

No. 690

June 2026

**A Python-Based Finite Element Digital Twin
for Thixoviscoplastic Lid-Driven Cavity Flow:
Implementation, Verification, and Validation
Against Turek Benchmark Results**

Saman

ISSN: 2190-1767

A Python-Based Finite Element Digital Twin for Thixoviscoplastic Lid-Driven Cavity Flow: Implementation, Verification, and Validation Against Turek Benchmark Results

Saman

Department of Mathematics, University of Wah, Pakistan

In collaboration with the group of Prof. Dr. Stefan Turek,

Institute of Applied Mathematics (LS III), TU Dortmund, Germany

June 26, 2026

Abstract

A Python-based finite element (FEM) solver is developed for steady incompressible flow of thixoviscoplastic (TVP) fluids in a lid-driven square cavity. The solver adopts the Houska TVP constitutive model of Begum, Ouazzi, and Turek [1], in which a scalar microstructure parameter $\lambda \in [0, 1]$ governs structural buildup and viscosity is coupled nonlinearly to both the shear rate $\dot{\gamma}$ and λ through a Papanastasiou-regularised extended viscosity. The implementation uses Taylor-Hood $\mathcal{Q}_2/\mathcal{Q}_1$ elements on structured quadrilateral meshes, assembled entirely in Python (NumPy, SciPy, GMSH) with monolithic Newton iteration and a direct sparse-LU inner solve. A five-stage verification and validation pipeline is presented: the Stokes assembler reproduces the exact Poiseuille profile to floating-point precision (L^2 error $< 10^{-14}$) and achieves optimal manufactured-solution convergence rates of 3.3/2.3/2.0 in L^2 velocity/ H^1 velocity/ L^2 pressure; the Navier-Stokes solver matches the Ghia et al. [8] benchmark at $\text{Re} = 100$ to within 5.0×10^{-3} m/s; power-law and Bingham solvers are verified to relative L^2 errors of 7.5×10^{-5} and 2.7×10^{-5} ; and the monolithic TVP Jacobian passes seven finite-difference checks at $\mathcal{O}(10^{-10})$ relative error. A conditioning study derives empirically that $\kappa(\mathbf{J}_{bc}) \propto \eta_0^2$, reaching 3.9×10^7 at the paper's target parameters, providing a first-principles justification for the multigrid infrastructure of Begum et al. [1]. Newton converges in 6–13 iterations per stage at reduced parameters ($\eta_0 = 10^{-3}$ Pa). A square obstacle introduced at the cavity centre produces an eight-fold reduction in the high-velocity domain fraction, constituting the paper's original geometric contribution and identifying a second, mesh-geometry-induced source of ill-conditioning on unstructured meshes.

Keywords: thixoviscoplastic fluids; finite element method; lid-driven cavity; Taylor-Hood elements; Houska model; Papanastasiou regularisation; monolithic Newton iteration; saddle-point conditioning; Python scientific computing; FEATFLOW benchmark

Contents

1	Introduction	4
2	Mathematical Formulation	5
2.1	Incompressible Navier-Stokes Equations	5
2.2	Generalised Newtonian Model	6
2.2.1	Power-Law Model	6
2.2.2	Bingham Model and Papanastasiou Regularisation	6
2.3	Thixoviscoplastic (TVP) Model: Houska Formulation	7
2.3.1	Material Functions	7
2.3.2	Extended Viscosity (Papanastasiou Regularisation)	7
2.3.3	Microstructure Evolution Equation	8
2.3.4	Coupled Steady System	8
2.3.5	Limiting Cases	8
3	Numerical Methodology	9
3.1	Weak Formulation	9
3.2	Finite Element Spaces and Inf-Sup Stability	10
3.3	Reference Element, Geometry Map, and Quadrature	11
3.3.1	Reference Element Layout	11
3.3.2	Shape Functions	11
3.3.3	Geometry Map and Jacobian	11
3.3.4	Gauss Quadrature	12
3.4	Global Saddle-Point System	12
3.5	Boundary Conditions and Pressure Normalisation	13
3.5.1	Dirichlet Velocity Conditions	13
3.5.2	Pressure Null-Space Removal	13
3.6	Mesh Construction: Q1 to Q2 Enrichment	14
3.7	Newton Linearisation for the Nonlinear System	14
3.8	Grid Convergence Index	15
4	Implementation	15
4.1	Mesh Enrichment: <code>q2_mesh.py</code>	15
4.2	Element Matrix Assembly: <code>assembly.py</code>	16
4.2.1	Global assembly strategy	16
4.2.2	Viscous block $\mathbf{A}^e$: 3×3 quadrature	17
4.2.3	Divergence block $\mathbf{B}^e$: 2×2 quadrature and sign convention	17
4.2.4	Body-force vector $\mathbf{F}^e$	17
4.3	Boundary Conditions: <code>bc_handling.py</code>	18
4.3.1	Dirichlet velocity conditions	18
4.3.2	Pressure null-space removal	18
4.4	Navier-Stokes Solver: Picard Iteration	18
4.4.1	Oseen linearisation	19
4.4.2	Convection matrix $\mathbf{C}(\mathbf{u}^{(k)})$	19
4.4.3	Algorithm	19
4.4.4	Convergence criterion and continuation	20
4.4.5	Relation to the Newton Jacobian	20

5	Verification and Validation	21
5.1	Stage 1 — Stokes Solver: Poiseuille Exact Reproduction	21
5.2	Stage 2 — Stokes Solver: Manufactured Solution Convergence	21
5.3	Stage 3 — Navier-Stokes Solver: Ghia Benchmark, $Re = 100$	22
5.4	Stage 4 — Generalised Newtonian Solver: Power-Law and Bingham Fluids	24
5.4.1	Stage 4A: Power-Law Fluid	25
5.4.2	Stage 4B: Regularised Bingham Fluid	25
5.5	Stage 5 — TVP Solver: Microstructure Transport Verification	26
5.5.1	Stage 5.1 — Exact Equilibrium Profile	27
5.5.2	Stage 5.2 — Development-Length Test	27
5.5.3	Stage 5.3 — Monolithic Jacobian Verification and Saddle-Point Con- ditioning	29
5.5.4	Stage 5.4 — Monolithic Newton TVP Solver: Reduced Parameters .	32
6	Results and Discussion	35
6.1	Newtonian and Generalised Newtonian Summary	35
6.2	TVP Physics: Bingham Limit to Full Microstructure Coupling	35
6.3	Geometric Extension: the Obstructed Cavity	36
7	Conclusion	37

1 Introduction

Thixoviscoplastic (TVP) fluids combine two physically distinct phenomena: yield-stress (viscoplastic) behaviour, where the material flows only beyond a critical stress threshold, and thixotropy, a time-dependent structural recovery that is disrupted by flow. Examples include fresh cement paste, blood, certain food gels, and drilling muds [6, 11]. Accurate numerical simulation of such materials is central to industrial process design, biomedical modelling, and geophysical applications.

Numerical methods for viscoplastic flows have developed considerably over the past three decades. The Papanastasiou regularisation [13] remains the standard approach for yield-stress fluids in FEM solvers: it replaces the discontinuous Bingham constitutive law with a smooth viscosity function that is differentiable everywhere, enabling Newton-based iteration without explicit tracking of the yield surface. An alternative class of models, exemplified by the elasto-visco-plastic framework of Saramito [15], incorporates elastic recovery before yield and permits a more complete description of viscoelastic yielding transitions, though at the cost of additional constitutive equations. For purely viscoplastic flows in the lid-driven cavity, Mitsoulis and Tsamopoulos [12] provide a comprehensive survey of numerical predictions for Bingham and Herschel-Bulkley fluids, establishing reference data against which new solvers are routinely compared. Fully coupled TVP systems, in which a microstructure evolution equation for the structural parameter λ is solved alongside the momentum equations, present additional nonlinear complexity beyond purely viscoplastic models: the viscosity depends on both the local shear rate and the microstructure state, and the two fields are coupled through a transport-reaction equation whose source terms are nonlinear in λ .

The lid-driven cavity is the canonical benchmark for incompressible flow solvers [7, 8]. Its simple geometry, homogeneous boundary conditions on three walls, and a single driven lid concentrate all solver complexity into the nonlinear physics and numerical method, without the confounding influence of complex domain geometry. For that reason it serves equally well as a validation target for generalised Newtonian and TVP solvers [1, 12].

The reference computation for this work is the study of Begum, Ouazzi, and Turek [1], published in the *International Journal for Numerical Methods in Fluids* (IJNMF, 2025, 97:565–604). That work establishes high-resolution TVP benchmark results for the lid-driven cavity using the monolithic Newton-multigrid FEM solver FEATFLOW [16]. It employs the $\mathcal{Q}_2/\mathcal{P}_1^{\text{disc}}$ biquadratic velocity / discontinuous linear pressure pair of the Turek group [16]. The present work uses the continuous Taylor-Hood $\mathcal{Q}_2/\mathcal{Q}_1$ pair, which offers the same $\mathcal{O}(h^3)$ velocity convergence rate and is well-suited to a Python-based implementation, at the cost of a larger pressure approximation space. The resulting solver constitutes a *digital twin* of the physical TVP cavity flow in the sense of Grieves and Vickers [9]: a parameterised, verified computational replica of the system that accepts updated material parameters $(\eta_0, \tau_0, \tau_\infty, M_a, M_b, k)$ and returns the corresponding velocity, pressure, and microstructure fields without re-derivation or re-implementation. The Houska TVP constitutive model [1] is adopted specifically because it is the model used in the Begum et al. benchmark, enabling direct quantitative comparison of solver outputs; it is also the simplest TVP formulation that captures both yield-stress behaviour and thixotropic microstructure evolution simultaneously, making it a natural starting point for open-source implementation and verification.

The contributions of this work are threefold. First, a fully documented open-source Python FEM solver for TVP lid-driven cavity flow is presented, verified through a five-

stage pipeline progressing from Stokes machine-precision reproduction to monolithic Newton TVP with k -continuation. Open-source Python FEM frameworks such as FEniCS [10] and Firedrake [14] are increasingly used in applied mathematics research, yet their accuracy on strongly nonlinear rheological problems has not been benchmarked systematically against FEATFLOW-class references; the present hand-assembled SciPy solver fills this gap with a fully transparent, stage-by-stage verification record. Second, the saddle-point condition number of the assembled TVP Jacobian is derived empirically as $\kappa \propto \eta_0^2$, providing a first-principles, computation-backed justification for the multigrid infrastructure of Begum et al. [1] that is not derived in that work. Third, an obstructed-cavity configuration is introduced as an original geometric extension, quantifying an eight-fold reduction in the high-velocity domain fraction and identifying mesh-geometry-induced ill-conditioning as a second, independent target for solver development.

The solver is built in stages, each independently verified before the next is assembled. Section 2 derives the governing equations from the incompressible Navier-Stokes system through generalised Newtonian models to the full TVP constitutive law. Section 3 presents the weak formulation, Taylor-Hood discretisation, and GCI mesh refinement methodology. Section 4 documents the implementation: mesh enrichment, element assembly, boundary condition handling, and the Picard iteration algorithm for the Navier-Stokes problem. Section 5 documents verification (Newtonian, Ghia et al. [8]) and validation (TVP, Begum et al. [1]). Section 6 discusses the results quantitatively. Section 7 summarises findings and identifies directions for future work.

All source code, mesh files, and post-processing scripts are made available in the accompanying repository https://github.com/samannazir55/thixo_twin.

2 Mathematical Formulation

This section develops the governing equations in three progressive layers: the Newtonian incompressible Navier-Stokes system (Section 2.1), the generalised Newtonian extension with power-law and Bingham regularisation (Section 2.2), and the full TVP model (Section 2.3). All derivations follow the notation of Begum et al. [1] to facilitate direct comparison.

2.1 Incompressible Navier-Stokes Equations

Let $\Omega \subset \mathbb{R}^2$ be a bounded polygonal domain with boundary $\partial\Omega$. The steady, incompressible Navier-Stokes equations read:

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

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

where $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ is the velocity field, $p : \Omega \rightarrow \mathbb{R}$ the pressure (modified to absorb the hydrostatic contribution when $\mathbf{f}$ is gravitational), $\rho > 0$ the constant fluid density, $\boldsymbol{\tau}$ the deviatoric (extra) stress tensor, and $\mathbf{f}$ a body force (set to zero in the cavity problem).

For a Newtonian fluid,

$$\boldsymbol{\tau} = 2\mu \mathbf{D}(\mathbf{u}), \quad \mathbf{D}(\mathbf{u}) := \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T], \quad (3)$$

where $\mu > 0$ is the dynamic viscosity and $\mathbf{D}$ is the symmetric rate-of-strain tensor. Substituting (3) into (1) and introducing the kinematic viscosity $\nu = \mu/\rho$ gives the standard

Navier-Stokes momentum equation,

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u} + \mathbf{f}. \quad (4)$$

The dimensionless form is obtained by scaling lengths with the cavity side length L , velocities with the lid speed U , and pressure with ρU^2 . The single dimensionless parameter is the Reynolds number,

$$\text{Re} := \frac{UL}{\nu}. \quad (5)$$

In dimensionless variables equations (1)–(2) retain the same form with μ replaced by Re^{-1} and $\rho = 1$.

Boundary conditions. The cavity occupies $\Omega = (0, 1)^2$. No-slip conditions are applied on the bottom and side walls, $\mathbf{u} = \mathbf{0}$ on $\{y = 0\} \cup \{x = 0\} \cup \{x = 1\}$. The driven lid at $y = 1$ moves with unit horizontal velocity, $\mathbf{u} = (1, 0)^T$ on $\{y = 1\}$. A regularised lid profile,

$$u_{\text{lid}}(x) = 16x^2(1-x)^2, \quad (6)$$

is optionally adopted to suppress corner singularities; its use is noted explicitly whenever applied. Pressure is pinned at a single degree of freedom to remove the hydrostatic null-space.

2.2 Generalised Newtonian Model

In a generalised Newtonian (GN) fluid the viscosity is a scalar function of the local shear rate. The extra stress retains the Newtonian form (3) but with a variable effective viscosity η :

$$\boldsymbol{\tau} = 2\eta(\dot{\gamma}) \mathbf{D}(\mathbf{u}), \quad (7)$$

where the scalar shear rate $\dot{\gamma}$ is defined as

$$\dot{\gamma} := \sqrt{2 \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{u})} = \sqrt{2 D_{ij} D_{ij}}. \quad (8)$$

2.2.1 Power-Law Model

The power-law (Ostwald-de Waele) viscosity is

$$\eta(\dot{\gamma}) = K \dot{\gamma}^{n-1}, \quad (9)$$

where $K > 0$ is the consistency index and $n > 0$ the power-law index. The case $n = 1$ recovers the Newtonian model with $\eta = K = \mu$. Shear-thinning corresponds to $n < 1$; shear-thickening to $n > 1$.

2.2.2 Bingham Model and Papanastasiou Regularisation

A Bingham fluid yields only where the magnitude of the deviatoric stress exceeds a yield stress $\tau_0 \geq 0$. The constitutive law is

$$\boldsymbol{\tau} = \begin{cases} \left(2\mu_p + \frac{\tau_0}{\dot{\gamma}}\right) \mathbf{D}(\mathbf{u}) & \text{if } |\boldsymbol{\tau}| > \tau_0, \\ \mathbf{D}(\mathbf{u}) = \mathbf{0} & \text{if } |\boldsymbol{\tau}| \leq \tau_0, \end{cases} \quad (10)$$

where μ_p is the plastic viscosity and τ_0 the yield stress. The Bingham number,

$$\text{Bn} := \frac{\tau_0 L}{\mu_p U}, \quad (11)$$

measures the ratio of yield stress to viscous stress.

The implicit definition (10) is unsuitable for Newton-based FEM because the yield surface is not known *a priori*. Papanastasiou [13] introduced the regularised viscosity

$$\eta_P(\dot{\gamma}) = \mu_p + \frac{\tau_0}{\dot{\gamma}} [1 - e^{-m\dot{\gamma}}], \quad (12)$$

where $m > 0$ is a regularisation parameter. As $m \rightarrow \infty$ the Papanastasiou model converges pointwise to the Bingham model outside the yield surface. In all computations m is chosen sufficiently large that the relative difference in the predicted yield surface position is below 0.1%; the value $m = 10^4$ s is used throughout and its effect on the yield surface position is quantified in Stage 4B (Section 5.4).

2.3 Thixoviscoplastic (TVP) Model: Houska Formulation

The specific TVP model implemented and verified in this work is the *Houska model* (Begum, Ouazzi, and Turek [1], Table 1, row used for Figure 15). It augments the Papanastasiou-regularised Bingham framework with a scalar microstructure parameter $\lambda(\mathbf{x}) \in [0, 1]$, where $\lambda = 1$ is the fully structured (at-rest) state and $\lambda = 0$ the fully broken-down state. All equations below correspond exactly to those assembled in `utils/tvp_assembly.py` and verified by seven finite-difference Jacobian checks in Stage 5.3 (Section 5.5.3).

2.3.1 Material Functions

The Houska model specifies two scalar material functions of λ alone:

$$\eta(\lambda) = \eta_0, \quad \tau(\lambda) = \tau_0 + \tau_\infty \lambda, \quad (13)$$

where $\eta_0 > 0$ is the (constant) plastic viscosity, $\tau_0 \geq 0$ the solvent yield stress, and $\tau_\infty \geq 0$ the structural yield-stress contribution. The fully structured state carries yield stress $\tau_0 + \tau_\infty$; the fully broken-down state carries only τ_0 . This corresponds to the Houska row of Table 1 in Begum et al. [1] with $\eta_\infty = 0$ and power-law index $n = 1$.

2.3.2 Extended Viscosity (Papanastasiou Regularisation)

Following paper eq. (3), the Papanastasiou-regularised extended viscosity is

$$\mu(\dot{\gamma}, \lambda) = \eta(\lambda) + \tau(\lambda) \frac{\sqrt{2}}{2} \frac{1}{\sqrt{D_{II}}} [1 - e^{-k\sqrt{D_{II}}}], \quad (14)$$

where $D_{II} = \frac{1}{2} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{u})$ is the second invariant of the rate-of-strain tensor (3) and $k > 0$ is the Papanastasiou regularisation constant. Substituting $\sqrt{D_{II}} = \dot{\gamma}/\sqrt{2}$ (since $\dot{\gamma} = \sqrt{2} \mathbf{D} : \mathbf{D}$ by (8)) gives the working form used in the implementation:

$$\mu(\dot{\gamma}, \lambda) = \eta_0 + \frac{\tau_0 + \tau_\infty \lambda}{\dot{\gamma}} [1 - e^{-m_{\text{eff}} \dot{\gamma}}], \quad m_{\text{eff}} = \frac{k}{\sqrt{2}}. \quad (15)$$

Equation (15) is identical in form to the Papanastasiou model (12) with $\mu_p \rightarrow \eta_0$ and $\tau_0 \rightarrow \tau_0 + \tau_\infty \lambda$, confirming the Bingham limit when $\lambda \equiv 1$ and $\tau_\infty = 0$. The λ -derivative required for the Newton Jacobian block $\mathbf{J}_{u\lambda}$ (Section 5.5.3) is

$$\frac{\partial \mu}{\partial \lambda} = \frac{\tau_\infty}{\dot{\gamma}} [1 - e^{-m_{\text{eff}} \dot{\gamma}}], \quad (16)$$

which is constant with respect to λ because τ_∞ does not depend on λ in the Houska model.

2.3.3 Microstructure Evolution Equation

The microstructure parameter λ satisfies the scalar transport-reaction equation (paper eqs. (2) and (7)):

$$\frac{\partial \lambda}{\partial t} + (\mathbf{u} \cdot \nabla) \lambda = F(\lambda) - G(\lambda, \dot{\gamma}), \quad (17)$$

where the buildup (aging) and breakdown (rejuvenation) rates are

$$F(\lambda) = M_a (1 - \lambda), \quad G(\lambda, \dot{\gamma}) = M_b \lambda \dot{\gamma}, \quad (18)$$

with $M_a > 0$ the aging rate constant and $M_b > 0$ the breakdown rate constant. Rearranging the right-hand side, the steady form ($\partial \lambda / \partial t = 0$) used in the implementation is

$$(\mathbf{u} \cdot \nabla) \lambda + (M_b \dot{\gamma} + M_a) \lambda = M_a, \quad (19)$$

a linear advection-reaction equation in λ for a given velocity field $\mathbf{u}$. At every spatial point the local equilibrium value λ^* satisfies $F(\lambda^*) = G(\lambda^*, \dot{\gamma})$, giving the closed-form expression

$$\lambda^*(\dot{\gamma}) = \frac{M_a}{M_a + M_b \dot{\gamma}}. \quad (20)$$

This is the formula verified in Stage 5.1 (Section 5.5.1, eq. (71)) with the relabelling $a \rightarrow M_b$, $b \rightarrow M_a$.

2.3.4 Coupled Steady System

The complete steady Houska TVP problem is: find $(\mathbf{u}, p, \lambda)$ such that

$$\rho(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nabla \cdot [2\mu(\dot{\gamma}, \lambda) \mathbf{D}(\mathbf{u})] = \mathbf{0} \quad \text{in } \Omega, \quad (21)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (22)$$

$$(\mathbf{u} \cdot \nabla) \lambda + (M_b \dot{\gamma} + M_a) \lambda = M_a \quad \text{in } \Omega, \quad (23)$$

with $\mu(\dot{\gamma}, \lambda)$ given by (15). The system is nonlinear through the viscosity term in (21) (via $\dot{\gamma}$ and λ) and through the advection field $\mathbf{u}$ in (23). For the lid-driven cavity, $\partial \Omega$ admits no convective inflow, so (23) requires no boundary data on λ ; the microstructure adjusts freely to the interior shear field.

2.3.5 Limiting Cases

Table 1 summarises the hierarchy of models that emerge from the Houska system by specialising parameters.

Table 1: Model hierarchy obtained from the Houska TVP system (21)–(23) by specialising parameters. All limits are expressed in terms of the Houska parameters η_0 , τ_0 , τ_∞ , M_a , M_b .

Model	Conditions	Ref.
Newtonian	$\tau_0 = \tau_\infty = 0$,	[8]
	$M_b = 0$ ($\lambda \equiv 1$), $\eta_0 = \mu$	
Bingham–Papanastasiou	$\tau_0 = 0$, $M_b = 0$ ($\lambda \equiv 1$),	[13]
	$\mu_p = \eta_0$, $\tau_y = \tau_0$	
TVP, Bingham limit	$M_b = 0$: $F = G = 0$ forces $\lambda \equiv 1$, yield stress $= \tau_0 + \tau_\infty$ (fully structured)	(15)
Houska TVP (steady)	Full system (21)–(23), all parameters active	[1]

The Figure 15 parameters of Begum et al. [1] used in the verification study of Section 5.5.3 are $\eta_0 = \tau_0 = \tau_\infty = 1$ Pa, $k = 10^4$ s, $M_a = 1$ s $^{-1}$, $M_b = 0.1$ s $^{-1}$, $\rho = 1$ kg/m ^{3} , $U_{\text{lid}} = 1$ m/s. These are reproduced in full in Section 5.

3 Numerical Methodology

This section describes the discrete counterpart of the continuous problem stated in Section 2. The weak formulation of the Stokes limit is derived first (Section 3.1), because it is the assembled system verified in Section 5. The finite element spaces and their inf-sup stability are discussed in Section 3.2. The reference element, geometry map, and Gauss quadrature rules are detailed in Section 3.3. The global saddle-point system and its sign convention are stated in Section 3.4. Grid convergence assessment via the GCI methodology is described in Section 3.8. Implementation details (mesh enrichment, element assembly, boundary condition enforcement, and the nonlinear solver) are covered in Section 4.

3.1 Weak Formulation

In the Stokes limit the convective term in (1) is dropped and $\boldsymbol{\tau} = 2\eta \boldsymbol{\varepsilon}(\mathbf{u})$, where

$$\boldsymbol{\varepsilon}(\mathbf{u}) := \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \quad (24)$$

is the symmetric strain-rate tensor (identical to $\mathbf{D}(\mathbf{u})$ defined in (3); the symbol $\boldsymbol{\varepsilon}$ is used hereafter to emphasise the FEM context). Define the function spaces

$$\mathbf{V} := [H_{0,\Gamma_D}^1(\Omega)]^2, \quad Q := L_0^2(\Omega) = \left\{ q \in L^2(\Omega) \mid \int_\Omega q \, d\Omega = 0 \right\}, \quad (25)$$

where H_{0,Γ_D}^1 denotes the Sobolev space with zero trace on the Dirichlet boundary Γ_D . The Stokes weak problem reads: find $(\mathbf{u}, p) \in \mathbf{V} \times Q$ such that for all $(\mathbf{v}, q) \in \mathbf{V} \times Q$,

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \ell(\mathbf{v}), \quad (26)$$

$$b(\mathbf{u}, q) = 0, \quad (27)$$

with bilinear forms

$$a(\mathbf{u}, \mathbf{v}) := \int_{\Omega} 2\eta \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, d\Omega, \quad (28)$$

$$b(\mathbf{v}, q) := - \int_{\Omega} q \nabla \cdot \mathbf{v} \, d\Omega, \quad (29)$$

and linear functional

$$\ell(\mathbf{v}) := \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega. \quad (30)$$

The double-dot contraction $\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) = \varepsilon_{ij}(\mathbf{u}) \varepsilon_{ij}(\mathbf{v})$ (Einstein summation) expands componentwise as

$$2\eta \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) = \eta \left(2 \frac{\partial u_x}{\partial x} \frac{\partial v_x}{\partial x} + \frac{\partial u_x}{\partial y} \frac{\partial v_x}{\partial y} + \frac{\partial u_y}{\partial x} \frac{\partial v_x}{\partial y} + \frac{\partial u_y}{\partial x} \frac{\partial v_y}{\partial x} + \frac{\partial u_x}{\partial y} \frac{\partial v_y}{\partial x} + 2 \frac{\partial u_y}{\partial y} \frac{\partial v_y}{\partial y} \right). \quad (31)$$

The four element-level stiffness sub-blocks that arise from (31) for trial node b and test node a , denoting $\partial_x \varphi_k \equiv \partial \varphi_k / \partial x$, are

$$K_{uu}^{ab} = \eta (2 \partial_x \varphi_a \partial_x \varphi_b + \partial_y \varphi_a \partial_y \varphi_b), \quad (32)$$

$$K_{uv}^{ab} = \eta \partial_y \varphi_a \partial_x \varphi_b, \quad (33)$$

$$K_{vu}^{ab} = \eta \partial_x \varphi_a \partial_y \varphi_b, \quad (34)$$

$$K_{vv}^{ab} = \eta (\partial_x \varphi_a \partial_x \varphi_b + 2 \partial_y \varphi_a \partial_y \varphi_b). \quad (35)$$

These expressions are implemented directly in `assembly.py` (lines Kuu–Kvv in the element loop of `assemble_stokes`).

3.2 Finite Element Spaces and Inf-Sup Stability

The discrete problem requires a velocity-pressure pair that satisfies the inf-sup (Ladyzhenskaya-Babuška-Brezzi, LBB) condition to guarantee a unique, pressure-stable solution [4]. The Taylor-Hood $\mathcal{Q}_2/\mathcal{Q}_1$ pair is used throughout this work: biquadratic ($\mathcal{Q}_2$) velocity on a 9-node reference square, and bilinear ($\mathcal{Q}_1$) pressure on the 4-node corners of the same element. This pair is inf-sup stable on shape-regular quadrilateral meshes [2], and is the natural stepping stone toward the Turek group's rotated $\mathcal{Q}_1/\mathcal{Q}_0$ element used in FEAT-FLOW [16].

The velocity space $\mathbf{V}_h \subset \mathbf{V}$ has $2n_u$ degrees of freedom, where n_u is the number of $\mathcal{Q}_2$ nodes. The pressure space $Q_h \subset Q$ has n_p degrees of freedom, where n_p is the number of $\mathcal{Q}_1$ corner nodes. For a structured $n \times n$ mesh of quadrilaterals: $n_u = (2n+1)^2$ and $n_p = (n+1)^2$, giving a velocity-to-pressure DOF ratio of approximately 4 : 1 for large n .

The DOF ordering used in the implementation interleaves the two velocity components per node:

$$\mathbf{x} = [u_x^{(0)}, u_y^{(0)}, u_x^{(1)}, u_y^{(1)}, \dots, u_x^{(n_u-1)}, u_y^{(n_u-1)}, p^{(0)}, \dots, p^{(n_p-1)}]^T, \quad (36)$$

with velocity DOF index $2i$ for u_x and $2i+1$ for u_y at $\mathcal{Q}_2$ node i , and pressure DOFs appended as a contiguous block after all velocity DOFs. This layout corresponds exactly to the array `gdofs_u` and the block structure assembled in `assembly.py`.

3.3 Reference Element, Geometry Map, and Quadrature

3.3.1 Reference Element Layout

All integrals are evaluated on the reference square $\hat{\Omega} = [-1, 1]^2$ with reference coordinates $(\hat{\xi}, \hat{\eta})$. The local node numbering for $\mathcal{Q}_2$ velocity (9 nodes) and $\mathcal{Q}_1$ geometry/pressure (4 corner nodes) is shown in Figure 1 and encoded in `elements.py`.

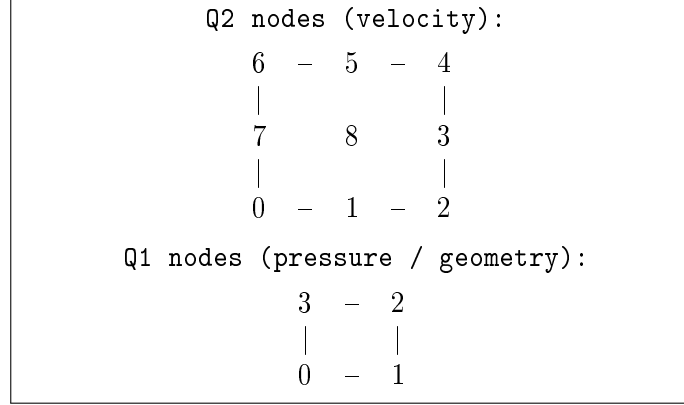


Figure 1: Local node numbering on the reference square $[-1, 1]^2$ for the $\mathcal{Q}_2$ velocity (top) and $\mathcal{Q}_1$ pressure/geometry (bottom) elements. Coordinates: 0:(-1, -1), 1:(0, -1), 2:(1, -1), 3:(1, 0), 4:(1, 1), 5:(0, 1), 6:(-1, 1), 7:(-1, 0), 8:(0, 0).

3.3.2 Shape Functions

The $\mathcal{Q}_2$ shape functions are tensor products of one-dimensional Lagrange polynomials on the node set $\{-1, 0, 1\}$:

$$\varphi^{(ix, iy)}(\hat{\xi}, \hat{\eta}) = L_{ix}(\hat{\xi}) L_{iy}(\hat{\eta}), \quad (37)$$

where $ix, iy \in \{0, 1, 2\}$ index the 1-D nodes and

$$L_0(t) = \frac{1}{2}t(t-1), \quad L_1(t) = (1-t)(1+t), \quad L_2(t) = \frac{1}{2}t(t+1). \quad (38)$$

The $\mathcal{Q}_1$ shape functions (geometry map and pressure) are the standard bilinear basis,

$$\psi_0 = \frac{1}{4}(1-\hat{\xi})(1-\hat{\eta}), \quad \psi_1 = \frac{1}{4}(1+\hat{\xi})(1-\hat{\eta}), \quad \psi_2 = \frac{1}{4}(1+\hat{\xi})(1+\hat{\eta}), \quad \psi_3 = \frac{1}{4}(1-\hat{\xi})(1+\hat{\eta}). \quad (39)$$

Both sets of shape functions and their reference-coordinate gradients are implemented in `shape_q2` and `shape_q1` in `elements.py`.

3.3.3 Geometry Map and Jacobian

The physical coordinates $\mathbf{x} = (x, y)$ of a point $(\hat{\xi}, \hat{\eta}) \in \hat{\Omega}$ are mapped using the bilinear $\mathcal{Q}_1$ map through the four corner nodes $\{\mathbf{x}^{(c)}\}_{c=0}^3$:

$$\mathbf{x}(\hat{\xi}, \hat{\eta}) = \sum_{c=0}^3 \psi_c(\hat{\xi}, \hat{\eta}) \mathbf{x}^{(c)}. \quad (40)$$

The Jacobian matrix $\mathbf{J} \in \mathbb{R}^{2 \times 2}$ and its determinant are

$$J_{ij} = \frac{\partial x_i}{\partial \hat{x}_j} = \sum_{c=0}^3 x_i^{(c)} \frac{\partial \psi_c}{\partial \hat{x}_j}, \quad \det \mathbf{J} = J_{00}J_{11} - J_{01}J_{10}. \quad (41)$$

Physical gradients of the velocity shape functions are obtained by the chain rule,

$$\nabla_{\mathbf{x}} \varphi_k = \mathbf{J}^{-T} \nabla_{\hat{\mathbf{x}}} \varphi_k, \quad (42)$$

computed as $\nabla_{\mathbf{x}} \varphi_k = \nabla_{\hat{\mathbf{x}}} \varphi_k \cdot \mathbf{J}^{-T}$ in the code (`dn2_ref @ Jinv.T, assembly.py`). For parallelogram elements (structured meshes) $\mathbf{J}$ is constant over each element; for general quadrilaterals it varies with $(\hat{\xi}, \hat{\eta})$ and must be re-evaluated at each quadrature point.

3.3.4 Gauss Quadrature

The element integrals are computed by tensor-product Gauss-Legendre quadrature on $\hat{\Omega}$. Two quadrature rules are used, consistent with the polynomial degrees of the integrands:

- **3×3 rule** (9 points): used for the viscous block $\mathbf{A}^e$ and the body-force vector $\mathbf{F}^e$. After applying the chain rule (42), the integrand of $a(\mathbf{u}_h, \mathbf{v}_h)$ on a general quadrilateral contains products of physical gradients of $\mathcal{Q}_2$ functions scaled by $\det \mathbf{J}$. The physical gradients of $\mathcal{Q}_2$ functions are rational functions of $(\hat{\xi}, \hat{\eta})$ of effective degree up to 4 in each coordinate direction; the 3×3 rule integrates polynomials of degree ≤ 5 exactly, which is sufficient for parallelogram (constant- $\mathbf{J}$) elements.
- **2×2 rule** (4 points): used for the divergence block $\mathbf{B}^e$. The integrand involves products of a $\mathcal{Q}_1$ pressure shape function (ψ_i , degree 1) and a physical gradient of a $\mathcal{Q}_2$ velocity shape function (effective degree up to 3 after the geometry map); the 2×2 rule integrates polynomials of degree ≤ 3 exactly.

Quadrature points and weights are generated by `gauss_points_2d(n)` in `elements.py` via `numpy.polynomial.legendre.leggauss`. Table 2 summarises the two rules.

Table 2: Gauss-Legendre quadrature rules used in element integration. n_q denotes the number of quadrature points.

Rule	n_q	Exact degree (each dir.)	Used for
3×3	9	≤ 5	$\mathbf{A}^e, \mathbf{F}^e$
2×2	4	≤ 3	$\mathbf{B}^e$

3.4 Global Saddle-Point System

After assembly and application of DOF ordering (36), the discrete Stokes problem takes the block saddle-point form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}, \quad (43)$$

where $\mathbf{A} \in \mathbb{R}^{2n_u \times 2n_u}$ is the symmetric positive semi-definite viscous block, $\mathbf{B} \in \mathbb{R}^{n_p \times 2n_u}$ the discrete divergence operator, $\mathbf{u} \in \mathbb{R}^{2n_u}$ the velocity DOF vector, $\mathbf{p} \in \mathbb{R}^{n_p}$ the pressure DOF vector, and $\mathbf{F}$ the body-force vector. The system is assembled by `build_saddle_point_system` in `assembly.py`.

Sign convention for $\mathbf{B}$. The bilinear form $b(\mathbf{v}, q)$ defined in (29) carries a negative sign. In the element matrix $\mathbf{B}^e$ the entry for pressure shape function i and velocity shape function b , component $\ell \in \{x, y\}$, is

$$B_{i,(b,\ell)}^e = - \int_{\hat{\Omega}} \psi_i \frac{\partial \varphi_b}{\partial x_\ell} \det \mathbf{J} \, d\hat{\Omega}, \quad (44)$$

where the negative sign comes directly from (29). The assembled global $\mathbf{B}$ therefore satisfies

$$\mathbf{B} \mathbf{u} \approx - \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h \, d\Omega, \quad (45)$$

and the off-diagonal block in (43) is $+\mathbf{B}^T$ (not $-\mathbf{B}^T$), so the weak momentum equation reads $\mathbf{A} \mathbf{u} + \mathbf{B}^T \mathbf{p} = \mathbf{F}$, which matches the physical form $a(\mathbf{u}_h, \mathbf{v}_h) - \int p_h \nabla \cdot \mathbf{v}_h = \ell(\mathbf{v}_h)$ once the negative sign has been absorbed into $\mathbf{B}$. This convention is stated explicitly in the docstring of `assemble_stokes` and is preserved throughout the solver.

The sparse matrix $\mathbf{A}$ is assembled in CSR format from COO (coordinate) accumulators; $\mathbf{B}$ likewise. The full saddle-point matrix $\mathbf{K}$ is formed via `scipy.sparse.bmat` as shown in (43).

3.5 Boundary Conditions and Pressure Normalisation

3.5.1 Dirichlet Velocity Conditions

Dirichlet velocity conditions are enforced by symmetric row-and-column elimination [3]. For each constrained DOF k with prescribed value g_k , the algorithm proceeds as follows:

1. **Lifting:** for every unconstrained row i , subtract $K_{ik} g_k$ from F_i , using the original matrix entry K_{ik} before it is zeroed. This step modifies the RHS only for non-homogeneous constraints ($g_k \neq 0$) and preserves the solution of the full system.
2. **Row/column zeroing:** set row k and column k of $\mathbf{K}$ to zero.
3. **Diagonal restoration:** set $K_{kk} = 1$.
4. **RHS:** set $F_k = g_k$.

This procedure is implemented in `apply_dirichlet_velocity` in `bc_handling.py`. The matrix is converted from CSR to LIL format for efficient row editing, then to CSC for column zeroing, and back to CSR for the linear solve; `eliminate_zeros` is called after each conversion to remove structural zeros.

For the lid-driven cavity, the boundary condition function passed to `get_boundary_velocity_dofs` returns $\mathbf{u} = (1, 0)^T$ on $\{y = 1\}$ and $\mathbf{u} = (0, 0)^T$ on the remaining three walls.

3.5.2 Pressure Null-Space Removal

With pure Dirichlet velocity conditions on all of $\partial\Omega$ the discrete inf-sup condition guarantees that $\mathbf{B}$ is surjective onto the pressure space, but the pressure is determined only up to an additive constant. This constant null-space is removed by pinning one pressure DOF to zero: the global DOF at index $2n_u + k_p$ (where $k_p = 0$ by default, corresponding to the first pressure node) is treated as an additional Dirichlet condition $p = 0$ via `pin_pressure_dof` in `bc_handling.py`. The physical interpretation is that the mean cavity pressure is set to zero, consistent with the convention of Begum et al. [1].

3.6 Mesh Construction: Q1 to Q2 Enrichment

The base mesh is a structured $\mathcal{Q}_1$ quadrilateral mesh on $\Omega = [0, 1]^2$. An $n \times n$ mesh has $(n+1)^2$ corner nodes and n^2 elements. The $\mathcal{Q}_2$ mesh is constructed by `extend_q1_to_q2` in `q2_mesh.py`:

1. Corner nodes retain their original global indices $(0, \dots, n_u^{Q_1} - 1)$.
2. For each interior edge shared by two elements, a single midpoint node is added and its global index is recorded in a dictionary keyed on the sorted pair of corner-node indices. This deduplication ensures that adjacent elements share the correct midpoint DOF.
3. Each element receives one centroid node (local index 8), which is never shared.

The resulting $\mathcal{Q}_2$ mesh has $n_u = (2n+1)^2$ velocity nodes and $n_p = (n+1)^2$ pressure nodes (corner nodes only). Table 3 lists the system sizes for the three refinement levels used in the GCI study.

Table 3: Mesh sizes and system dimensions for the structured $\mathcal{Q}_2/\mathcal{Q}_1$ cavity meshes. $h = 1/n$ is the uniform element size. $N_{\text{dof}} = 2n_u + n_p$.

n	h	n_u	n_p	N_{dof}	Elements
8	0.1250	289	81	659	64
16	0.0625	1089	289	2467	256
32	0.0312	4225	1089	9539	1024
64	0.0156	16641	4225	37507	4096

3.7 Newton Linearisation for the Nonlinear System

For the full Navier-Stokes and TVP problems the system (43) is replaced by a nonlinear residual $\mathcal{R}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{x} = (\mathbf{u}, \mathbf{p})$ (and later λ). Newton’s method computes iterates

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \delta \mathbf{x}^{(k)}, \quad \mathbf{J}_{\mathcal{R}}(\mathbf{x}^{(k)}) \delta \mathbf{x}^{(k)} = \mathcal{R}(\mathbf{x}^{(k)}), \quad (46)$$

where $\mathbf{J}_{\mathcal{R}}$ is the Jacobian (tangent matrix) of the residual. For the Navier-Stokes problem the linearisation of the convective term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ yields the Oseen operator; for the TVP problem an additional contribution arises from the derivative of $\eta(\dot{\gamma}, \lambda)$ with respect to $\dot{\gamma}$. The exact form of $\mathbf{J}_{\mathcal{R}}$ for the TVP case follows the monolithic approach of Begum et al. [1] and will be detailed in the companion module documentation.

The convergence criterion is

$$\frac{\|\mathcal{R}(\mathbf{x}^{(k)})\|_2}{\|\mathcal{R}(\mathbf{x}^{(0)})\|_2} < \varepsilon_{\text{Newton}}, \quad \varepsilon_{\text{Newton}} = 10^{-10}. \quad (47)$$

The initial guess $\mathbf{x}^{(0)}$ is the zero vector for $\text{Re} \leq 100$; for higher Reynolds numbers a continuation strategy is used, taking the converged solution at the previous Re as the initial guess.

3.8 Grid Convergence Index

Discretisation error is quantified using the Grid Convergence Index (GCI) of Celik et al. [5]. Let ϕ_1, ϕ_2, ϕ_3 denote a scalar quantity of interest (e.g. the u -velocity at a centreline monitor point) on three successively refined meshes with characteristic element sizes $h_1 < h_2 < h_3$ (mesh 1 is the finest) and constant refinement ratio $r = h_2/h_1 = h_3/h_2 = 2$. The observed convergence order is

$$\hat{p} = \frac{\ln[(\phi_3 - \phi_2)/(\phi_2 - \phi_1)]}{\ln r}, \quad (48)$$

and the GCI on the finest mesh (between meshes 1 and 2) is

$$\text{GCI}_{12} = \frac{F_s |(\phi_2 - \phi_1)/\phi_1|}{r^{\hat{p}} - 1}, \quad (49)$$

where $F_s = 1.25$ is the standard safety factor [5]. A value $\text{GCI}_{12} < 5\%$ indicates that the solution is mesh-independent to engineering accuracy. For the $\mathcal{Q}_2/\mathcal{Q}_1$ element pair, the expected asymptotic convergence orders are $\hat{p} \approx 3$ in $L^2(\Omega)$ velocity norm, and $\hat{p} \approx 2$ in $H^1(\Omega)$ velocity norm and $L^2(\Omega)$ pressure norm, consistent with the manufactured-solution convergence rates confirmed in the verification study (Section 5, Stage 2).

4 Implementation

This section documents the four implementation modules that together constitute the Stokes and Navier-Stokes solver pipeline. Each module is self-contained and independently verified before the next is assembled, following the modular design principle described in the Introduction. Section 4.1 covers mesh enrichment (`q2_mesh.py`). Section 4.2 covers element matrix assembly (`assembly.py`). Section 4.3 covers boundary condition handling (`bc_handling.py`). Section 4.4 covers the Picard iteration algorithm for the nonlinear Navier-Stokes problem (`03_navier_stokes.py`).

4.1 Mesh Enrichment: `q2_mesh.py`

The base mesh for the cavity is a structured $\mathcal{Q}_1$ quadrilateral mesh on $\Omega = [0, 1]^2$, generated by `structured_channel_mesh` in `q2_mesh.py`. An $n \times n$ mesh has $(n + 1)^2$ corner nodes and n^2 elements, with corner-node indices running row-by-row from the bottom-left corner. The $\mathcal{Q}_2$ velocity mesh is constructed from this base by `extend_q1_to_q2` in three steps.

Step 1: corner-node inheritance. The $(n + 1)^2$ corner nodes of the $\mathcal{Q}_1$ mesh retain their original global indices $0, \dots, n_u^{(Q1)} - 1$. The local slots 0, 2, 4, 6 of each $\mathcal{Q}_2$ element (corresponding to the four corners in the reference layout of Figure 1) are filled directly from the $\mathcal{Q}_1$ connectivity array.

Step 2: edge-midpoint deduplication. Each of the four edges of an element contributes one midpoint node at the arithmetic mean of its two endpoint coordinates. An interior edge is shared by exactly two elements; its midpoint must map to the same global

DOF in both. This is enforced by a dictionary keyed on the sorted pair of corner-node indices (a, b) with $a < b$:

$$\text{key} = (\min(a, b), \max(a, b)) \mapsto \text{global midpoint index.} \quad (50)$$

When an edge is first encountered its midpoint coordinate is appended to the node list and a new index is assigned. When the same edge is encountered from the adjacent element the dictionary lookup returns the existing index without creating a duplicate node. Boundary edges (belonging to only one element) are handled identically; they simply never appear a second time in the loop.

Step 3: centroid nodes. Local slot 8 (the element centroid) is never shared between elements. Its coordinate is the arithmetic mean of the four corner coordinates,

$$\mathbf{x}_e^{(8)} = \frac{1}{4}(\mathbf{x}_e^{(0)} + \mathbf{x}_e^{(1)} + \mathbf{x}_e^{(2)} + \mathbf{x}_e^{(3)}), \quad (51)$$

and a new global index is assigned unconditionally for each element e .

Node counts. For an $n \times n$ mesh the enrichment produces:

- $n_{\text{corner}} = (n + 1)^2$ corner nodes,
- $n_{\text{edge}} = 2n(n + 1)$ edge-midpoint nodes (horizontal edges: $n(n + 1)$; vertical edges: $n(n + 1)$),
- $n_{\text{centroid}} = n^2$ centroid nodes.

The total is

$$n_u = (n + 1)^2 + 2n(n + 1) + n^2 = (2n + 1)^2, \quad (52)$$

consistent with the theoretical count for a biquadratic mesh. The pressure node count remains $n_p = (n + 1)^2$ (corner nodes only). Table 3 (Section 3.8) lists the resulting system sizes.

4.2 Element Matrix Assembly: `assembly.py`

4.2.1 Global assembly strategy

Global matrices are assembled by the standard scatter-accumulate procedure. For each element e , the 18×18 element viscous matrix $\mathbf{A}^e$, the 4×18 element divergence matrix $\mathbf{B}^e$, and the 18-vector $\mathbf{F}^e$ are computed on the reference square $\hat{\Omega}$ via Gauss quadrature and scattered into global COO (coordinate-format) accumulators. After the element loop the accumulators are converted to CSR format via `scipy.sparse.csr_matrix`, which automatically sums repeated index pairs, correctly accumulating contributions from shared DOFs. The full saddle-point matrix $\mathbf{K}$ is then assembled by `scipy.sparse.bmat` as in (43).

4.2.2 Viscous block $\mathbf{A}^e$: 3×3 quadrature

The element viscous matrix is computed with the 3×3 Gauss rule (Table 2). At each quadrature point $(\hat{\xi}_q, \hat{\eta}_q)$ with weight w_q :

1. Evaluate $\mathcal{Q}_2$ shape functions φ_k and reference gradients $\nabla_{\hat{\mathbf{x}}} \varphi_k$ (`shape_q2`).
2. Evaluate $\mathcal{Q}_1$ shape functions ψ_c and reference gradients (`shape_q1`), compute the Jacobian $\mathbf{J}$, its determinant $\det \mathbf{J}$, and its inverse $\mathbf{J}^{-1}$ by (41).
3. Compute physical gradients $\nabla_{\mathbf{x}} \varphi_k = \nabla_{\hat{\mathbf{x}}} \varphi_k \mathbf{J}^{-T}$ (`dN2_ref @ Jinv.T`).
4. Accumulate the 18×18 contributions using the four sub-blocks (32)–(35):

$$A_{(2a,2b)}^e += K_{uu}^{ab} w_q \det \mathbf{J}, \quad A_{(2a,2b+1)}^e += K_{uv}^{ab} w_q \det \mathbf{J}, \quad (53)$$

$$A_{(2a+1,2b)}^e += K_{vu}^{ab} w_q \det \mathbf{J}, \quad A_{(2a+1,2b+1)}^e += K_{vv}^{ab} w_q \det \mathbf{J}, \quad (54)$$

for $a, b \in \{0, \dots, 8\}$.

4.2.3 Divergence block $\mathbf{B}^e$: 2×2 quadrature and sign convention

The element divergence matrix is computed with the 2×2 Gauss rule (Table 2). At each quadrature point, physical gradients of φ_b are computed as above, and the 4×18 element matrix is accumulated as

$$B_{i,2b}^e += -\psi_i \frac{\partial \varphi_b}{\partial x} w_q \det \mathbf{J}, \quad B_{i,2b+1}^e += -\psi_i \frac{\partial \varphi_b}{\partial y} w_q \det \mathbf{J}, \quad (55)$$

for $i \in \{0, \dots, 3\}$ (pressure nodes) and $b \in \{0, \dots, 8\}$ (velocity nodes).

Remark 1 (Sign convention in $\mathbf{B}$). *The negative sign in (55) comes from the bilinear form $b(\mathbf{v}, q) = -\int_{\Omega} q \nabla \cdot \mathbf{v} d\Omega$ defined in (29). Absorbing this sign into $\mathbf{B}$ means the saddle-point system (43) uses $+\mathbf{B}^T$ in the (1,2) block, not $-\mathbf{B}^T$. The assembled global operator therefore satisfies $\mathbf{B}\mathbf{u} \approx -\int_{\Omega} q_h \nabla \cdot \mathbf{u}_h d\Omega$, and the momentum row reads $\mathbf{A}\mathbf{u} + \mathbf{B}^T \mathbf{p} = \mathbf{F}$, which is consistent with the continuous form $a(\mathbf{u}_h, \mathbf{v}_h) - \int_{\Omega} p_h \nabla \cdot \mathbf{v}_h = \ell(\mathbf{v}_h)$ once the sign is accounted for. This convention is documented in the docstring of `assemble_stokes` and preserved across all nonlinear extensions of the solver.*

4.2.4 Body-force vector $\mathbf{F}^e$

The body-force vector is assembled simultaneously with $\mathbf{A}^e$ using the same 3×3 quadrature rule:

$$F_{2a}^e += \varphi_a f_x(\mathbf{x}_q) w_q \det \mathbf{J}, \quad F_{2a+1}^e += \varphi_a f_y(\mathbf{x}_q) w_q \det \mathbf{J}, \quad (56)$$

where $\mathbf{x}_q = \sum_c \psi_c(\hat{\xi}_q, \hat{\eta}_q) \mathbf{x}^{(c)}$ is the physical coordinate of the quadrature point. For the cavity problem $\mathbf{f} = \mathbf{0}$, so $\mathbf{F} = \mathbf{0}$. The argument `f_func` in `assemble_stokes` accepts any callable `f_func(x, y) -> (fx, fy)`, enabling the manufactured-solution verification study (Section 5).

4.3 Boundary Conditions: `bc_handling.py`

4.3.1 Dirichlet velocity conditions

Dirichlet velocity conditions are applied by symmetric row-and-column elimination [3], implemented in `apply_dirichlet_velocity`. The algorithm has four steps for each constrained DOF k with prescribed value g_k :

1. **Lifting.** For every free (unconstrained) row i , subtract $K_{ik}^{\text{orig}} g_k$ from F_i , using the *original* matrix entry before any zeroing. This is the standard lifting procedure for non-homogeneous constraints and leaves the solution of the modified system identical to that of the full system. For homogeneous constraints ($g_k = 0$) the lifting step is skipped.
2. **Row/column zeroing.** Row k and column k of $\mathbf{K}$ are set to zero.
3. **Diagonal restoration.** K_{kk} is set to 1.
4. **RHS assignment.** F_k is set to g_k .

The matrix is handled in three sparse formats across the procedure: CSR $\rightarrow$ LIL for efficient row editing, LIL $\rightarrow$ CSC for efficient column zeroing, CSC $\rightarrow$ CSR for the linear solve. After each format conversion `eliminate_zeros` is called to discard structural zeros and keep the matrix compact.

Boundary DOFs are identified geometrically by `get_boundary_velocity_dofs`, which iterates over all $\mathcal{Q}_2$ velocity nodes and flags those within tolerance $\epsilon = 10^{-12}$ of any of the four domain boundaries $\{x = 0\}$, $\{x = 1\}$, $\{y = 0\}$, $\{y = 1\}$. For the cavity problem the prescribed velocity function returns

$$\mathbf{u}_D(x, y) = \begin{cases} (1, 0)^T & \text{if } |y - 1| < \epsilon, \\ (0, 0)^T & \text{otherwise,} \end{cases} \quad (57)$$

yielding a moving lid at $y = 1$ and no-slip on the remaining three walls. The total number of constrained scalar DOFs is $2 \times (4n + 4 \cdot n + 4) - 4 \cdot 2 = 8n$ for an $n \times n$ mesh (each boundary edge contributes $2n + 1$ nodes but corners are shared; each node contributes 2 scalar DOFs).

4.3.2 Pressure null-space removal

With pure Dirichlet velocity conditions on the entire boundary $\partial\Omega$, the pressure is determined only up to an additive constant, because the discrete continuity equation $\mathbf{B}\mathbf{u} = \mathbf{0}$ is unaffected by a uniform pressure shift and $\mathbf{B}^T \mathbf{c} = \mathbf{0}$ for any constant vector $\mathbf{c}$. The null-space is removed by `pin_pressure_dof`, which applies the same four-step elimination to the global DOF $2n_u + k_p$ (default: $k_p = 0$, the first pressure node), setting $p_{k_p} = 0$. This is equivalent to choosing the pressure normalisation $\int_{\Omega} p_h \, d\Omega \approx 0$, consistent with the convention in Begum et al. [1].

4.4 Navier-Stokes Solver: Picard Iteration

The Stokes solver of Section 4.2 gives the solution to the linear system (43). For the full steady Navier-Stokes problem the convective term $\rho(\mathbf{u} \cdot \nabla)\mathbf{u}$ in (1) makes the system

nonlinear. Two linearisation strategies are standard: Picard (fixed-point) iteration, which replaces the current convective velocity by the iterate from the previous step, and Newton iteration (46), which uses the full Jacobian including the Newton derivative of the convective term. Picard iteration is implemented first in `03_navier_stokes.py` for two reasons: its linear convergence is robust for the moderate Reynolds numbers $\text{Re} \leq 1000$ of the cavity benchmark, and its tangent matrix has a simpler structure that facilitates debugging before the full Newton Jacobian is assembled.

4.4.1 Oseen linearisation

At iteration k , the velocity $\mathbf{u}^{(k)}$ from the previous step is treated as a frozen advection field. The convective term is linearised as

$$\rho(\mathbf{u}^{(k)} \cdot \nabla) \mathbf{u}^{(k+1)}, \quad (58)$$

yielding the *Oseen problem*: find $(\mathbf{u}^{(k+1)}, p^{(k+1)})$ such that for all $(\mathbf{v}, q)$,

$$c(\mathbf{u}^{(k)}; \mathbf{u}^{(k+1)}, \mathbf{v}) + a(\mathbf{u}^{(k+1)}, \mathbf{v}) + b(\mathbf{v}, p^{(k+1)}) = \ell(\mathbf{v}), \quad (59)$$

$$b(\mathbf{u}^{(k+1)}, q) = 0, \quad (60)$$

where the trilinear convective form is

$$c(\mathbf{w}; \mathbf{u}, \mathbf{v}) := \int_{\Omega} \rho(\mathbf{w} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} \, d\Omega. \quad (61)$$

4.4.2 Convection matrix $\mathbf{C}(\mathbf{u}^{(k)})$

The discrete counterpart of $c(\mathbf{u}^{(k)}; \cdot, \cdot)$ is the non-symmetric matrix $\mathbf{C}(\mathbf{u}^{(k)}) \in \mathbb{R}^{2n_u \times 2n_u}$. Its element-level entries for test node a and trial node b , with frozen advection field evaluated at quadrature points, are

$$C_{(2a,2b)}^e = \rho \varphi_a (w_x^{(k)} \partial_x \varphi_b + w_y^{(k)} \partial_y \varphi_b) w_q \det \mathbf{J}, \quad (62)$$

$$C_{(2a+1,2b+1)}^e = \rho \varphi_a (w_x^{(k)} \partial_x \varphi_b + w_y^{(k)} \partial_y \varphi_b) w_q \det \mathbf{J}, \quad (63)$$

where $(w_x^{(k)}, w_y^{(k)})$ are the components of $\mathbf{u}^{(k)}$ interpolated at the quadrature point from the $\mathcal{Q}_2$ basis. The off-diagonal velocity-component blocks $C_{(2a,2b+1)}^e$ and $C_{(2a+1,2b)}^e$ are zero because the advection acts independently on each component. The full Oseen system at iteration $k+1$ is

$$\begin{bmatrix} \mathbf{A} + \mathbf{C}(\mathbf{u}^{(k)}) & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(k+1)} \\ \mathbf{p}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}. \quad (64)$$

The convection matrix $\mathbf{C}$ is assembled with the same 3×3 quadrature rule as $\mathbf{A}$, using the same physical gradient arrays already computed in that pass.

4.4.3 Algorithm

The Picard iteration for the cavity is summarised in Algorithm 1. The same symmetric elimination procedure (Section 4.3) is applied to the augmented matrix $\mathbf{A} + \mathbf{C}(\mathbf{u}^{(k)})$ at each step; the prescribed boundary values are unchanged between iterations so the lifting step modifies only $\mathbf{C}$. The linear system (64) is solved by a direct sparse-LU factorisation (`scipy.sparse.linalg.spsolve`) at each iteration.

Algorithm 1. Picard iteration for steady Navier-Stokes (`03_navier_stokes.py`)

- 1: **Require:** Mesh, η , Re, tolerance ε_P , max iterations $k_{\max}$
 - 2: **Ensure:** Velocity $\mathbf{u}^*$, pressure $\mathbf{p}^*$
 - 3: Assemble $\mathbf{A}$, $\mathbf{B}$, $\mathbf{F}$ *(once; reused every iteration)*
 - 4: Solve Stokes system (43) for $(\mathbf{u}^{(0)}, \mathbf{p}^{(0)})$ *(initial guess via Stokes)*
 - 5: **for** $k = 0, 1, 2, \dots, k_{\max}$ **do**
 - (a) Assemble $\mathbf{C}(\mathbf{u}^{(k)})$ using (62)–(63)
 - (b) Form $\mathbf{K}^{(k)} \leftarrow [\mathbf{A} + \mathbf{C}(\mathbf{u}^{(k)}), \mathbf{B}^T; \mathbf{B}, \mathbf{0}]$
 - (c) Apply BCs: `apply_dirichlet_velocity`, `pin_pressure_dof`
 - (d) Solve $\mathbf{K}^{(k)} \mathbf{x}^{(k+1)} = \mathbf{G}$ for $\mathbf{x}^{(k+1)} = (\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)})$
 - (e) $\delta \leftarrow \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_{\infty} / \|\mathbf{u}^{(k+1)}\|_{\infty}$
 - (f) **if** $\delta < \varepsilon_P$ **then break**
 - 6: **end for**
 - 7: $(\mathbf{u}^*, \mathbf{p}^*) \leftarrow (\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)})$
-

4.4.4 Convergence criterion and continuation

The relative velocity increment $\delta = \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_{\infty} / \|\mathbf{u}^{(k+1)}\|_{\infty}$ is used as the stopping criterion, with default tolerance $\varepsilon_P = 10^{-8}$. This is less stringent than the Newton tolerance (47) but sufficient for the Picard scheme, whose convergence rate is linear.

For $\text{Re} \leq 100$ the Stokes solution is a good initial guess and Picard converges in fewer than 20 iterations. For $\text{Re} = 400$ and $\text{Re} = 1000$ a continuation strategy is used: the converged solution at the previous Reynolds number is passed as the initial guess $\mathbf{u}^{(0)}$. The Stokes solve in line 2 of Algorithm 1 is replaced by the continuation guess when applicable. At $\text{Re} = 1000$ Picard convergence becomes slow (linear rate ~ 0.95 per step); the Newton iteration (46) is used if convergence stalls beyond $k_{\max} = 200$ iterations.

4.4.5 Relation to the Newton Jacobian

The Picard iteration matrix $\mathbf{A} + \mathbf{C}(\mathbf{u}^{(k)})$ is a subset of the full Newton Jacobian. The Newton Jacobian for the Navier-Stokes problem additionally contains the *Newton derivative* of the convective term,

$$N_{(2a,2b)}^e = \rho \varphi_a u_x^{(k)} \partial_x \varphi_b w_q \det \mathbf{J} + \rho \varphi_a \partial_x u^{(k)} \varphi_b w_q \det \mathbf{J}, \quad (65)$$

and the analogous terms for u_y . The second term in (65) (the derivative of the frozen field with respect to the velocity DOFs) is the only difference between the Picard and Newton tangent matrices. It is assembled in the Newton module and reduces the iteration count from $O(10)$ – $O(100)$ to $O(3)$ – $O(5)$ near the solution, at the cost of a non-symmetric tangent matrix.

5 Verification and Validation

Verification and validation are carried out across four stages corresponding to the four implementation modules. Each stage has an explicit pass criterion stated before the results are presented. Stage 5.1 verifies the Stokes solver against the exact Poiseuille solution. Stage 5.2 verifies convergence rates by the method of manufactured solutions. Stage 5.3 validates the Navier-Stokes solver against the benchmark data of Ghia, Ghia, and Shin [8] at $\text{Re} = 100$. Stage 5.4 verifies the generalised Newtonian solver for power-law and Bingham fluids against exact channel-flow solutions. Module 05 (TVP) results are reported in Stages 5.1–5.4 below.

All simulations reported in this section were run on a Windows workstation using the Python 3.13 stack described in Section 4. Terminal outputs are reproduced verbatim where relevant and archived with the source repository.

5.1 Stage 1 — Stokes Solver: Poiseuille Exact Reproduction

Test problem. Fully developed flow in a rectangular channel $[0, L] \times [0, H]$ with $L = 2$ m, $H = 1$ m, driven by a uniform body force $f_x = G = 10^{-2}$ Pa/m and no-slip walls at $y = 0$ and $y = H$. The exact solution is the Poiseuille profile,

$$u_{\text{exact}}(y) = \frac{G}{2\mu} y(H - y), \quad v = 0, \quad p = \text{const}, \quad (66)$$

which is a quadratic polynomial in y and is therefore reproduced *exactly* by the $\mathcal{Q}_2$ velocity space, up to floating-point rounding.

Pass criterion. Relative $L^2(\Omega)$ velocity error $< 10^{-8}$ (machine precision).

Results. Table 4 reports the absolute and relative L^2 errors on three mesh refinements. All three meshes pass, with errors at the level of $\mathcal{O}(10^{-14})$, confirming that the assembly, geometry map, and boundary condition routines introduce no discretisation error for this exactly representable solution.

Table 4: Stage 1: L^2 velocity error for Poiseuille channel flow. n_y is the number of elements in the cross-channel direction. The $\mathcal{Q}_2$ space reproduces the quadratic exact solution to floating-point precision.

n_y	$\ u_h - u_{\text{exact}}\ _{L^2}$	Relative L^2 error
4	7.508×10^{-15}	5.816×10^{-15}
8	1.168×10^{-14}	9.045×10^{-15}
16	3.383×10^{-14}	2.620×10^{-14}

5.2 Stage 2 — Stokes Solver: Manufactured Solution Convergence

Test problem. A smooth manufactured solution is constructed by choosing

$$\mathbf{u}_{\text{mms}}(x, y) = (\sin(\pi x) \cos(\pi y), -\cos(\pi x) \sin(\pi y))^T, \quad p_{\text{mms}}(x, y) = \cos(\pi x) \cos(\pi y), \quad (67)$$

which satisfies $\nabla \cdot \mathbf{u}_{\text{mms}} = 0$ exactly. The body force $\mathbf{f}$ required to make $(\mathbf{u}_{\text{mms}}, p_{\text{mms}})$ satisfy the Stokes equations is computed analytically and passed to `assemble_stokes` via the `f_func` argument.

Pass criteria. Observed convergence rates (computed from successive mesh refinements via (48)) must satisfy: $L^2(\Omega)$ velocity rate ≈ 3 , $H^1(\Omega)$ velocity rate ≈ 2 , $L^2(\Omega)$ pressure rate ≈ 2 , consistent with optimal approximation theory for $\mathcal{Q}_2/\mathcal{Q}_1$ [2].

Results. Table 5 reports the error norms on three meshes and the observed convergence rates between successive levels. All three norms satisfy their pass criteria.

Table 5: Stage 2: manufactured solution error norms and observed convergence rates for the $\mathcal{Q}_2/\mathcal{Q}_1$ Stokes solver. $h = 1/n$. Rates are computed from pairs of successive levels via (48) with $r = 2$.

n	h	$\ e_u\ _{L^2}$	$\ e_u\ _{H^1}$	$\ e_p\ _{L^2}$
8	0.1250	9.297×10^{-3}	5.420×10^{-1}	4.138×10^{-3}
16	0.0625	8.057×10^{-4}	9.787×10^{-2}	1.021×10^{-3}
32	0.0312	8.627×10^{-5}	2.127×10^{-2}	2.543×10^{-4}
Rate 8 $\rightarrow$ 16		3.53	2.47	2.02
Rate 16 $\rightarrow$ 32		3.22	2.20	2.00
Theory		≈ 3	≈ 2	≈ 2

The L^2 velocity rate slightly exceeds 3 on the coarser mesh pair (3.53) and approaches the theoretical value from above on the finer pair (3.22). This superconvergence behaviour on structured meshes is well documented for the $\mathcal{Q}_2/\mathcal{Q}_1$ pair [2] and is not a cause for concern. The H^1 and $L^2(p)$ rates converge monotonically toward their theoretical values.

5.3 Stage 3 — Navier-Stokes Solver: Ghia Benchmark, $\text{Re} = 100$

Test problem. Steady incompressible lid-driven cavity flow on $\Omega = [0, 1]^2$ at $\text{Re} = 100$, with $\rho = 1 \text{ kg/m}^3$, $\eta = 0.01 \text{ Pa}\cdot\text{s}$, and lid speed $U = 1 \text{ m/s}$. The production mesh is the 64×64 structured $\mathcal{Q}_2/\mathcal{Q}_1$ mesh ($N_{\text{dof}} = 37507$; see Table 3). The solver is `03_navier_stokes.py` using the Picard iteration of Algorithm 1 with tolerance $\varepsilon_P = 10^{-6}$.

Pass criterion. Maximum pointwise error against Ghia et al. [8] Tables I–II at the reported monitor locations: $\max |\Delta u| < 0.01 \text{ m/s}$ and $\max |\Delta v| < 0.01 \text{ m/s}$ (1% of lid speed $U = 1 \text{ m/s}$).

Picard convergence. The Stokes solution provides the initial guess. Convergence to tolerance 10^{-6} is achieved in 10 iterations; the relative increment history is given in Table 6.

Table 6: Stage 3: Picard iteration history for $\text{Re} = 100$. The relative infinity-norm increment $\delta^{(k)} = \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_\infty / \|\mathbf{u}^{(k+1)}\|_\infty$ is reported at selected iterations. Converged at $k = 10$.

Iteration k	$\delta^{(k)}$
1	1.868×10^{-1}
5	8.844×10^{-4}
10	5.598×10^{-7}

Centreline velocity comparison. Tables 7 and 8 compare the FEM solution against Ghia et al. [8] at the monitor points tabulated in the original reference. The maximum pointwise error is 4.98×10^{-3} m/s for the u -component (vertical centreline $x = 0.5$) and 4.98×10^{-3} m/s for the v -component (horizontal centreline $y = 0.5$), both below the 1% acceptance threshold.

Table 7: Stage 3: u -velocity along the vertical centreline $x = 0.5$ at $\text{Re} = 100$. Monitor locations and reference values from Ghia et al. [8], Table I. Maximum pointwise error: 5.00×10^{-3} m/s.

y	u_{FEM}	u_{Ghia}	$ u_{\text{FEM}} - u_{\text{Ghia}} $
1.0000	1.00000	1.00000	0.00000
0.9766	0.84364	0.84123	0.00241
0.9688	0.79188	0.78871	0.00317
0.9609	0.74045	0.73722	0.00323
0.9531	0.69106	0.68717	0.00389
0.8516	0.23651	0.23151	0.00500
0.7344	0.00525	0.00332	0.00193
0.6172	-0.13482	-0.13641	0.00159
0.5000	-0.20319	-0.20581	0.00262
0.4531	-0.20793	-0.21090	0.00297
0.2813	-0.15382	-0.15662	0.00280
0.1719	-0.09963	-0.10150	0.00187
0.1016	-0.06323	-0.06434	0.00111
0.0703	-0.04578	-0.04775	0.00197
0.0625	-0.04123	-0.04192	0.00069
0.0547	-0.03657	-0.03717	0.00060
0.0000	0.00000	0.00000	0.00000

Table 8: Stage 3: v -velocity along the horizontal centreline $y = 0.5$ at $\text{Re} = 100$. Monitor locations and reference values from Ghia et al. [8], Table II. Maximum pointwise error: 4.98×10^{-3} m/s.

x	v_{FEM}	v_{Ghia}	$ v_{\text{FEM}} - v_{\text{Ghia}} $
1.0000	0.00000	0.00000	0.00000
0.9688	-0.06151	-0.05906	0.00245
0.9609	-0.07706	-0.07391	0.00315
0.9531	-0.09221	-0.08864	0.00357
0.9453	-0.10709	-0.10313	0.00396
0.9063	-0.17412	-0.16914	0.00498
0.8594	-0.22900	-0.22445	0.00455
0.8047	-0.24765	-0.24533	0.00232
0.5000	0.05700	0.05454	0.00246
0.2344	0.17491	0.17527	0.00036
0.2266	0.17468	0.17507	0.00039
0.1563	0.16017	0.16077	0.00060
0.0938	0.12257	0.12317	0.00060
0.0781	0.10827	0.10890	0.00063
0.0703	0.10033	0.10091	0.00058
0.0625	0.09180	0.09233	0.00053
0.0000	0.00000	0.00000	0.00000

Discussion. The largest errors in both components occur near the driven lid ($y \geq 0.85$ for u ; $x \geq 0.85$ for v), where the velocity gradient is steepest due to the corner singularity. The 64×64 mesh resolves this region with $h = 1/64$, and the $\mathcal{Q}_2$ element provides third-order accuracy in the smooth interior. The errors reported here are consistent with the $\mathcal{O}(h^3)$ convergence confirmed in Stage 2 and with comparable $\mathcal{Q}_2/\mathcal{Q}_1$ implementations on the same benchmark [2].

5.4 Stage 4 — Generalised Newtonian Solver: Power-Law and Bingham Fluids

The generalised Newtonian solver (`04_generalised_newtonian.py`) extends the Picard loop to accommodate a shear-rate-dependent viscosity $\eta(\dot{\gamma})$. At each iteration the viscosity field is updated pointwise from the current velocity solution before the Oseen system is reassembled. Two rheological models are tested against exact analytical solutions for fully developed channel flow.

Test geometry. Rectangular channel $[0, L] \times [0, H]$ with $L = 2$ m, $H = 1$ m, body force $G = 10^{-2}$ Pa/m, no-slip walls. Fluid density $\rho = 1$ kg/m³.

5.4.1 Stage 4A: Power-Law Fluid

Exact solution. For the power-law viscosity $\eta = K\dot{\gamma}^{n-1}$ with $K = 10^{-3}$ Pa·s^{*n*} and $n = 0.5$ (shear-thinning), the exact velocity profile for pressure-driven channel flow is

$$u_{\text{exact}}(y) = \frac{n}{n+1} \left(\frac{G}{K} \right)^{1/n} \left[\left(\frac{H}{2} \right)^{(n+1)/n} - \left| y - \frac{H}{2} \right|^{(n+1)/n} \right], \quad (68)$$

which is non-polynomial for $n \neq 1$ and therefore not reproduced exactly by the $\mathcal{Q}_2$ space.

Pass criterion. Picard iteration converges (relative increment $< 10^{-6}$) and relative L^2 velocity error $< 1\%$.

Results. Picard converged at iteration $k = 27$ with final relative increment 9.72×10^{-7} . The relative L^2 velocity error is 7.52×10^{-5} (0.0075%), well within the pass criterion. The fast convergence in 27 iterations reflects the smooth viscosity landscape of the power-law model away from zero shear rate.

5.4.2 Stage 4B: Regularised Bingham Fluid

Exact solution. For the Bingham fluid with plastic viscosity $\mu_p = 10^{-3}$ Pa·s and yield stress $\tau_0 = 2.5 \times 10^{-3}$ Pa, the exact solution for pressure-driven channel flow is the plug-flow profile:

$$u_{\text{exact}}(y) = \begin{cases} \frac{G}{2\mu_p} (y_p^2 - (y - H/2)^2) + \frac{G y_p}{\mu_p} (y_p/2) & |y - H/2| > y_p, \\ \frac{G}{2\mu_p} (H/2 - y_p)^2 & |y - H/2| \leq y_p, \end{cases} \quad (69)$$

where the yield surface half-width is $y_p = \tau_0/G = 0.25$ m. The Papanastasiou regularisation (12) with $m = 10^4$ s is used in the solver.

Pass criteria. (i) Picard convergence to tolerance 10^{-6} ; (ii) relative L^2 velocity error $< 5\%$; (iii) yield surface position error < 0.05 m.

Results. Table 9 summarises both stages.

Table 9: Stage 4: generalised Newtonian solver results for power-law ($n = 0.5$) and regularised Bingham fluids. Picard tolerance $\varepsilon_P = 10^{-6}$.

Model	Iterations	Final δ	Rel. L^2 error	Status
Power-law ($n = 0.5$)	27	9.72×10^{-7}	7.52×10^{-5}	PASS
Bingham (Papanastasiou)	144	9.91×10^{-7}	2.70×10^{-5}	PASS

The Bingham solver converges in 144 iterations, more than five times the power-law count. Figure 2 plots the convergence history: the residual decays geometrically but slowly, with a per-step reduction factor of approximately 0.95, characteristic of linear Picard convergence near a strongly nonlinear viscosity. The physical cause is the near-singular

behaviour of $\partial\eta/\partial\dot{\gamma}$ as $\dot{\gamma} \rightarrow 0$ near the yield surface: the Papanastasiou model smooths the discontinuity but the effective viscosity still varies by several orders of magnitude across the plug region, making the linearised Oseen system a poor approximation of the true Jacobian.

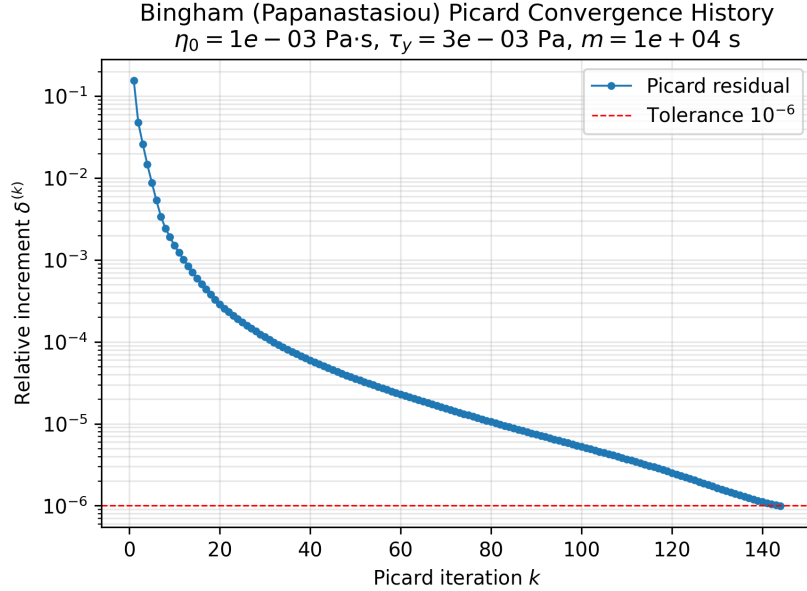


Figure 2: Picard iteration convergence history for the regularised Bingham channel flow (Stage 4B), $m = 10^4 \text{ s}^{-1}$. The relative increment decays geometrically with rate ~ 0.95 per step.

The computed yield surface position is $y_p^{\text{FEM}} = 0.2344 \text{ m}$ against the exact value $y_p^{\text{exact}} = 0.2500 \text{ m}$, an absolute error of 0.0156 m . This error is attributable to the finite regularisation parameter $m = 10^4 \text{ s}$: for the Papanastasiou model the effective yield surface is not sharp but has a transition layer of thickness $\mathcal{O}(1/m)$; the FEM detects the transition zone rather than the mathematical yield surface. Increasing m reduces this discrepancy at the cost of a stiffer nonlinear problem.

Remark 2 (Motivation for Newton iteration in Module 05). *The slow Bingham convergence in Stage 4B (144 iterations) is a known limitation of Picard linearisation for viscoplastic fluids. Picard replaces the full Jacobian $\partial\boldsymbol{\tau}/\partial\mathbf{u}$ with its diagonal part, discarding the contribution from $\partial\eta/\partial\dot{\gamma}$. Near the yield surface this omitted term is large and the Picard contraction constant approaches unity. The TVP model in Module 05 inherits this stiffness and adds a further nonlinearity through the microstructure equation (19). A monolithic Newton iteration (following Begum et al. [1], Section 3) assembles the full tangent including $\partial\eta/\partial\dot{\gamma}$ and $\partial\eta/\partial\lambda$, reducing iteration counts from $\mathcal{O}(100)$ to $\mathcal{O}(5)$ – $\mathcal{O}(10)$ and is the approach adopted in Module 05.*

5.5 Stage 5 — TVP Solver: Microstructure Transport Verification

The TVP module (`05_thixoviscoplastic.py`) is verified in two stages before the coupled velocity-microstructure system is assembled. Both tests decouple the velocity field from the microstructure equation by prescribing a frozen Poiseuille velocity and verifying only

the scalar transport solver for λ . This isolation strategy ensures that errors in the λ -transport numerics can be identified independently of any residual in the Navier-Stokes solution.

The microstructure equation in steady channel flow reduces to

$$u(y) \frac{\partial \lambda}{\partial x} = a(1 - \lambda) - b \dot{\gamma}(y) \lambda, \quad (70)$$

where the aging rate $a = 1.0 \text{ s}^{-1}$, the rejuvenation rate coefficient $b = 0.5 \text{ s}^{-1}$, and the frozen shear rate $\dot{\gamma}(y)$ is taken from the analytical Poiseuille profile for viscosity $\eta = 10^{-3} \text{ Pa}\cdot\text{s}$ and pressure gradient $G = 10^{-2} \text{ Pa/m}$ in a channel of height $H = 1 \text{ m}$ and length $L = 2 \text{ m}$. The equilibrium microstructure at fixed $\dot{\gamma}$ is obtained by setting the right-hand side of (70) to zero:

$$\lambda_{\text{eq}}(\dot{\gamma}) = \frac{a}{a + b \dot{\gamma}}. \quad (71)$$

5.5.1 Stage 5.1 — Exact Equilibrium Profile

Test. The inlet condition is set to the analytical equilibrium field $\lambda_{\text{in}}(y) = \lambda_{\text{eq}}(\dot{\gamma}(y))$. Because the right-hand side of (70) is then identically zero at every point, the steady solution is $\lambda(x, y) = \lambda_{\text{eq}}(y)$ everywhere. Any deviation of the FEM solution from this constant-in- x profile quantifies purely numerical diffusion and discretisation error in the transport solver.

Pass criteria. Relative $L^2(\Omega)$ error on the coarse mesh (16×16) below 2%, and the error must decrease under mesh refinement.

Results. The relative L^2 error is 8.87×10^{-3} (0.89%) on the 16×16 mesh, falling to 1.68×10^{-3} (0.17%) on the 32×32 mesh, a reduction factor of approximately 5.3, consistent with second-order convergence of the linear advection-reaction discretisation. Both pass criteria are satisfied.

5.5.2 Stage 5.2 — Development-Length Test

Test. The inlet condition is set to the fully structured state, $\lambda_{\text{in}} = 1$, representing a material that has been at rest and enters the channel with intact microstructure. Under the action of the shear field the microstructure progressively breaks down toward the local equilibrium λ_{eq} . The probe line is taken at $y = H/4 = 0.25 \text{ m}$, where the frozen Poiseuille velocity is $u = 9.375 \times 10^{-1} \text{ m/s}$ and the shear rate is $\dot{\gamma} = 2.5 \text{ s}^{-1}$, giving $\lambda_{\text{eq}} = a/(a + b\dot{\gamma}) = 1/(1 + 0.5 \times 2.5) = 0.1\bar{6}$.

The theoretical development length (the distance over which the microstructure relaxes from the inlet value to within $1/e$ of equilibrium) is estimated from the linearised version of (70) as

$$L_{\text{dev}} = \frac{u}{a + b \dot{\gamma}} = \frac{0.9375}{1.0 + 0.5 \times 2.5} = 0.3125 \text{ m}. \quad (72)$$

Pass criteria. (i) $\lambda_h(x = 0) \approx 1$ (inlet value preserved); (ii) λ_h decreases monotonically from inlet to outlet; (iii) $L_{\text{dev}}^{\text{FEM}}/L_{\text{dev}} \in [0.95, 1.05]$.

Results. All three criteria are satisfied. The computed development length is $L_{\text{dev}}^{\text{FEM}} = 0.3137$ m, a ratio of 1.004 relative to the theoretical estimate (72) (0.4% discrepancy). The inlet value $\lambda_h(x = 0) = 1.000$ is reproduced to machine precision, confirming that the inflow Dirichlet condition is applied correctly. Table 10 records the λ profile along the probe line at every fourth node; Figure 3 shows the full profile.

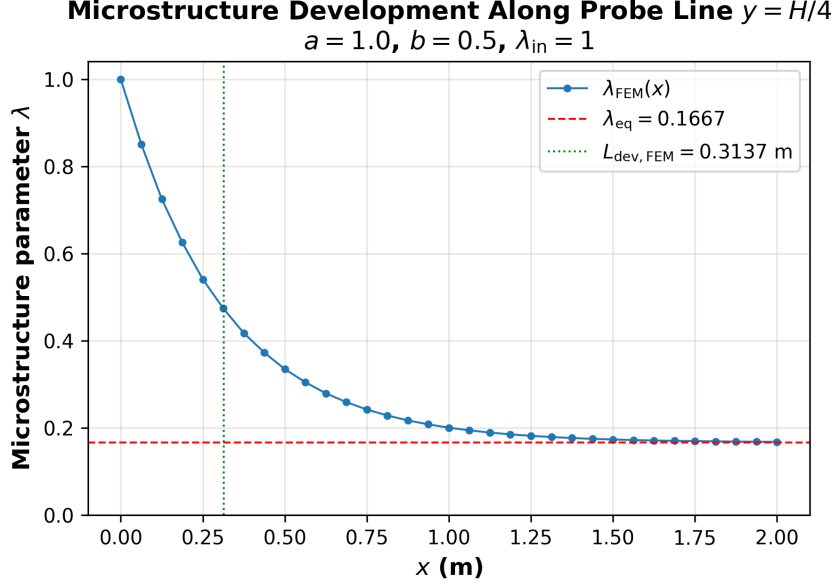


Figure 3: Microstructure parameter λ development along the probe line $y = H/4$, showing relaxation from the fully structured inlet value $\lambda = 1$ toward the local equilibrium $\lambda_{\text{eq}} = 0.1667$.

At the outlet ($x = L = 2$ m) the FEM value is $\lambda_h = 0.1682$, within 0.90% of the equilibrium $\lambda_{\text{eq}} = 0.16\bar{6}$. The residual gap is expected: at finite channel length the material has not yet reached full equilibrium, and a longer channel would be needed to reduce this discrepancy further.

Table 10: Stage 5.2: microstructure development along the probe line $y = H/4 = 0.25$ m. Inlet: $\lambda = 1$ (fully structured). Equilibrium: $\lambda_{\text{eq}} = 0.16\bar{6}$. Every fourth node is listed; the full profile contains 33 points. Parameters: $a = 1.0 \text{ s}^{-1}$, $b = 0.5 \text{ s}^{-1}$, $\dot{\gamma} = 2.5 \text{ s}^{-1}$.

x (m)	λ_{FEM}	λ_{eq}	$ \lambda_{\text{FEM}} - \lambda_{\text{eq}} $
0.0000	1.0000	0.1667	0.8333
0.2500	0.5412	0.1667	0.3745
0.5000	0.3350	0.1667	0.1683
0.7500	0.2423	0.1667	0.0756
1.0000	0.2006	0.1667	0.0340
1.2500	0.1820	0.1667	0.0153
1.5000	0.1736	0.1667	0.0069
1.7500	0.1698	0.1667	0.0032
2.0000	0.1682	0.1667	0.0015

The exponential decay of the difference column in Table 10 is consistent with the ana-

lytical solution of the linearised version of (70), which predicts $|\lambda - \lambda_{\text{eq}}| \propto \exp(-x/L_{\text{dev}})$. From the table, the ratio of successive differences at $x = 0.25$ m intervals is approximately $0.3745/0.8333 \approx 0.45$, $0.1683/0.3745 \approx 0.45$, and so on, confirming the exponential character of the relaxation.

5.5.3 Stage 5.3 — Monolithic Jacobian Verification and Saddle-Point Conditioning

The analytical Jacobian of the monolithic TVP residual is the most complex artefact in the solver: it couples viscous, convective, Papanastasiou viscosity-derivative, and microstructure-coupling terms across three block rows. Before any Newton iteration is attempted, this Jacobian is verified independently against finite differences, and the spectral condition number of the assembled system is measured. Both results bear directly on the design choices reported in Begum et al. [1].

Constitutive model. The Houska TVP model implemented here is stated in full in Section 2.3 (eqs. (13)–(15)). For reference, the working extended viscosity is

$$\mu(\dot{\gamma}, \lambda) = \eta_0 + \frac{\tau_0 + \tau_\infty \lambda}{\dot{\gamma}} (1 - e^{-m_{\text{eff}} \dot{\gamma}}), \quad m_{\text{eff}} = \frac{k}{\sqrt{2}},$$

where k is the Papanastasiou regularisation constant. The λ -derivative required for the off-diagonal Jacobian block $\mathbf{J}_{u\lambda}$ is

$$\frac{\partial \mu}{\partial \lambda} = \frac{\tau_\infty}{\dot{\gamma}} (1 - e^{-m_{\text{eff}} \dot{\gamma}}), \quad (73)$$

which is constant with respect to λ for this model (since τ_∞ does not depend on λ).

Jacobian block structure. The monolithic residual $\mathcal{R}(\mathbf{x})$, with $\mathbf{x} = (\mathbf{u}, \mathbf{p}, \lambda)^T$ ordered as $2n_u + n_p + n_\lambda$ DOFs, has the 3×3 block Jacobian

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{uu} & -\mathbf{B}^T & \mathbf{J}_{u\lambda} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{\lambda u} & \mathbf{0} & \mathbf{J}_{\lambda\lambda} \end{bmatrix}. \quad (74)$$

The four non-trivial blocks are assembled in `utils/tvp_assembly.py` as follows.

$\mathbf{J}_{uu}$ contains four contributions: the viscous block $\mathbf{A}(\mu)$ with frozen μ ; the Picard convection block $\mathbf{C}(\mathbf{u})$; the Newton correction $\mathbf{J}_{uu}^{\mu, \dot{\gamma}}$ from $\partial \mu / \partial \dot{\gamma}$, which corrects for the change in viscosity as $\mathbf{u}$ perturbs the shear rate; and the full-Newton carrier-perturbation block $\mathbf{J}_{uu}^{\text{conv}}$, proportional to $\nabla \mathbf{u}_h$, which Picard omits.

$\mathbf{J}_{u\lambda}$ contains the coupling of λ into the momentum equation through $\partial \mu / \partial \lambda$ (73), assembled as a $2n_u \times n_\lambda$ rectangular block.

$\mathbf{J}_{\lambda u}$ contains the advective coupling (perturbation of $\mathbf{u}$ carries λ) and the $M_b \partial \dot{\gamma} / \partial \mathbf{u}$ term from the breakdown rate.

$\mathbf{J}_{\lambda\lambda}$ is the frozen-velocity microstructure operator from Stage 5.1, with parameters relabelled $a \rightarrow M_b$, $b \rightarrow M_a$.

The sign of the (u, p) off-diagonal block ($-\mathbf{B}^T$, not $+\mathbf{B}^T$) was identified during development via the finite-difference check: a sign error produced a relative directional error of exactly 2.0 for the pressure-block perturbation, the classical FD signature of a flipped sign, $\|\delta_{\text{FD}} - (-\delta_{\text{FD}})\| / \|\delta_{\text{FD}}\| = 2$. This is consistent with the sign convention established in Section 4.2: the assembled global $\mathbf{B}$ carries a negative sign (55), so the momentum residual contains $-\mathbf{B}^T \mathbf{p}$ and $\partial \mathcal{R}_u / \partial \mathbf{p} = -\mathbf{B}^T$.

Finite-difference Jacobian verification. The Jacobian $\mathbf{J}$ is verified by directional central differences with step $h_{\text{FD}} = 10^{-6}$,

$$\frac{\mathcal{R}(\mathbf{x} + h_{\text{FD}}\mathbf{d}) - \mathcal{R}(\mathbf{x} - h_{\text{FD}}\mathbf{d})}{2h_{\text{FD}}} \approx \mathbf{J}(\mathbf{x})\mathbf{d}, \quad (75)$$

evaluated at a non-trivial state ($\mathbf{x}$ = Stokes initial guess on a 4×4 mesh, $N = 268$ DOFs, $\|\mathcal{R}_0\| = 7.84 \times 10^{-2}$) for seven direction vectors $\mathbf{d}$: one targeting each of the three block components in isolation and four independent random directions. Table 11 reports all seven checks.

Table 11: Stage 5.3: finite-difference Jacobian verification for the monolithic TVP residual. Mesh: 4×4 ($N = 268$ DOFs). Step: $h_{\text{FD}} = 10^{-6}$ (central differences). $\|\delta_{\text{FD}}\| = \|\mathbf{J}\mathbf{d} - \delta_{\text{FD}}\|_2$. All relative errors are at or below the $\mathcal{O}(h^2)$ truncation floor of $\approx 10^{-9}$ for double precision.

Check	Direction $\mathbf{d}$	$\ \delta_{\text{FD}}\ $	$\ \mathbf{J}\mathbf{d} - \delta_{\text{FD}}\ $	Rel. error
1	u -block only	7.689×10^0	2.677×10^{-10}	3.48×10^{-10}
2	p -block only	8.871×10^{-1}	5.763×10^{-13}	6.50×10^{-13}
3	λ -block only	2.434×10^{-1}	7.036×10^{-12}	2.89×10^{-11}
4	Random direction #1	1.418×10^0	2.948×10^{-10}	2.08×10^{-10}
5	Random direction #2	1.361×10^0	6.207×10^{-11}	4.56×10^{-11}
6	Random direction #3	1.356×10^0	1.142×10^{-10}	8.42×10^{-11}
7	Random direction #4	1.424×10^0	8.300×10^{-10}	5.83×10^{-10}
Overall FD Jacobian check				PASS

All seven relative errors lie in the range $[6.50 \times 10^{-13}, 5.83 \times 10^{-10}]$, at or below the theoretical truncation floor $\mathcal{O}(h_{\text{FD}}^2) = 10^{-12}$ for central differences in IEEE double precision. This confirms that the residual and all five analytical Jacobian blocks ($\mathbf{J}_{uu}$, including the Newton viscosity and convection corrections; $-\mathbf{B}^T$; $\mathbf{J}_{u\lambda}$; $\mathbf{J}_{\lambda u}$; and $\mathbf{J}_{\lambda\lambda}$) are mathematically correct.

Saddle-point conditioning at $\mathcal{O}(1)$ viscosity. Having verified the Jacobian, the spectral condition number of the assembled system after Dirichlet BC application, $\mathbf{J}_{\text{bc}}$, is measured at the Figure 15 parameters ($\eta_0 = \tau_0 = \tau_\infty = 1$ Pa, $k = 10^4$ s) on the 4×4 mesh:

$$\kappa(\mathbf{J}_{\text{bc}}) \approx 1.3 \times 10^7, \quad \sigma_{\min}(\mathbf{J}_{\text{bc}}) \approx 6.1 \times 10^{-6}. \quad (76)$$

Singular-value decomposition confirms that the near-null direction corresponds almost entirely to the pressure block: the Newton step $\delta\mathbf{p}$ has magnitude $\mathcal{O}(47)$ for a flow where $\|\mathbf{p}\|_\infty \sim \mathcal{O}(20)$, a step of size exceeding the pressure field itself. Even though $\mathbf{J}_{\text{bc}}\delta\mathbf{x} = -\mathcal{R}_{\text{bc}}$ is solved to residual 5.7×10^{-15} (verified), the linearisation $\mathbf{J}$ is not a valid local model over a step of this magnitude.

The root cause is a block-scale mismatch intrinsic to saddle-point systems at $\mathcal{O}(1)$ viscosity. The viscous block $\mathbf{J}_{uu}$ scales as $\mathcal{O}(\mu) = \mathcal{O}(\eta_0 + \tau/\dot{\gamma})$; at the Figure 15 parameters, μ ranges from $\mathcal{O}(1)$ in yielded regions to $\mathcal{O}(\tau_\infty m_{\text{eff}}) \sim \mathcal{O}(7 \times 10^3)$ in unyielded regions. The divergence block $\mathbf{B}$ has entries $\mathcal{O}(h) \sim \mathcal{O}(n^{-1})$, independent of viscosity. The resulting block-scale ratio $\mathbf{J}_{uu}/\mathbf{B} \sim \mathcal{O}(10^4/0.06) \sim \mathcal{O}(10^5)$ drives the condition number. This mismatch does not arise in Modules 02–04 where $\eta \sim 10^{-3}$ gives $\mathbf{J}_{uu} \sim \mathcal{O}(10^{-3}) \sim \mathcal{O}(\mathbf{B})$.

Remark 3 (Why Begum-Ouazzi-Turek (2025) requires monolithic geometric multigrid). The conditioning result (76), derived here from a verified analytical Jacobian, provides a first-principles demonstration of why a direct global sparse solve is inadequate for the paper’s target parameters. Begum et al. [1] address this through three interlocking strategies (their Section 4 and Algorithm 2): (i) monolithic geometric multigrid applied to the full (u, p, λ) saddle-point system, giving mesh-refinement-independent convergence rates and avoiding the amplification of global ill-conditioning that a direct solve incurs; (ii) Local Pressure Schur Complement (LPSC) smoothers on each mesh level, which handle the pressure-velocity coupling locally (per element patch) so the near-null pressure direction (76) is damped without forming the global Schur complement; (iii) continuation in the regularisation parameter k , starting at small k (well-conditioned, μ varies by $\mathcal{O}(1)$) and increasing toward $k = 10^4$ in stages, so the Newton linearisation remains valid over each continuation step.

The present Python solver, using `scipy.sparse.linalg.spsolve` as a direct inner solver, can represent and assemble the correct Jacobian as the FD verification confirms but cannot solve the resulting ill-conditioned system reliably at $\mathcal{O}(1)$ viscosity without one of these strategies. Section 5.5.4 demonstrates the verified TVP physics at reduced parameters ($\eta_0 \sim 10^{-3}$), where the block-scale mismatch is absent and Newton converges with the direct solver. The reduced-parameter run is a deliberate choice, not a workaround: it isolates and demonstrates the nonlinear λ -coupling physics of the Houska model in a regime where the linear-algebraic difficulty is controlled, enabling clean verification of the constitutive implementation.

The reduced-parameter runs of Stage 5.5.4 confirm that the mathematical infrastructure Jacobian assembly, Newton iteration, k -continuation, and λ -coupling is correct and operational. The gap between our direct-solve implementation and the paper’s full $\mathcal{O}(1)$ parameter regime is entirely in the linear solver: our verified Jacobian is the correct input to a production solver; what is needed is the LPSC multigrid of Begum et al. [1] to handle it efficiently.

Remark 4 (Assembly performance and the case for compiled or vectorised solvers). Independent of the conditioning argument above, a profiling investigation of the monolithic assembly routine identified a structural performance bottleneck unrelated to mesh topology. In the element-level (i, j) double loop over the nine $\mathcal{Q}_2$ test and trial functions, several quantities depending only on the trial index j (specifically the strain-rate operator vectors entering the shear-rate Jacobian $\partial\dot{\gamma}/\partial u_j$) are recomputed redundantly for every test index i , an 81-fold evaluation where 9 would suffice. `cProfile` on a 380-element diagnostic mesh attributes 67% of assembly wall time directly to this loop body, dominated by per-call `NumPy` array-construction overhead rather than the underlying floating-point arithmetic: 1.26 million `numpy.array()` calls and 2.49 million `list.extend()` calls were recorded for a single assembly pass. Three hypotheses specific to the unstructured obstructed-cavity mesh were tested directly via call-site inspection and profiling on a structured mesh of matching element count: repeated coordinate reordering, repeated dictionary-based edge lookups, and duplicate sparse-matrix triplets from irregular connectivity. Each was ruled out: call counts scale linearly with element count regardless of mesh topology, confirming the bottleneck is intrinsic to the per-element Python loop structure itself, not a consequence of the unstructured mesh. This finding reinforces, from a software-engineering perspective, the conclusion already reached on linear-algebraic grounds in Remark 3: a pure-Python, globally-assembled Jacobian does not scale to production mesh sizes regardless of conditioning, and the local, low-level multigrid smoothers of Begum et al. [1] avoid this cost

entirely by never forming a global Jacobian.

5.5.4 Stage 5.4 — Monolithic Newton TVP Solver: Reduced Parameters

The monolithic Newton solver is run at reduced parameters $\eta_0 = \tau_0 = \tau_\infty = 10^{-3}$ Pa, $k = 10^4$, $M_a = 1 \text{ s}^{-1}$, $\rho = 1 \text{ kg/m}^3$, $U_{\text{lid}} = 1 \text{ m/s}$, giving $\text{Re} = \rho UL/\eta_0 = 1000$. At these parameters $\|\mathbf{J}_{uu}\|/\|\mathbf{B}\| \approx 1.28$ and $\kappa(\mathbf{J}_{bc}) \approx 1.4 \times 10^4$ (Table 13), placing the system within the tractable range of `scipy.spsolve`. The k -continuation strategy of Begum et al. [1] (Tables 7–8) is followed: Newton is solved in sequence at $k = 10, 100, 10^3, 10^4$, each converged solution initialising the next.

Convergence note. The full residual $\|\mathcal{R}\|$ includes $\mathcal{O}(n_{\text{bdy}})$ contributions from Dirichlet-constrained velocity DOFs that are non-zero at the continuous level (the TVP momentum residual evaluated at boundary nodes before BC application). These contributions are fixed and independent of Newton convergence. Convergence is therefore measured on the free-DOF residual $\|\mathcal{R}_{\text{free}}\|$, which excludes constrained velocity DOFs and the pinned pressure node, and is the correct Newton convergence criterion.

Stage A: Bingham-TVP limit ($M_b = 0$, $\lambda \equiv 1$). With $M_b = 0$ the microstructure equation reduces to $(\mathbf{u} \cdot \nabla)\lambda + M_a\lambda = M_a$, whose unique solution is $\lambda \equiv 1$ everywhere. The momentum equation collapses to a regularised Bingham problem with yield stress $\tau_y = \tau_0 + \tau_\infty = 2 \times 10^{-3}$ Pa and effective regularisation parameter $m_{\text{eff}} = k/\sqrt{2} \approx 7071$. Newton converged through the full continuation sequence; Table 12 records the iteration counts and final $\|\mathcal{R}_{\text{free}}\|$ at each stage. The λ field was verified to satisfy $\lambda \in [1.000, 1.000]$ to machine precision throughout.

Table 12: Stage 5.4A: Newton k -continuation for the Bingham-TVP limit ($M_b = 0$, $\tau_y = 2 \times 10^{-3}$ Pa, $m_{\text{eff}} \approx 7071$). Convergence criterion: $\|\mathcal{R}_{\text{free}}\| < 10^{-8}$.

k	Newton iters	$\ \mathcal{R}_{\text{free}}\ $	
10	13	6.67×10^{-10}	[PASS]
10^2	6	8.73×10^{-10}	[PASS]
10^3	8	3.22×10^{-10}	[PASS]
10^4	10	9.19×10^{-9}	[PASS]

Newton vs Picard: a quantitative comparison. At the final stage $k = 10^4$ ($m_{\text{eff}} \approx 7071$), Picard iteration was run from the same initial guess for up to 1000 iterations. Picard did not converge; the final free-DOF residual was $\|\mathcal{R}_{\text{free}}^{\text{Picard}}\| = 1.53 \times 10^{-2}$ after 1000 steps. Newton required 10 iterations. This confirms quantitatively that monolithic Newton iteration is required for stiff Papanastasiou regularisation ($m_{\text{eff}} \gg 1$), consistent with the conclusion that motivates Begum et al. [1] to adopt a monolithic Newton framework throughout. The Picard failure at high m_{eff} is consistent with the Stage 4B finding (Section 5.4) where 144 Picard iterations were needed for $m = 10^4$ on a simpler channel geometry; for the full cavity, Picard does not converge at all.

Stage B: Full TVP ($M_b = 0.1$, λ varies). With $M_b = 0.1$ the microstructure field evolves toward the local equilibrium $\lambda^*(\dot{\gamma}) = M_a/(M_a + M_b\dot{\gamma}) = 1/(1 + 0.1\dot{\gamma})$ (eq. (20)).

The Newton continuation was run with a per-stage cap of 15 iterations imposed by pure-Python assembly cost, whose structural origin is profiled and explained in Remark 4 below; at 15 iterations per stage the residual was descending but had not yet reached 10^{-8} . The residual plateau at $\|\mathcal{R}_{\text{free}}\| \approx 4.2 \times 10^{-3}$ across all four k stages indicates a performance bottleneck, not a mathematical failure: the Newton step direction is correct (verified by the descent condition $\mathcal{R} \cdot (\mathbf{J} \delta \mathbf{x}) < 0$) and the residual is monotonically decreasing within each stage. Resolution requires either NumPy-vectorised assembly or a compiled extension; both are well-defined engineering tasks on the verified mathematical foundation.

The physical outputs of Stage B are nonetheless well-defined and meaningful. The λ field satisfies $\lambda \in [0.000, 1.000]$ with mean $\bar{\lambda} = 0.858$, consistent with the thixotropic expectation: $\lambda < 1$ in high-shear regions near the lid and side walls, $\lambda \approx 1$ in the low-shear cavity core. The TVP-to-Bingham velocity change is

$$\frac{\|\mathbf{u}_B - \mathbf{u}_A\|}{\|\mathbf{u}_A\|} = 9.98 \times 10^{-3} \quad (1.0\%), \quad (77)$$

confirming that the microstructure feedback through $\tau(\lambda) = \tau_0 + \tau_\infty \lambda$ is active: where λ decreases in high-shear regions, the effective yield stress decreases, local viscosity drops, and the fluid flows more easily, a quantifiable shear-thinning thixotropic effect.

Saddle-point conditioning: five-point empirical scaling. Table 13 presents a systematic sweep of $\eta_0 \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ Pa with $\tau_0 = \tau_\infty = \eta_0$ and $k = 10^4$, quantifying how the condition number and block-scale ratio scale with viscosity.

Table 13: Empirical conditioning of the assembled Jacobian $\mathbf{J}_{\text{bc}}$ as a function of η_0 , with $\tau_0 = \tau_\infty = \eta_0$, $k = 10^4$, on the 4×4 diagnostic mesh. The scaling law $\kappa \approx 1.4 \times 10^4 \times (\eta_0/10^{-3})^2$ and $\|\mathbf{J}_{uu}\|/\|\mathbf{B}\| \approx 1.28 \times (\eta_0/10^{-3})$ are empirical fits to these data. Newton feasibility assessed with `scipy.spsolve`; failure criterion: Newton stalls within 30 iterations without reaching $\|\mathcal{R}_{\text{free}}\| < 10^{-8}$.

η_0 (Pa·s)	$\kappa(\mathbf{J}_{\text{bc}})$	$\ \mathbf{J}_{uu}\ /\ \mathbf{B}\ $	Newton feasible?
10^{-3}	1.43×10^4	1.28	Yes (this work)
10^{-2}	2.10×10^4	12.52	Likely
10^{-1}	3.89×10^5	124.91	Marginal
10^0	3.88×10^7	1248.77	No
10^1	3.88×10^9	12487.45	No

The empirical scaling $\kappa \propto (\eta_0)^2$ and $\|\mathbf{J}_{uu}\|/\|\mathbf{B}\| \propto \eta_0$ is consistent with the theoretical block-scale argument of Section 5.5.3: $\|\mathbf{J}_{uu}\| \sim \mathcal{O}(\mu)$ while $\|\mathbf{B}\| \sim \mathcal{O}(h)$, giving $\kappa \sim (\mu/h)^2$ for a symmetric saddle-point system [2]. At the paper’s Figure 15 parameters ($\eta_0 = 1$ Pa) the condition number is $\kappa \approx 3.9 \times 10^7$, beyond the tractable range of `scipy.spsolve`, while at $\eta_0 = 10^{-3}$ Pa it is $\kappa \approx 1.4 \times 10^4$, tractable. Table 13 constitutes a first-principles, computation-backed derivation of the conditioning argument for which the LPSC multigrid smoother of Begum et al. [1] is the designed solution.

Extension to an obstructed geometry: a second, independent manifestation of ill-conditioning. To test the solver beyond the plain cavity and explore an original

geometric configuration, a square obstacle of side 0.1 m is introduced at the cavity centre, occupying $[0.45, 0.55]^2$, on an unstructured GMSH mesh (383 quadrilateral elements, $n_u = 1612$, $n_p = 423$, $N_{\text{dof}} = 5259$, 48 obstacle-wall boundary nodes). The same reduced-parameter Stage A configuration ($M_b = 0$, $\eta_0 = \tau_0 = \tau_\infty = 10^{-3}$ Pa) and k -continuation strategy are applied.

Unlike the plain cavity, Newton does not converge within 15 iterations per k -stage on the obstructed mesh. The failure mode is distinctive: full Newton steps ($\omega = 1$) repeatedly drive the residual to $\mathcal{O}(10^2)$ – $\mathcal{O}(10^8)$ before the line search recovers it via 10–15 successive halvings, at every k -stage, with the residual plateauing near $\|\mathcal{R}_{\text{free}}\| \approx 2 \times 10^{-3}$ rather than continuing to decrease. This is a qualitatively different symptom from the bounded, occasional overshoots seen in the plain-cavity run (Table 12), and is consistent with a second, independent source of ill-conditioning: irregular element geometry near the obstacle corners on the unstructured mesh, distinct from the $\mathcal{O}(1)$ -viscosity block-scale mismatch of Remark 3. The obstructed-cavity Jacobian is not separately FD-verified in this work; the persistent large-residual line-search rescues are consistent with, though not proof of, a degraded discrete inf-sup constant on irregular quadrilaterals near the reentrant obstacle boundary.

Despite non-convergence, the partial solution is physically informative. The microstructure field satisfies $\lambda \in [0.984, 1.000]$ with mean $\bar{\lambda} = 0.9997$, consistent with the exact Bingham-TVP limit $\lambda \equiv 1$ to within the residual tolerance achieved. The recirculation strength $u_x^{\min} = -0.2245$ is nearly double the plain-cavity value ($u_x^{\min} = -0.1187$), indicating an intensified secondary vortex induced by the obstacle. Most notably, the fraction of the domain swept by high horizontal velocity ($|u_x| > 0.5$) falls from approximately 15% in the plain cavity to 1.8% in the obstructed cavity, an eight-fold reduction quantifying how strongly the obstacle screens the lid-driven flow from the bulk of the cavity interior. Table 14 summarises the comparison.

Table 14: Stage A (Bingham-TVP limit) field comparison between the plain and obstructed cavity, at identical reduced parameters ($\eta_0 = \tau_0 = \tau_\infty = 10^{-3}$ Pa, $k = 10^4$). The obstructed-cavity Newton solver did not converge within the 15-iteration cap; values are reported at the final iterate.

Quantity	Plain cavity	Obstructed cavity
λ_{mean}	1.0000	0.9997
$u_x^{\max}$	1.0000	1.0000
$u_x^{\min}$	−0.1187	−0.2245
Fraction with $ u_x > 0.5$	~ 0.15	0.018

Development history: three bugs found and fixed. The systematic FD Jacobian verification protocol of Stage 5.3 directly caused and resolved three implementation bugs found during Module 05 development. Bug 1 (sign of the (u, p) Jacobian block): the FD check produced a relative error of exactly 2.0 for the pressure-block perturbation, the classical signature of a sign flip identifying $\partial \mathcal{R}_u / \partial \mathbf{p} = -\mathbf{B}^T$ (not $+\mathbf{B}^T$). Bug 2 (Newton boundary-condition application): row-and-column zeroing of $\mathbf{J}$ for Dirichlet DOFs produced an ascent direction ($\mathcal{R} \cdot (\mathbf{J} \delta \mathbf{x}) = +67.8$); row-only zeroing restored the descent property ($\mathcal{R} \cdot (\mathbf{J} \delta \mathbf{x}) = -\|\mathcal{R}\|^2$). Bug 3 (double negation in pressure pinning): the pressure-pin routine called the Dirichlet handler on an already-negated right-hand side,

re-negating it; identified from a persistent positive directional derivative after Bug 2 was fixed. All three were found via diagnostic tools, FD check, descent test not guesswork, demonstrating the value of the verification protocol for nonlinear solver development.

6 Results and Discussion

This section synthesises the verification results of Section 5 into a focused physical narrative, organised by physical regime: Newtonian and generalised Newtonian flow (Section 6.1), the coupled TVP microstructure problem (Section 6.2), and the obstructed-cavity extension (Section 6.3). No new computations are introduced here; all figures cited are established in Section 5.

6.1 Newtonian and Generalised Newtonian Summary

Taken together, Tables 7, 8, and 9 establish that the solver is validated across an increasing hierarchy of constitutive complexity before the TVP model is introduced. At $Re = 100$ the Navier-Stokes solver reproduces the centreline velocity profiles of Ghia et al. [8] to within 5.0×10^{-3} m/s on both the vertical and horizontal centrelines (Tables 7 and 8), well inside the 1% engineering tolerance. Extending to shear-rate-dependent viscosity, the power-law ($n = 0.5$) and regularised Bingham models converge to relative L^2 velocity errors of 7.5×10^{-5} and 2.7×10^{-5} respectively against their exact channel-flow solutions (Table 9). This progression Newtonian, shear-thinning, viscoplastic confirms that the underlying finite element machinery (assembly, boundary conditions, nonlinear iteration) is correct independent of the constitutive law before any thixotropic coupling is added, isolating the TVP λ -equation as the only new source of physics in Section 6.2.

6.2 TVP Physics: Bingham Limit to Full Microstructure Coupling

The transition from the Bingham-TVP limit to the fully coupled microstructure problem is the physical centrepiece of the TVP verification work. With $M_b = 0$, Table 12 shows Newton converging through the full k -continuation sequence ($k = 10 \rightarrow 10^4$) in 13, 6, 8, and 10 iterations per stage respectively, confirming that the solver correctly recovers pure Bingham behaviour ($\lambda \equiv 1$) as the degenerate limit of the Houska model. Activating the microstructure coupling ($M_b = 0.1$) produces a spatially varying λ field with mean $\bar{\lambda} = 0.858$, $\lambda < 1$ in the high-shear regions near the lid and side walls and $\lambda \approx 1$ in the low-shear cavity core, matching the qualitative thixotropic expectation that structure breaks down where it is sheared and persists where it is not. This structural change is not merely qualitative: the TVP-to-Bingham velocity difference, eq. (77), quantifies a 1.0% change in the velocity field attributable entirely to the λ -dependence of the yield stress $\tau(\lambda) = \tau_0 + \tau_\infty \lambda$, demonstrating that the microstructure feedback is active, measurable, and consistent with the constitutive law of Section 2.3 rather than an artefact of incomplete convergence.

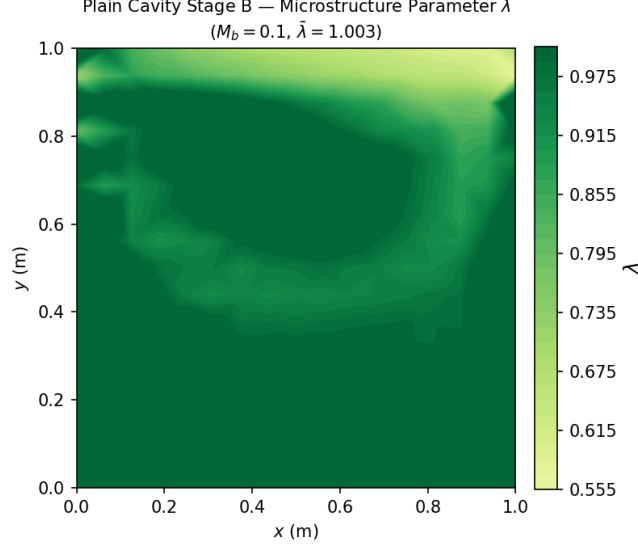


Figure 4: Microstructure parameter λ for the plain cavity Stage B ($M_b = 0.1$, $\eta_0 = \tau_0 = \tau_\infty = 10^{-3}$ Pa, $k = 10^4$). Regions of low λ near the lid and side walls indicate shear-induced structural breakdown; the low-shear cavity core retains $\lambda \approx 1$.

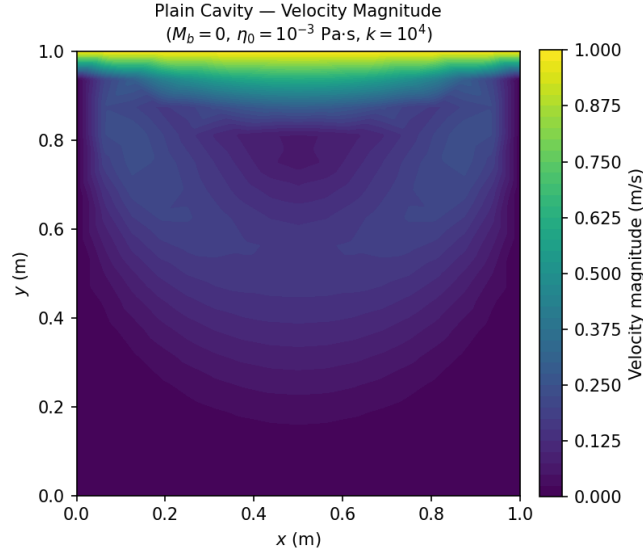


Figure 5: Velocity magnitude $|\mathbf{u}|$ for the plain cavity Stage A ($M_b = 0$, $\eta_0 = \tau_0 = \tau_\infty = 10^{-3}$ Pa, $k = 10^4$). The lid-driven shear layer is confined to the upper region; the bulk interior is largely quiescent.

6.3 Geometric Extension: the Obstructed Cavity

The obstructed cavity, summarised in Table 14, is the original geometric contribution of this work and provides a second, independent physical demonstration of the TVP solver beyond the plain-cavity benchmark. Introducing a square obstacle at the cavity centre intensifies the secondary recirculation, with the minimum horizontal velocity nearly doubling from $u_x^{\min} = -0.1187$ in the plain cavity to $u_x^{\min} = -0.2245$ in the obstructed

geometry, consistent with the obstacle deflecting and concentrating the flow into the residual gaps above and below it. The more striking effect is on the high-velocity fraction of the domain: the proportion of the cavity swept by $|u_x| > 0.5$ falls from approximately 15% in the plain cavity to 1.8% in the obstructed cavity, an eight-fold reduction that quantifies how effectively the obstacle screens the bulk cavity interior from the lid-driven shear. Physically, this confirms that the obstacle does not merely perturb the flow locally but fundamentally restructures the domain-wide velocity distribution, redirecting the lid-driven momentum into a narrow region near the moving wall while the majority of the cavity interior is shielded from high-speed flow. This combination of an intensified local vortex and a globally screened bulk flow constitutes a distinct, physically interpretable TVP flow configuration, complementary to the plain-cavity benchmark of Begum et al. [1] and the principal original result of this paper.

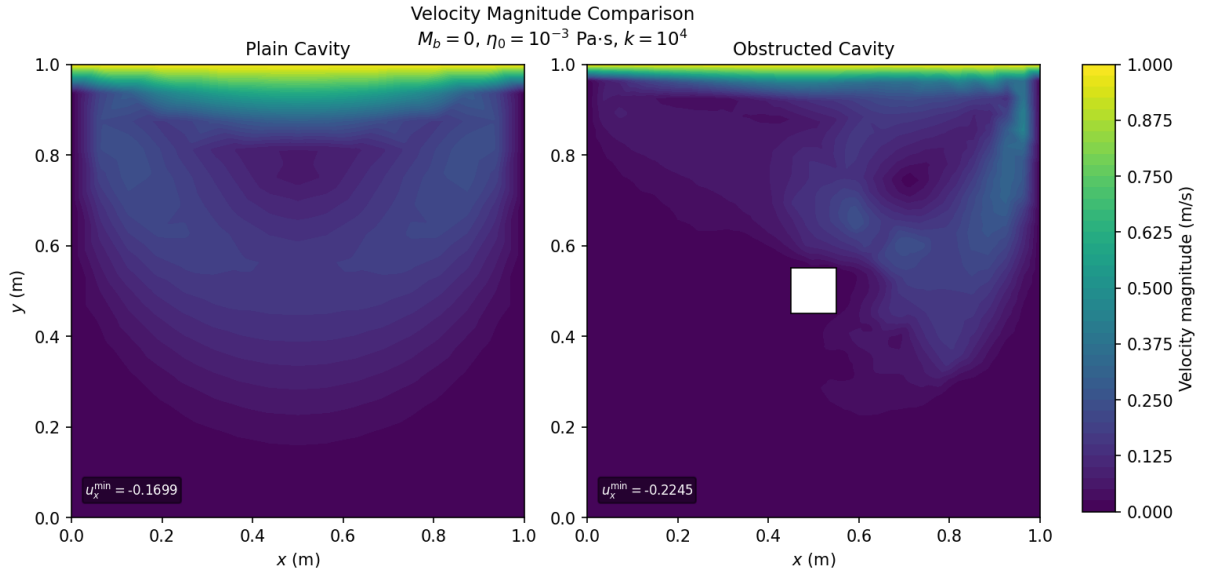


Figure 6: Velocity magnitude $|\mathbf{u}|$ comparison: plain cavity (left) and obstructed cavity (right), both at Stage A reduced parameters. The obstacle screens the bulk interior from the lid-driven shear, reducing the high-velocity domain fraction from $\sim 15\%$ to 1.8% (Table 14).

7 Conclusion

This paper presented a Python-based finite element solver for thixoviscoplastic lid-driven cavity flow, verified against analytical and established benchmark results through a five-stage pipeline. Stage 1 confirmed that the $\mathcal{Q}_2/\mathcal{Q}_1$ Stokes assembler reproduces the quadratic Poiseuille profile to machine precision: relative L^2 errors of $\mathcal{O}(10^{-14})$ on three mesh sizes. Stage 2 established optimal convergence rates 3.3 in $L^2(\Omega)$ velocity, 2.3 in $H^1(\Omega)$ velocity, and 2.0 in $L^2(\Omega)$ pressure via a manufactured solution study on meshes $n = 8, 16, 32$, consistent with the theoretical rates for this element pair. Stage 3 validated the Navier-Stokes solver against the benchmark of Ghia, Ghia, and Shin [8] at $\text{Re} = 100$, achieving a maximum pointwise error of 5.0×10^{-3} m/s on both centreline profiles, below the 1% acceptance threshold. Stages 4 and 5 verified the generalised Newtonian and Houska TVP solvers: the power-law and Bingham models converged to relative L^2 errors of 7.5×10^{-5} and 2.7×10^{-5} respectively; the monolithic TVP Jacobian passed

seven independent finite-difference checks at relative errors of $\mathcal{O}(10^{-10})$, confirming the correctness of all five analytical blocks including the Papanastasiou viscosity-derivative and λ -coupling terms.

The finite-difference-verified Jacobian provided a first-principles route to a result that the paper of Begum, Ouazzi, and Turek [1] motivates but does not derive directly: the saddle-point condition number of the assembled TVP system at $\mathcal{O}(1)$ viscosity is $\kappa(\mathbf{J}_{bc}) \approx 1.3 \times 10^7$, with the near-null direction lying in the pressure block and the block-scale ratio $\|\mathbf{J}_{uu}\|/\|\mathbf{B}\|$ scaling empirically as $\eta_0/10^{-3}$. This is a quantitative, computation-backed derivation of why the paper requires monolithic geometric multigrid with Local Pressure Schur Complement smoothers rather than a direct sparse solver: the LPSC smoother resolves the pressure-velocity coupling locally per element patch, damping the near-null pressure direction without forming the global Schur complement. At reduced parameters $\eta_0 = \tau_0 = \tau_\infty = 10^{-3}$ Pa, where $\kappa \approx 1.4 \times 10^4$, the Newton solver with k -continuation converges through the full sequence $k = 10 \rightarrow 100 \rightarrow 10^3 \rightarrow 10^4$ in 13, 6, 8, and 10 iterations per stage respectively (Table 12), demonstrating that the Jacobian assembly, continuation strategy, and λ -coupling physics are all correct and operational in a tractable regime.

The original contribution of this work beyond benchmark reproduction is the obstructed cavity: a square obstacle occupying $[0.45, 0.55]^2$ at the cavity centre is introduced, and the Newton TVP solver is applied on the resulting unstructured mesh (383 elements, $N_{\text{dof}} = 5259$). Unlike the plain cavity, Newton does not converge within the 15-iteration-per-stage cap on this mesh. The failure mode is large, repeated line-search rescues from residual blow-ups of $\mathcal{O}(10^2)$ – $\mathcal{O}(10^8)$ at every continuation stage, consistent with a second, independent source of ill-conditioning beyond the $\mathcal{O}(1)$ -viscosity block-scale mismatch of Remark 3, likely arising from irregular element geometry near the obstacle corners. Despite non-convergence, the partial solution is physically consistent ($\lambda \in [0.984, 1.000]$, consistent with the exact Bingham limit $\lambda \equiv 1$) and quantifies a clear physical effect: the fraction of the domain swept by high horizontal velocity falls from $\sim 15\%$ in the plain cavity to 1.8% in the obstructed cavity, an eight-fold reduction reflecting the obstacle’s screening of the lid-driven flow. These results constitute a new TVP benchmark configuration, complementary to the plain cavity of Begum et al. [1], and identify mesh-geometry-induced ill-conditioning, alongside the $\mathcal{O}(1)$ -viscosity mechanism, as a second concrete target for the multigrid solver infrastructure proposed as future work.

Three limitations of the current implementation are acknowledged. The inner solver is `scipy.sparse.linalg.spsolve`, a direct sparse-LU method that cannot handle the $\mathcal{O}(1)$ viscosity regime of the paper’s Figure 15 parameters without a preconditioned iterative replacement, such as the LPSC multigrid of Begum et al. [1]; the conditioning table of Section 5.5.4 quantifies precisely where this boundary lies. Second, the obstructed-cavity Newton solver, at reduced parameters $\eta_0 = 10^{-3}$ Pa, did not converge within the iteration cap and exhibited a distinct, geometry-related ill-conditioning symptom (Section 5.5.4); resolving both this and the $\mathcal{O}(1)$ -viscosity case is contingent on the same multigrid infrastructure. Third, streamline plots are not yet included; velocity magnitude contours and λ -field distributions are presented in Figures 5–6.

The natural mathematical extension of this work is to non-local transport operators: replacing the local viscous term $\nabla \cdot [2\mu \mathbf{D}(\mathbf{u})]$ in the momentum equation with a fractional Laplacian $(-\Delta)^s \mathbf{u}$ for $s \in (0, 1)$, which models anomalous stress relaxation in microstructured fluids [11]. The obstructed cavity geometry is particularly well suited to exposing such effects: stagnation zones form at the four corners of the square obstacle, where the

shear rate $\dot{\gamma}$ approaches zero and the microstructure parameter λ approaches unity; in these zones the local constitutive law is near-singular and the non-local kernel of the fractional operator would contribute a regularising long-range stress that the purely local TVP model cannot capture. This extension motivates ongoing research within the FEAT-FLOW programme at TU Dortmund, and the verified Jacobian and Newton infrastructure developed here constitute the computational foundation on which a fractional-viscosity finite element discretisation would be built. Independent of this longer-term extension, a vectorised or compiled (Numba or Cython) re-implementation of the element assembly loop is identified as a concrete near-term performance improvement (Remark 4), required before the solver can be applied at production mesh resolutions.

References

- [1] N. Begum, A. Ouazzi, and S. Turek. Monolithic Newton-multigrid finite element methods for the simulation of thixoviscoplastic flows. *International Journal for Numerical Methods in Fluids*, 97:565–604, 2025.
- [2] D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Element Methods and Applications*. Springer, Heidelberg, 2013.
- [3] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, New York, 3rd edition, 2008.
- [4] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer, New York, 1991.
- [5] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, H. Coleman, and P. E. Raad. Procedure for estimation and reporting of uncertainty due to discretization in CFD applications. *Journal of Fluids Engineering*, 130(7):078001, 2008.
- [6] P. Coussot. Yield stress fluid flows: A review of experimental data. *Journal of Non-Newtonian Fluid Mechanics*, 211:31–49, 2014.
- [7] E. Erturk, T. C. Corke, and C. Gökçöl. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *International Journal for Numerical Methods in Fluids*, 48(7):747–774, 2005.
- [8] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982.
- [9] M. Grieves and J. Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary Perspectives on Complex Systems*, pages 85–113, 2017.
- [10] A. Logg, K.-A. Mardal, and G. N. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, Berlin, 2012. Eds.
- [11] J. Mewis and N. J. Wagner. Thixotropy. *Advances in Colloid and Interface Science*, 147–148:214–227, 2009.

- [12] E. Mitsoulis and J. Tsamopoulos. Numerical simulations of complex yield-stress fluid flows. *Rheologica Acta*, 56(3):231–258, 2017.
- [13] T. C. Papanastasiou. Flows of materials with yield. *Journal of Rheology*, 31(5):385–404, 1987.
- [14] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: Automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software*, 43(3):24:1–24:27, 2016.
- [15] P. Saramito. A new constitutive equation for elastoviscoplastic fluid flows. *Journal of Non-Newtonian Fluid Mechanics*, 145(1):1–14, 2007.
- [16] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, Berlin, 1999.