


## RESEARCH ARTICLE

# Simulation of two- and three-dimensional viscoplastic flows using adaptive mesh refinement

Jordi Cotela-Dalmau<sup>1,2</sup>  | Riccardo Rossi<sup>1,2</sup> | Antonia Larese<sup>1,2</sup>

<sup>1</sup>CIMNE - Centre Internacional de Metodes Numerics en Enginyeria, Campus Nord UPC, Barcelona, c/Gran Capità s.n. 08034, Spain

<sup>2</sup>Universitat Politècnica de Catalunya, UPC Barcelona Tech, Barcelona, c/Gran Capità s.n. 08034, Spain

## Correspondence

Jordi Cotela-Dalmau, CIMNE, Campus Nord UPC, Ed. B0, c/Gran Capità s.n. 08034 Barcelona, Spain.  
Email: jcotela@cimne.upc.edu

## Funding information

Spanish Ministry of Economy and Competitiveness (Ministerio de Economía y Competitividad, MINECO), Grant/Award Number: MAT2013-48624-C2-1-P; ERC Advanced grant project SAFECON, Grant/Award Number: AdG-267521; European Research Council and of the T-MAPPP project, Grant/Award Number: FP7 PEOPLE 2013 ITN-G.A.n607453

## Summary

This paper presents a finite element solver for the simulation of steady non-Newtonian flow problems, using a regularized Bingham model, with adaptive mesh refinement capabilities.

The solver is based on a stabilized formulation derived from the variational multi-scale framework. This choice allows the introduction of an a posteriori error indicator based on the small scale part of the solution, which is used to drive a mesh refinement procedure based on element subdivision.

This approach applied to the solution of a series of benchmark examples, which allow us to validate the formulation and assess its capabilities to model 2D and 3D non-Newtonian flows.

## KEYWORDS

adaptive mesh refinement, Bingham model, variational multiscale stabilization, viscoplastic fluids

## 1 | INTRODUCTION

The choice of a computational mesh is a crucial aspect of practical finite element analysis. The use of unstructured meshes allows us to obtain a higher level of detail for parts of the domain where the solution changes rapidly, or around regions of interest. At the same time, a coarser resolution can be used where the solution is smooth to reduce the computational requirements. Clearly, the use of a single *static* mesh limits the advantages of the approach for evolving solutions and requires the analyst to adequately design the mesh, guided by an intuition on the expected behavior. Indeed, a good choice is often not possible a priori, especially when studying complex geometries. This represents a particularly problematic issue in the case of viscoplastic fluids, where adequate mesh resolution is crucial to properly describe the flow and, in particular, to capture the transition between yielded and unyielded regions.

For complex problems or in cases where it is difficult to draw from previous experience, dynamically adapting the mesh may be the only viable method to reduce computational time. For this reason, a variety of adaptive mesh refinement (AMR) techniques have been developed over the years to adjust the computational mesh in response to the obtained solution (see for example Carey<sup>1</sup> or Ainsworth and Oden<sup>2</sup>).

An AMR technique typically involves 2 components: an indicator that can be used to identify regions where the obtained solution has a large error and a refinement strategy to locally modify mesh resolution in response to a large detected error. Both aspects of the problem are addressed in turn in the current paper to provide a picture of our proposed technique.

An extensive literature exists on error estimation techniques, starting from the pioneering work of Babuška and Rheinboldt.<sup>3</sup> One of the most widespread options is the use of a posteriori error estimation techniques to quantify the error in the solution (as for example, in previous studies<sup>2,4-7</sup>). The general idea in such approaches is to use the computed solution to assess its own accuracy.

The use of AMR techniques for viscoplastic flows remains comparatively unexplored. In the specific case of Bingham fluids, previous studies have applied adaptive remeshing techniques to plane and axisymmetric flows.<sup>8-10</sup> The same approach has been extended to three-dimensional free surface flows,<sup>11</sup> described using a shallow water approximation. In these cases, refinement is driven by an indicator based on the dissipative energy in the solution, which is larger in areas with high-velocity gradients, and the mesh is completely regenerated using a size distribution obtained from the values of the indicator. Another application of AMR techniques for viscoplastic materials can be found in forging processes.<sup>12</sup> In this reference, mesh refinement was used to preserve mesh quality in the simulation of metal casting.

In the current paper, we explore the use of an error indicator motivated by variational multiscale (VMS) stabilization techniques,<sup>13,14</sup> which provide the basis for the underlying finite element formulation. The forerunner of the technique used here was originally presented for convection-diffusion problems<sup>15</sup> and more recently has been extended to the Navier-Stokes equations<sup>16,17</sup> and used to solve incompressible Newtonian flow problems. The key idea for the proposed error indicator is actually rather simple: VMS solvers are based on the construction of a model for the fine scale variables (representing the part of the solution that is not resolved by the mesh) in terms of the large scale ones (the part that can be described using the finite element mesh). Since the small scale model is designed to provide consistent solutions, it should tend to zero as the mesh is refined. Thus, the small scale can be used to construct an error indicator: If, for a given element or patch of elements, the magnitude of the small scale solution is large, the mesh is likely to be relatively coarse for that region of the model. This provides an indicator that can be used to identify areas in need of a refinement.

Such approach is closely related to the solver technology being used. Crucially, it can be derived naturally from any VMS formulation once it has been defined for the problem at hand. As such, it can be applied to the case of nonlinear rheologic behavior, which is the focus in the current work, by extending the mentioned Newtonian version of the method. To our knowledge, the approach we describe has not been applied before in this context.

Once we are able to identify areas with large error within the domain, we can modify the computational mesh to increase the overall accuracy. Many different grid modification strategies have been proposed since the late eighties,<sup>18-20</sup> involving for example local mesh modifications,<sup>21</sup> movement of mesh nodes through the solution of an auxiliary elasticity problem,<sup>22</sup> mesh morphing techniques,<sup>23</sup> or edge stretching.<sup>24</sup> Many of such techniques require the user to specify a nodal or edge-based refinement target, which involves obtaining a measure of the error (or its indicator) at the nodes or edges of the mesh to be improved.

On the contrary, the technique we use here is based on an elemental (rather than nodal) refinement strategy.<sup>16,25</sup> The approach followed is thus that the error indicator signals the list of elements to be refined. Such elements are subdivided and their neighbor modified so that mesh conformity is preserved. The approach thus represents a good match for the chosen error indicator, which can be evaluated on each element independently. The proposed procedure was originally developed for refinement in a distributed memory context<sup>16</sup> and allows MPI-parallel refinement.

The present paper is organized as follows: In Section 2, the VMS finite element formulation used for the solution of the stationary Bingham flow problem is introduced. Section 3 describes the mesh refinement algorithm and the error indicator used to identify the regions that will be refined. We present the results of the simulation of several two- and three-dimensional Bingham flow test cases in Section 4, which allow us to evaluate the capabilities of the present approach. Finally, some concluding remarks are given in Section 5.

## 2 | NON-NEWTONIAN FLOW SOLVER

### 2.1 | Governing equations

The problems considered in the present work can be described using the stationary form of the Navier-Stokes equations. Given a domain  $\Omega \subset \mathbb{R}^d$  (with  $d = 2, 3$ ) containing a fluid with density  $\rho$  and external forces  $\mathbf{f}$  acting on it, the relationship between the fluid stresses  $\boldsymbol{\sigma}$  and the velocities  $\mathbf{u}$  is given by

$$\rho \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega. \quad (2)$$

Equations 1 and 2 are combined with conditions over the boundary of the domain  $\Omega$ , which is defined as  $\partial\Omega = \Gamma_D \cup \Gamma_N$ , where  $\Gamma_D$  and  $\Gamma_N$  represent the Dirichlet and Neumann parts of the boundary, respectively. Such boundary conditions can be expressed as

$$\mathbf{u} = \mathbf{u}_0 \quad \text{on } \Gamma_D \quad (3)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} \quad \text{on } \Gamma_N, \quad (4)$$

where  $\mathbf{u}_0$  is the imposed velocity on the Dirichlet boundary  $\Gamma_D$ ,  $\mathbf{n}$  is the outwards unit normal on the Neumann boundary  $\Gamma_N$ , and  $\mathbf{t}$  is the imposed traction.

The stress tensor in Equation 1 can be decomposed into a volumetric part, which depends on the pressure,  $p$ , and the deviatoric stress tensor,  $\boldsymbol{\tau}$ , as

$$\boldsymbol{\sigma} = -p \mathbf{I} + \boldsymbol{\tau}, \quad (5)$$

where  $\mathbf{I}$  is the second-order identity tensor.

A constitutive model is required to close the formulation, giving an expression for the deviatoric stresses  $\boldsymbol{\tau}$ . The behavior of a broad class of fluids can be described using a generalized Newtonian law with apparent viscosity  $\eta$ , given by

$$\boldsymbol{\tau} = 2\eta\mathbf{S}, \quad (6)$$

where the apparent viscosity is, in general, variable and depends on the characteristics of the flow, and  $\mathbf{S}$  is the strain rate tensor, defined as

$$\mathbf{S} = \nabla^s \mathbf{u} = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T). \quad (7)$$

Since stress and strain rate are tensorial quantities, we introduce the following invariants to measure their magnitude

$$\dot{\gamma} = \sqrt{2 \mathbf{S} : \mathbf{S}} \quad \bar{\tau} = \sqrt{\frac{1}{2} \boldsymbol{\tau} : \boldsymbol{\tau}}, \quad (8)$$

which allow us to use the constitutive relation of Equation 6 in a multidimensional setting.

In the present work, we use a regularized Bingham model to describe the flow. A Bingham fluid<sup>26,27</sup> is a material that resists shear stresses lower than a threshold value, known as the yield stress  $\tau_0$ , and flows with a constant viscosity  $\mu_p$ , called plastic viscosity, when larger shear stresses are applied. This behavior can be represented using the notation of Equation 6 if the apparent viscosity is defined as

$$\begin{aligned} \eta &\rightarrow \infty && \text{if } \bar{\tau} < \tau_0 \\ \eta &= \mu_p + \frac{\tau_0}{\dot{\gamma}} && \text{if } \bar{\tau} \geq \tau_0. \end{aligned} \quad (9)$$

Unfortunately, the discontinuous nature of the Bingham model introduces numerical difficulties, as the apparent viscosity  $\eta$  is infinite for strain rates lower than the yield stress. Two possible approaches have been explored in the literature to address this issue. First is to reformulate the problem as a variational inequality<sup>28-30</sup> and to use either operator splitting methods<sup>31-34</sup> or an augmented Lagrangian approach<sup>35,36</sup> to solve the problem. Second is the introduction of a regularized constitutive relation.<sup>37-41</sup> We use the latter technique, replacing Equation 9 with a regularized expression that is better suited to numerical simulation. We have adopted the expression proposed by Papanastasiou,<sup>39</sup> given by

$$\eta = \mu_p + \frac{\tau_0}{\dot{\gamma}} (1 - e^{-m\dot{\gamma}}), \quad (10)$$

where  $m$  is a regularization coefficient with dimensions of time. The relationship between the 2 laws is shown in Figure 1, where it can be seen that the regularized law of Equation 10, represented by the dotted lines, tends to the ideal model of Equation 9 as the value of the regularization coefficient  $m$  is increased.

It must be remarked that the use of a regularized model introduces an error in the solution, which may be significant in determining, for example, if a particular problem is stable (the solution is static) or yields, developing a flowing solution. This was quantified in Frigaard and Nouar,<sup>42</sup> which addresses the Papanastasiou regularization, among others. This effect is magnified for small values of the regularization coefficient  $m$  and is at odds with the fact that large values of  $m$  result in a problem that is harder to solve numerically. As such, one must be careful in choosing a value for  $m$  to avoid compromising the obtained solution.

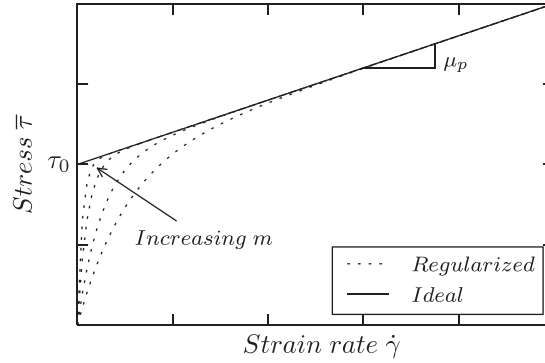


FIGURE 1 Bingham model

## 2.2 | Weak formulation

We use the standard Galerkin approach to obtain a variational form of our problem, multiplying Equations 1 and 2 by test functions  $\mathbf{w}$ ,  $q$  and integrating over the simulation domain  $\Omega$ . If we integrate by parts the terms corresponding to stresses, we can incorporate the Neumann boundary condition of Equation 4 in the expression, obtaining

$$\int_{\Omega} \mathbf{w} \rho \mathbf{u} \cdot \nabla \mathbf{u} \, d\Omega + \int_{\Omega} 2\eta \nabla^s \mathbf{w} : \nabla^s \mathbf{u} \, d\Omega - \int_{\Omega} \nabla \cdot \mathbf{w} p \, d\Omega = \int_{\Omega} \mathbf{w} \mathbf{f} \, d\Omega + \int_{\Gamma_N} \mathbf{w} \mathbf{t} \, d\Gamma \quad (11)$$

$$\int_{\Omega} q \nabla \cdot \mathbf{u} \, d\Omega = 0. \quad (12)$$

To ensure that the problem given by Equations 11 and 12 is well posed, we need to specify appropriate function spaces for the unknowns  $\mathbf{u}$ ,  $p$  and test functions  $\mathbf{w}$ ,  $q$  to guarantee that all integrals are bounded. In the case of velocities, we require that both the functions and their spatial derivatives are square-integrable, that is,  $\mathbf{u} \in V_D \subset [H^1(\Omega)]^d$ . Furthermore, we require that all functions in  $V_D$  verify the Dirichlet boundary conditions of the problem, given by Equation 3. A similar requirement is made on the momentum test functions  $\mathbf{w}$ , which have to be zero on the Dirichlet boundary  $\Gamma_D$ . This is denoted as  $\mathbf{w} \in V_0 \subset [H^1(\Omega)]^d$ . In the case of the pressure and the continuity equation test function  $q$ , it is enough to ensure that they are square-integrable in the domain, which is denoted by  $p, q \in Q \subset L_2(\Omega)$ .

The finite element solution of the problem defined by Equations 11 and 12 is known to suffer from stability issues that, in particular, preclude the use of equal-order interpolations for velocity and pressure. To solve this problem, we introduce a VMS formulation.<sup>13,14</sup>

Denoting the finite element discretization of the problem domain as  $\Omega_h = \cup_e \Omega_e$ , where  $\Omega_e$  represents the subdomain corresponding to element  $e$ , we introduce the discrete solution of the problem  $\mathbf{u} \in V_h, p \in Q_h$ , where  $V_h \subset V_D$  and  $Q_h \subset Q$  are the discrete spaces defined by the finite element interpolation. The VMS approach is based on decomposing the problem variables into a large scale part, identified with the finite element solution, and a small scale part

$$\mathbf{u} = \mathbf{u} + \tilde{\mathbf{u}} \quad p = p + \tilde{p}, \quad (13)$$

where the small scale variables  $\tilde{\mathbf{u}}$ ,  $\tilde{p}$  represent the part of the continuous solution that is not resolved by the finite element approximation.

The scale separation of Equation 13 can be introduced in Equations 11 and 12 to obtain the VMS formulation of the problem. Using test functions belonging to the discrete spaces  $\mathbf{w} \in V_{h0} \subset V_0$  and  $q \in Q_h$ , we obtain

$$\int_{\Omega} \mathbf{w}_h \rho \mathbf{u}_h \cdot \nabla \mathbf{u}_h \, d\Omega + \int_{\Omega} 2\eta \nabla^s \mathbf{w}_h : \nabla^s \mathbf{u}_h \, d\Omega - \int_{\Omega} \nabla \cdot \mathbf{w}_h p_h \, d\Omega - \sum_e \int_{\Omega_e} \rho \mathbf{u}_h \cdot \nabla \mathbf{w}_h \tilde{\mathbf{u}} \, d\Omega \quad (14)$$

$$- \sum_e \int_{\Omega_e} \nabla \cdot (2\eta \nabla^s \mathbf{w}_h) \tilde{\mathbf{u}} \, d\Omega - \sum_e \int_{\Omega_e} \nabla \cdot \mathbf{w}_h \tilde{p} \, d\Omega = \int_{\Omega} \mathbf{w}_h \mathbf{f} \, d\Omega + \int_{\Gamma_N} \mathbf{w}_h \mathbf{t} \, d\Gamma \quad (15)$$

$$\int_{\Omega} q_h \nabla \cdot \mathbf{u} = \sum_e \int_{\Omega_e} \nabla q_h \tilde{\mathbf{u}} \, d\Omega.$$

In deriving Equations 14 and 15 from Equations 11 and 12, we have integrated by parts on elemental subdomains  $\Omega_e$  for several terms involving the small scale variables  $\mathbf{u}$  and  $p$ . This allows us to obtain an expression that depends on small scale values but not on small scale spatial derivatives.

Note that, to do so, we have neglected terms involving elemental boundary integrals. This can be understood as considering that the small scale functions vanish on the elemental boundaries (see for example reference herein<sup>43</sup> for a detailed derivation).

A model for the small scale part of the solution is required to close the formulation. A possibility is to use algebraic subgrid scales (ASGS), defined in Codina.<sup>44</sup> In ASGS, the small scales are considered proportional to the residual of the strong form of the momentum and continuity equations, when they are evaluated using the discrete solution. This results in the following expression for the small scales:

$$\tilde{\mathbf{u}} = \tau_1 \mathbf{R}^m(\mathbf{u}_h, p_h) = \tau_1 (\mathbf{f} - \rho \mathbf{u}_h \cdot \nabla \mathbf{u} + \nabla \cdot 2\eta \nabla^s \mathbf{u}_h - \nabla p_h) \quad (16)$$

$$p = \tau_2 R^c(\mathbf{u}_h) = \tau_2 (-\nabla \cdot \mathbf{u}_h), \quad (17)$$

where  $\mathbf{R}^m(\mathbf{u}_h, p_h)$  and  $R^c(\mathbf{u}_h)$  correspond to the residual forms of Equations 1 and 2 evaluated using the finite element solution  $\mathbf{u}_h, p_h$ , and  $\tau_1$  and  $\tau_2$  are scalar stabilization parameters. The value of  $\tau_1$  and  $\tau_2$  is chosen following,<sup>45,46</sup> that is, we take the parameters originally designed for the Newtonian version of the method<sup>44</sup> and replace the Newtonian viscosity by the apparent viscosity  $\eta$ . This can be seen as equivalent to neglecting the dependence of  $\eta$  on the velocity to design the stabilization parameters. Following this approach, the stabilization parameters are defined in terms of a characteristic element length  $h$  as

$$\tau_1 = \left( \frac{2\rho \|\mathbf{u}_h\|}{h} + \frac{4\eta}{h^2} \right)^{-1} \quad \tau_2 = \eta + \frac{\rho \|\mathbf{u}_h\| h}{2}, \quad (18)$$

where  $\|\cdot\|$  is used to denote the standard Euclidean vector norm.

An alternative choice for the small scales is the use of orthogonal subgrid scales (OSS).<sup>43,46-49</sup> In the OSS formulation, only the part of the large scale residuals that is orthogonal to the finite element space is used to model the small scale variables. This can be expressed as

$$\tilde{\mathbf{u}} = \tau_1 (\mathbf{f} - \rho \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla \cdot 2\eta \nabla^s \mathbf{u}_h - \nabla p_h - \boldsymbol{\pi}^m) \quad (19)$$

$$\tilde{p} = \tau_2 (-\nabla \cdot \mathbf{u}_h - \pi^c), \quad (20)$$

where  $\boldsymbol{\pi}^m(\mathbf{u}_h, p_h)$  and  $\pi^c(\mathbf{u}_h)$  are the  $L_2$  projections of the residuals onto the finite element space, that is, the solution of the auxiliary projection problem

$$\int_{\Omega} \mathbf{w}_h \boldsymbol{\pi}^m \, d\Omega = \int_{\Omega} \mathbf{w}_h (\mathbf{f} - \rho \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla \cdot 2\eta \nabla^s \mathbf{u}_h - \nabla p_h) \, d\Omega \quad (21)$$

$$\int_{\Omega} q_h \pi^c \, d\Omega = - \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h \, d\Omega. \quad (22)$$

It is worth mentioning that the viscous term  $\nabla \cdot 2\eta \nabla^s \mathbf{u}_h$  that appears in Equations 16 and 19, as well as in the momentum projection in Equation 21, is zero on element interiors when evaluated using linear finite elements, since it involves second derivatives of piecewise linear functions. Since all examples in the present work use linear elements, this term will be neglected in the following.

Introducing the OSS small scale model into the variational form of the problem, given by Equations 14 and 15, the following weak formulation is obtained

$$\begin{aligned} & \int_{\Omega} \rho \mathbf{w}_h \mathbf{u}_h \nabla \mathbf{u}_h \, d\Omega + \int_{\Omega} 2\eta \nabla^s \mathbf{w}_h : \nabla^s \mathbf{u}_h \, d\Omega - \int_{\Omega} \nabla \cdot \mathbf{w}_h p_h \, d\Omega \\ & + \sum_e \int_{\Omega_e} \rho \mathbf{u}_h \nabla \mathbf{w}_h \tau_1 (\rho \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla p_h) \, d\Omega - \sum_e \int_{\Omega_e} \nabla \cdot \mathbf{w}_h \tau_2 (-\nabla \cdot \mathbf{u}_h - \Pi^c) \, d\Omega \end{aligned} \quad (23)$$

$$\begin{aligned} & = \int_{\Omega} \mathbf{w}_h \mathbf{f} \, d\Omega + \int_{\Gamma_N} \mathbf{w}_h \mathbf{t} \, d\Gamma + \sum_e \int_{\Omega_e} \rho \mathbf{u}_h \nabla \mathbf{w}_h \tau_1 (\mathbf{f} - \Pi^m) \, d\Omega \\ & \int_{\Omega} q_h \nabla \cdot \mathbf{u} + \sum_e \int_{\Omega_e} \nabla q_h \tau_1 (\rho \mathbf{u}_h \nabla \mathbf{u}_h + \nabla p_h) \, d\Omega = \sum_e \int_{\Omega_e} \nabla q_h \tau_1 (\mathbf{f} - \Pi^m) \, d\Omega. \end{aligned} \quad (24)$$

Similarly, the ASGS formulation can be recovered by dropping all terms involving  $\Pi^m$  and  $\Pi^c$  in Equations 23 and 24.

As a final remark, note that VMS represents a general framework to write stabilized weak formulations. In this sense, the final weak problems engendered by both the ASGS and OSS can be seen as encompassing other well-known stabilization techniques such as SUPG/PSPG<sup>50</sup> or Galerkin-Least Squares.<sup>51</sup>

### 2.3 | Matrix formulation

We use standard linear finite element functions to interpolate the large scale velocity and pressure solutions. Introducing this interpolation in Equations 23 and 24 and using the shape functions of each node as test functions  $\mathbf{w}_h, p_h$  in succession, we obtain the following system of equations

$$\begin{bmatrix} \mathbf{C} + \mathbf{K} + \mathbf{S}_K & \mathbf{G} + \mathbf{S}_G \\ \mathbf{D} + \mathbf{S}_D & \mathbf{L} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} + \mathbf{S}_M \\ \mathbf{S}_C \end{bmatrix}, \quad (25)$$

where  $\mathbf{U}$  and  $\mathbf{P}$  represent the vectors of nodal values for velocity and pressure, respectively. The blocks that appear in the system matrix and the right hand side vector of Equation 25 are obtained by the finite element assembly of the different integrals that appear in the stabilized equations. If  $a$  and  $b$  represent node indices,  $i, j$  correspond to the spatial directions and  $N_a$  is the finite element function associated with node  $a$ , the Galerkin terms in Equations 23 and 24 give rise to the the following block matrices

$$\begin{aligned} \mathbf{C}_{ij}^{ab} &= \int_{\Omega_e} N_a \left( \rho u_{h k} \frac{\partial N_b}{\partial x_k} \right) \delta_{ij} \, d\Omega & \mathbf{K}^{ab} &= \int_{\Omega_e} \mathbf{B}_a^T \mathbf{C}_\eta \mathbf{B}_b \, d\Omega \\ \mathbf{G}_i^{ab} &= - \int_{\Omega_e} \frac{\partial N_a}{\partial x_i} N_b \, d\Omega & \mathbf{D}_j^{ab} &= \int_{\Omega_e} N_a \frac{\partial N_b}{\partial x_j} \, d\Omega \\ \mathbf{F}_i^a &= \int_{\Omega_e} N_a f_i \, d\Omega + \int_{\Gamma_N} N_a t_i \, d\Gamma, \end{aligned} \quad (26)$$

where  $\delta_{ij}$  corresponds to the Kronecker delta. To define the viscous matrix  $\mathbf{K}^{ab}$ , we used the strain rate matrix  $\mathbf{B}$  and the constitutive matrix  $\mathbf{C}_\eta$ , which in 2D are given by

$$\mathbf{B}_a = \begin{bmatrix} \partial_x N_a & 0 \\ 0 & \partial_y N_a \\ \partial_y N_a & \partial_x N_a \end{bmatrix} \quad \mathbf{C}_\eta = \eta \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (27)$$

and in 3D can be defined as

$$\mathbf{B}_a = \begin{bmatrix} \partial_x N_a & 0 & 0 \\ 0 & \partial_y N_a & 0 \\ 0 & 0 & \partial_z N_a \\ \partial_y N_a & \partial_x N_a & 0 \\ 0 & \partial_z N_a & \partial_y N_a \\ \partial_z N_a & 0 & \partial_x N_a \end{bmatrix} \quad \mathbf{C}_\eta = \eta \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (28)$$

where we introduced the notation  $\partial_{(\cdot)} N_a = \partial N_a / \partial (\cdot)$  for compactness.

Analogously, the discretization of the terms arising from the VMS stabilization allows us to write

$$\begin{aligned} \mathbf{S}_{K_{ij}^{ab}} &= \int_{\Omega_e} \left( \rho u_{h k} \frac{\partial N_a}{\partial x_k} \right) \tau_1 \left( \rho u_{h l} \frac{\partial N_b}{\partial x_l} \right) \delta_{ij} \, d\Omega + \int_{\Omega_e} \frac{\partial N_a}{\partial x_i} \tau_2 \frac{\partial N_b}{\partial x_j} \, d\Omega \\ \mathbf{S}_{G_i^{ab}} &= \int_{\Omega_e} \left( \rho u_{h k} \frac{\partial N_a}{\partial x_k} \right) \tau_1 \frac{\partial N_b}{\partial x_i} \, d\Omega \\ \mathbf{S}_{D_j^{ab}} &= \int_{\Omega_e} \frac{\partial N_a}{\partial x_j} \tau_1 \left( \rho u_{h k} \frac{\partial N_b}{\partial x_k} \right) \, d\Omega \\ \mathbf{L}^{ab} &= \int_{\Omega_e} \frac{\partial N_a}{\partial x_k} \tau_1 \frac{\partial N_b}{\partial x_k} \, d\Omega \\ \mathbf{S}_{M_i^a} &= \int_{\Omega_e} \left( \rho u_{h k} \frac{\partial N_a}{\partial x_k} \right) \tau_1 (f_i - \Pi_i^m) \, d\Omega - \int_{\Omega_e} \frac{\partial N_a}{\partial x_i} \tau_2 \Pi^c \, d\Omega \\ \mathbf{S}_{C^a} &= \int_{\Omega_e} \frac{\partial N_a}{\partial x_k} \tau_1 (f_k - \Pi_k^m) \, d\Omega. \end{aligned} \quad (29)$$

The system of Equation 25 contains multiple nonlinear terms: Besides the convective term, terms involving either the apparent viscosity  $\eta$ , the stabilization parameters  $\tau_1$ ,  $\tau_2$ , and, for OSS, the projections, all have a nonlinear dependency on velocity. The problem is linearized using a fixed-point iteration.

In the case of OSS stabilization, an associated problem is solved to calculate the projections. This problem corresponds to the discrete form of Equations 21 and 22, and can be defined as

$$\begin{bmatrix} \mathbf{M}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_C \end{bmatrix} \begin{bmatrix} \Pi_M \\ \Pi_C \end{bmatrix} = \begin{bmatrix} \mathbf{R}_M \\ \mathbf{R}_C \end{bmatrix}, \quad (30)$$

where  $\Pi_M$  and  $\Pi_C$  represent the vectors of nodal values for the momentum and mass projections, respectively, and the different terms in the matrix and right hand side vector can be obtained by the finite element assembly of local contributions given by

$$\begin{aligned} \mathbf{M}_{M_{ij}^{ab}} &= \int_{\Omega_e} N_a N_b \delta_{ij} \, d\Omega & \mathbf{R}_{M_i^a} &= \int_{\Omega_e} N_a \left( f_i - \rho u_h^k \frac{\partial u_h^i}{\partial x_k} - \frac{\partial p}{\partial x_i} \right) \, d\Omega \\ \mathbf{M}_{C^{ab}} &= \int_{\Omega_e} N_a N_b \, d\Omega & \mathbf{R}_{C^a} &= \int_{\Omega_e} N_a \left( -\frac{\partial u_h^i}{\partial x_i} \right) \, d\Omega. \end{aligned} \quad (31)$$

Note that both nonzero matrix blocks on the left hand side of Equation 30 have the structure of a mass matrix. In practice, they are approximated by a diagonal mass matrix, which allows us to obtain the projections through the solution of a diagonal linear system.

### 3 | MESH REFINEMENT STRATEGY

#### 3.1 | Error estimation

The mesh refinement strategy is driven by an error indicator, which identifies regions of the simulation domain where the mesh resolution has to be increased. We use an error indicator based on the scale separation introduced by VMS models, following the ideas of Hauke et al.<sup>15</sup> This approach, originally presented for the convection-diffusion problem, has been extended to the Navier-Stokes equations in Rossi et al.<sup>16</sup> and Hauke et al.<sup>17</sup>

The main idea behind this approach is to use the magnitude of the small scale velocity model, which, in our case, is given by either Equation 16 for ASGS or Equation 19 for OSS as the error indicator. Since the small scale velocity is ultimately proportional to the residual form of the original momentum equation (or its projection onto the mesh), its magnitude can be interpreted as a measure of the error in the solution. With this in mind, we define our indicator of the error within element  $e$  as

$$(I_e)^2 = \|\tilde{\mathbf{u}}\|_{\Omega_e}^2 = \int_{\Omega_e} \tilde{\mathbf{u}} \cdot \tilde{\mathbf{u}} \, d\Omega. \quad (32)$$

Introducing the definition of the small scale velocity in Equation 32, this expression can be particularized for the considered stabilized formulations as

$$(I_e^{ASGS})^2 = \int_{\Omega_e} \tau_1^2 \|\mathbf{R}^m\|^2 \, d\Omega \quad (33)$$

$$(I_e^{OSS})^2 = \int_{\Omega_e} \tau_1^2 \|\mathbf{R}^m - \Pi^m\|^2 \, d\Omega. \quad (34)$$

The indicator thus obtained is derived from the small scale model used in the stabilization and, on the element level, is proportional to the local residual, weighted by the square of the stabilization parameter  $\tau_1$ . In the OSS variant, the indicator of Equation 34 involves the difference between the local residual and the projected residual. This idea of comparing local and smooth fields for the same quantity can be related to the error estimator of Zienkiewicz and Zhu.<sup>52</sup>

By measuring  $I_e$  on all elements, we obtain a distribution of the error indicator across the domain. We identify the elements where the estimate is larger than a prescribed tolerance, which are then split according to the algorithm described in the following section.

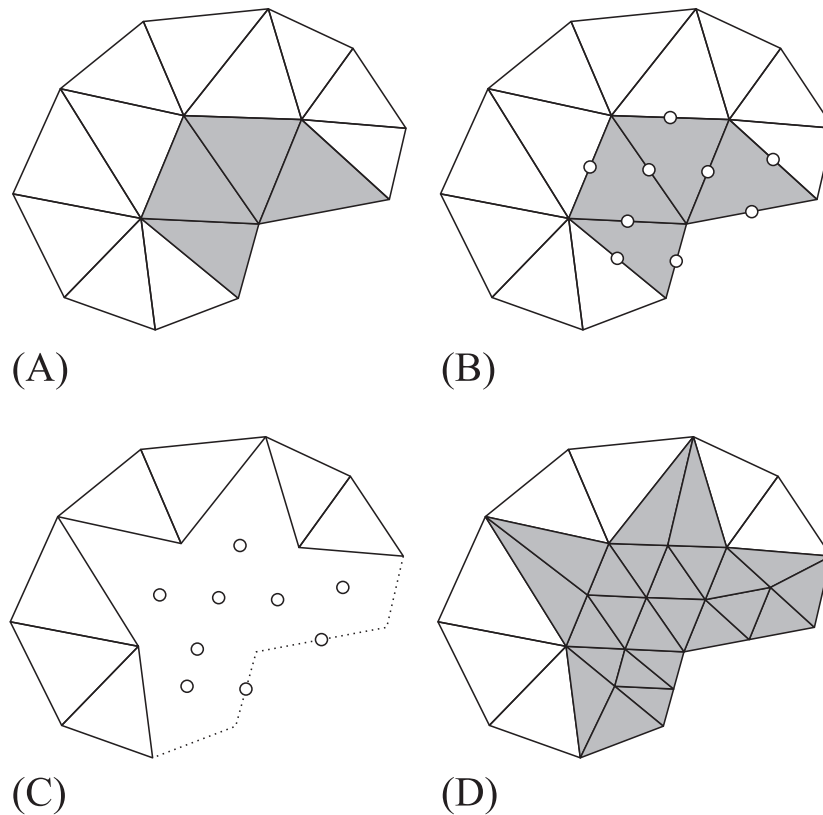


### 3.2 | Refinement algorithm

Once the elements that have larger error have been identified, we use a mesh refinement algorithm to improve mesh resolution. We have adopted the algorithm presented in et al,<sup>16</sup> which can be used to improve the resolution of an existing triangular or tetrahedral mesh. This refinement strategy was originally developed to be easily applicable in a distributed memory environment and is designed to require a minimal amount of information that is not local to the elements to be refined.

Given a triangular mesh, the idea of the procedure is to refine the identified elements by introducing new nodes on the midpoints of its edges and quadrisectioning each original triangle. Neighboring elements are then divided to maintain mesh consistency: Those that have one refined edge are bisected along that edge, while elements that have 2 refined edges are divided into 3 new triangles. The same procedure can be applied to tetrahedral meshes, with fully refined elements being divided into 8 new tetrahedra and adjacent elements refined accordingly. The main steps of the refinement procedure are illustrated by Figure 2 for a 2D case and can be described as follows:

1. Iterate over mesh elements, evaluating the error indicator given by Equation 33 for ASGS or that of Equation 34 for OSS.
2. If the error indicator is larger than a predefined tolerance in a given element, mark it and its edges as *needing refinement*. This step is represented by Figure 2A.
3. A new node is created in the midpoint of all the edges that have been identified as *needing refinement*, as shown in Figure 2B. All nodal data and, in particular, initial guesses for velocity and pressure are interpolated from the nodes that define the original edge.
4. All elements with refined edges are deleted (see Figure 2C). Note that this includes elements where the error indicator was not larger than the tolerance.
5. New elements are created using predefined patterns, depending on which and how many edges of the original element were split. This step is represented in Figure 2D. While this step is relatively straightforward in the 2D case, it is more involved for 3D problems, where the number of possible edge splitting patterns is much larger. We direct the interested reader to the original reference on this algorithm<sup>16</sup> for details. Note that, in some cases, 3D refinement involves the creation of an extra node on the center of the original tetrahedron.



**FIGURE 2** Refinement procedure: A, Identify elements to refine; B, divide edges and insert new nodes; C, remove all elements with split edges; D, create new elements to recover a conforming mesh.



Note that the refinement procedure preserves the quality of fully refined elements, since the new elements have the same angles as the original one. Unfortunately, this is not the case for elements where only some of the edges are refined, and mesh quality can be significantly degraded if this type of partial refinement is performed repeatedly over the same patch of elements.

There are multiple techniques that can alleviate such issue. Mesh quality could be trivially preserved by only allowing complete refinement (quadrisection) of elements, but maintaining conformance under such requirement would imply refining the entire domain. It is possible to relax this by introducing nonconforming meshes with hanging nodes, but this typically requires constraining successive refinements so that 2 neighboring elements do not differ by more than one refinement level.

Alternatively, one could use a different algorithm for marking and refining elements (see for example Dörfler and Nochetto et al.<sup>53,54</sup>) that can limit the mesh degradation produced by successive refinement. While such alternatives are viable and well documented in the literature, they require comparatively more information about its neighbors. Given that this is not always easily available in a distributed memory environment, which was the main consideration when designing the used algorithm,<sup>16</sup> they will not be used here.

In the present paper, we follow a different strategy: We allow repeated partial refinement but apply local improvement strategies after each refinement iteration to improve the quality of the final mesh. For the 2D examples presented in Section 4, we use the algorithm described in Frey and George,<sup>55</sup> which is based on reconnecting the nodes of adjacent elements. For the 3D cases, we use the procedure presented in D'Amato and Vénere,<sup>56</sup> which consists in taking clusters of adjacent elements and changing the local topology, adding or deleting nodes, and reconnecting elements to improve mesh quality.

## 4 | NUMERICAL RESULTS

The formulation presented in the previous pages has been implemented within the Kratos multiphysics framework for finite element solvers.<sup>57</sup> We use this implementation to simulate several benchmark problems with the intention of evaluating the capabilities of the adaptive refinement techniques. The same finite element formulation, without mesh refinement, has been used by the authors in combination with the particle finite element method to simulate non-Newtonian flows.<sup>58-60</sup> The choice of parameters, including both the regularization parameter  $m$  and problem-specific values, such as load increments, are based on the previous experience of the authors on these or similar problems (see Larese<sup>45</sup>). The linear systems of equations that appear in the solution procedure have been solved using the AMGCL algebraic multigrid solver.<sup>61</sup>

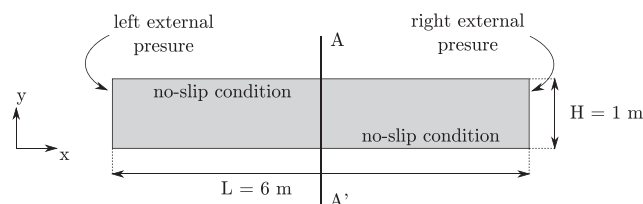
### 4.1 | Poiseuille flow

The first test case is a simple Poiseuille flow under an imposed pressure gradient. We define a  $6 \times 1m$  plane channel, shown in Figure 3, and prescribe a pressure variation  $\Delta p = -2 \times 10^3 \text{Pa}$  between its right and left extremes. A no-slip condition is imposed along the remaining sides of the channel. The fluid density is  $\rho = 1 \text{Kg}/m^3$  while the plastic viscosity is  $\mu_p = 10 \text{Pa} \cdot s$  and the regularization coefficient is set to  $m = 10^3 s$ .

We consider 2 cases: a Bingham flow with yield stress  $\tau_0 = 100 \text{Pa}$  and a Newtonian case, recovered by taking a viscosity  $\eta = \mu_p$  in Equation 6.

We start the simulation using the unstructured mesh shown in Figure 4, containing 66 nodes and 92 triangular elements. The problem is solved iteratively: The solution of the flow problem is followed by the mesh refinement algorithm, and this procedure is repeated until the error indicator is smaller than a fixed tolerance for all elements in the mesh.

We simulate a series of cases with absolute indicator tolerances in the range  $10^{-3} - 10^{-6} m^2 s^{-1}$  to study the sensitivity of the proposed approach to the tolerance for the value of the error indicator, testing both the ASGS and OSS approaches. The number of elements obtained in each case is shown in Figure 5, while the velocity profiles on the central transversal section of the domain, which corresponds to section A–A' in Figure 3, are shown in Figure 6 for the Newtonian fluid test and in Figure 7 for



**FIGURE 3** Poiseuille flow: geometry and boundary conditions

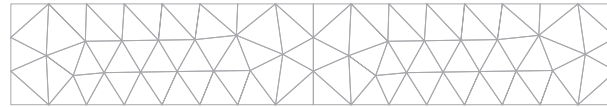


FIGURE 4 Poiseuille flow: initial mesh

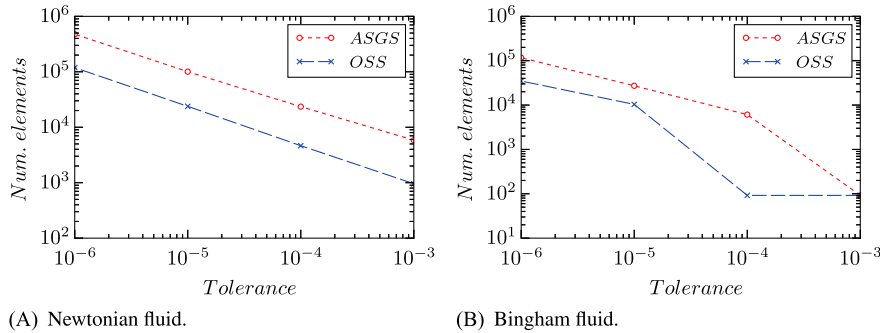


FIGURE 5 Poiseuille flow: number of elements at the end of the simulation for different error indicator tolerances. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales

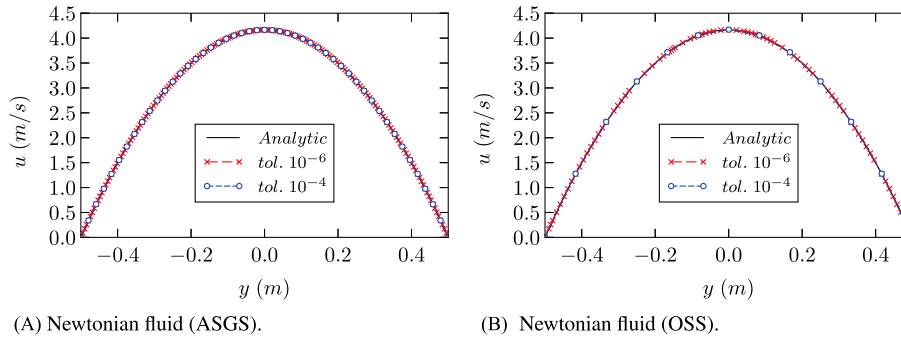


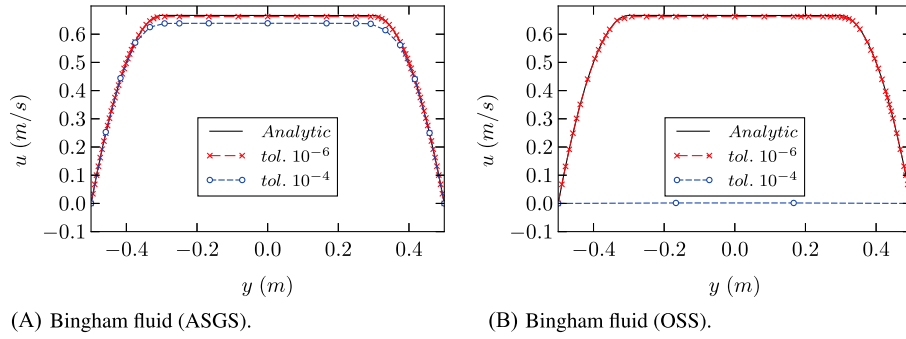
FIGURE 6 Newtonian Poiseuille flow: streamwise velocity profiles for different error indicator tolerances. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales

the Bingham flow. In both cases, the obtained results are compared to the analytic solution of the Poiseuille flow, which is given by Equation 35 (see for example Moreno<sup>62</sup>).

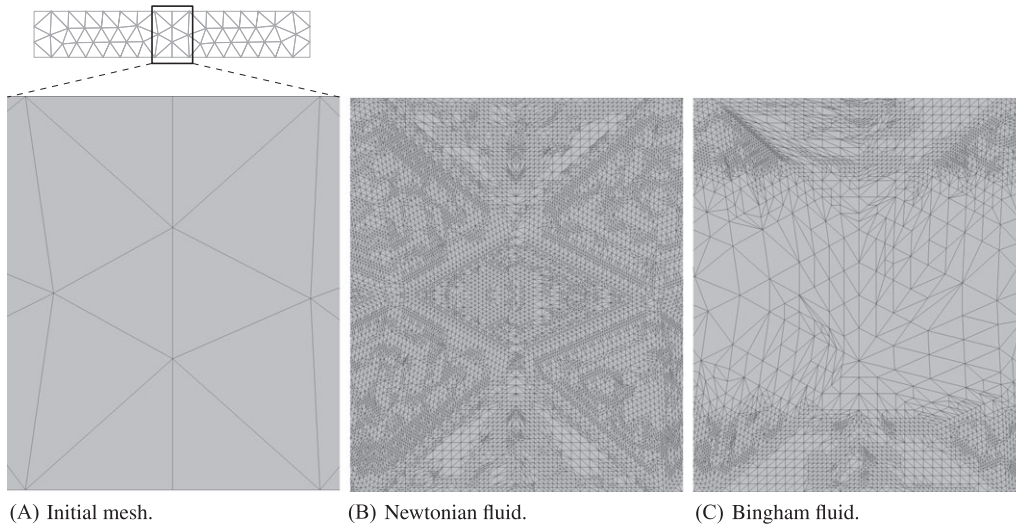
$$u_x(y) = \begin{cases} \frac{1}{2\mu_p} \left(-\frac{\Delta p}{\Delta x}\right) \left( \left(h_1 + \frac{H}{2}\right)^2 - (h_1 - y)^2 \right) & \text{if } -\frac{H}{2} \leq y < h_1 = -\tau_0 \left(-\frac{\Delta p}{\Delta x}\right)^{-1} \\ \frac{1}{2\mu_p} \left(-\frac{\Delta p}{\Delta x}\right) h_1^2 & \text{if } h_1 \leq y < h_2 = \tau_0 \left(-\frac{\Delta p}{\Delta x}\right)^{-1} \\ \frac{1}{2\mu_p} \left(-\frac{\Delta p}{\Delta x}\right) \left( \left(\frac{H}{2} - h_2\right)^2 - (y - h_2)^2 \right) & \text{if } h_2 \leq y \leq \frac{H}{2} \end{cases} \quad (35)$$

$$u_y = 0$$

In the Newtonian case, the solution reduces to a parabolic profile while, in the Bingham case, the flow has 2 yielded regions close to the wall, where shear stresses are concentrated, and a central region that moves rigidly. This central region has a constant streamwise velocity and can be solved with small error using relatively coarse meshes. As a result, refinement is concentrated close to the walls in the Bingham case and the overall number of elements is lower, as can be observed in Figure 5B. This is confirmed by Figure 8, which shows a detail of the final mesh obtained using the lowest tolerance and the OSS formulation for both Newtonian and Bingham fluids. As can be seen in the figure, the spatial distribution of the refinement is different in both cases, uniform for the Newtonian fluid and concentrated close to the walls, corresponding to the yielded regions, in the Bingham case.



**FIGURE 7** Bingham Poiseuille flow: streamwise velocity profiles for different error indicator tolerances. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales



**FIGURE 8** Poiseuille flow: refined meshes in the orthogonal subgrid scales case with error indicator tolerance  $10^{-6} m^2 s^{-1}$

As can be seen in Figure 5A, the OSS formulation results in roughly 5 times less elements than the ASGS one for a given tolerance. In the same figure, we can also observe that, for small tolerances, refinement may fail to start. This can be seen in in Figure 5B for the ASGS simulation with a tolerance of  $10^{-3} m^2 s^{-1}$  and for the OSS cases with tolerances  $10^{-3} m^2 s^{-1}$  and  $10^{-4} m^2 s^{-1}$ , where no new elements are added. These cases produce a solution that is close to zero everywhere, where the yielded regions fail to appear.

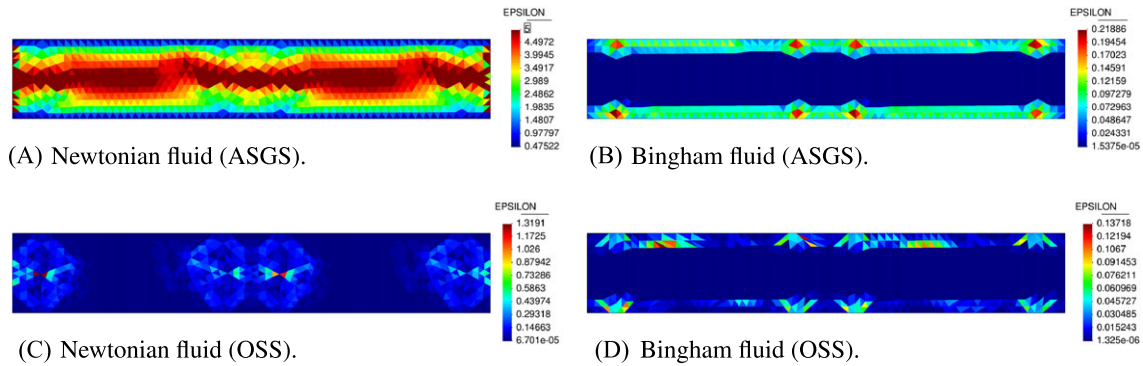
In Figure 6, we can observe a good agreement with the analytic solution for the Newtonian case, but the number of elements required is quite large, as can be seen in Figure 5A. To understand the results, we calculate the efficiency of our chosen indicators. The efficiency  $\varepsilon$  of an error indicator is the ratio between the estimator and the real error, which in our case is given by

$$\varepsilon = \frac{I_e}{\|\mathbf{u} - \mathbf{u}\|_{\Omega_e}} \quad \text{with} \quad \|\mathbf{u} - \mathbf{u}\|_{\Omega_e} = \left( \int_{\Omega_e} (\mathbf{u} - \mathbf{u}) \cdot (\mathbf{u} - \mathbf{u}) \, d\Omega \right)^{1/2}, \quad (36)$$

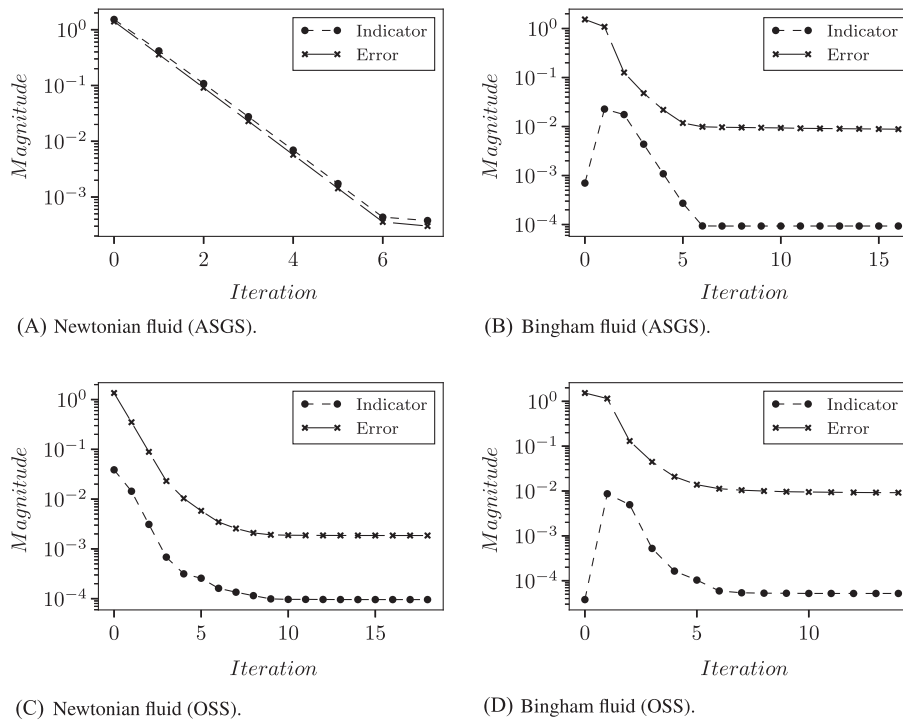
where  $\mathbf{u}$  is the exact solution, given by Equation 35, and  $\mathbf{u}_h$  is the approximate finite element solution. We are also interested in comparing the global value of the indicator to the global error, computed from the sums of the square of the values on each element as

$$I_g = \left( \sum_e I_e^2 \right)^{1/2} \quad E_g = \left( \sum_e \|\mathbf{u} - \mathbf{u}\|_{\Omega_e}^2 \right)^{1/2}. \quad (37)$$

The distribution of the efficiency index on a given solution step for the different cases is shown in Figure 9. It can be observed that, except for the Newtonian ASGS case, the efficiency is far from the ideal value of 1. In spite of this, the efficiency index tends to be larger in the areas that are in need of refinement, as for example close to the borders in the Bingham cases, where the gradients of the solution are larger.



**FIGURE 9** Poiseuille flow: distribution of the efficiency index  $\varepsilon$  for the different test cases. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales.



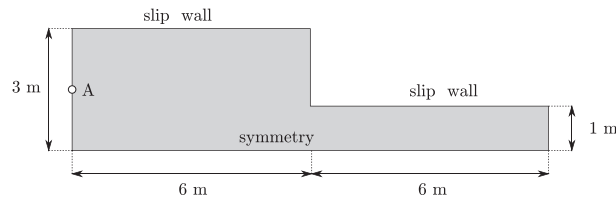
**FIGURE 10** Poiseuille flow: global indicator values and errors for the different test cases. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales

The evolution of the global value of the indicator and the error, evaluated according to Equation 37, for the cases with an indicator tolerance of  $10^{-6}m^2s^{-1}$  is presented in Figure 10. As can be observed in the plots, the error indicator has the same behavior as the true error, but its value is not directly related to the magnitude of the true error. This can explain how the general behavior of the refinement algorithm is correct, even when the efficiency index for a given case is not necessarily close to one.

## 4.2 | Plane extrusion

We simulated the plane extrusion of a Bingham fluid through a die with a 3 to 1 reduction of the cross-section. This problem was presented in Peric and Slijepcevic<sup>63</sup> and revisited in Larese,<sup>45</sup> where it was solved using a fixed fluid mesh and the ASGS formulation presented in Section 2 and in Moreno et al<sup>46</sup> and Moreno and Cervera<sup>64</sup> where it was simulated using OSS stabilization.

The simulation domain can be seen in Figure 11, where we use the symmetry of the problem to simulate only one-half of the domain. A slip boundary condition is imposed on the wall and only the wall-normal component of the velocity is restricted.



**FIGURE 11** Plane extrusion: geometry and boundary conditions

**TABLE 1** Plane extrusion: flow parameters for the problem

Parameter	Value
Fluid density	$\rho = 100\text{Kg}/\text{m}^3$
Yield stress	$\tau_0 = 1000\text{Pa}$
Fluid viscosity	$\mu_p = 10^{-6}\text{Pa} \cdot \text{s}$
Regularization coefficient	$m = 1000\text{s}$

The flow is driven by a ram pressure applied on the left side of the domain, which generates a pressure gradient. The fluid parameters are reported in Table 1.

As argued in Peric and Slijepcevic,<sup>63</sup> using the present settings, with smooth walls and a very small plastic viscosity, the problem is analog to a perfect plasticity problem. An exact solution for the plasticity problem, obtained using slip line theory, is reported by Lubliner.<sup>65</sup> This solution predicts the formation of slip lines once the applied pressure reaches

$$p = \frac{4}{3} \left( 1 + \frac{\pi}{2} \right) \tau_0 \approx 3427 \text{ Pa.} \quad (38)$$

The simulation is performed using both the ASGS and OSS variants of our formulation. An increasing normal pressure is applied on the left end of the domain in increments of 2Pa per step, starting from 0 to a maximum value of  $p_{max} = 5000\text{Pa}$ . After each step, the mesh refinement algorithm is used to improve the mesh resolution, using an absolute tolerance for the error indicator of  $10^{-6}\text{m}^2\text{s}^{-1}$ . In light of the results of the previous example, we set an additional control, limiting the minimum area of refined elements to  $10^{-4}\text{m}^2$  to prevent excessive refinement in localized regions. The domain is initially discretized with an unstructured mesh composed of 102 nodes and 152 linear triangles.

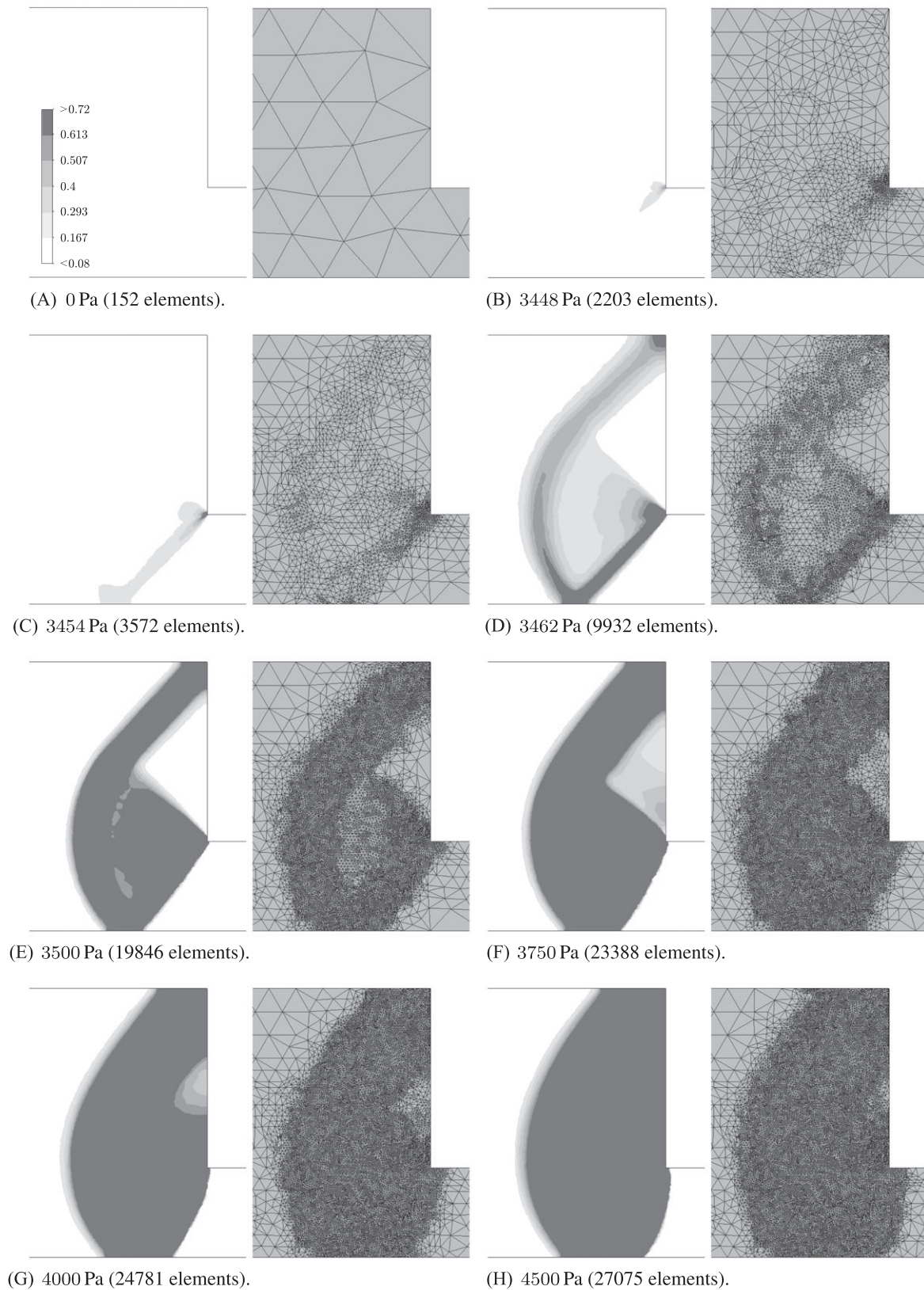
The evolution of the strain rate and the refined mesh for different values of the ram pressure in the OSS case is shown in Figure 12. As the ram pressure increases, a yielded region characterized by high strain rates develops, matching the slip line mechanism. The finite element mesh is refined accordingly, following the high strain rates.

Figure 13 shows the evolution of mesh size as the simulation evolves. The number of elements required to solve the problem is uniform for low values of the ram pressure until the yielded zone develops. At this point, the number of elements increases rapidly as the material starts to flow. The new elements are concentrated on the yielded regions, as can be observed in Figure 12, and the refinement process continues as the yielded region expands. The mesh at the end of the simulation, corresponding to an external pressure of 5000Pa, is composed by approximately 14 500 nodes and 28 700 elements for the OSS test and 15 100 nodes and 30 100 elements for the ASGS case.

The velocity of the fluid on the left boundary (measured on point A in Figure 11) is related to the ram pressure in Figure 14. The value of yield pressure expected from perfect plasticity, given by Equation 38, is indicated in Figure 14 using a dotted line. We can observe that velocity is very low until the pressure reaches roughly 3460Pa, when the material starts to flow, accelerating rapidly. This change corresponds to the formation of a yielded zone just before the extrusion section, which can be observed in Figure 12. This is found to be in agreement with the expected behavior, although the material starts flowing at higher pressures than predicted using perfect plasticity theory. While plasticity theory predicts yielding for  $p = 3427\text{Pa}$ , the simulated solution predicts yielding for pressures in the range of  $p = 3450 - 3460\text{Pa}$ . In this sense, we recall that this plasticity solution does not correspond exactly to our problem, as it does not account for the presence of a convective term in the equations nor for the fact that we are using a *regularized* constitutive equation instead of a purely plastic behavior.

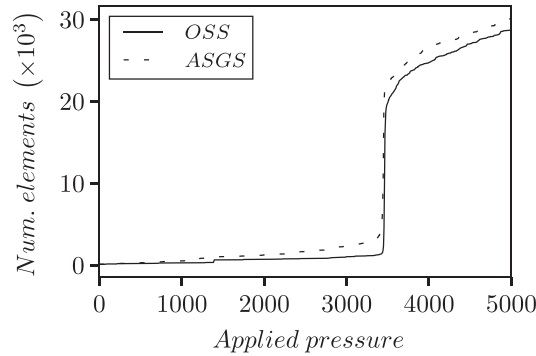
To evaluate the effect of the refinement procedure, we compare the results obtained using adaptive mesh refinement to those obtained using a uniform mesh size. With the same numerical settings, we simulate the problem using 2 uniform triangular meshes. The first is designed to contain about 28 000 elements, a number close to that of the finest mesh used during the refinement procedure, and predicts yielding on the  $p = 3540 - 3550\text{Pa}$  range. The second mesh is designed so that all elements have a size similar to that of the smallest element produced by the refinement procedure and contains about 700 000 elements



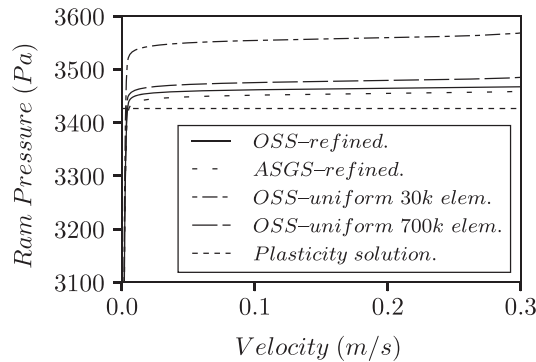


**FIGURE 12** Plane extrusion, orthogonal subgrid scales case: detail of the evolution of the strain rate (left) and computational mesh (right) for increasing values of the applied ram pressure

and predicts yielding for  $p = 3460\text{--}3470\text{Pa}$  range, much closer to the dynamically refined result. The velocity-pressure relation for each of these cases is also plotted in Figure 14. Comparing the solution obtained using adaptive mesh refinement to that



**FIGURE 13** Plane extrusion: evolution of the number of elements. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales



**FIGURE 14** Plane extrusion: applied pressure vs inlet velocity. ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales

produced by a comparable number of uniform size elements, we can observe that adaptive mesh refinement has a positive effect in the quality of the solution, producing a yielding pressure that is closer to that obtained using a much finer mesh.

### 4.3 | Cavity flow

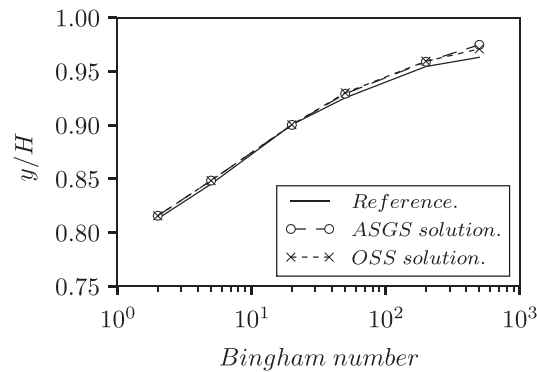
Next, we consider the flow in a 2D cavity filled with a Bingham fluid. The problem is set up following Mitsoulis and Zisis.<sup>66</sup> Defining a square domain  $\Omega = (0, H) \times (0, H)$ , we impose a horizontal velocity  $u_x = 1\text{ m/s}$  on the  $y = H$  side and zero velocity on the remaining contour nodes. Our model is configured as a leaky cavity, that is, the top left and top right corner nodes have a fixed horizontal velocity. This condition is reimposed after each refinement step, so that the wall node immediately next to the corner always has zero velocity, even if this node did not exist in previous iterations.

The fluid density is set to  $\rho = 1\text{ Kg/m}^3$  and the dynamic viscosity for the yielded region to  $\mu_p = 1\text{ Pa}\cdot\text{s}$ . We simulate multiple cases with different yield stresses, which correspond to a range of values of the Bingham number, defined as

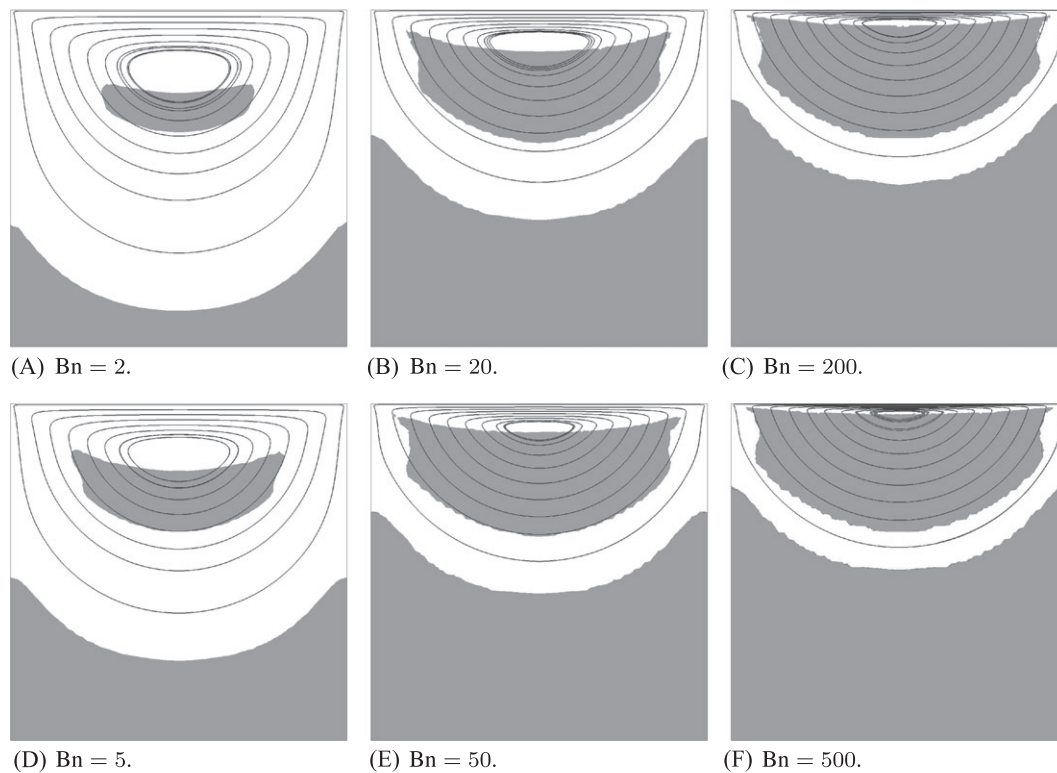
$$\text{Bn} = \frac{\tau_0 H}{\mu_p u_x}. \quad (39)$$

The regularization coefficient is set to  $m = 300\text{ s}$ . Again, we start from a relatively coarse uniform mesh composed of 2900 nodes and 5600 triangular elements, and we solve the problem iteratively, with a mesh refinement phase after each solution. The absolute tolerance for the error indicator is set to  $10^{-6}\text{ m}^2\text{ s}^{-1}$ , and a maximum of 10 refinement steps are allowed over the same original element. For the range of Bingham numbers considered, this problem results in a single recirculating vortex in the interior of the cavity, which develops closer to the top wall as the Bingham number increases. The vertical position of the vortex center for the different cases is compared to the results reported by Mitsoulis and Zisis<sup>66</sup> in Figure 15. The results are in agreement with the reference, although we obtain a slightly higher position for the center in the higher Bingham numbers. The final distribution of yielded and unyielded regions and the corresponding velocity streamlines are shown in Figure 16 for the OSS simulations. The number of elements obtained after each refinement is shown in Figure 17 for the ASGS cases and in Figure 18 for the OSS cases. In this case, we observe that the ASGS indicator results on a significantly larger number of elements for a given tolerance.





**FIGURE 15** 2D cavity flow: vertical position of the vortex center, compared to the results of Mitsoulis and Zisis.<sup>66</sup> ASGS, algebraic subgrid scales; OSS, orthogonal subgrid scales



**FIGURE 16** 2D cavity flow: velocity streamlines and distribution of yielded (light) and unyielded (dark) regions for the orthogonal subgrid scales simulations

Although the cavity flow is essentially a 2D problem for the range of values we are testing, we also simulated a 3D case to validate our approach for tetrahedra. The geometry of the problem is shown in Figure 19 and follows the definition of Elias et al.<sup>67</sup> The domain is a cube of side  $H = 1m$ , where velocity is fixed to  $(u_x, 0, 0)$  on the top side. Taking  $y$  to be the vertical axis, the velocity is set to zero on the bottom and on the sides of the cube normal to the flow. On the 2 remaining sides, parallel to the flow on the top, only the normal ( $z$ ) component of velocity is restricted.

For the 3D case, we use a Reynolds number  $Re = 1$  and a Bingham number  $Bn = 5$ . All fluid parameters are defined as in the 2D case, fixing the top velocity to  $u_x = 1m/s$  and the yield stress is  $\tau_0 = 5Pa$ . The regularization coefficient is set to  $m = 1000s$ .

Given the overrefinement observed for the ASGS case, the 3D simulation has been performed using only OSS. The flow is simulated in 10 solution steps, refining after each solution. Starting from a uniform tetrahedral mesh with 30 divisions along each edge, containing approximately 51 000 nodes and 277 000 elements, a final mesh with 113 000 nodes and 612 000 elements is obtained. The final distribution of yielded and unyielded regions and velocity streamlines is shown in Figure 20. The vortex center in this case is placed at a vertical position  $y/H = 0.848$ , in agreement with the 2D results shown in Figure 15.

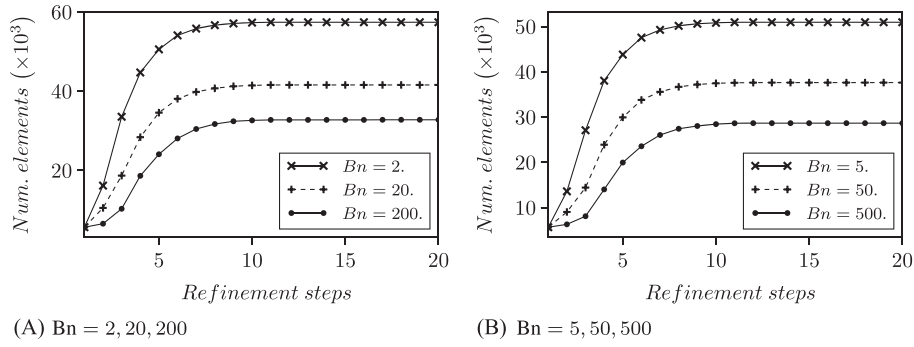


FIGURE 17 2D cavity flow: evolution of the number of elements for the algebraic subgrid scales simulations

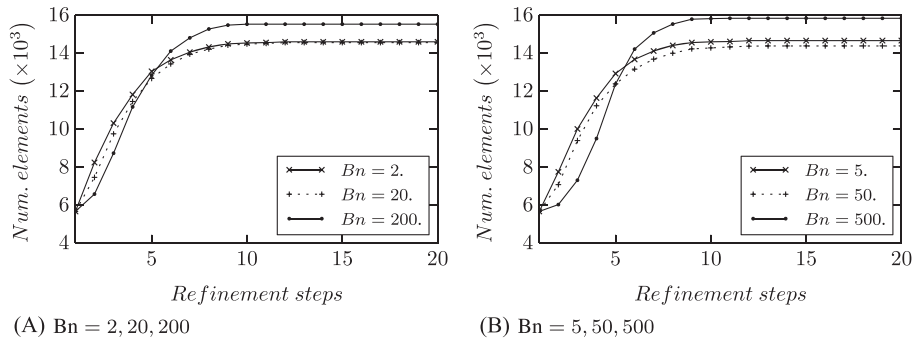


FIGURE 18 2D cavity flow: evolution of the number of elements for the orthogonal subgrid scales simulations

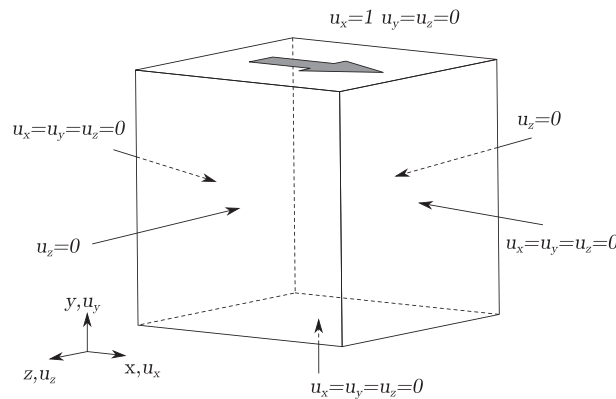
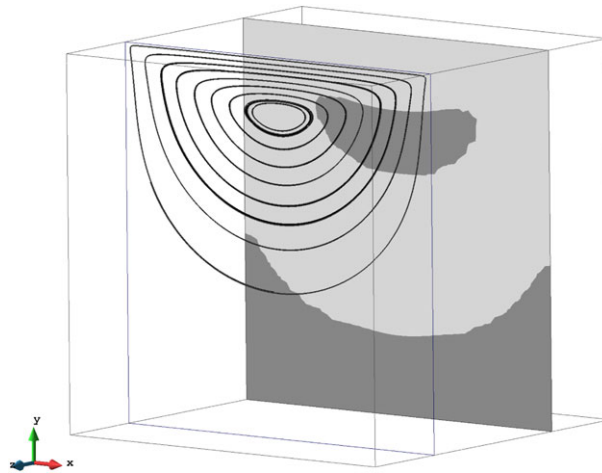


FIGURE 19 3D cavity flow: geometry and velocity boundary conditions

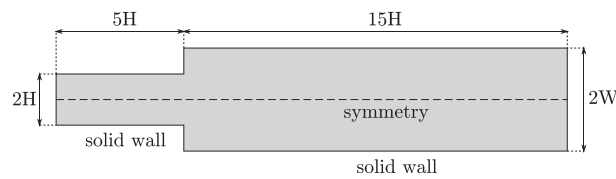
### 4.4 | Flow through a sudden expansion

As a final 3D test case, we simulate the flow through a square sudden expansion using OSS. This problem is studied in Burgos and Alexandrou<sup>68</sup> and Alexandrou et al<sup>69</sup> for Herschel-Bulkley fluids and represents a three-dimensional version of the more common planar or axisymmetric expansions (see for example reference herein<sup>70-72</sup>). The cross-section of the problem is shown in Figure 21. We model the flow through 2 different geometries, corresponding to expansions with width ratios  $W/H = 2$  and  $W/H = 4$ .

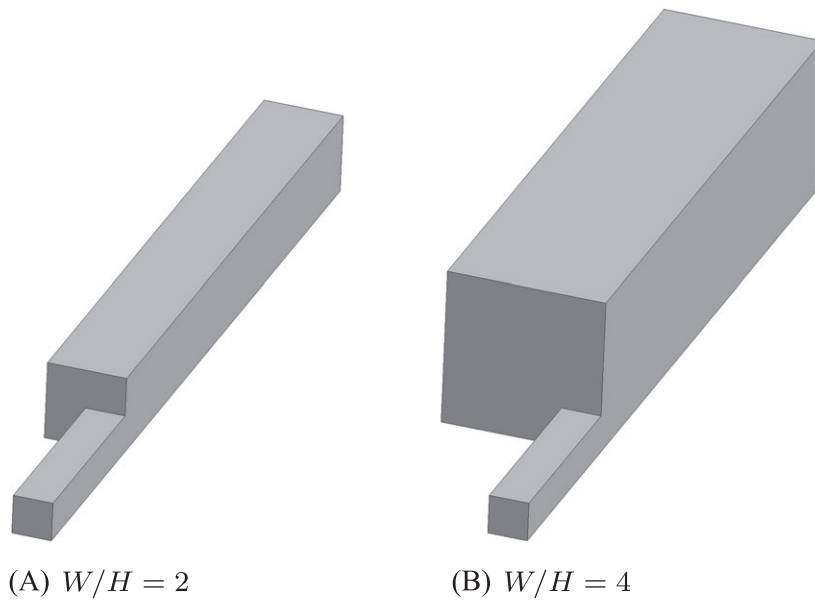
Taking into account the symmetries of the problem, only one-fourth of the expansion is simulated, resulting in the computational domains shown in Figure 22. No-slip boundary conditions are used to model the solid walls, while a no-penetration condition is set for the symmetry planes. The flow is driven by a pressure gradient imposed between the inflow and the outlet sections of the domain.



**FIGURE 20** 3D cavity flow: velocity streamlines and distribution of yielded (light) and unyielded (dark) regions



**FIGURE 21** Sudden expansion: geometry

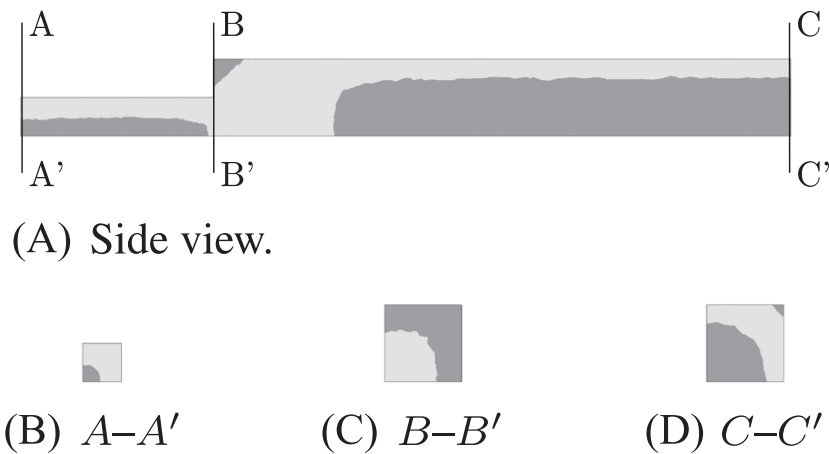


**FIGURE 22** Sudden expansion: simulation domains

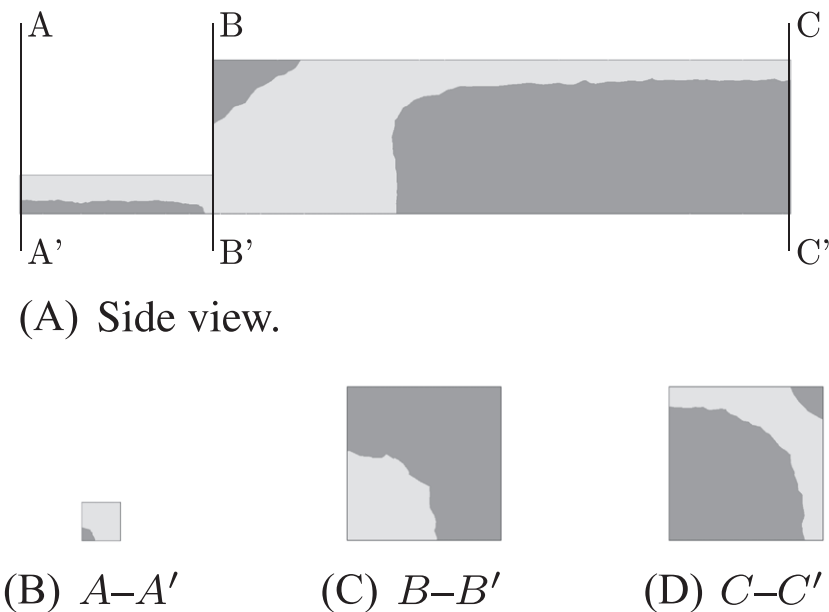
The problem is solved for Reynolds and Bingham numbers equal to one, calculated using  $H$  as the reference length and a reference velocity  $U_0$  defined in Burgos and Alexandrou<sup>68</sup> as

$$U_0 = \frac{1}{\mu_p} \left( H \left| \frac{\Delta p}{\Delta x} \right| - \tau_0 \right) H, \quad (40)$$

where  $\Delta p/\Delta x$  is the imposed pressure gradient. [Correction added on 8 November 2017, after first online publication: vertical bars were missing in Equation 40 and have since been added]



**FIGURE 23** Sudden expansion,  $W/H = 2$ : yielded (dark) and unyielded (light) regions

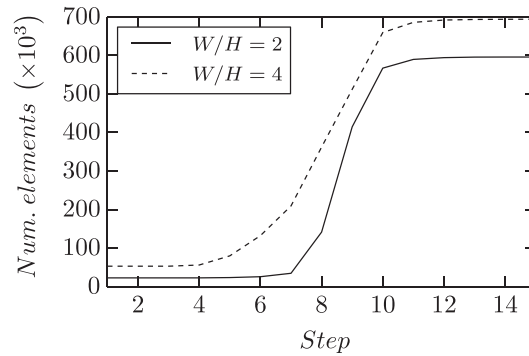


**FIGURE 24** Sudden expansion,  $W/H = 4$ : yielded (dark) and unyielded (light) regions

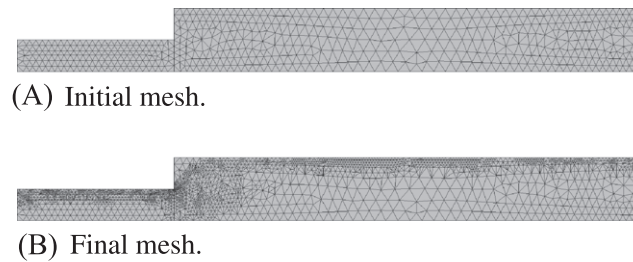
We apply the external pressure in 10 incremental load steps, refining after each iteration. Once the loading process is finished, we simulate 5 extra steps under full load to ensure that the final solution does not require additional refinement. The distribution of yielded and unyielded regions on different sections can be observed in Figure 23 for the  $W/H = 2$  expansion and in Figure 24 for the  $W/H = 4$  case.

Both cases exhibit a qualitatively similar behavior, in agreement with the results obtained in the references. Far from the expansion, a yielded region appears close to the walls because of the shear produced by wall friction, while a core of unyielded material remains in the center. Close to the expansion, high-velocity gradients develop as the flow adapts to the change in cross-section and the central core has yielded completely. On the corner of the expansion, a region of stationary unyielded material appears, unaffected by the main flow.

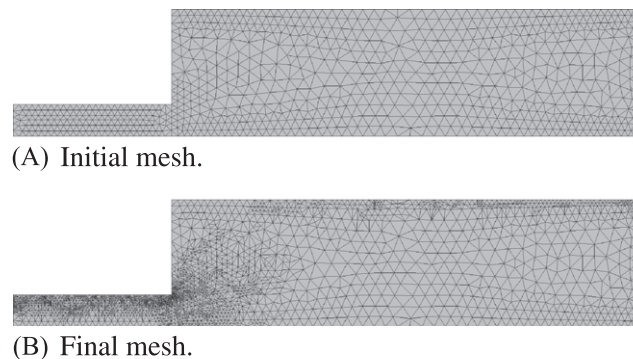
As in the previous cases, the simulation is initialized with a uniform tetrahedral mesh, composed of approximately 5000 nodes and 22 000 elements for the  $W/H = 2$  case or 10 000 nodes and 53 000 elements for the  $W/H = 4$  case. The evolution of the number of elements during the solution is shown in Figure 25. The number of elements grows as the applied pressure gradient increases and stabilizes once the loading process finishes, resulting in a final grid of 106 000 nodes and 596 000 elements for the  $W/H = 2$  case and 123 000 nodes and 694 000 elements for the  $W/H = 4$  expansion.



**FIGURE 25** Sudden expansion: evolution of the number of elements during the simulation



**FIGURE 26** Sudden expansion,  $W/H = 2$ : side view of the computational meshes



**FIGURE 27** Sudden expansion,  $W/H = 4$ : side view of the computational meshes

The initial and final meshes for the 2 expansion cases are shown in Figure 26 for the  $W/H = 2$  test and Figure 27 for the  $W/H = 4$  case. It can be observed that refined areas coincide with yielded regions, where higher velocity gradients are generally found: close to the solid walls and just after the expansion section.

## 5 | CONCLUDING REMARKS

We have presented a VMS finite element formulation for non-Newtonian flow problems. Using an error indicator based on the scale separation introduced by VMS methods, we have combined this formulation with a mesh refinement procedure to obtain a solver with adaptive mesh refinement. The chosen mesh refinement strategy, based on edge splitting, works on triangular and tetrahedral meshes, which has allowed us to simulate both  $2D$  and  $3D$  problems.

We have used this approach to solve multiple benchmark examples found in the literature, starting from uniform, relatively coarse meshes, and letting the refinement procedure increase mesh resolution on regions of interest. Results suggest that the use of adaptive mesh refinement helps obtain good quality solutions starting from meshes without sufficient resolution. This validation procedure has also allowed us to identify some issues with the proposed error indicator, in particular when evaluating its efficiency in Section 4.1. While the indicator tends to be larger in areas with a larger error, its value is not directly related to the magnitude of the error, and the ratio between the 2 quantities appears to be problem-dependent. This introduces a difficulty

in choosing the tolerances for the error indicator, since the magnitude of the estimator can not be immediately deduced from quantities of interest. Most notably, it can not be understood as a maximum admissible local error.

In practice, this introduces potential over- or under-refinement issues, since the indicator tolerance is difficult to set a priori. We have observed this in Section 4.3, where, for a fixed tolerance, the use of the ASGS indicator results in significantly more elements than the OSS one for the same simulation, although this has little effect on the accuracy of the obtained solution. A first step to mitigate this issue has been to introduce controls for the minimum element size allowed, ensuring that the refinement procedure will always stop.

In view of these issues, an obvious venue for improvement would be to test the same refinement approach with other indicators. Since the refinement procedure itself is independent of the choice of indicator, this should be a straightforward modification. In addition, the refinement procedure could be extended to more complex geometries, such as quadrilateral and hexahedral elements, by introducing splitting patterns for these geometries. Similarly, by introducing more nodes on each split edge, the scheme could be used with higher order elements.

We note that, although we have limited our tests to Bingham flow problems, the presented formulation is directly applicable to other fluids with yield and, in particular, would be straightforward to use for the more general Herschel-Bulkley model. Another venue for future work is the application of the present solver to time-dependent problems. The OSS formulation used here is extended to (fixed mesh) dynamic problems in Moreno et al.,<sup>46</sup> and the refinement procedure was used in the setting of evolving turbulent flows in Rossi et al.<sup>16</sup> However, as noted in this last reference, expanding the adaptive refinement procedure to allow the possibility of coarsening the mesh in response to small values of the error indicator would be a welcome addition in this context. This would allow us to *move* the refined areas in accordance with the evolution of the flow, optimizing the number of elements used throughout the simulation.

Finally, the use of a regularized formulation raises some questions that deserve further investigation. While necessary to eliminate the singularity on the Bingham law and accelerate the convergence of the method to a solution, regularization does introduce an error on the solution, independent of the mesh and of our error indicator. Gaining a better understanding of this fact could open the door to an improved approach, where the value of the regularization parameter  $m$  is adjusted as part of the adaptive procedure.

## ACKNOWLEDGMENTS

The first author gratefully acknowledges the support of the Spanish *Ministerio de Educación* through a doctoral grant in the FPU program. The research was partially supported by the Spanish Ministry of Economy and Competitiveness (*Ministerio de Economía y Competitividad*, MINECO) through the project EACY (MAT2013-48624-C2-1-P). The financial support of the ERC Advanced grant project SAFECON (AdG-267521) of the European Research Council and of the T-MAPPP project (FP7 PEOPLE 2013 ITN-G.A.n607453) is also greatly acknowledged.

## REFERENCES

1. Carey GF. *Computational grids: generations, adaptation & solution strategies*. Washington, DC: CRC Press; 1997.
2. Ainsworth M, Oden JT. *A posteriori error estimation in finite element analysis*. New York, NY: John Wiley & Sons, Ltd; 2000.
3. Babuška I, Rheinboldt WC. Error estimates for adaptive finite element computations. *SIAM J Numer Anal*. 1978;15(4):736-754.
4. Babuška I, Rheinboldt WC. A-posteriori error estimates for the finite element method. *Int J Numer Methods Eng*. 1978;12(10):1597-1615.
5. Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. *Comput Methods Appl Mech Eng*. 1997;142(12):1-88.
6. John V. A numerical study of a posteriori error estimators for convection-diffusion equations. *Comput Methods Appl Mech Eng*. 2000;190(5-7):757-781.
7. Papastavrou A, Verfürth R. A posteriori error estimators for stationary convection-diffusion problems: a computational comparison. *Comput Methods Appl Mech Eng*. 2000;189(2):449-462.
8. Saramito P, Roquet N. An adaptive finite element method for viscoplastic fluid flows in pipes. *Comput Methods Appl Mech Eng*. 2001;190(40-41):5391-5412.
9. Roquet N, Saramito P. An adaptive finite element method for Bingham fluid flows around a cylinder. *Comput Methods Appl Mech Eng*. 2003;192(31-32):3317-3341.
10. Roquet N, Saramito P. An adaptive finite element method for viscoplastic flows in a square pipe with stick-slip at the wall. *J Nonnewton Fluid Mech*. 2008;155(3):101-115.
11. Bernabeu N, Saramito P, Smutek C. Numerical modeling of non-Newtonian viscoplastic flows: part II. Viscoplastic fluids and general tridimensional topographies. *Int J Numer Anal Model*. 2014;11(1):213-228. Dedicated to Professor Francisco J. Lisbona on the occasion of his 65th Birthday.
12. Boussetta R, Coupeux T, Fourment L. Adaptive remeshing based on a posteriori error estimation for forging simulation. *Comput Methods Appl Mech Eng*. 2006;195(4849):6626-6645. Computational Metal Forming.



13. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comput Methods Appl Mech Eng.* 1995;127(1-4):387-401.
14. Hughes TJR, Feijóo GR, Mazzei L, Quincy J-B. The variational multiscale method—a paradigm for computational mechanics. *Comput Methods Appl Mech Eng.* 1998;166(1-2):3-24. *Advances in Stabilized Methods in Computational Mechanics.*
15. Hauke G, Doweidar MH, Miana M. The multiscale approach to error estimation and adaptivity. *Comput Methods Appl Mech Eng.* 2006;195(13-16):1573-1593. *A Tribute to Thomas J.R. Hughes on the Occasion of his 60th Birthday.*
16. Rossi R, Cotela J, Lafontaine NM, Dadvand P, Idelsohn SR. Parallel adaptive mesh refinement for incompressible flow problems. *Comput Fluids.* 2013;80(0):324-355.
17. Hauke G, Fuster Da, Lizarraga F. Variational multiscale a posteriori error estimation for systems: the Euler and Navier-Stokes equations. *Comput Methods Appl Mech Eng.* 2015;283:1493-1524.
18. Peraire J, Vahdati M, Morgan K, Zienkiewicz OC. Adaptive remeshing for compressible flow computations. *J Comput Phys.* 1987;72(2):449-466.
19. Löhner R. Adaptive remeshing for transient problems. *Comput Methods Appl Mech Eng.* 1989;75(1-3):195-214.
20. Selmin V, Formaggia L. Simulation of hypersonic flows on unstructured grids. *Int J Numer Methods Eng.* 1992;34(2):569-606.
21. Zienkiewicz OC, Wu J. Automatic directional refinement in adaptive analysis of compressible flows. *Int J Numer Methods Eng.* 1994;37(13):2189-2210.
22. Chiandussi G, Bugada G, Oñate E. A simple method for automatic update of finite element meshes. *Commun Numer Methods Eng.* 2000;16(1):1-19.
23. Sieger D, Menzel S, Botsch M. RBF morphing techniques for simulation-based design optimization. *Eng Comput.* 2014;30(2):161-174.
24. Hachem E, Feghali S, Codina R, Coupez T. Anisotropic adaptive meshing and monolithic variational multiscale method for fluid-structure interaction. *Comput Struct.* 2013;122(0):88-100. *Computational Fluid and Solid Mechanics 2013 Proceedings Seventh MIT Conference on Computational Fluid and Solid Mechanics.*
25. Cotela Dalmau J. Applications of turbulence modelling in civil engineering. *Ph.D. Thesis: Universitat Politècnica de Catalunya;* 2016.
26. Bingham EC. *Fluidity and plasticity.* New York, NY: McGraw-Hill; 1922.
27. Oldroyd JG. A rational formulation of the equations of plastic flow for a Bingham solid. *Math Proc Cambridge Philos Soc.* 1947(1);43:100-105.
28. Duvaut G, Lions JL. *Inequalities in mechanics and physics.* Berlin, Germany: Springer-Verlag Berlin Heidelberg; 1976.
29. Glowinski R. *Numerical methods for nonlinear variational problems.* Berlin, Germany: Springer; 1984.
30. Huilgol RR. *Fluid mechanics of viscoplasticity.* Berlin Heidelberg: Springer; 2015.
31. Dean EJ, Glowinski R, Guidoboni G. On the numerical simulation of Bingham visco-plastic flow: old and new results. *J Nonnewton Fluid Mech.* 2007;142:36-62.
32. Huilgol RR, You Z. Prolegomena to variational inequalities and numerical schemes for compressible viscoplastic fluids. *J Nonnewton Fluid Mech.* 2009;158:113-126.
33. Huilgol RR, Kefayati GHR. Natural convection problem in a Bingham fluid using the operator-splitting method. *J Nonnewton Fluid Mech.* 2015;220:22-32.
34. Sanchez FJ. Application of first-order operator splitting method to Bingham fluid flow simulation. *Comput Math Appl.* 1998;36(3):71-86.
35. Glowinski R, Tallec PL. Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. *SIAM, Philadelphia.* 1989;287-295.
36. Zhang J. An augmented Lagrangian approach to Bingham fluid flows in a lid-driven square cavity with piecewise linear equal-order finite elements. *Comput Methods Appl Mech Eng.* 2010;199:3051-3057.
37. Bercovier M, Engelman M. A finite-element method for incompressible non-Newtonian flows. *J Comput Phys.* 1980;36(3):313-326.
38. Tanner RI, Milthorpe JF. Numerical simulation of the flow of fluids with yield stress. In: Taylor C, Johnson JA, Smith WR, eds. *Numerical methods for laminar and turbulent flow.* Swansea, UK: Pineridge Press; 1983:680-690. *Proceedings of the Third International Conference, Seattle.*
39. Papanastasiou TC. Flows of materials with yield. *J Rheol (1978-present).* 1987;31(5):385-404.
40. Souza Mendes PR, Dutra ESS. Viscosity function for yield-stress liquids. *Appl Rheol.* 2004;14(6):296-302.
41. Mitsoulis E. Flows of viscoplastic materials: models and computations. In: *Rheology Reviews 2007.* British Society of Rheology. Swansea, UK;2007:135-178.
42. Frigaard IA, Nouar C. On the usage of viscosity regularisation methods for visco-plastic fluid flow computation. *J Nonnewton Fluid Mech.* 2005;127(1):1-26.
43. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput Methods Appl Mech Eng.* 2002;191(39-40):4295-4321.
44. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Comput Methods Appl Mech Eng.* 2001;190(20-21):2681-2706.
45. Larese A. A coupled Eulerian-PFEM Thesis: Universitat Politècnica de Catalunya. UPC BarcelonaTech; 2012.
46. Moreno E, Larese An, Cervera M. Modelling of Bingham and Herschel-Bulkley flows with mixed P1/P1 finite elements stabilized with orthogonal subgrid scale. *J Nonnewton Fluid Mech.* 2016;228:1-16.
47. Rossi R, Larese A, Dadvand P, Oñate E. An efficient edge-based level set finite element method for free surface flow problems. *Int J Numer Meth Fluids, DOI: 10.1002/fld.3680.* 2013;71 (6):687-716.
48. Larese A, Rossi R, Oñate E. Finite element modeling of free surface flow in variable porosity media. *Arch Numer Meth Eng DOI: 10.1007/s11831-014-9140-x.* 2014;22:637-653.
49. Moreno E, Cervera M. Elementos finitos mixtos estabilizados para flujos confinados de Bingham y de Herschel-Bulkley parte I: Formulación. *Revista Int de Métodos Numéricos para Cálculo y Diseño en Ingeniería.* 2016;32(2):100-109.



50. Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. In: Hutchinson JW, Wu TY, eds. *Advances in Applied Mechanics* 28. Cambridge, MA: Elsevier; 1991:1-44.
51. Hughes TJR, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Comput Methods Appl Mech Eng*. 1989;73(2):173-189.
52. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *Int J Numer Methods Eng*. 1987;24(2):337-357.
53. Dörfler W. A convergent adaptive algorithm for Poissons equation. *SIAM J Numer Anal*. 1996;33(3):1106-1124.
54. Nochetto RH, Siebert KG, Veeser A. Multiscale, nonlinear and adaptive approximation. In: DeVore R, Kunoth A, eds. *Multiscale, Nonlinear and Adaptive Approximation: Dedicated to Wolfgang Dahmen on the Occasion of his 60th Birthday*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009:409-542.
55. Frey PJ, George PL. *Mesh generation: application to finite elements*. 2nd ed.. Hoboken, NJ: John Wiley & Sons, Ltd; 2008.
56. D'Amato JP, Vénere M. A CPU-GPU framework for optimizing the quality of large meshes. *J Parallel Distrib Comput*. 2013;73(8):1127-1134.
57. Dadvand P, Rossi R, Gil M, et al.. Migration of a generic multi-physics framework to HPC environments. *Comput Fluids*. 2013;80(0):301-309.
58. Larese A, Rossi R, Oñate E, Toledo MA. Physical and numerical modelization of the behavior of rockfill dams during overtopping scenarios. *Dam Maintenance and Rehabilitation II. CRCpress Balkema*. ISBN:978-0-415-61648-5. 2010;2:479-487.
59. Larese A, Rossi R, Oñate E, Idelsohn SR. A coupled PFEM-Eulerian approach for the solution of porous FSI problems. *Comput Mech, DOI: 10.1007/s00466-012-0768-9*. 2012;50 (6):805-819.
60. Larese A, Rossi R, Oñate E, Toledo MA, Moran R, Campos H. Numerical and experimental study of overtopping and failure of rockfill dams. *Int J Geomechanics (ASCE)*. 2013,15(4):10.1061/(ASCE)GM.1943-5622.0000345.
61. Demidov DE, Shevchenko DV. Modification of algebraic multigrid for effective gpgpu-based solution of nonstationary hydrodynamics problems. *J Comput Sci*. 2012;3(6):460-462. Next Generation Computational Scientists: Russian Federation.
62. Moreno E. Elementos finitos mixtos estabilizados para flujos viscoplásticos. *Ph.D. Thesis*: Universitat Politècnica de Catalunya. UPC BarcelonaTech; 2014.
63. Peric D, Slijepcevic S. Computational modelling of viscoplastic fluids based on a stabilised finite element method. *Eng Comput*. 2001;18(3/4):577-591.
64. Moreno E, Cervera M. Elementos finitos mixtos estabilizados para flujos confinados de Bingham y de Herschel-Bulkley parte II: Soluciones numéricas. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*. 2016;32(3):131-138.
65. Lubliner J. *Plasticity theory*. New York: Macmillan Publishing Company; 1990.
66. Mitsoulis E, Zisis T. Flow of Bingham plastics in a lid-driven square cavity. *J Nonnewton Fluid Mech*. 2001;101(1-3):173-180.
67. Elias R, Martins M, Coutinho A. Parallel edge-based solution of viscoplastic flows with the SUPG/PSPG formulation. *Comput Mech*. 2006;38(4-5):365-381.
68. Burgos GR, Alexandrou AN. Flow development of Herschel-Bulkley fluids in a sudden three-dimensional square expansion. *J Rheol*. 1999;43(3):485-498.
69. Alexandrou AN, McGilvrey TM, Burgos G. Steady Herschel-Bulkley fluid flow in three-dimensional expansions. *J Nonnewton Fluid Mech*. 2001;100(13):77-96.
70. Sheen HJ, Chen WJ, Wu JS. Flow patterns for an annular flow over an axisymmetric sudden expansion. *J Fluid Mech*. 1997;350:177-188.
71. Alleborn N, Nandakumar K, Raszillier H, Durst F. Further contributions on the two-dimensional flow in a sudden expansion. *J Fluid Mech*. 1997;330:169-188.
72. Mitsoulis E, Huilgol RR. Entry flows of Bingham plastics in expansions. *J Nonnewton Fluid Mech*. 2004;122(1-3):45-54. *XIIIth International Workshop on Numerical Methods for Non-Newtonian Flows*.

**How to cite this article:** Cotela-Dalmau J, Rossi R, Larese A. Simulation of two- and three-dimensional viscoplastic flows using adaptive mesh refinement. *Int J Numer Meth Engng*. 2017;112:1636-1658. <https://doi.org/10.1002/nme.5574>