

# Author’s Response to Reviewers

Manuscript: “Efficient Uzawa algorithms with projection strategies  
for geodynamic Stokes flow”

by Jang, Lee, Thieulot, Choi and So

## Cover Letter to the Editor

Dear Editor,

We sincerely thank both reviewers for their constructive and insightful comments, which have substantially improved the manuscript. We have carefully addressed all comments from both reviews. The detailed point-by-point responses to each reviewer are provided in the subsequent sections of this document.

Below, we summarize the main changes that will be reflected in the revised manuscript.

### Changes in response to Reviewer 1:

- Revised the abstract and introduction to position the contribution as a complementary algorithmic perspective rather than a replacement for modern block-preconditioned Krylov solvers.
- Added a mesh refinement study using the SolKz benchmark, demonstrating optimal convergence rates for P2-P1 Taylor-Hood elements up to  $1024 \times 1024$  resolution (Appendix A).
- Added a multisinker benchmark following Rudi et al. (2017), demonstrating solver robustness for dynamic ratios up to  $10^{10}$ .
- Added a systematic comparison of projection post-processing versus additional iterations, showing that projection achieves superior accuracy at minimal computational overhead (Appendix B).
- Introduced the adaptive AAR strategy for high-resolution cases where the preconditioner stagnates.

### Changes in response to Reviewer 2:

- Added subplot labels and improved captions to all figures.
- Added structured algorithm summary boxes for all algorithms in Sections 2.2 and 2.3.
- Added a three-dimensional thermochemical convection benchmark using Mini elements, confirming the effectiveness of the projection post-processing in three dimensions (Appendix C).
- Clarified the definition of thermal diffusivity  $\kappa$  and specified representative mantle viscosity values.
- Revised inner solver tolerance discussion with references to ASPECT default parameters.

All modifications can be reviewed in the accompanying latexdiff PDF, which presents a marked-up version of the revised manuscript with additions and deletions clearly indicated. While our finalized author comments on EGU sphere referenced color-coded highlighting per reviewer (blue for Reviewer 1, green for Reviewer 2), we have adopted the latexdiff markup format here following the journal's standard practice for revised submissions.

We believe the revised manuscript addresses all concerns raised by the reviewers and hope it is now suitable for publication.

Sincerely,

Deok-Kyu Jang, Kyeong-Min Lee, Cedric Thieulot, Whan-Hyuk Choi, and Byung-Dal So

## Response to Reviewer 1 (Professor Stadler)

Dear Editor and Professor Stadler,

We sincerely thank Professor Stadler for the thorough and constructive review of our manuscript. We appreciate the positive feedback regarding the clarity of writing and careful preparation, and the insightful comments have helped us significantly improve the manuscript. Below, we provide detailed point-by-point responses to all comments and describe the corresponding changes that will be reflected in the revised manuscript. All modifications can be reviewed in the accompanying latexdiff PDF.

### General Comments

**Reviewer:** *As the authors acknowledge in the introduction, Uzawa methods are no longer regarded as state-of-the-art for Stokes problems; contemporary approaches typically handle velocity and pressure simultaneously and generally achieve superior convergence. The authors argue that Uzawa methods are simple, modular, and therefore easy to implement. I am not entirely persuaded by this claim. Given the solver technology available in modern FEM and linear algebra libraries (such as FEniCS and PETSc), implementing a full-space Stokes solver with a block-diagonal preconditioner is not particularly difficult.*

#### Response:

We thank the reviewer for this observation. Classical Uzawa iterations are indeed no longer the standard approach. However, our reformulation through variational principles enables conjugate gradient methods for the Schur complement, achieving robust convergence for highly heterogeneous viscosity problems. The key insight is that the inverse  $\eta$ -weighted operations in our solver correspond directly to  $L^2$  inner products in the finite element space, making the approach a natural extension of the variational framework rather than an ad hoc modification. Additionally, we introduce a projection post-processing step that reduces error accumulation in time-dependent problems at minimal computational cost. Our numerical experiments demonstrate competitive performance across various benchmarks, and we believe the proposed method is competitive with modern full-space approaches while retaining the modularity of segregated solvers. We have revised the introduction accordingly.

#### Changes in manuscript:

The abstract and introduction have been revised to explicitly state that the proposed method reformulates the Uzawa iteration through weighted  $L^2$  inner products, yielding a robust preconditioned conjugate gradient iteration for the Schur complement system, and to position this contribution as a complementary algorithmic perspective rather than a replacement for modern block-preconditioned Krylov solvers.

### Specific Comments

#### Comment 1: Line 71 – Vector Laplacian and boundary conditions

**Reviewer:** *As for the statement that the constant-viscosity Stokes system reduces to the vector Laplacian: disregarding boundary conditions this is certainly true, and it also holds for Dirichlet conditions. But is it still generally valid for more complex boundary conditions?*

**Response:**

The reviewer is correct to point out this important distinction. The reduction to vector Laplacian is exact for Dirichlet boundary conditions, but for Neumann or mixed boundary conditions, the pressure gradient term must be handled differently and the pressure-velocity coupling must be retained. We have qualified the statement in line 71 to avoid confusion.

**Changes in manuscript:**

The statement on the reduction to the vector Laplacian has been qualified to: *“In the case of constant viscosity with Dirichlet boundary conditions, the momentum equation reduces to a vector Laplacian.”*

**Comment 2:  $\eta$ -weighted preconditioner notation**

**Reviewer:** *Throughout the paper, the authors refer to the  $\eta$ -weighted version of the preconditioner. Is this weight not actually the inverse viscosity?*

**Response:**

The reviewer is absolutely correct. As shown in Eq. (18),  $\langle \frac{1}{\eta} w^k, r \rangle = \langle q^k, r \rangle$ , the weighting is indeed by  $1/\eta$  (inverse viscosity). We have added clarification in the manuscript to avoid confusion while keeping the notation CD-U- $\eta$  for brevity.

**Changes in manuscript:**

A clarification has been added noting that the weighting is by  $1/\eta$  (inverse viscosity), denoted as  $\eta$ -weighted for brevity: *“Note that the weighting is by  $1/\eta$ , i.e., the inverse of viscosity, which we denote as  $\eta$ -weighted for brevity throughout this paper.”* The notation CD-U- $\eta$  remains unchanged throughout.

**Comment 3: Line 148 – Divergence-free condition and projection**

**Reviewer:** *The authors argue that the divergence-free condition is slightly violated due to inexact solves and that this is corrected by the Helmholtz projection. I can see this being the case if Taylor–Hood elements are also used in the projection step, which appears to be what is done. Would this issue vanish entirely if more iterations were used so that the system is solved more accurately? This could be investigated further in the numerical experiments.*

**Response:**

The reviewer raises two important points, both of which are correct.

First, regarding the projection step, the reviewer is correct that we use Taylor-Hood (P2-P1) elements throughout, including in the projection. The projection solves a Poisson equation for the scalar potential  $\phi$  in the P1 pressure space, then updates the velocity through  $u^{k+1} = \hat{u} + \nabla \phi$  in the P2 velocity space. While projection is theoretically designed to enforce  $\nabla \cdot u = 0$  exactly, the measured divergence residual does not reach machine zero in practice. This happens because the discrete Laplacian operator, which is assembled from the variational form  $\langle \nabla \phi, \nabla \psi \rangle$ , does not exactly equal the composition  $BG$ , where  $B : V_h \rightarrow Q_h$  and  $G : Q_h \rightarrow V_h$  denote the discrete divergence and gradient operators, respectively. The effect of projection in a single time step is small. However, in long time-dependent simulations with thousands of steps, these small improvements accumulate to produce substantial differences in accuracy.

Second, the reviewer correctly notes that with sufficiently many iterations, projection becomes unnecessary. In principle, if we iterate until very tight tolerances, the divergence constraint is satisfied without projection. However, we deliberately use relaxed tolerances in time-dependent simulations to maintain efficiency. This raises a key question: which is more cost-effective—more iterations or fewer iterations with projection?

To answer this, we tested both strategies on two time-dependent benchmarks: mantle convection and block sinking. In the results below, “CD-U- $\eta(niter)$ ” denotes  $n$  iterations of the CD-U- $\eta$  solver per time step, and “+Proj” indicates projection post-processing. Numbers in parentheses show the projection overhead.

**Mantle convection (5000 steps,  $\Delta t = 10^{-6}$ ):**

|                           | Uzawa time   | $E_u$     | $R_{div}$ | $E_T$     |
|---------------------------|--------------|-----------|-----------|-----------|
| CD-U- $\eta(2iter)$       | 1277s        | 0.0177    | 0.0138    | 0.00447   |
| CD-U- $\eta(2iter)$ +Proj | 1355s (+78s) | 0.000633  | 0.00144   | 0.000120  |
| CD-U- $\eta(3iter)$       | 1704s        | 0.00775   | 0.00975   | 0.00187   |
| CD-U- $\eta(3iter)$ +Proj | 1783s (+79s) | 0.0000970 | 0.000263  | 0.0000178 |

**Block sinking (1500 steps,  $\Delta t = 5 \times 10^{-6}$ ):**

For this benchmark, we used a practical strategy: the initial step iterates to tolerance, while subsequent steps use fixed iteration counts with the previous solution as initial guess.

|                           | Uzawa time   | $E_u$   | $R_{div}$ | $E_\rho$ |
|---------------------------|--------------|---------|-----------|----------|
| CD-U- $\eta(5iter)$       | 1253s        | 0.0449  | 0.0234    | 0.0236   |
| CD-U- $\eta(5iter)$ +Proj | 1278s (+25s) | 0.00200 | 0.00384   | 0.00127  |
| CD-U- $\eta(6iter)$       | 1463s        | 0.0336  | 0.0198    | 0.0160   |
| CD-U- $\eta(6iter)$ +Proj | 1485s (+22s) | 0.00875 | 0.00608   | 0.00113  |

Both benchmarks show that projection adds minimal overhead (2-6% of total runtime) while improving accuracy by 1-2 orders of magnitude. This efficiency comes from the cost difference: each additional iteration requires solving the momentum equation, while projection requires only one additional Poisson solve per time step. Although projection can only be applied once per time step, it can enhance any iterative Stokes solver as a post-processing step.

**Changes in manuscript:**

An appendix has been added presenting a comparison of projection post-processing versus additional iterations for two time-dependent benchmarks (mantle convection and block sinking), demonstrating that projection achieves superior accuracy at lower computational cost than simply increasing the iteration count.

**Comment 4: Mesh refinement studies**

**Reviewer:** *Figure 1 (and, in fact, all experiments) use a \*fixed\* discretization. Yet, one of the most important properties of numerical methods is their behavior under mesh refinement. It would be valuable to show results for at least one level of finer mesh.*

**Response:**

We have conducted mesh refinement studies using the SolKz benchmark from Duretz et al. (2011). We chose SolKz over the SolCx benchmark presented in our manuscript because SolCx’s

discontinuous viscosity at  $x = 0.5$  reduces solution regularity, preventing observation of optimal convergence rates with standard finite elements. SolKz provides smooth exponentially varying viscosity  $\eta(y) = \exp(cy)$  with dynamic ratio  $10^6$ , enabling proper verification of convergence rates using P2-P1 Taylor-Hood elements.

We tested resolutions from  $32 \times 32$  to  $1024 \times 1024$  with reference solutions obtained from the Underworld software. The following table shows the mesh refinement results. Our method achieves the optimal convergence rates for Taylor-Hood elements (order 3 for velocity and order 2 for pressure).

| Resolution         | $E_u$                  | Order | $E_p$                  | Order |
|--------------------|------------------------|-------|------------------------|-------|
| $32 \times 32$     | $1.877 \times 10^{-3}$ | -     | $1.591 \times 10^{-2}$ | -     |
| $64 \times 64$     | $2.316 \times 10^{-4}$ | 3.02  | $3.970 \times 10^{-3}$ | 2.00  |
| $128 \times 128$   | $2.888 \times 10^{-5}$ | 3.00  | $9.872 \times 10^{-4}$ | 2.01  |
| $256 \times 256$   | $3.610 \times 10^{-6}$ | 3.00  | $2.401 \times 10^{-4}$ | 2.04  |
| $512 \times 512$   | $4.514 \times 10^{-7}$ | 3.00  | $5.845 \times 10^{-5}$ | 2.04  |
| $1024 \times 1024$ | $5.648 \times 10^{-8}$ | 3.00  | $1.490 \times 10^{-5}$ | 1.97  |

#### Adaptive AAR strategy for high-resolution cases.

At high resolution ( $1024 \times 1024$ ), the conjugate gradient iteration occasionally stagnates. This occurs when the denominators in the step-size and conjugacy coefficients become too small (near machine precision), causing the updates to effectively stop since we set the coefficients to zero to avoid division by near-zero values. To overcome this, we introduce an adaptive AAR parameter that scales the preconditioner weights. Instead of using  $1/\eta(x)$  directly in the  $\eta$ -weighted mass matrix, we use  $\text{AAR}/\eta(x)$ .

The general strategy is as follows. We start from  $\text{AAR} = 1.0$  and reduce it when stagnation is detected. However, as AAR is progressively reduced, the weights can become excessively small in high-viscosity regions, which introduces numerical instability. Therefore, we enforce a pointwise lower bound on the weights.

For the SolKz benchmark, we implement this strategy as follows. When  $|R_{\text{div}}^{k+1} - R_{\text{div}}^k| < 10^{-10}$  for 2 consecutive iterations, AAR is reduced by a factor of 0.1. To maintain numerical stability, we enforce  $\min_x(\text{AAR}/\eta(x)) \geq 10^{-6}$ .

The following table illustrates the adaptive AAR strategy for the  $1024 \times 1024$  case. The solver initially converges normally (iterations 4-7) but stagnates at iteration 8. Adaptive AAR reduction immediately resumes progress, ultimately achieving  $E_u \sim 5.6 \times 10^{-8}$  and  $R_{\text{div}} \sim 1.6 \times 10^{-8}$  by iteration 500.

| Iteration | AAR    | AAR change                 | $E_u$    | $E_p$    |
|-----------|--------|----------------------------|----------|----------|
| 4         | 1.0    | —                          | 1.99e-03 | 2.03e-01 |
| 5         | 1.0    | —                          | 2.55e-04 | 1.77e-02 |
| 6         | 1.0    | —                          | 2.57e-05 | 1.22e-03 |
| 7         | 1.0    | —                          | 2.61e-07 | 3.10e-05 |
| 8         | 1.0    | —                          | 2.61e-07 | 3.10e-05 |
| 9         | 1.0    | —                          | 2.61e-07 | 3.10e-05 |
| 10        | 0.1    | 1.0 $\rightarrow$ 0.1      | 5.75e-08 | 2.17e-05 |
| 13        | 0.01   | 0.1 $\rightarrow$ 0.01     | 5.75e-08 | 2.17e-05 |
| 15        | 0.001  | 0.01 $\rightarrow$ 0.001   | 5.67e-08 | 2.16e-05 |
| 388       | 0.0001 | 0.001 $\rightarrow$ 0.0001 | 5.65e-08 | 1.50e-05 |
| 500       | 0.0001 | —                          | 5.65e-08 | 1.49e-05 |

For resolutions up to  $256 \times 256$ , the adaptive strategy was not triggered and the solver converged with  $\text{AAR} = 1.0$  throughout. At  $512 \times 512$ , AAR adjustment occurred occasionally. At  $1024 \times 1024$ ,

the adaptive strategy was actively used and proved essential for convergence. We also apply this technique in the multisinker benchmark discussed in Comment 5.

### Changes in manuscript:

A mesh refinement study using the SolKz benchmark has been added, demonstrating convergence across increasing resolutions up to  $1024 \times 1024$ . The adaptive AAR strategy is described in the main text, and the corresponding numerical results are provided in an appendix.

### Comment 5: More challenging benchmarks

**Reviewer:** *I would conjecture that the benchmarks in Sec. 3.3.1 and 3.3.2 have rather smooth solutions, so I am not convinced that these are the most informative problems to consider. The May/Moresi sinker example (Sec. 3.4.2) is more demanding, although a multi-sinker setup (e.g., May et al., SC Proceedings, 2014; Rudi et al., SISC 2017) is known to be even more challenging and would definitely be insightful.*

### Response:

We thank the reviewer for this insightful suggestion. Following the recommendation, we conducted multisinker benchmark experiments based on the setup described in Rudi et al. (2017, SISC, “Weighted BFBT preconditioner for Stokes flow problems with highly heterogeneous viscosity”).

#### Experimental setup:

- Domain: unit square  $\Omega = [0, 1]^2$  with 8 high-viscosity sinkers at prescribed locations
- Viscosity field:  $\mu(x) = (\mu_{\max} - \mu_{\min})(1 - \chi_8(x)) + \mu_{\min}$ , where  $\chi_8(x) = \prod_{i=1}^8 [1 - \exp(-\delta \max(0, |c_i - x| - \omega)^2)]$  with  $\delta = 200$ ,  $\omega = 0.05$
- Body force:  $\mathbf{f}(x, y) = (0, \beta(\chi_8(x, y) - 1))$  with  $\beta = 10^5$
- Dynamic ratios:  $\text{DR}(\mu) = 10^2$  to  $10^{10}$ , with  $\mu_{\min} = \text{DR}^{-1/2}$  and  $\mu_{\max} = \text{DR}^{1/2}$
- Discretization:  $256 \times 256$  triangular P2-P1 Taylor-Hood elements
- Reference solution: MUMPS direct solver
- Adaptive AAR strategy: same as Comment 4, with stagnation threshold  $10^{-8}$  and lower bound  $\min_x(\text{AAR}/\eta(x)) \geq 10^{-8}$ .

**Results:** Since this multisinker problem has no analytical solution, we compute reference solutions using the MUMPS direct solver to evaluate convergence. Unlike the original study by Rudi et al. (2017), which used residual reduction as the stopping criterion, we measure convergence using relative errors in velocity and pressure. This approach provides a more direct assessment of solution accuracy. The following table presents the number of CD-U- $\eta$  iterations required to achieve three different accuracy levels ( $10^{-3}$ ,  $10^{-6}$ , and  $10^{-9}$ ) across dynamic ratios ranging from  $10^2$  to  $10^{10}$ . The reported iteration count is the number of iterations needed for both velocity and pressure relative errors to simultaneously fall below the specified threshold.

| $\text{DR}(\mu)$ | $E_u, E_p < 10^{-3}$ | $E_u, E_p < 10^{-6}$ | $E_u, E_p < 10^{-9}$ |
|------------------|----------------------|----------------------|----------------------|
| $10^2$           | 20                   | 37                   | 52                   |
| $10^4$           | 65                   | 83                   | 93                   |
| $10^6$           | 84                   | 93                   | 103                  |
| $10^8$           | 83                   | 94                   | 108                  |
| $10^{10}$        | 83                   | 95                   | 109                  |

**Changes in manuscript:**

Multisinker benchmark results have been added, including a figure illustrating the viscosity field and the corresponding velocity and pressure solutions, to demonstrate solver robustness under strongly heterogeneous viscosity configurations.

## Response to Reviewer 2 (Professor Schmalholz)

Dear Editor and Professor Schmalholz,

We sincerely thank Professor Schmalholz for the positive and constructive review of our manuscript. We appreciate the recognition that the manuscript is clearly structured and of interest for readers developing numerical algorithms for geodynamic processes. The helpful comments and suggestions have further improved the manuscript. Below, we provide detailed point-by-point responses to all comments and describe the corresponding changes that will be reflected in the revised manuscript. All modifications can be reviewed in the accompanying latexdiff PDF.

### Main Comment

#### 3D benchmark comparison

**Reviewer:** *The authors motivate their work in line 18 with 3D problems which indeed represent a major challenge. However, the authors do not show the computational performance and advantage of their new algorithm for 3D problems (except the ABC Flow). The presented examples are useful, but I think a 3D example, for example a simple 3D thermal convection or subduction scenario, would increase the relevance of the presented algorithms. One could show results of a 3D simulation done with a “standard” solver and compare it to the results of a 3D simulation done with the solver presented in the manuscript. Such comparison could maybe better show the advantage and potential computational gain of the solver presented. But this is just a suggestion.*

#### Response:

We thank the reviewer for this constructive suggestion. Following this recommendation, we have added a three-dimensional thermochemical convection benchmark in Appendix C. The problem extends the two-dimensional configuration of Section 3.4 to three dimensions by adding a unit-depth lateral dimension, using the same boundary-layer initial temperature field, basal dense layer, and temperature-dependent viscosity. The domain  $[0, 2] \times [0, 1] \times [0, 1]$  was discretized with a  $100 \times 50 \times 50$  structured tetrahedral mesh using Mini elements (P1+Bubble/P1), which satisfy the LBB stability condition and thus provide an independent verification that the CD-U- $\eta$  algorithm operates correctly with a different stable element pair than the Taylor-Hood elements used in the two-dimensional benchmarks. The transport equations for temperature and composition were discretized with P1 discontinuous Galerkin elements using upwind fluxes and advanced in time with the second-order BDF2 scheme, which ensures stable advection without requiring additional stabilization. Since the purpose of this three-dimensional test is to verify the algorithmic behavior of CD-U- $\eta$  and the projection post-processing rather than to assess solution accuracy, we adopted a fixed time step of  $\Delta t = 10^{-5}$  and advanced the solution to the same target time  $t = 5 \times 10^{-3}$  as in the two-dimensional configuration.

Since a direct solver reference solution is not feasible at this mesh resolution in three dimensions, the comparison focuses on the effect of the projection strategy rather than on accuracy relative to a reference. The Stokes system was solved at each time step using CD-U- $\eta$  with and without the projection post-processing, and convergence was monitored through the normalized weak divergence residual  $R_{\text{div}}$ . The results show that the projection consistently reduces  $R_{\text{div}}$  across all time steps, confirming that the effectiveness of the projection post-processing observed in the two-dimensional benchmarks carries over to three-dimensional problems.

Figure 1 shows the temperature and composition fields at  $t = 5 \times 10^{-3}$ , where thermal plumes and entrainment of the basal dense layer are consistent with the two-dimensional results.

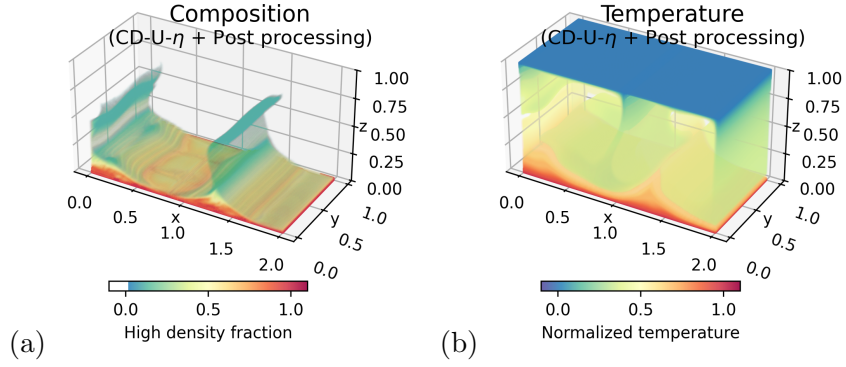


Figure 1: Three-dimensional thermochemical convection at  $t = 5 \times 10^{-3}$  solved with CD-U- $\eta$  + projection on a  $100 \times 50 \times 50$  Mini element mesh. (a) Composition field showing entrainment of the basal dense layer. (b) Normalized temperature field.

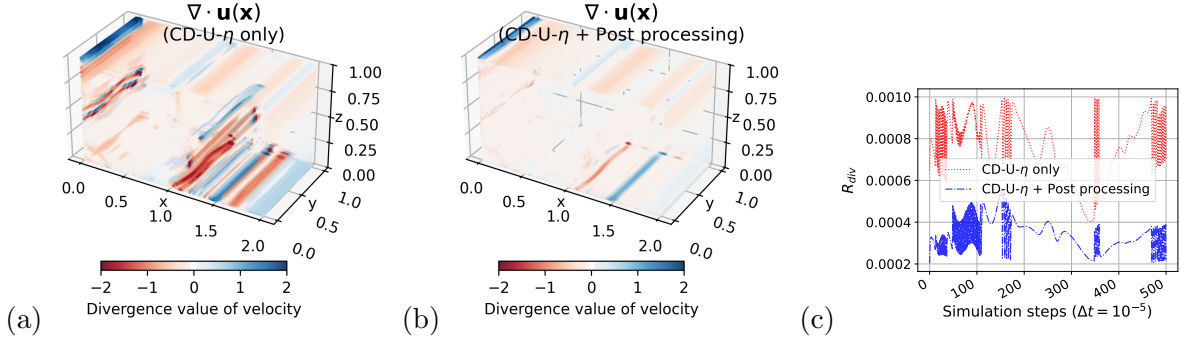


Figure 2: Spatial distribution of  $\nabla \cdot \mathbf{u}$  at  $t = 5 \times 10^{-3}$  for (a) CD-U- $\eta$  only and (b) CD-U- $\eta$  + projection. (c) Evolution of the normalized weak divergence residual  $R_{\text{div}}$  over 501 time steps for CD-U- $\eta$  only and CD-U- $\eta$  + projection.

Figure 2(a) and (b) show that the projection markedly suppresses the divergence error throughout the domain, with the localized concentrations near the plume regions in (a) largely eliminated in (b). The  $R_{\text{div}}$  history in Figure 2(c) confirms that this reduction is maintained consistently across all time steps, with the projection lowering  $R_{\text{div}}$  by approximately a factor of two, consistent with the two-dimensional results.

### Changes in manuscript:

Added Appendix C presenting the three-dimensional thermochemical convection benchmark, including the problem setup, convergence history of  $R_{\text{div}}$  with and without projection, solution snapshots, and the spatial distribution of  $\nabla \cdot \mathbf{u}$  at the final time step.

### Minor Comments

#### Comment 1: Subplot labels and captions

**Reviewer:** *General: Subplots in the figures should be labeled (e.g. a), b) etc.) and the captions could be improved to better explain the figures.*

**Response:**

We thank the reviewer for this suggestion. All figures have been updated with subplot labels (a), (b), etc., and the captions have been revised to provide more detailed descriptions of the quantities shown, including mesh resolution and element type where applicable.

**Changes in manuscript:**

Subplot labels and improved captions added to all figures (Figures 1–8).

**Comment 2: Algorithm presentation**

**Reviewer:** *General: Maybe flowcharts could help to explain the algorithms presented in sections 2.2 and 2.3.*

**Response:**

We thank the reviewer for this suggestion. Since the mathematical formulations are central to the algorithms and the iterations are purely sequential without branching, we opted for structured summary boxes rather than flowcharts. Each box presents the algorithmic steps with descriptive labels alongside the full variational forms, which we believe conveys the sequential structure clearly while preserving the mathematical detail that would be difficult to accommodate in a flowchart format. As an example, the CD-U- $\eta$  algorithm box is shown below.

**CD-U- $\eta$** 

*Initialization:* Set  $p^0 = 0$

*Initial solve:* Find  $u^1$  satisfying  $Ku^1 = f - Gp^0$

*Initial residual:* Find  $q^1$  satisfying  $\langle q^1, r \rangle = \langle h + \nabla \cdot u^1, r \rangle \quad \forall r \in Q$

*Preconditioning:* Find  $w^1$  satisfying  $\langle \frac{1}{\eta} w^1, r \rangle = \langle q^1, r \rangle \quad \forall r \in Q$ , set  $d^1 = -w^1$

*Iteration ( $k \geq 1$ ):*

*Direction solve:* Find  $z^k$  satisfying  $Kz^k = Gd^k$

*Step-size selection:* Compute  $\alpha_k = \langle q^k, w^k \rangle / \langle \nabla d^k, z^k \rangle$

*Pressure update:* Compute  $p^{k+1} = p^k + \alpha_k d^k$

*Velocity update:* Compute  $u^{k+1} = u^k - \alpha_k z^k$

*Residual update:* Find  $q^{k+1}$  satisfying  $\langle q^{k+1}, r \rangle = \langle h + \nabla \cdot u^{k+1}, r \rangle \quad \forall r \in Q$

*Preconditioning:* Find  $w^{k+1}$  satisfying  $\langle \frac{1}{\eta} w^{k+1}, r \rangle = \langle q^{k+1}, r \rangle \quad \forall r \in Q$

*Conjugate direction:* Compute  $\beta_k = \langle q^{k+1}, w^{k+1} \rangle / \langle q^k, w^k \rangle$ , set  $d^{k+1} = -w^{k+1} + \beta_k d^k$

**Changes in manuscript:**

Structured algorithm boxes added for the Standard Uzawa, Adaptive Uzawa, CD-U, CD-U- $\eta$ , and projection post-processing in Sections 2.2 and 2.3.

**Comment 3: Line 12 – High viscosity value**

**Reviewer:** *Maybe provide a value of what you consider as high viscosity.*

**Response:**

We have specified a representative range of mantle viscosity ( $10^{19}$ – $10^{23}$  Pa s) in the revised text.

**Changes in manuscript:**

Line 12: “The mantle has high viscosity” replaced with “The mantle has a viscosity on the order of  $10^{19}$ – $10^{23}$  Pa s.”

**Comment 4: Line 150 – Equation numbering**

**Reviewer:** *Equations are not numbered anymore.*

**Response:**

In our manuscript, we follow the convention that only equations cross-referenced elsewhere in the text receive numbers. Display equations that are self-contained within their local context are left unnumbered. In the revised manuscript, we applied this convention consistently: equations that carried numbers but were never referenced have been changed to unnumbered display equations. In addition, the algorithm presentations in Sections 2.2 and 2.3 have been restructured into summary boxes (see our response to Comment 2), which removed some previously numbered equations. Cross-references that pointed to equations now contained in algorithm boxes have been replaced with descriptive references to the corresponding boxes.

**Changes in manuscript:**

Unreferenced numbered equations have been changed to unnumbered form, and cross-references that pointed to equations now contained in algorithm boxes have been replaced with descriptive references to the corresponding boxes (e.g., “CD-U algorithm box in Section 2.2”).

**Comment 5: Line 198 – Strong viscosity contrast**

**Reviewer:** *Please specify what is a strong viscosity contrast, for example four orders of magnitude or similar.*

**Response:**

We agree that the original phrasing was vague. The viscosity contrasts used in our benchmarks are:  $10^6$  for SolCx (Section 3.3.2),  $10^6$  for SolKz (Appendix A),  $10^2$ – $10^{10}$  for the multisinker problem (Section 3.3.3), and  $10^3$  for the block sinking problem (Section 3.4.2). We have revised the sentence to remove the unspecified claim, since the concrete values are already stated in each benchmark section.

**Changes in manuscript:**

Line 198: “strong viscosity contrasts and heterogeneous distributions” replaced with “across the benchmark problems considered here.”

**Comment 6: Line 203 – Typical tolerances**

**Reviewer:** *“These tolerances are stricter than typical values in geodynamic simulations.” Please provide references for this statement and provide a number for “typical values”.*

**Response:**

We thank the reviewer for pointing this out. The tolerances  $10^{-10}$  (relative) and  $10^{-12}$  (absolute) refer to the inner linear solves within each Uzawa iteration. Our intent was to make these inner solves effectively exact so that the convergence behavior observed in our benchmarks is determined by the outer Uzawa iteration alone, not by inner solver errors. We have revised the text to clarify this rationale. For reference, the ASPECT code (Kronbichler et al., 2012) uses default inner solve tolerances of  $10^{-2}$  for the velocity block and  $10^{-6}$  for the Schur complement block.

**Changes in manuscript:**

Revised the sentence on inner solver tolerances to clarify the rationale and added the ASPECT solver parameters as a reference point (Kronbichler et al., 2012).

**Comment 7: Figure 1 – Numerical resolution**

**Reviewer:** *What was the numerical resolution?*

**Response:**

We thank the reviewer for noting this omission. The ABC flow benchmark in Figure 1 was computed on a  $32 \times 32 \times 32$  structured tetrahedral mesh with P2-P1 Taylor-Hood elements. This information has been added to the figure caption.

**Changes in manuscript:**

Figure 1 caption revised to include the mesh resolution and element type (see also Comment 1).

**Comment 8: Line 283 – Constant  $\kappa$**

**Reviewer:** *If  $\kappa$  is constant you could bring it out of the spatial derivative, since it should also include density and specific heat.*

**Response:**

We thank the reviewer for this remark. We have clarified that  $\kappa$  denotes the thermal diffusivity, defined as  $\kappa = k/(\rho c_p)$  where  $k$  is the thermal conductivity,  $\rho$  the density, and  $c_p$  the specific heat capacity. In the nondimensional formulation used for this benchmark,  $\kappa$  is set to unity.

**Changes in manuscript:**

Clarified the definition of  $\kappa$  as thermal diffusivity with its explicit form  $\kappa = k/(\rho c_p)$ , and noted the nondimensional setting  $\kappa = 1$ .

**Comment 9: Figure 6 – Caption**

**Reviewer:** *The caption should state what simulation the figures shows; the block sinking problem.*

**Response:**

We have revised the caption to explicitly state that Figure 6 shows the block sinking benchmark (see also Comment 1).

**Changes in manuscript:**

Figure 6 caption revised to include “Block sinking benchmark” (see Comment 1).