas for the L_2 error norm p_{ls} is roughly halved. These results agree with the verification of the stand-alone volume-fraction equation with a discontinuous initial solution in Ref. [31].

A common practice in code verification is to assume $e_0 = 0$ in Eqs. (8) and (11) (see e.g., Refs. [12], [13], [31], and [33]). The implications of making such assumption are discussed in detail in Ref. [15]. The main advantage is that the observed order can be simply determined from data on grid doublets as

$$p^*[e(\phi_i)] = \frac{\log\left(e(\phi_{i-1})/e(\phi_i)\right)}{\log(2)} \tag{21}$$

where, in this case, the selected grids cover a refinement ratio of 2.

For the present exercise we assumed $e_0 = 0$ for two reasons. First, solving a nonlinear system of three equations to determine e_0, α and p was not possible because of the oscillatory grid/time

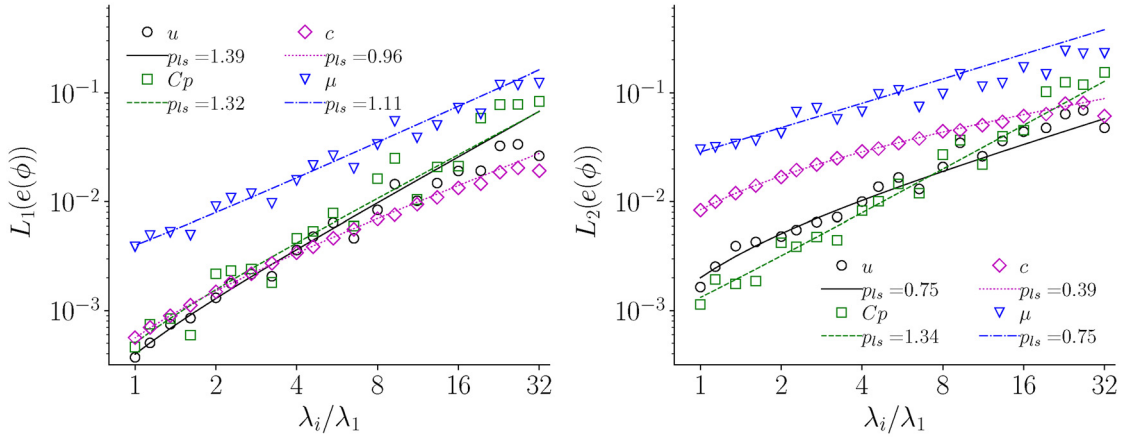

Fig. 8 L_1 error norms as function of the refinement factor. p_{ls} is obtained from the best weighted least-square fitting of Eq. (11) to data on the 10 finest grids.

Table 3 L_1 and L_2 error norms and observerd order of convergence p^* (Eq. (21)) for case 3

λ_i/λ_1	$L_1[e(u)]$	p^*	$L_1[e(C_p)]$	p^*	$L_1[e(c)]$	p^*	$L_1[e(\mu)]$	p^*	$L_2[e(u)]$	p^*	$L_2[e(C_p)]$	p^*	$L_2[e(c)]$	p^*	$L_2[e(\mu)]$	p^*
32	2.64×10^{-2}	–	8.37×10^{-2}	–	1.92×10^{-2}	–	1.23×10^{-1}	–	4.78×10^{-2}	–	1.54×10^{-1}	–	6.13×10^{-2}	–	2.30×10^{-1}	–
16	1.91×10^{-2}	0.46	2.12×10^{-2}	1.98	1.34×10^{-2}	0.53	7.24×10^{-2}	0.76	4.40×10^{-2}	0.12	4.55×10^{-2}	1.76	6.14×10^{-2}	0.00	1.71×10^{-1}	0.43
8	8.40×10^{-3}	1.19	1.63×10^{-2}	0.38	6.97×10^{-3}	0.94	3.35×10^{-2}	1.11	2.09×10^{-2}	1.08	2.70×10^{-2}	0.75	4.45×10^{-2}	0.47	9.77×10^{-2}	0.80
4	3.60×10^{-3}	1.22	4.60×10^{-3}	1.82	3.40×10^{-3}	1.04	1.57×10^{-2}	1.10	1.00×10^{-2}	1.06	8.29×10^{-3}	1.71	2.90×10^{-2}	0.62	6.74×10^{-2}	0.54
2	1.31×10^{-3}	1.46	2.18×10^{-3}	1.08	1.49×10^{-3}	1.19	9.03×10^{-3}	0.79	4.80×10^{-3}	1.06	4.22×10^{-3}	0.97	1.68×10^{-2}	0.79	4.26×10^{-2}	0.66
1	3.71×10^{-4}	1.82	4.63×10^{-4}	2.23	5.63×10^{-4}	1.40	3.87×10^{-3}	1.22	1.65×10^{-3}	1.54	1.14×10^{-3}	1.89	8.35×10^{-3}	1.01	3.00×10^{-2}	0.51

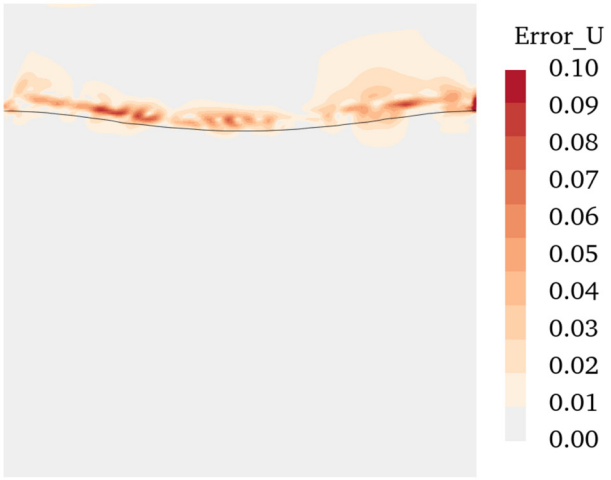

Fig. 9 Velocity errors on the grid with 160^2 cells. The black isoline corresponds to $c = 0.5$.

convergence caused by the above-mentioned spurious velocities. Second, the present results can be more easily compared with the work of Klaij et al. [31], where the order of convergence was also determined assuming $e_0 = 0$. With this in mind, the discretization error and order of convergence p^* are reported in Table 3.

The convergence of the L_1 error of pressure and volume fraction is similar to Ref. [31]: pressure errors converge with order oscillating between 1 and 2, whereas the volume fraction is well-behaved and its order of convergence reaches 1.4. On the other hand, our velocity errors appear to be reaching second-order accuracy, in contrast with the order close to 2/3 in Ref. [31]. Such difference is due to the different boundary conditions applied on the bottom boundary. In Ref. [31], a Dirichlet condition for pressure was imposed on the bottom using the exact pressure at $t = 0$, whereas in the present work we imposed a Dirichlet condition for velocity using the exact velocity at each time-step. This, combined with the slightly finer grids of the present work, produced a better convergence of the velocity error compared to Ref. [31].

To summarize, while results of this code verification exercise are less conclusive than case 2, errors of all quantities have a clear decreasing trend upon grid/time-step refinement, and the order p^* compares favorably with an earlier verification exercise performed with only Newtonian fluids [31]. We have also shown that the oscillatory convergence is due to the presence of spurious velocities, a known issue for many free-surface calculations

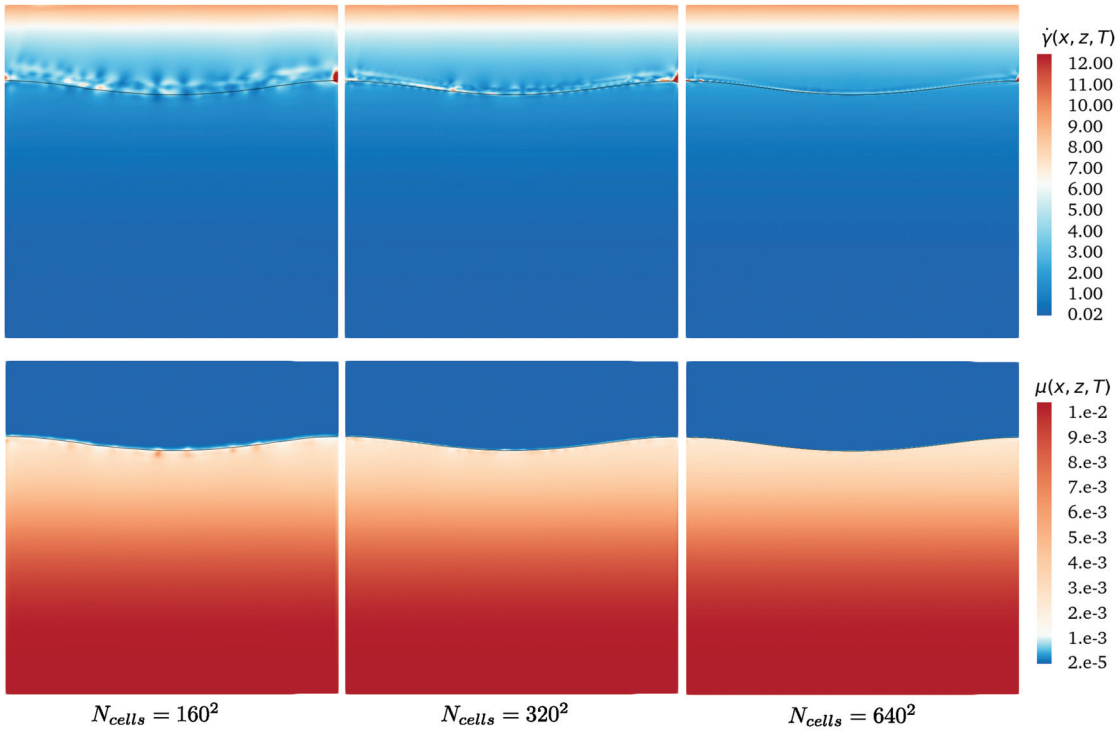

Fig. 10 Contour plots of the shear rate $\dot{\gamma}$ (s^{-1}) (top panel) and the viscosity $\mu(\dot{\gamma})$ (Pa·s) (bottom panel) for different refinement levels at computational time $t = T$. The artifacts caused by spurious velocities are cleary visible near the free surface on the grid with 160^2 cells.

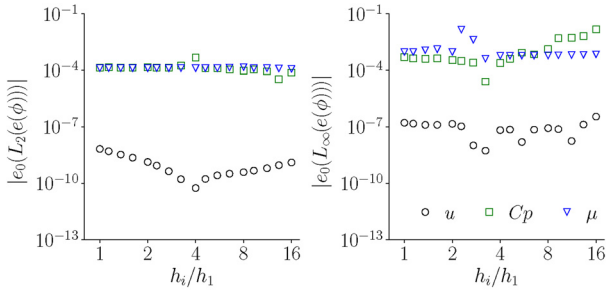

Fig. 11 Extrapolated error e_0 for case 1 with err = 0.01%

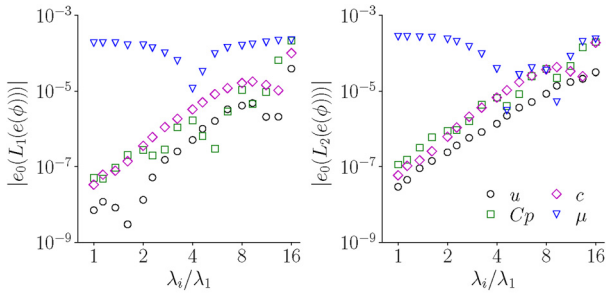

Fig. 12 Extrapolated error e_0 for case 2 with err = 0.01%

performed using the VOF method and the density-weighted interpolation scheme.

Finally, the exercise revealed that spurious velocities produce artifacts in the viscosity of the non-Newtonian fluid, but it was just shown that such artifacts tend to disappear with grid/time-step refinement. For practical applications where high grid resolution around the free surface is too expensive, spurious viscosities in the non-Newtonian fluid might be even visible to the naked eye. Fortunately, spurious velocities mainly occur in low density fluids, which are typically Newtonian, thus the viscosity of the non-Newtonian fluid is only moderately affected. Furthermore, the impact of these spurious viscosities on the flow field depends on how large the viscous forces are in comparison with other forces, and this varies from application to application.

9 Conclusions

In this study, code verification of an Herschel-Bulkley fluid solver for laminar flows has been performed and discussed through three exercises with increasing complexity: steady single-phase flow (case 1), unsteady two-phase flow with a smooth interface (case 2) and with a free surface (case 3).

For cases 1 and 2, the two goals of code verification are achieved: the observed order of convergence matches the expected order and the extrapolated error to cell size/time-step zero tends to zero with grid/time-step refinement. Therefore, the code performs as intended for both single- and two-phase laminar flows of Herschel-Bulkley fluids. Code verification in case 3 is less conclusive due to the lack of asymptotic grid/time convergence. Nevertheless, this test case can be used as an extension of case 2 to check that, in presence of a free surface, errors of all quantities have at least an overall decreasing trend with grid/time-step refinement. Furthermore, the exercise revealed that the well-known "spurious velocities" typical of free-surface calculations with the VoF model induce "spurious viscosities" in the non-Newtonian fluid. We have however demonstrated that these artifacts disappear with grid/time-step refinement.

Additionally, it is shown in Appendix A that examining the convergence properties of the viscosity is a valuable tool for detecting coding mistakes in the rheological model, especially for

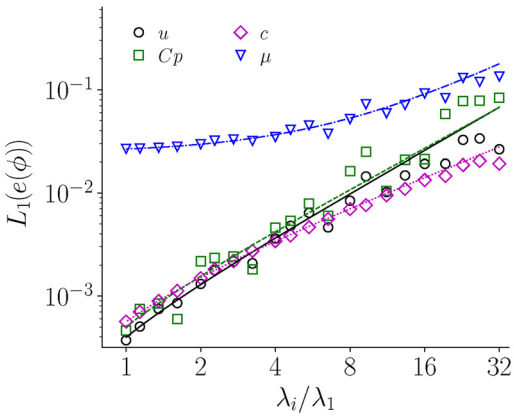

Fig. 13 L_1 error norms as function of the refinement factor with err = 1%. Lines are obtained from the best weighted least-square fitting of Eq. (11) to data on the 10 finest grids.

case 2 and case 3, where viscous effects have little influence on the convergence properties of the dependent variables.

Finally, it should be noted that the procedure illustrated here on a finite volume code for Herschel-Bulkley fluids can also be used to verify and demonstrate the correct implementation of other GNF models and in any type of CFD code. Further research is required to extend this work to more complex rheological models, such as viscoelastic models, and to nonuniform/unstructured grids.

Acknowledgment

This work is part of the research project SailFM, which is supported by the Dutch Research Council (NWO), Port of Rotterdam (PoR), Rijkswaterstaat, Deltares, MARIN and the Dutch Ministry of Economic Affairs. Calculations were performed on the Reynolds (TU Delft) and Marclus4 (MARIN) clusters. This study was carried out within the framework of the MUDNET academic network.[7]

Appendix A: Sensitivity to Coding Mistakes in the Rheological Model

This exercise illustrates the sensitivity of the three test cases to coding errors in the implementation of the apparent viscosity. For this purpose, we have simulated the presence of bugs in the viscosity by inserting the following command right below the line of code where the apparent viscosity is calculated:

```
mu(:) = mu(:) * (1.0 + err/100.0)
```

where `err` represents the (small) coding error in percentage. This error mimics the presence of coding mistakes in the apparent viscosity that produce a uniform viscosity error equal to `err`.

For case 1, the presence of a coding error is clearly visible from both the observed order of accuracy p (Table 4) and the convergence of e_0 (Fig. 11) of all quantities.

Table 4 Observed order p for the five finest grids of case 1 with err=0.01%

h_i/h_1	$p(L_2[e(\phi)])$			$p(L_\infty[e(\phi)])$		
	u	C_p	μ	u	C_p	μ
2.00	2.01	−0.11	2.32	2.05	0.96	9.21
1.61	2.01	2.02	2.24	2.03	0.97	−0.04
1.35	2.02	−0.13	2.19	2.03	0.97	−0.04
1.14	2.04	−0.10	2.14	2.07	0.97	1.05
1.00	2.08	0.73	2.11	2.11	0.98	1.03

[7] https://www.tudelft.nl/mudnet/.

Table 5 Observed order p for the five finest grids of case 2 with err=0.01%

λ_i/λ_1	$p(L_1[e(\phi)])$				$p(L_2[e(\phi)])$			
	u	C_p	c	μ	u	C_p	c	μ
2.00	2.00	2.02	2.13	5.05	2.04	2.05	2.10	2.82
1.61	2.00	2.02	2.08	5.05	2.04	2.04	2.06	3.45
1.35	2.00	2.02	2.06	9.09	2.03	2.03	2.05	4.10
1.14	2.00	2.01	2.06	9.09	2.03	2.03	2.04	5.37
1.00	2.00	2.01	2.04	9.09	2.03	2.02	2.03	8.65

For case 2, the observed order of accuracy of viscosity appears to be highly sensitive to the small coding error (Table 5), whereas for velocity, pressure and volume fraction, p is virtually unaffected. Likewise, the extrapolated error $e0$ (Fig. 12) shows an alarming trend for the viscosity, whereas for the other variables e_0 appears to be the same as in the exercise without the coding error.

For case 3, the presence of an anomaly in the rheological model is suggested by the convergence of the viscosity error, which appears to stagnate instead of decreasing with grid refinement (Fig. 13). However, as for case 2, the grid/time convergence properties of the other variables appear to be unaffected by the coding error.

The insensitivity of the dependent variables to coding mistakes in the viscosity for cases 2 and 3 is due to the fact that the diffusion term in the momentum equations is much smaller than the other terms. This is simply due to the nature of the manufactured solution, which represents a gravity wave. Therefore, for cases 2 and 3, it is recommended to examine the grid/time convergence properties of the viscosity for detection of bugs in the rheological model.

Appendix B: Parameter Values Used for the Three Test Cases

Table 6 Parameters used for the three test cases

Parameter	Value
Case 1	
Computational domain (m)	$(x, y) \in [0, 0.6] \times [0.1, 0.7]$
Finest grid	640×640
Density (kg m^{-3})	$\rho = 1$
Flow index	$n = 0.8$
Consistency parameter (Pa s^n)	$k = 0.5$
Yield stress (Pa)	$\tau_0 = 5$
Regularization parameter (s)	$m = 1$
Convergence tolerance for residuals	L_∞ norm $< 2 \times 10^{-14}$
Case 2	
Computational domain (m)	$(x, z) \in [0, 1] \times [-0.75, 0.25]$
Finest grid and time step	640×640, $\tau_1 = T/4800$
Wave speed (m s^{-1})	$c = 1.25$
Wave amplitude (m)	$A = 0.02$
Gravity (m s^{-2})	$g = 9.81$
Steepening coefficient (m^{-1})	$b = 12$
Wavelength (m) and period (s)	$\lambda = 1, T = 0.8$
Molecular viscosity fluid 2 (Pa s)	$\mu_2 = 0.001$
Consistency parameter fluid 1 (Pa s^n)	$k = 0.002$
Flow index fluid 1	$n = 0.8$
Yield stress fluid 1 (Pa)	$\tau_0 = 0.002$
Regularization parameter fluid 1 (s)	$m = 4$
Density (kg m^{-3})	$\rho_1 = 1200$, $\rho_2 = 1000$
Convergence tolerance for residuals	L_∞ norm $< 10^{-10}$
Convection scheme volume-fraction equation	TVD Harmonic [30]
Case 3 (other parameters are as case 2)	
Steepening coefficient (m^{-1})	$b = 1200$
Molecular viscosity fluid 2 (Pa s)	$\mu_2 = 2 \times 10^{-5}$
Density fluid 2 (kg m^{-3})	$\rho_2 = 1$
Convection scheme volume-fraction equation	ReFRICS [31]

Appendix C: Scripts for Generation of the Source Terms

The source terms for the three verification exercises can be obtained by running the scripts Listings 1 and 2 with the computer algebra system Maxima [23].

Listing 1: Maxima [23] script to generate the source terms for case 1 in Sec. 6.

```
/* the manufactured solution */
u(x,y):=sin(x^2+y^2)+0.001;
v(x,y):=cos(x^2+y^2)+0.001;
p(x,y):=sin(x^2+y^2)+0.001;

/* shear rate */
S(x,y):=sqrt(2*(diff(u(x,y),x))^2+2*(diff(v(x,y),y))^2+(diff(u(x,y),y)+diff(v(x,y),x))^2);

/* viscosity for the regularised Herschel-Bulkley model */
mu(x,y):= (tau_0+K*S(x,y)^n )/S(x,y)*(1-exp(-m*S(x,y)));

/* body force of u-momentum equation */
q1:diff(rho*u(x,y)*u(x,y),x)+diff(rho*u(x,y)*v(x,y),y)+diff(p(x,y),x)
  -diff(mu(x,y)*(diff(u(x,y),x)+diff(u(x,y),x)\
   -2.0/3.0*(diff(u(x,y),x)+diff(v(x,y),y))),x)
  -diff(mu(x,y)*(diff(u(x,y),y)+diff(v(x,y),x)),y) ;

/* body force of v-momentum equation */
q2:diff(rho*v(x,y)*u(x,y),x)+diff(rho*v(x,y)*v(x,y),y)+diff(p(x,y),y)
  -diff(mu(x,y)*(diff(v(x,y),x)+diff(u(x,y),y)),x)
  -diff(mu(x,y)*(diff(v(x,y),y)+diff(v(x,y),y)\
   -2.0/3.0*(diff(u(x,y),x)+diff(v(x,y),y))),y) ;

/* mass source of continuity equation */
q3:diff(u(x,y),x)+diff(v(x,y),y);

/* Export to fortran */
fortran(q0);fortran(q1);fortran(q2);
```

Listing 2: Maxima [23] script used to generate the source terms for case 2 and 3 in Secs. 7 and 8. This script is taken from Ref. [31] and it has been modified to account for the Herschel-Bulkley model used for fluid 1.

```
/* the potential flow solution */
phi(x,z,t):=c*A*exp(g*z/c**2)*sin(g*x/c**2-g*t/c);
u(x,z,t):=diff(phi(x,z,t),x);
w(x,z,t):=diff(phi(x,z,t),z);
hw(x,z,t):=-1/g*diff(phi(x,z,t),t); /* wave elevation */

/* shear rate */
S(x,z,t):=sqrt(2*(diff(u(x,z,t),x))^2+2*(diff(w(x,z,t),z))^2
               +(diff(u(x,z,t),z)+diff(w(x,z,t),x))^2);

/* viscosity for the regularised Herschel-Bulkley model */
muF1(x,z,t):=(tau_0+K*S(x,z,t)^n)/(S(x,z,t))*(1-exp(-m*S(x,z,t)));

/* the volume fraction, material properties and pressure field */
c(x,z,t):=(1/2)*(1+erf(b*(z-hw(x,z,t))));
rho(x,z,t):=rhoF2*c(x,z,t)+rhoF1*(1-c(x,z,t));
mu(x,z,t):=muF2*c(x,z,t)+muF1(x,z,t)*(1-c(x,z,t));
p(x,z,t):=rho(x,z,t)*g*(hw(x,z,t)-z);

/* source term of the volume fraction equation */
q1:diff(c(x,z,t),t)+diff(u(x,z,t)*c(x,z,t),x)+diff(w(x,z,t)*c(x,z,t),z);

/* body force of x-momentum equation */
q2:diff(rho(x,z,t)*u(x,z,t),t) +diff(rho(x,z,t)*u(x,z,t)*u(x,z,t),x)
  +diff(rho(x,z,t)*u(x,z,t)*w(x,z,t),z)
  +diff(p(x,z,t),x)
  -diff(mu(x,z,t)*(diff(u(x,z,t),x)+diff(u(x,z,t),x)),x)
  -diff(mu(x,z,t)*(diff(u(x,z,t),z)+diff(w(x,z,t),x)),z) ;

/* body force of z-momentum equation */
q3:diff(rho(x,z,t)*w(x,z,t),t) +diff(rho(x,z,t)*w(x,z,t)*u(x,z,t),x)
  +diff(rho(x,z,t)*w(x,z,t)*w(x,z,t),z)
  +diff(p(x,z,t),z) + rho(x,z,t)*g
  -diff(mu(x,z,t)*(diff(w(x,z,t),x)+diff(u(x,z,t),z)),x)
  -diff(mu(x,z,t)*(diff(w(x,z,t),z)+diff(w(x,z,t),z)),z) ;

/* Export to fortran */
fortran(q1);fortran(q2);fortran(q3);
```

References

[1] Chhabra, R. P., and Richardson, J. F., 2008, *Non-Newtonian Flow and Applied Rheology*, Elsevier, Amsterdam, The Netherlands.

[2] Irgens, F., 2014, *Rheology and Non-Newtonian Fluids*, Springer International Publishing, Cham, Switzerland.

[3] Roache, P. J., 1998, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, NM, pp. 107–240.

[4] Oberkampf, W. L., and Roy, C. J., 2010, *Verification and Validation in Scientific Computing*, Cambridge University Press, Cambridge, UK.

[5] Wu, W., Huang, X., Yuan, H., Xu, F., and Ma, J., 2017, "A Modified Lattice Boltzmann Method for Herschel-Bulkley Fluids," Rheol. Acta, **56**(4), pp. 369–376.

[6] Grasinger, M., Overacker, S., and Brigham, J., 2018, "Numerical Investigation of the Accuracy, Stability, and Efficiency of Lattice Boltzmann Methods in Simulating non-Newtonian Flow," Comput. Fluids, **166**, pp. 253–274.

[7] Saramito, P., and Wachs, A., 2017, "Progress in Numerical Simulation of Yield Stress Fluid Flows," Rheol. Acta, **56**(3), pp. 211–230.

[8] Roache, P. J., 2002, "Code Verification by the Method of Manufactured Solutions," ASME J. Fluids Eng., **124**(1), pp. 4–10.

[9] Turgeon, E., and Pelletier, D., 2002, "Verification and Validation in CFD Using an Adaptive Finite-Element Method," Can. Aeronaut. Space J., **48**(4), pp. 219–231.

[10] Knupp, P., 2002, *Verification of Computer Codes in Computational Science and Engineering*, Chapman and Hall/CRC, New York.

[11] Eça, L., Hoekstra, M., Hay, A., and Pelletier, D., 2007, "Verification of RANS Solvers With Manufactured Solutions," Eng. Comput., **23**(4), pp. 253–270.

[12] Veluri, S. P., Roy, C. J., and Luke, E. A., 2012, "Comprehensive Code Verification Techniques for Finite Volume CFD Codes," Comput. Fluids, **70**, pp. 59–72.

[13] Blais, B., and Bertrand, F., 2015, "On the Use of the Method of Manufactured Solutions for the Verification of CFD Codes for the Volume-Averaged Navier-Stokes Equations," Comput. Fluids, **114**, pp. 121–129.

[14] Choudhary, A., Roy, C. J., Dietiker, J.-F., Shahnam, M., Garg, R., and Musser, J., 2016, "Code Verification for Multiphase Flows Using the Method of Manufactured Solutions," Int. J. Multiphase Flow, **80**, pp. 150–163.

[15] Eça, L., Klaij, C. M., Vaz, G., Hoekstra, M., and Pereira, F. S., 2016, "On Code Verification of RANS Solvers," J. Comput. Phys., **310**, pp. 418–439.

[16] Venkatesan, J., and Ganesan, S., 2017, "A Three-Field Local Projection Stabilized Formulation for Computations of Oldroyd-B Viscoelastic Fluid Flows," J. Non-Newtonian Fluid Mech., **247**, pp. 90–106.

[17] Kim, N., and Reddy, J., 2019, "3-D Least-Squares Finite Element Analysis of Flows of Generalized Newtonian Fluids," J. Non-Newtonian Fluid Mech., **266**, pp. 143–159.

[18] Carrozza, M., Hulsen, M., and Anderson, P., 2020, "Benchmark Solutions for Flows With Rheologically Complex Interfaces," J. Non-Newtonian Fluid Mech., **286**, p. 104436.

[19] Kim, J., and Park, J. D., 2020, "The Non-Homogeneous Flow of a Thixotropic Fluid Around a Sphere," Appl. Math. Modell., **82**, pp. 848–866.

[20] Mitsoulis, E., and Tsamopoulos, J., 2017, "Numerical Simulations of Complex Yield-Stress Fluid Flows," Rheol. Acta, **56**(3), pp. 231–258.

[21] Souza Mendes, P. R., and Dutra, E. S. S., 2004, "Viscosity Function for Yield-Stress Liquids," Appl. Rheol., **14**(6), pp. 296–302.

[22] Hirt, C., and Nichols, B., 1981, "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," J. Comput. Phys., **39**(1), pp. 201–225.

[23] Maxima, "Maxima, A Computer Algebra System, Version 5.37.2," accessed Mar. 2, 2021, https://maxima.sourceforge.io/

[24] Eça, L., Vaz, G., Toxopeus, S. L., and Hoekstra, M., 2019, "Numerical Errors in Unsteady Flow Simulations," ASME. J. Verif. Valid. Uncert., **4**(2): p. 021001.

[25] Roache, P. J., 2009, *Fundamentals of Verification and Validation*, Hermosa Publishers, Albuquerque, NM.

[26] Wesseling, P., 2001, *Principles of Computational Fluid Dynamics*, Vol. 29, Springer Berlin Heidelberg, Berlin, Germany.

[27] Vaz, G., Jaouen, F., and Hoekstra, M., 2009, "Free-Surface Viscous Flow Computations: Validation of URANS Code FRESCO," Proceedings of OMAE2009, Honolulu, HI, May 31–June 5, pp. 425–437.
[28] Miller, T. F., and Schmidt, F. W., 1988, "Use of a Pressure-Weighted Interpolation Method for the Solution of the Incompressible Navier-Stokes Equations on a Non-staggered Grid System," Numer. Heat Transfer, **14**(2), pp. 213–233.
[29] Klaij, C. M., and Vuik, C., 2013, "SIMPLE-Type Preconditioners for Cell-Centered, Colocated Finite Volume Discretization of Incompressible Reynolds-Averaged Navier-Stokes Equations," Int. J. Numer. Methods Fluids, **71**(7), pp. 830–849.
[30] van Leer, B., 1979, "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," J. Comput. Phys., **32**(1), pp. 101–136.
[31] Klaij, C., Hoekstra, M., and Vaz, G., 2018, "Design, Analysis and Verification of a Volume-of-Fluid Model With Interface-Capturing Scheme," Comput. Fluids, **170**, pp. 324–340.
[32] Ferziger, J. H., and Perić, M., 1996, *Computational Methods for Fluid Dynamics*, Springer, Berlin.
[33] Salari, K., and Knupp, P., 2000, "Code Verification by the Method of Manufactured Solutions,".
[34] Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A., and Tsamopoulos, J., 2017, "A Critical Analysis of Some Popular Methods for the Discretisation of the Gradient Operator in Finite Volume Methods," Phys. Fluids, **29**(12), p. 127103.
[35] Syrakos, A., Georgiou, G. C., and Alexandrou, A. N., 2014, "Performance of the Finite Volume Method in Solving Regularised Bingham Flows: Inertia Effects in the Lid-Driven Cavity Flow," J. Non-Newtonian Fluid Mech., **208–209**, pp. 88–107.
[36] Larsson, L., and Raven, H. C., 2010, *Ship Resistance and Flow*, Society of Naval Architects and Marine Engineers, Jersey City, NJ.
[37] Duraisamy, K., Baeder, J. D., and Liu, J. G., 2003, "Concepts and Application of Time-Limiters to High Resolution Schemes," J. Sci. Comput., **19**(1/3), pp. 139–162.
[38] Filali, A., Khezzar, L., and Mitsoulis, E., 2013, "Some Experiences With the Numerical Simulation of Newtonian and Bingham Fluids in Dip Coating," Comput. Fluids, **82**, pp. 110–121.