

IMEX-HDG dicsretisation of incompressible Euler equations

September 19, 2024

The goal of this note is to derive a hybridisable discontinuous Galerkin (HDG) projection method [1] for the time-dependent incompressible Euler equation, similar to [2] where this approach is applied to the Navier-Stokes equations. While in [2] the presence of the Stokes term requires the hybridisation of both the tentative velocity computation and the pressure solve, for the incompressible Euler equations this is only required in the latter step. The starting point of our derivation is the fully implicit DG formulation of the incompressible Euler equations in [3]. Similar to [2], we also discuss higher-order IMEX timestepping methods [4]. To solve the resulting mixed pressure equation we use a Schur-complement approach and the non-nested multigrid preconditioner described in [5].

1 Continuum equations

We consider the incompressible Euler equations for velocity $Q(x, t)$ and pressure $p(x, t)$

$$\begin{aligned} \partial_t Q + (Q \cdot \nabla)Q + \nabla p &= f \\ \nabla \cdot Q &= 0 \end{aligned} \tag{1}$$

in the domain $\Omega = [0, 1]^d$ with the initial condition $Q(x, 0) = Q_0(x)$ and the boundary condition $n \cdot Q|_{\partial\Omega} = 0$ on $\partial\Omega$. f is a time dependent forcing function.

2 Notation

We write Ω_h for the mesh and \mathcal{E}_h for the skeleton of this mesh, i.e. the set of all facets. The set of interior facets is denoted by \mathcal{E}_h^i and the set of boundary facets by \mathcal{E}_h^∂ such that $\mathcal{E}_h = \mathcal{E}_h^i \cup \mathcal{E}_h^\partial$. Further, for scalar quantities a and vector-valued quantities b we define the average $\{\{ \cdot \} \}$ and jump $\llbracket \cdot \rrbracket$ as

$$\{\{a\}\} := \frac{1}{2}(a^+ + a^-), \quad \{\{b\}\} := \frac{1}{2}(b^+ + b^-), \quad \llbracket b \cdot n \rrbracket := b^+ \cdot n^+ + b^- \cdot n^- \tag{2}$$

on interior facets $F \in \mathcal{E}_h^i$ and

$$\{\{a\}\} := a, \quad \{\{b\}\} := b, \quad \llbracket b \cdot n \rrbracket := b \cdot n \tag{3}$$

on boundary facets $F \in \mathcal{E}_h^\partial$. The inner product of two tensors A, B is

$$A : B := \sum_{ij} A_{ij} B_{ji} = \text{trace}(AB^\top). \tag{4}$$

We use the shorthand

$$(f)_A = \int_A f \, dx \quad \text{for } A \subset \Omega \tag{5}$$

for integrals over d -dimensional domains A (typically grid cells) and

$$\langle f \rangle_S = \int_S f \, ds \quad \text{for } S \subset \mathcal{E}_h \tag{6}$$

for integrals over $d - 1$ -dimensional domains (typically facets, boundaries of grid cells).

3 DG discretisation

We follow the DG discretisation with a central flux as described in [3]. Let $P_k(A)$ be the space of polynomials of degree k on A , then we define the DG space as

$$\text{DG}_k := \{v \in L^2(\Omega) : v|_K \in P_k(K) \text{ for all } K \in \Omega_h\} \quad (7)$$

and assume that

$$\begin{aligned} Q &\in V_Q := [\text{DG}_{k+1}]^d, \\ p &\in Q_p := \text{DG}_k \cap \{v : (v)_\Omega = 0\}. \end{aligned} \quad (8)$$

We start by post-processing the velocity Q into the divergence free advection velocity Q^\star which is defined as follows:

3.1 Post-processed advection velocity

Let $\text{ND}_{k-1}(K)$ be the space of Nedelec functions of the first kind [6, Section 3.5.1]¹ on the cell K . In each cell $K \in \Omega_h$ and for a given $Q \in V_Q$ we look for $(Q_K^\star, \sigma_K, r_K) \in W_K = [P_{k+1}(K)]^d \times \text{ND}_{k-1}(K) \times (\oplus_{F \in \partial K} P_{k+1}(F))$ which satisfy the following weak form:

$$(Q_K^\star \cdot \omega_K)_K + \langle (Q_K^\star \cdot n) s_K \rangle_{\partial K} + (v_K \cdot \sigma_K)_K + \langle (v_K \cdot n) r_K \rangle_{\partial K} = (Q \cdot \omega_K)_K + \langle (\{Q\} \cdot n) s_K \rangle_{\partial K \cap \mathcal{E}_h^i} \quad (9)$$

for all test functions $(v_K, \omega_K, s_K) \in W_K$. Note that since the solution satisfies $\sigma_K = 0$ and $r_K = 0$ this is equivalent to Eqs. (2.25) and (2.26) in [3], but to implement these two constraints in Firedrake we need to make sure that the weak form corresponds to a square matrix. In fact, it turns out that a much simpler and entirely equivalent approach is to *interpolate* Q to the $\text{BDM}_{k+1}^0(\Omega_h)$ space. For this, the dofs of the BDM element (which are dual functions in the sense of Ciarlet's definition [7] of a finite element), are evaluated on the function Q , which is averaged on the facets. Explicit expressions for the dofs of the BDM element are given in [6, Section 3.4.2, Definition 3.6].

For future reference we write $Q^\star = \mathcal{P}(Q)$, observing that $\mathcal{P} : V_Q \rightarrow V_Q$ is a linear operator; in fact as argued in [3] \mathcal{P} projects onto the H-div conforming subspace $\text{BDM}_{k+1}^0(\Omega_h)$. This, however, is not a *global* L_2 projection (in the sense of $(Q_K^\star \cdot w)_{\Omega_h} = (Q \cdot w)_{\Omega_h}$ for all $w \in \text{BDM}_{k+1}^0(\Omega_h)$), but rather a local L_2 projection in the sense of Eq. (9).

The normal component of Q^\star is continuous ($\llbracket Q^\star \cdot n \rrbracket = 0$). Since we only consider integrals over interior facets in the final term on the right hand side of Eq. (9), we have that $(n \cdot Q^\star)_{\partial \Omega} = 0$ on the boundary $\partial \Omega$ of the domain. As shown in [3] the post-processed velocity Q^\star is divergence free in the strong sense $\nabla \cdot Q^\star = 0$.

3.2 Original scheme

The discretisation proposed in [3] (see Eqs. (2.27) and (2.28) there) is: for given forcing function f find $(Q, p) \in V_Q \times V_p$ such that

$$\begin{aligned} (\partial_t Q \cdot w)_K - (Q \otimes Q^\star : \nabla w)_K - (p \nabla \cdot w)_K + \langle \hat{\sigma} : n \otimes w \rangle_{\partial K} &= (f \cdot w)_K \\ -(Q \cdot \nabla \psi)_K + \langle (\hat{Q} \cdot n) \psi \rangle_{\partial K} &= 0 \end{aligned} \quad (10)$$

for all test functions $(w, \psi) \in V_Q \times V_p$ and all cells. The numerical fluxes $\hat{\sigma}$ and \hat{Q} are defined as

$$\begin{aligned} \hat{\sigma} &:= \{\{p\}\} \text{Id} + \alpha h_F^{-1} \llbracket Q \cdot n \rrbracket \text{Id} + \begin{cases} \{\{Q\}\} \otimes Q^\star & \text{for the central flux,} \\ Q^{\text{up}} \otimes Q^\star & \text{for the upwind flux} \end{cases} \\ \hat{Q} \cdot n &:= \{\{Q\}\} \cdot n. \end{aligned} \quad (11)$$

¹Note that in [6] the lowest order Nedelec element is $\text{ND}_1(K)$ whereas it is $\text{ND}_0(K)$ in [3]; here we use the counting employed in [3].

with the upwind velocity defined as

$$Q^{\text{up}} = \begin{cases} Q^+ & \text{if } Q \cdot n \geq 0 \\ Q^- & \text{otherwise} \end{cases} \quad (12)$$

This leads to the time-dependent scheme for computing $(Q^{n+1}, p^{n+1}) \in V_Q \times V_p$ from Q^n and f^n , the value of the forcing function at timestep n :

$$(Q^{n+1} \cdot w)_{\Omega_h} + \Delta t \left[(w \cdot (Q^* \cdot \nabla) Q^{n+1})_{\Omega_h} + \alpha \left(\sum_{F \in \mathcal{E}_h^i} h_F^{-1} \langle [Q^{n+1} \cdot n] [w \cdot n] \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} h_F^{-1} \langle (Q^{n+1} \cdot n) (w \cdot n) \rangle_F \right) \right] \quad (13a)$$

$$\begin{aligned} & - \sum_{F \in \mathcal{E}_h^i} \langle (Q^* \cdot n^+) (Q_+^{n+1} - Q_-^{n+1}) \cdot \{w\} \rangle_F \\ & + \delta_{\text{up}} \sum_{F \in \mathcal{E}_h^i} \langle |Q^* \cdot n^+| (Q_+^{n+1} - Q_-^{n+1}) \cdot (w^+ - w^-) \rangle_F \\ & - (p^{n+1} \nabla \cdot w)_{\Omega_h} + \sum_{F \in \mathcal{E}_h^i} \langle [w \cdot n] \{p^{n+1}\} \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle (w \cdot n) p^{n+1} \rangle_F \Big] = (Q^n \cdot w)_{\Omega_h} + \Delta t (f^n \cdot w)_{\Omega_h} \\ & (\psi \nabla \cdot Q^{n+1})_{\Omega_h} - \sum_{F \in \mathcal{E}_h^i} \langle [Q^{n+1} \cdot n] \{\psi\} \rangle_F - \sum_{F \in \mathcal{E}_h^\partial} \langle (Q^{n+1} \cdot n) \psi \rangle_F = 0 \end{aligned} \quad (13b)$$

for all test functions $(w, \psi) \in V_Q \times V_p$ where $\delta_{\text{up}} = 1$ for the upwind flux and $\delta_{\text{up}} = 0$ for the central flux. In contrast to [3] the boundary condition $(n \cdot Q^{n+1})_{\partial\Omega} = 0$ is enforced weakly through the penalty term in the second line of Eq. (13a), this introduces additional boundary integrals.

3.3 Hybridised scheme

To hybridise we replace the numerical fluxes in Eq. (11) by

$$\begin{aligned} \widehat{\sigma} &:= \lambda \text{Id} + \alpha h_F^{-1} [Q \cdot n] \text{Id} + \begin{cases} \{\{Q\}\} \otimes Q^* & \text{for the central flux,} \\ Q^{\text{up}} \otimes Q^* & \text{for the upwind flux} \end{cases} \\ \widehat{Q} \cdot n &:= Q \cdot n + \tau(p - \lambda). \end{aligned} \quad (14)$$

for some stability parameter $\tau > 0$ and the function $\lambda \in V_{\text{trace}}$ defined on the trace space

$$V_{\text{trace}} = \{v \in L^2(\mathcal{E}_h) : v|_F \in P_k(F) \text{ for all } F \in \mathcal{E}_h\}. \quad (15)$$

We also introduce a transmission condition which guarantees that the normal component of \widehat{Q} is single-valued:

$$0 = \langle [\widehat{Q} \cdot n] \mu \rangle_F = \langle ([Q \cdot n] + 2\tau\{p - \lambda\}) \mu \rangle_F \quad \text{for all } F \in \mathcal{E}_h^i \text{ and all } \mu \in V_{\text{trace}} \quad (16)$$

and enforce the boundary condition weakly by requiring that

$$0 = \langle (\widehat{Q} \cdot n) \mu \rangle_F = \langle (Q \cdot n + \tau(p - \lambda)) \mu \rangle_F \quad \text{for all } F \in \mathcal{E}_h^\partial \text{ and all } \mu \in V_{\text{trace}} \quad (17)$$

As a result, Eqs. (13a) and (13a) get replaced by: for given $Q^n, f^n \in V_Q$ find $(Q^{n+1}, p^{n+1}, \lambda^{n+1}) \in V_Q \times V_p \times V_{\text{trace}}$

$$\begin{aligned}
(Q^{n+1} \cdot w)_{\Omega_h} + \Delta t \Big[& (w \cdot (Q^* \cdot \nabla) Q^{n+1})_{\Omega_h} - \sum_{F \in \mathcal{E}_h^i} \langle (Q^* \cdot n^+) (Q_+^{n+1} - Q_-^{n+1}) \cdot \{w\} \rangle_F \\
& + \delta_{\text{up}} \sum_{F \in \mathcal{E}_h^i} \langle |Q^* \cdot n^+| (Q_+^{n+1} - Q_-^{n+1}) \cdot (w^+ - w^-) \rangle_F \\
& + \alpha \left(\sum_{F \in \mathcal{E}_h^i} h_F^{-1} \langle \llbracket Q^{n+1} \cdot n \rrbracket \llbracket w \cdot n \rrbracket \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} h_F^{-1} \langle (Q^{n+1} \cdot n) (w \cdot n) \rangle_F \right) \\
& - (p^{n+1} \nabla \cdot w)_{\Omega_h} + \sum_{F \in \mathcal{E}_h^i} \langle \llbracket w \cdot n \rrbracket \lambda^{n+1} \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle (w \cdot n) \lambda^{n+1} \rangle_F \Big] = (Q^n \cdot w)_{\Omega_h} + \Delta t (f^n \cdot w)_{\Omega_h}
\end{aligned} \tag{18a}$$

$$(\psi \nabla \cdot Q^{n+1})_{\Omega_h} + 2 \sum_{F \in \mathcal{E}_h^i} \langle \{ \tau (p^{n+1} - \lambda^{n+1}) \psi \} \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle \tau (p^{n+1} - \lambda^{n+1}) \psi \rangle_F = 0 \tag{18b}$$

$$\sum_{F \in \mathcal{E}_h^i} \langle (\llbracket Q^{n+1} \cdot n \rrbracket + 2\tau \{ \{ p^{n+1} - \lambda^{n+1} \} \}) \mu \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle (Q^{n+1} \cdot n + \tau (p^{n+1} - \lambda^{n+1})) \mu \rangle_F = 0 \tag{18c}$$

for all test functions $(w, \psi, \mu) \in V_Q \times V_p \times V_{\text{trace}}$. Observe that Eqs. (13a) and (18a) only differ in the final two terms on the left hand side.

3.4 Split hybridised scheme

Finally, we split the update $Q^n \mapsto (Q^{n+1}, p^{n+1}, \lambda^{n+1})$ in Eqs. (18a), (18b), (18c) into two steps:

- The computation of a tentative velocity \bar{Q} which is not necessarily divergence-free
- The solution of a mixed Poisson equation for the pressure p^{n+1} at the next time step and a velocity correction δQ which renders the velocity $Q^{n+1} = \bar{Q} - \Delta t \delta Q$ at the next timestep divergence free (in the weak sense)

Tentative velocity. To obtain \bar{Q} , replace $Q^{n+1} \mapsto \bar{Q}$ in Eq. (18a) and drop all terms which contain the pressure p^{n+1} and hybridised pressure λ^{n+1} (these will be dealt with in the next step):

$$\begin{aligned}
(\bar{Q} \cdot w)_{\Omega_h} + \Delta t \Big[& (w \cdot (Q^* \cdot \nabla) \bar{Q})_{\Omega_h} - \sum_{F \in \mathcal{E}_h^i} \langle (Q^* \cdot n^+) (\bar{Q}_+ - \bar{Q}_-) \cdot \{w\} \rangle_F \\
& + \delta_{\text{up}} \sum_{F \in \mathcal{E}_h^i} \langle |Q^* \cdot n^+| (\bar{Q}_+ - \bar{Q}_-) \cdot (w^+ - w^-) \rangle_F \\
& + \alpha \left(\sum_{F \in \mathcal{E}_h^i} h_F^{-1} \langle \llbracket \bar{Q} \cdot n \rrbracket \llbracket w \cdot n \rrbracket \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} h_F^{-1} \langle (\bar{Q} \cdot n) (w \cdot n) \rangle_F \right) \Big] = (Q^n \cdot w)_{\Omega_h} + \Delta t (f^n \cdot w)_{\Omega_h}
\end{aligned} \tag{19}$$

Mixed Poisson. In the continuum, the mixed Poisson equation that we need to solve can be written as

$$\delta Q + \nabla p^{n+1} = 0, \quad \nabla \cdot \delta Q = -\frac{1}{\Delta t} \nabla \cdot \bar{Q}. \tag{20}$$

In a single cell K the weak form is

$$\begin{aligned}
(w \cdot \delta Q)_K - (p^{n+1} \nabla \cdot w)_K + \langle (n \cdot w) \widehat{p}^{n+1} \rangle_{\partial K} &= 0, \\
(\psi \nabla \cdot \delta Q)_K + \langle \psi (\widehat{\delta Q} \cdot n - \delta Q \cdot n) \rangle_{\partial K} &= -\frac{1}{\Delta t} ((\psi \nabla \cdot \bar{Q})_K - \langle \psi (\bar{Q} - \mathcal{A}(\bar{Q})) \cdot n \rangle_{\partial K})
\end{aligned} \tag{21}$$

with the hybridised fluxes

$$\begin{aligned}\widehat{p}^{n+1} &= \lambda^{n+1} \\ \widehat{\delta Q} \cdot n &= \delta Q \cdot n + \tau(p^{n+1} - \lambda^{n+1})\end{aligned}\tag{22}$$

where τ is the stabilisation parameter as before. The velocity $\mathcal{A}(\overline{Q})$ is single valued on each facet and defined as

$$\mathcal{A}(Q) := \begin{cases} 0 & \text{on boundary facets } F \subset \partial\Omega \\ \{\{Q\}\} & \text{on interior facets } F \not\subset \partial\Omega. \end{cases} \tag{23}$$

We also require the numerical flux $\widehat{\delta Q}$ to be unique on interior facets and enforce the flux boundary condition $(Q \cdot n)_{\partial\Omega} = 0$ weakly:

$$\begin{aligned}\langle \llbracket \widehat{\delta Q} \cdot n \rrbracket \mu \rangle_F &= \langle (\llbracket \delta Q \cdot n \rrbracket + 2\tau\{p^{n+1} - \lambda^{n+1}\}) \mu \rangle_F = 0 & \text{for all } F \in \mathcal{E}_h^i \\ \langle (\widehat{\delta Q} \cdot n) \mu \rangle_F &= \langle (\delta Q \cdot n + \tau(p^{n+1} - \lambda^{n+1})) \mu \rangle_F = 0 & \text{for all } F \in \mathcal{E}_h^\partial.\end{aligned}\tag{24}$$

for all test functions $\mu \in V_{\text{trace}}$. Putting everything together we need to solve the following weak problem: find $(\delta Q, p^{n+1}, \lambda^{n+1}) \in V_Q \times V_p \times V_{\text{trace}}$ such that

$$(\delta Q \cdot w)_{\Omega_h} - (p^{n+1} \nabla \cdot w)_{\Omega_h} + \sum_{F \in \mathcal{E}_h^i} \langle \llbracket w \cdot n \rrbracket \lambda^{n+1} \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle (w \cdot n) \lambda^{n+1} \rangle_F = 0 \tag{25a}$$

$$(\psi \nabla \cdot \delta Q)_{\Omega_h} + 2 \sum_{F \in \mathcal{E}_h^i} \langle \{\tau(p^{n+1} - \lambda^{n+1})\psi\} \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle \tau(p^{n+1} - \lambda^{n+1})\psi \rangle_F = -\frac{1}{\Delta t} \text{Div}(\psi, \overline{Q}) \tag{25b}$$

$$\sum_{F \in \mathcal{E}_h^i} \langle (\llbracket \delta Q \cdot n \rrbracket + 2\tau\{p^{n+1} - \lambda^{n+1}\}) \mu \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle (\delta Q \cdot n + \tau(p^{n+1} - \lambda^{n+1})) \mu \rangle_F = 0 \tag{25c}$$

for all test functions $(w, \psi, \mu) \in V_Q \times V_p \times V_{\text{trace}}$ where the weak divergence is given by

$$\text{Div}(\psi, Q) := (\psi \nabla \cdot Q)_{\Omega_h} - 2 \sum_{F \in \mathcal{E}_h^i} \langle (\{\psi Q\} - \{\psi\}\{Q\}) \cdot n \rangle_F - \sum_{F \in \mathcal{E}_h^\partial} \langle \psi(Q \cdot n) \rangle_F. \tag{26}$$

Final update. Having computed \overline{Q} with Eq. (19) and $\delta Q, p^{n+1}$ with Eqs. (25a), (25b), (25c), the velocity at the next timestep is computed as

$$Q^{n+1} = \overline{Q} + \Delta t \delta Q. \tag{27}$$

4 IMEX timestepping

The above method uses a simple timestepping approach and we therefore expect it to be first order in time. This can be improved by using the higher-order IMEX [4] scheme described in [2, Section 4].

Let $(Q, p, \lambda) \in V_Q \times V_p \times V_{\text{trace}}$ be the time dependent state of the system. Consider the time-evolution equation in weak form

$$(w \cdot \partial_t Q)_{\Omega_h} = f^{(\text{im})}(w, Q, \mathcal{P}(Q)) + f^{(\text{ex})}(w; t) + g(w, p, \lambda) \tag{28}$$

subject to the weak incompressibility constraint

$$\Gamma(\psi, \mu, Q, p, \lambda) = 0 \tag{29}$$

where $w \in V_Q$, $\psi \in V_p$, $\mu \in V_{\text{trace}}$ are test functions. Looking at Eqs. (18a), (18b) and (18c) the weak forms are

$$f^{(\text{ex})}(w; t) := (w \cdot f(t))_{\Omega_h} \quad (30a)$$

$$\begin{aligned} f^{(\text{im})}(w, Q, Q^*) &:= -(w \cdot (Q^* \cdot \nabla)Q)_{\Omega_h} + \sum_{F \in \mathcal{E}_h^i} \langle (Q^* \cdot n^+)(Q_+ - Q_-) \cdot \{\{w\}\} \rangle_F \\ &\quad - \delta_{\text{up}} \sum_{F \in \mathcal{E}_h^i} \langle |Q^* \cdot n^+| (Q_+ - Q_-) \cdot (w^+ - w^-) \rangle_F \\ &\quad - \alpha \left(\sum_{F \in \mathcal{E}_h^i} h_F^{-1} \langle \llbracket Q \cdot n \rrbracket \llbracket w \cdot n \rrbracket \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} h_F^{-1} \langle (Q \cdot n)(w \cdot n) \rangle_F \right) \end{aligned} \quad (30b)$$

$$g(w, p, \lambda) := (p \nabla \cdot w)_{\Omega_h} - \sum_{F \in \mathcal{E}_h^i} \langle \llbracket w \cdot n \rrbracket \lambda \rangle_F - \sum_{F \in \mathcal{E}_h^\partial} \langle (w \cdot n) \lambda \rangle_F \quad (30c)$$

$$\begin{aligned} \Gamma(\psi, \mu, Q, p, \lambda) &:= (\psi \nabla \cdot Q)_{\Omega_h} + 2 \sum_{F \in \mathcal{E}_h^i} \langle \{\{\tau(p - \lambda)\psi\}\} \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle \tau(p - \lambda)\psi \rangle_F \\ &\quad + \sum_{F \in \mathcal{E}_h^i} \langle (\llbracket Q \cdot n \rrbracket + 2\tau\{\{p - \lambda\}\}) \mu \rangle_F + \sum_{F \in \mathcal{E}_h^\partial} \langle (Q \cdot n + \tau(p - \lambda)) \mu \rangle_F \end{aligned} \quad (30d)$$

Note that $g(w, p, \lambda)$ in Eqs. (28), (30c) is the (weak) pressure gradient which, together with the constraint in Eqs. (29), (30d) ensures that the velocity is divergence free ($\nabla \cdot Q = 0$) in the weak sense. Restricting ourselves to the special case $b_0^{(\text{im})} = a_{i,0}^{(\text{im})} = 0$, an s -stage IMEX-RK scheme is given by: find $(Q^{n+1}, p^{n+1}, \lambda^{n+1}) \in V_Q \times V_p \times V_{\text{trace}}$ such that

$$\begin{aligned} (w \cdot Q^{n+1})_{\Omega_h} &= (w \cdot Q^n)_{\Omega_h} + \Delta t \sum_{i=1}^{s-1} b_i^{(\text{im})} \left(f^{(\text{im})}(w, Q_i, \mathcal{P}(Q_{i-1})) + g(w, p_i, \lambda_i) \right) \\ &\quad + \Delta t \sum_{i=0}^{s-1} b_i^{(\text{ex})} f^{(\text{ex})}(w; t^n + c_i \Delta t) + \Delta t b_{s-1}^{(\text{ex})} g(w, \delta p, \delta \lambda) \end{aligned} \quad (31)$$

subject to the constraint

$$\Gamma(\psi, \mu, Q^{n+1}, \delta p, \delta \lambda) = 0 \quad (32)$$

for all test functions $(w, \psi, \mu) \in V_Q \times V_p \times V_{\text{trace}}$ and set

$$p^{n+1} = p_{s-1} + \delta p, \quad \lambda^{n+1} = \lambda_{s-1} + \delta \lambda. \quad (33)$$

Observe that the final pressure correction $\delta p, \delta \lambda$ appears on the right hand side of Eq. (31).

The coefficients that define a particular IMEX method are usually written down in the form of Butcher Tableaus, see Section 4.2 for details.

The stage variables $Q_1, Q_2, \dots, Q_{s-1} \in V_Q$, $p_1, p_2, \dots, p_{s-1} \in V_p$ and $\lambda_1, \lambda_2, \dots, \lambda_{s-1} \in V_{\text{trace}}$ are obtained by: for $i = 1, 2, \dots, s-1$ find $(Q_i, p_i, \lambda_i) \in V_Q \times V_p \times V_{\text{trace}}$ such that

$$(Q_i \cdot w)_{\Omega_h} - \Delta t a_{i,i}^{(\text{im})} (f^{(\text{im})}(w, Q_i, \mathcal{P}(Q_{i-1})) + g(w, p_i, \lambda_i)) = r_i(w) \quad \text{for all } w \in V_Q \quad (34)$$

with $Q_0 = Q^n$, $p_0 = p^n$ subject to the constraint

$$\Gamma(\psi, \mu, Q_i, p_i, \lambda_i) = 0 \quad \text{for all } \psi \in V_p, \mu \in V_{\text{trace}}. \quad (35)$$

At each stage the residual r_i is a one-form defined by

$$r_i(w) := (w \cdot Q^n)_{\Omega_h} + \Delta t \sum_{j=1}^{i-1} a_{i,j}^{(\text{im})} \left(f^{(\text{im})}(w, Q_j, \mathcal{P}(Q_{j-1})) + g(w, p_j, \lambda_j) \right) + \Delta t \sum_{j=0}^{i-1} a_{i,j}^{(\text{ex})} f^{(\text{ex})}(w; t^n + c_j \Delta t). \quad (36)$$

However, to avoid re-evaluating $f^{(\text{im})}$ and to ensure that corresponding terms cancel in the projection method, we use Eq. (34) to express $f^{(\text{im})}(w, Q_j, \mathcal{P}(Q_{j-1}))$ in terms of $(Q_j \cdot w)_{\Omega_h}$ and $r_j(w)$. With this, we arrive at the following recursive definition of the residual in Eq. (36)

$$r_i(w) := (w \cdot Q^n)_{\Omega_h} + \sum_{j=1}^{i-1} \frac{a_{i,j}^{(\text{im})}}{a_{j,j}^{(\text{im})}} ((Q_j \cdot w)_{\Omega_h} - r_j(w)) + \Delta t \sum_{j=0}^{i-1} a_{i,j}^{(\text{ex})} f^{(\text{ex})}(w; t^n + c_j \Delta t). \quad (37)$$

Eq. (34) together with the constraint in Eq. (35) is a linear equation for (Q_i, p_i, λ_i) . Instead of solving this linear equation in one go as in Section 3.3, we might proceed in two stages by using the projection method from Section 3.4 as a preconditioner for a Richardson iteration.

The resulting algorithm is shown in Alg. 1. Note that in the calculation of the final residual in Eq. (39) we again expressed $f^{(\text{im})}(w, Q_i, \mathcal{P}(Q_{i-1}))$ in terms of $(Q_i \cdot w)_{\Omega_h}$ and $r_i(w)$, compare the manipulations that transform Eq. (36) into Eq. (37). The pressures p' and λ' that are computed when solving (38) are only required to ensure that Q^{n+1} is divergence free.

Algorithm 1 IMEX-HDG timestepping with s stages (optionally based on the projection method): compute Q^{n+1}, p^{n+1} at the next timestep given Q^n, p^n at the current time $t^n = n\Delta t$.

- 1: Initialise $Q_0 := Q^n, p_0 := p^n$
- 2: **for** $i = 1, \dots, s-1$ **do**
- 3: Compute the residual $r_i(w)$ defined in Eq. (37)
- 4: Set $Q_{i-1}^* := \mathcal{P}(Q_{i-1})$
- 5: **if** use_projection_method **then**
- 6: Iteratively solve Eq. (34) subject to the constraint in Eq. (35) with Alg. 2.
- 7: **else**
- 8: Solve Eq. (34) subject to the constraint in Eq. (35) exactly in one step.
- 9: **end if**
- 10: **end for**
- 11: Compute $Q^{n+1}, \delta p, \delta \lambda$ by solving Eq. (31) subject to the constraint in Eq. (32). For this, solve

$$\begin{aligned} (Q^{n+1} \cdot w)_{\Omega_h} - \Delta t b_{s-1}^{(\text{im})} g(w, \delta p, \delta \lambda) &= r^{n+1}(w) \\ \Gamma(\phi, \mu, Q^{n+1}, \delta p, \delta \lambda) &= 0 \end{aligned} \quad (38)$$

with

$$r^{n+1}(w) := (w \cdot Q^n)_{\Omega_h} + \sum_{i=1}^{s-1} \frac{b_i^{(\text{im})}}{a_{i,i}^{(\text{im})}} ((Q_i \cdot w)_{\Omega_h} - r_i(w)) + \Delta t \sum_{i=0}^{s-1} b_i^{(\text{ex})} f^{(\text{ex})}(w; t^n + c_i \Delta t). \quad (39)$$

- 12: Reconstruct the (hybridised) pressure at the next timestep by solving Eq. (49) for $p = p^{n+1}, \lambda = \lambda^{n+1}$.
-

4.1 Pressure reconstruction

The pressure p can be reconstructed from the velocity Q at a given time by solving an elliptic problem. To see this, take the divergence of the first line of Eq. (1) in the domain Ω and multiply it by the outward normal n on the boundary $\partial\Omega$. Since $\nabla \cdot \partial_t Q = 0$ in Ω and $n \cdot \partial_t Q = 0$ on $\partial\Omega$, this results in the following boundary value problem

$$\begin{aligned} -\Delta p &= \nabla \cdot f_p & \text{in } \Omega \\ -n \cdot \nabla p &= n \cdot f_p & \text{on } \partial\Omega \end{aligned} \quad (44)$$

with the function

$$f_p := -f + (Q \cdot \nabla)Q. \quad (45)$$

Algorithm 2 Preconditioned Richardson iteration. Approximately solve Eq. (34) subject to the constraint in Eq. (35), using the projection method as a preconditioner with a fixed number of N iterations.

1: Initialise $Q_i^{(0)} = Q_{i-1}$, $p_i^{(0)} = p_{i-1}$, $\lambda_i^{(0)} := \lambda_{i-1}$.

2: **for** $k = 1, 2, \dots, N$ **do**

3: Set

$$\delta r_i^{(k)}(w) = r_i(w) - (Q_i^{(k-1)} \cdot n)_{\Omega_h} + a_{i,i}^{(\text{im})} \Delta t \left(f^{(\text{im})}(w, Q_i^{(k-1)}, Q_{i-1}^*) + g(w, p_i^{(k-1)}, \lambda_i^{(k-1)}) \right) \quad (40)$$

4: **Step I:** Compute the tentative velocity \bar{Q} by solving (cf. Eq. (19))

$$(\bar{Q} \cdot w)_{\Omega_h} - a_{i,i}^{(\text{im})} \Delta t f^{(\text{im})}(w, \bar{Q}, Q_{i-1}^*) = \delta r_i(w). \quad (41)$$

5: **Step II:** given \bar{Q} , compute increments δp , $\delta \lambda$ and δQ by solving (cf. Eqs. (25a) - (25c))

$$\begin{aligned} (\delta Q \cdot w)_{\Omega_h} - g(w, \delta p, \delta \lambda) &= -\frac{1}{a_{i,i}^{(\text{im})} \Delta t} \text{Div}(w, \bar{Q}) \\ \Gamma(\psi, \mu, \delta p, \delta \lambda) &= 0 \end{aligned} \quad (42)$$

6: **Step III:** Update

$$Q_i^{(k)} = Q_i^{(k-1)} + \bar{Q} + a_{i,i}^{(\text{im})} \Delta t \delta Q, \quad p_i^{(k)} = p_i^{(k-1)} + \delta p, \quad \lambda_i^{(k)} = \lambda_i^{(k-1)} + \delta \lambda. \quad (43)$$

7: **end for**

8: **return** $Q_i^{(N)}, p_i^{(N)}, \lambda_i^{(N)}$

To make the solution p of Eq. (44) unique, we require that $\int_{\Omega} p \, dx = 0$. The expression $n \cdot f_p|_{\partial\Omega}$ can be simplified as follows: Since $n \cdot Q|_{\partial\Omega} = 0$, we can write on the boundary

$$\begin{aligned} n \cdot f_p &= -n \cdot f + (Q \cdot \nabla)(n \cdot Q) \\ &= -n \cdot f + (Q^{\parallel} \cdot \nabla^{\parallel})(n \cdot Q) \\ &= -n \cdot f \end{aligned} \quad (46)$$

where Q^{\parallel} is the part of Q that is tangential to the boundary and similarly ∇^{\parallel} is the derivative in the tangential direction. The final line in Eq. (46) follows since on the boundary $n \cdot Q = 0$ obviously does not change in the tangential direction.

By introducing the new variable U , the second order problem in Eq. (44) can be written in mixed form as

$$U + \nabla p = 0, \quad \nabla \cdot U = \nabla \cdot f_p \quad (47)$$

subject to the boundary condition $n \cdot U|_{\partial\Omega} = -n \cdot f$. Eq. (47) can be solved through hybridisation in exactly the same way as the corresponding problem in Eq. (20). For this, we solve the following weak problem (c.f. Eqs. (25a) - (25c)): find $(U, p, \lambda) \in V_Q \times V_p \times V_{\text{trace}}$ such that

$$(U \cdot w)_{\Omega_h} - (p \nabla \cdot w)_{\Omega_h} + \sum_{F \in \mathcal{E}_h^i} \langle \llbracket w \cdot n \rrbracket \lambda \rangle_F + \sum_{F \in \mathcal{E}_h^{\partial}} \langle (w \cdot n) \lambda \rangle_F = 0 \quad (48a)$$

$$(\psi \nabla \cdot U)_{\Omega_h} + 2 \sum_{F \in \mathcal{E}_h^i} \langle \{\tau(p - \lambda)\psi\} \rangle_F + \sum_{F \in \mathcal{E}_h^{\partial}} \langle \tau(p - \lambda)\psi \rangle_F = \text{Div}(\psi, f_p) \quad (48b)$$

$$\sum_{F \in \mathcal{E}_h^i} \langle (\llbracket U \cdot n \rrbracket + 2\tau\{\{p - \lambda\}\}) \mu \rangle_F + \sum_{F \in \mathcal{E}_h^{\partial}} \langle (U \cdot n + \tau(p - \lambda)) \mu \rangle_F = - \sum_{F \in \mathcal{E}_h^{\partial}} \langle \mu n \cdot f \rangle_F \quad (48c)$$

0	0	...	0	0	...	0
c_1	$a_{1,0}^{(\text{ex})}$	0		\vdots		
c_2	$a_{2,0}^{(\text{ex})}$	$a_{2,1}^{(\text{ex})}$	0			
\vdots	\vdots		\ddots	\ddots		
c_{s-2}	$a_{s-2,0}^{(\text{ex})}$...			0	
c_{s-1}	$a_{s-1,0}^{(\text{ex})}$	$a_{s-1,1}^{(\text{ex})}$...	$a_{s-1,s-2}^{(\text{ex})}$	0	
	$b_0^{(\text{ex})}$	$b_1^{(\text{ex})}$...	$b_{s-2}^{(\text{ex})}$	$b_{s-1}^{(\text{ex})}$	

0	...	0	...	0
0	$a_{1,1}^{(\text{im})}$			\vdots
0	$a_{2,1}^{(\text{im})}$	$a_{2,2}^{(\text{im})}$		
\vdots		\ddots	\ddots	
0	...		$a_{s-2,s-2}^{(\text{im})}$	
0	$a_{s-1,1}^{(\text{im})}$...	$a_{s-1,s-2}^{(\text{im})}$	$a_{s-1,s-1}^{(\text{im})}$
0	$b_1^{(\text{im})}$...	$b_{s-2}^{(\text{im})}$	$b_{s-1}^{(\text{im})}$

Table 1: Generic form of Butcher Tableaus for IMEX methods considered here (recall that we assume that $a_{i,0}^{(\text{im})} = b_0^{(\text{im})} = 0$ and hence the first column of the implicit tableau on the right hand side contains only zeros).

0	0	0
1	1	0
	1	0

0	0
0	1
0	1

Table 2: Butcher Tableaus for the IMEX Euler method

for all test functions $(w, \psi, \mu) \in V_Q \times V_p \times V_{\text{trace}}$. The weak divergence operator $\text{Div}(\psi, \cdot)$ that appears on the right hand side of Eq. (48b) is defined in Eq. (26). The three equations in Eqs. (48a), (48b) and (48c) can be written in compact form as

$$\begin{aligned}
(U \cdot w)_{\Omega_h} - g(w, p, \lambda) &= \text{Div}(\psi, f_p) \\
\Gamma(\psi, \mu, U, p, \lambda) &= - \sum_{F \in \mathcal{E}_h^\partial} \langle \mu n \cdot f \rangle_F
\end{aligned} \tag{49}$$

which has the same structure as Eqs. (38) and (42).

4.2 Butcher Tableaus

The coefficients $a_{i,j}^{(\text{im})}$, $a_{i,j}^{(\text{ex})}$, $b_i^{(\text{im})}$, $b_i^{(\text{ex})}$ and c_j that define a particular IMEX method can be written in the form of Butcher-Tableaus, see Tab. 1. The Butcher-Tableaus for a range of IMEX methods can be found for example in [8]. For reference the Butcher tableaus for the timestepping method that are used for our numerical experiments are shown in Tabs. 2, 3 and 4.

5 Results

5.1 Experimental setup

To test the IMEX timestepping methods described in Section 4 we solve the incompressible Euler equations in Eq. (1) with initial conditions and forcing function chosen such that in the continuum $Q(x, y, t)$ and

0	0	0	0
1	1/2	0	0
1/2	1/2	1/2	0
	1/3	1/3	1/3

1/4	0	0
0	1/4	0
1/3	1/3	1/3
1/3	1/3	1/3

Table 3: Butcher Tableaus for the SSP2(3,3,2) method, see [8, Fig. 2]

0	0	0	0	0	α	0	0	0	$\alpha = 0.2416942608$ $\beta = 0.0604235652$ $\eta = 0.1291528696$ $\delta = 1/2 - \alpha - \beta - \eta$
0	0	0	0	0	$-\alpha$	α	0	0	
1	0	1	0	0	0	$1 - \alpha$	α	0	
$1/2$	0	$1/4$	$1/4$	0	β	η	δ	α	
	0	$1/6$	$1/6$	$2/3$	0	$1/6$	$1/6$	$2/3$	

Table 4: Butcher Tableaus for the SSP3(4,3,3) method, see [8, Fig. 2]

$p(x, y, t)$ are given by the exact solution written down in Appendix A. Setting $\Psi(t) = e^{-\kappa t}$, the initial condition Q_0 and forcing function f can be read off from Eq. (56) as

$$Q_0(x, y) = \begin{pmatrix} -\cos\left(\frac{2x-1}{2}\pi\right) \sin\left(\frac{2y-1}{2}\pi\right) \\ \sin\left(\frac{2x-1}{2}\pi\right) \cos\left(\frac{2y-1}{2}\pi\right) \end{pmatrix}, \quad f(x, y, t) = -\kappa e^{-\kappa t} Q_0(x, y). \quad (50)$$

We set $\kappa = 0.5$ in all numerical experiments and fix the final time to $T = 1$.

Timestepping methods. We pick timesteppers such that the order of the method is the same as the polynomial degree k used to discretise the pressure field; in this case we would expect that the temporal and spatial order of convergence are the same. The following three setups are considered here:

1. The two-stage implicit Euler method (see Tab. 2) with $k = 1$
2. The three-stage SSP2(3,3,2) method (see Tab. 3) with $k = 2$
3. The four-stage SSP3(4,3,3) method (see Tab. 4) with $k = 3$

Recall that according to Eq. (8) the components of the velocity field are represented by polynomials of degree $k + 1$. Numerical experiments are carried out on grids of size $n \times n$. Since the cells are right-angled triangles with two sides of length $h = 1/n$ and one side of length $\sqrt{2}h$, each grid contains $2n^2$ cells; we will refer to n as the “grid size” in the following. The number of timesteps is set to $n_t = n$ which results in a timestep size of $\Delta t = h$.

Solvers and preconditioners. To compute the tentative velocity (see Eq. (19)) we use a GMRES solver with ILU0 preconditioner. The L_2 projection $\mathcal{P} : Q \rightarrow Q^*$ of the velocity Q to the H-Div BDM_{k+1}^0 subspace is performed with a GMRES solver preconditioned with a simple Jacobi iteration. To solve the mixed problem in Eqs. (38), (42) and (49) we first use a Schur-complement reduction method to reduce the problem to an equation for the facet variables λ ; in Firedrake this is realised with the static condensation preconditioner class `firedrake.SPC`. The resulting system is then solved with a GMRES iteration, preconditioned with the non-nested multigrid method described in [5] (class `firedrake.GTMGPC`). On the finest level, a Chebyshev smoother based on the block-diagonal is used such that the matrix which couples the unknowns on each facet is inverted exactly; this is achieved with `firedrake.ASMStarPC`. For simplicity, the coarse level problem is solved with a standard AMG method with Chebyshev smoothers. Two pre- and post-smoothing iterations are used on all levels. In the non-nested multigrid algorithm the coarse level space consists of piecewise linear conforming elements on the original mesh, i.e. it is formulated in the space

$$V_c = P_1 := \{v \in C(\Omega) : v|_K \in P_1(K) \text{ for all } K \in \Omega_h\}. \quad (51)$$

The weak form on the coarse level is obtained by re-discretisation of the Laplace operator

$$a_c(\phi_c, \psi_c) = -(\nabla \psi_c \cdot \nabla \phi_c)_{\Omega_h} \quad \text{for all } \phi_c, \psi_c \in V_c \quad (52)$$

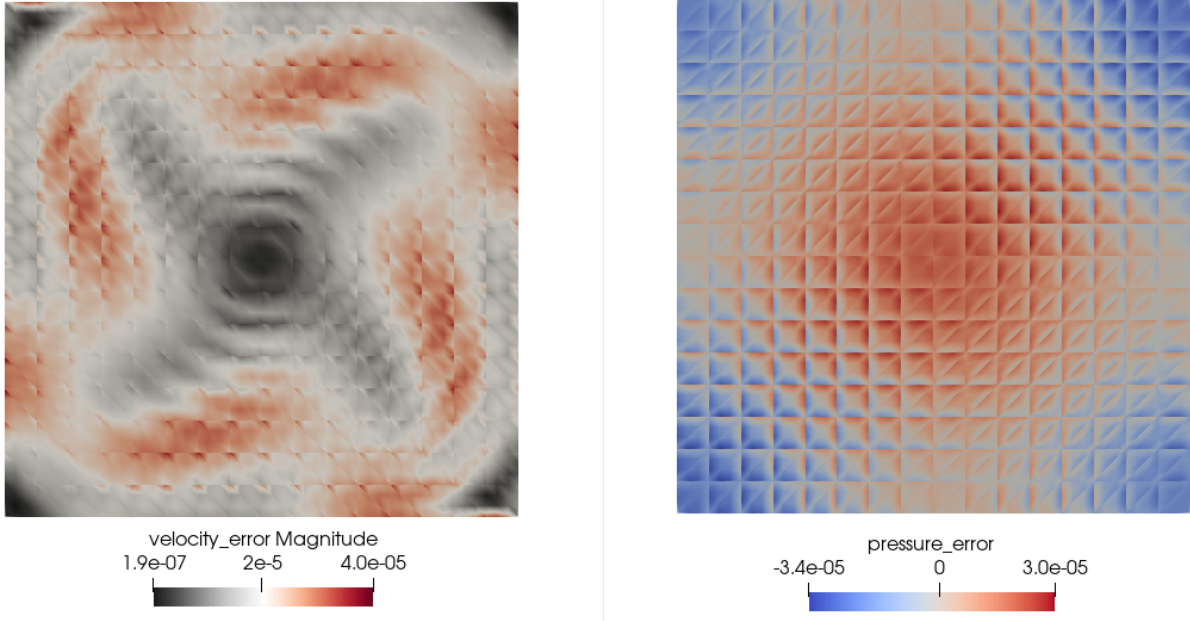


Figure 1: Spatial distribution of the error in velocity (left) and pressure (right) on a 16×16 grid. Results are obtained with a SSP2(3,3,2) timestepper and a polynomial degree of $k = 2$.

and the prolongation $\mathcal{P} : V_c \rightarrow V_{\text{trace}}$ is given by the injection on the skeleton \mathcal{E}_h

$$\mathcal{P} : \phi_c \mapsto \phi, \quad \langle \mu \phi_c \rangle_{\mathcal{E}_h} = \langle \mu \phi \rangle_{\mathcal{E}_h} \quad \text{for all } \mu \in V_{\text{trace}}. \quad (53)$$

As will be demonstrated below (see Fig. 4), this non-nested multigrid algorithm is h - and p - robust.

5.2 Convergence

We start by verifying that the numerical solution converges to the exact solution given in Eqs. (57). To illustrate that errors are uniformly distributed across the domain, Fig. 1 shows the spatial distribution of the error on a 16×16 grid with a SSP2(3,3,2) timestepper and $k = 2$. To quantify convergence with this grid spacing h we compute the L_2 error norm for increasing resolution. Fig. 2 shows how the error in the L_2 norm depends on the grid size for different polynomial degrees. For both velocity and pressure we observe convergence with the asymptotic rate $\propto h^k \propto (\Delta t)^k$.

5.3 Performance

Next, we investigate the performance of the different time integrators. For this, we first identify the computational bottlenecks.

Breakdown of time per iteration. Recall that an s -stage method with n_R Richardson iterations requires $s - 1$ projections of the velocity to the BDM_{k+1}^0 space, $n_R(s - 1)$ solves for the tentative velocity and $n_R(s - 1) + 2$ mixed pressure solves. We expect these operations to dominate the total cost and this is confirmed by Fig. 3 which shows a breakdown of the runtime; more detailed results can be found in Tab. 5.

implicit Euler, $k = 1$

grid	pressure solve				tentative velocity solve				BDM projection				timestep
4×4	4	\times	0.07	= 0.28	2	\times	0.01	= 0.03	1	\times	0.014	= 0.014	0.34
8×8	4	\times	0.06	= 0.25	2	\times	0.01	= 0.02	1	\times	0.008	= 0.008	0.30
16×16	4	\times	0.17	= 0.66	2	\times	0.02	= 0.04	1	\times	0.005	= 0.005	0.73
32×32	4	\times	1.01	= 4.04	2	\times	0.07	= 0.14	1	\times	0.004	= 0.004	4.21
64×64	4	\times	4.17	= 16.70	2	\times	0.33	= 0.65	1	\times	0.004	= 0.004	17.38

SSP2(3,3,2), $k = 2$

grid	pressure solve				tentative velocity solve				BDM projection				timestep
4×4	6	\times	0.06	= 0.38	4	\times	0.01	= 0.06	2	\times	0.008	= 0.017	0.49
8×8	6	\times	0.07	= 0.40	4	\times	0.02	= 0.07	2	\times	0.005	= 0.010	0.50
16×16	6	\times	0.18	= 1.08	4	\times	0.05	= 0.21	2	\times	0.004	= 0.007	1.34
32×32	6	\times	1.09	= 6.53	4	\times	0.25	= 0.99	2	\times	0.003	= 0.006	7.56
64×64	6	\times	4.55	= 27.32	4	\times	1.04	= 4.16	2	\times	0.004	= 0.009	31.54

SSP3(4,3,3), $k = 3$

grid	pressure solve				tentative velocity solve				BDM projection				timestep			
4×4	8	\times	0.06	=	0.46	6	\times	0.02	=	0.11	3	\times	0.007	=	0.020	0.62
8×8	8	\times	0.07	=	0.56	6	\times	0.04	=	0.21	3	\times	0.004	=	0.013	0.82
16×16	8	\times	0.21	=	1.66	6	\times	0.15	=	0.91	3	\times	0.003	=	0.010	2.62
32×32	8	\times	1.19	=	9.54	6	\times	0.67	=	4.02	3	\times	0.004	=	0.011	13.62
64×64	8	\times	4.92	=	39.35	6	\times	2.71	=	16.26	3	\times	0.007	=	0.022	55.71

Table 5: Breakdown of the time spent in the solver components for each timestep. For each component, the time per call and the number of calls per timestep is given. The total time per timestep is listed in the final column. Results are shown for the 2-stage implicit Euler method, the 3-stage SSP2(3,3,2) integrator and the 4-stage SSP3(4,3,3) timestepper.

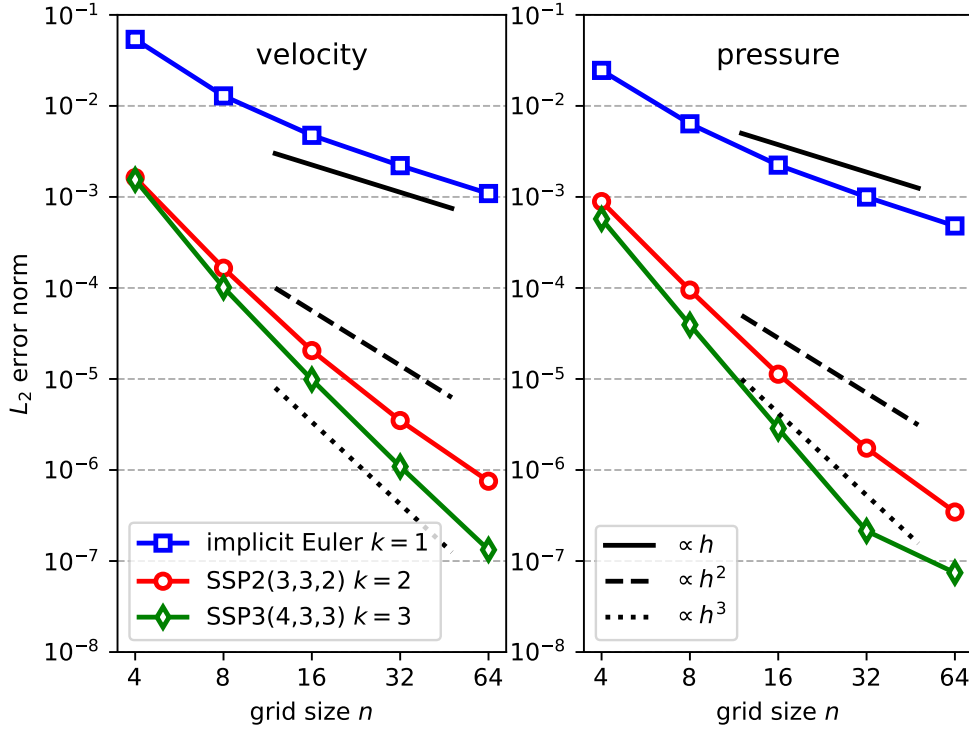


Figure 2: L_2 error norm in velocity (left) and pressure (right) for different grid sizes and polynomial degrees.

Robustness of the solvers. In the following we discuss the h - and p - robustness of the pressure- and tentative velocity solves. Since both solves are performed with iterative methods, it is crucial to quantify the number of solver iterations that are required to achieve convergence. We manually set at relative tolerance of 10^{-12} on the preconditioned residual for the pressure solve and use the PETSc default setting for the tentative velocity solver. Fig. 4 shows the average number of iterations² as a function of the grid size both for the tentative velocity solver and the pressure solver. The figure demonstrates that empirically both solvers are robust as the grid size n and polynomial degree k increases. Since (ideally) each solver iteration incurs a cost proportional to the number of unknowns, we therefore expect the cost of a single timestep to be proportional to number of unknowns for a given timestepper and polynomial degree k . This time per timestep is plotted as a function of the grid size for different polynomial degrees in Fig. 5. Naturally, this cost increases for higher polynomial degrees k . Asymptotically, for fixed k the time per timestep grows in proportion to the number of unknowns, which is proportional to $n^2 \propto h^{-2}$.

Total runtime. The most useful metric to quantify the performance of a given setup is the total runtime for a given error. We plot this time as a function of both the L_2 velocity error norm and the L_2 pressure error norm in Fig. 6. For a fixed error budget higher order integrator have a significantly smaller overall runtime as the bound on the error decreases.

²For the pressure solver, we only consider the final pressure solve in each timestep. However, we verified that the other solves require approximately the same number of iterations.

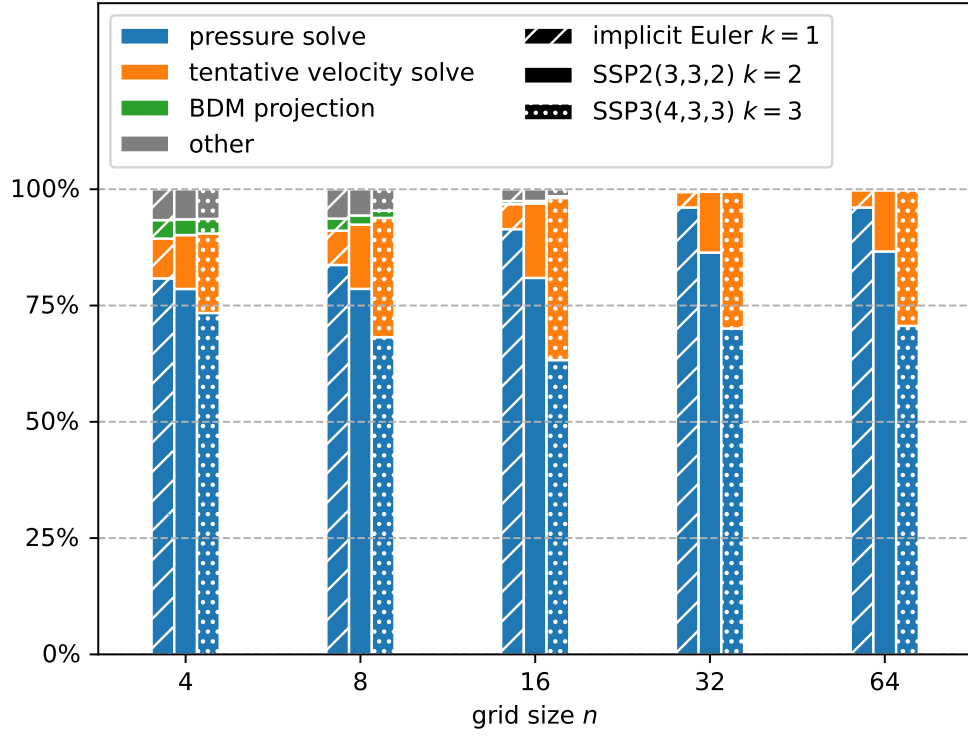


Figure 3: Fraction of runtime spent in the different solver components for varying grid size and different polynomial degrees.

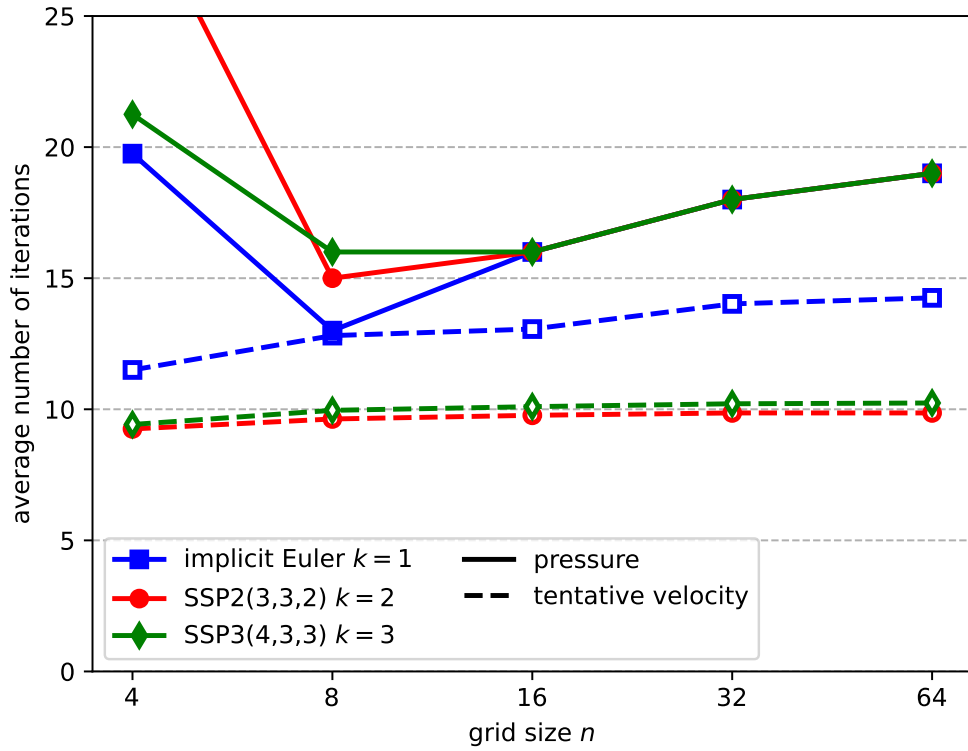


Figure 4: Number of solver as a function of the grid size for different polynomial degrees.

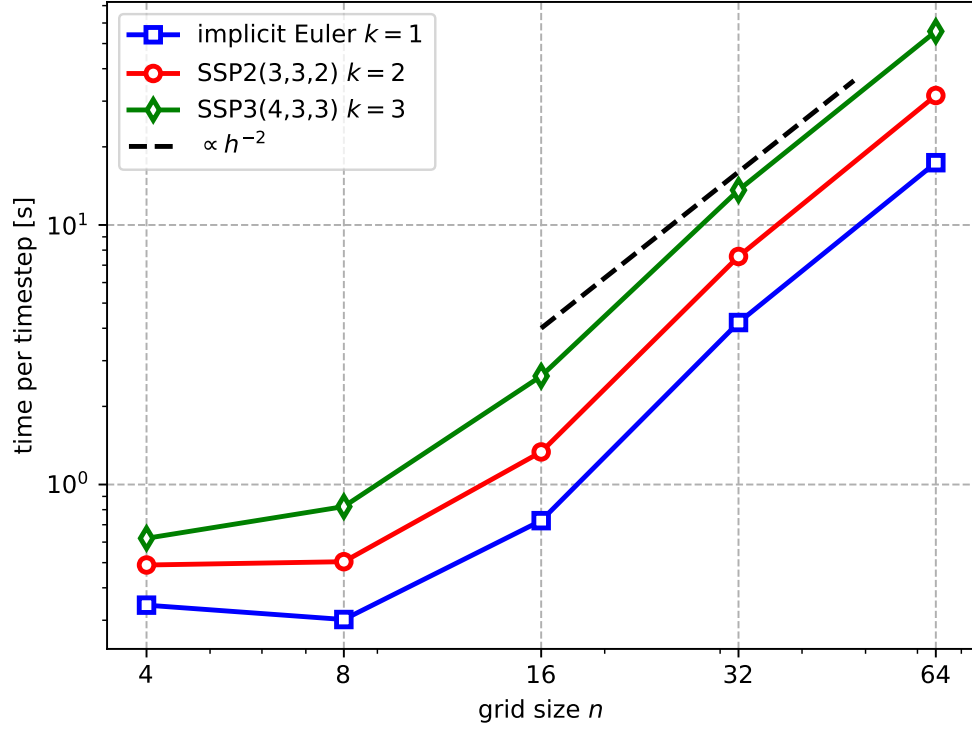


Figure 5: Time per timestep as a function of the grid size for different polynomial degrees.

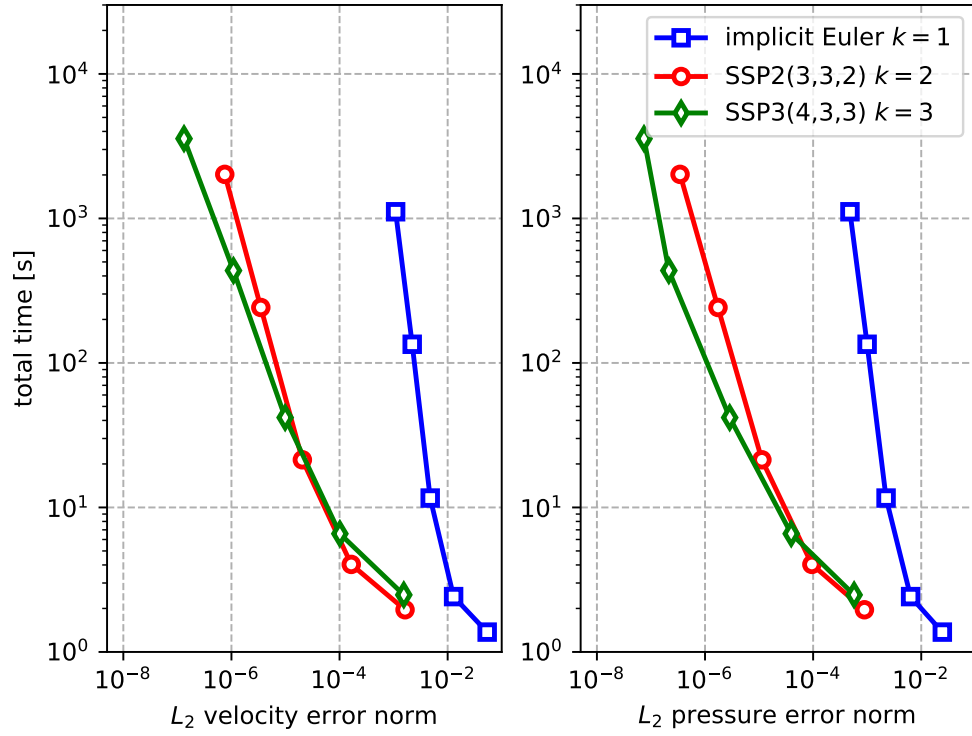


Figure 6: Total runtime for different values of the L_2 error in velocity (left) and pressure (right).

A Exact solution

An exact solution of the two-dimensional incompressible Euler equations in Eq. (1) is given by for the stationary case $f = 0$ by

$$\begin{aligned} Q_s(x, y) &= \begin{pmatrix} -C(x)S(y) \\ S(x)C(y) \end{pmatrix} \\ p_s(x, y) &= p_0 - C(x)C(y) \end{aligned} \quad (54)$$

where

$$S(z) = \sin\left(\frac{2z-1}{2}\pi\right), \quad C(z) = \cos\left(\frac{2z-1}{2}\pi\right). \quad (55)$$

Note that the derivatives of these functions satisfy $S'(z) = \pi C(z)$ and $C'(z) = -\pi S(z)$. From this a divergence-free time-dependent solution for the special case

$$f = \frac{d\Psi}{dt} Q_s(x, y). \quad (56)$$

with arbitrary scalar function $\Psi(t)$ can be constructed as

$$Q(x, y, t) = \Psi(t) Q_s(x, y), \quad p(x, y, t) = \Psi(t)^2 p_s(x, y). \quad (57)$$

B Double shear layer flow

The initial condition for the double shear layer flow in [9] (which is also used as a testcase in [3]) is given by

$$\begin{aligned} Q_0(x, y) &= \begin{pmatrix} Q_{0,x}(y) \\ Q_{0,y}(x) \end{pmatrix} \quad \text{with} \\ Q_{0,x}(y) &= \begin{cases} \tanh\left((y - \frac{1}{4})/\rho\right) & \text{for } 0 \leq y \leq \frac{1}{2}, \\ \tanh\left((\frac{3}{4} - y)/\rho\right) & \text{for } \frac{1}{2} < y \leq 1. \end{cases} \\ Q_{0,y}(x) &= \delta \sin(2\pi x) \end{aligned} \quad (58)$$

with the parameters $\rho = 1/30$ and $\delta = 0.05$. Periodic boundary conditions are assumed in both dimensions. The corresponding initial pressure $p_0(x, y)$ can be obtained by multiplying Eq. (1) by ∇ . This results in following elliptic problem for the initial pressure

$$-\Delta p_0 = \nabla \cdot (Q_0 \cdot \nabla) Q_0 \quad (59)$$

With the explicit expressions for the initial velocity in Eq. (58), Eq. (59) can be written as

$$-\Delta p_0(x, y) = 2 (\partial_x Q_{0,y}(x)) (\partial_y Q_{0,x}(y)). \quad (60)$$

We now make the ansatz

$$p_0(x, y) = \delta \cos(2\pi x) p_{0,y}(y) + P_0, \quad (61)$$

where the constant P_0 is chosen such that the average of p_0 is zero across the domain. Inserting Eq. (61) into Eq. (60) results in an second elliptic problem in one dimension for $p_{0,y}$:

$$4\pi^2 p_{0,y} - p_{0,y}'' = 4\pi^2 g(y) \quad \text{with } g(y) := \frac{1}{\pi} \partial_y Q_{0,x}(y). \quad (62)$$

Using the definition of $Q_{0,x}(y)$ in Eq. (58), the function $g(y)$ can be written down explicitly as

$$g(y) = \begin{cases} (\pi\rho)^{-1} (1 - \tanh^2((y - \frac{1}{4})/\rho)) & \text{for } 0 \leq y \leq \frac{1}{2}, \\ -(\pi\rho)^{-1} (1 - \tanh^2((\frac{3}{4} - y)/\rho)) & \text{for } \frac{1}{2} < y \leq 1. \end{cases} \quad (63)$$

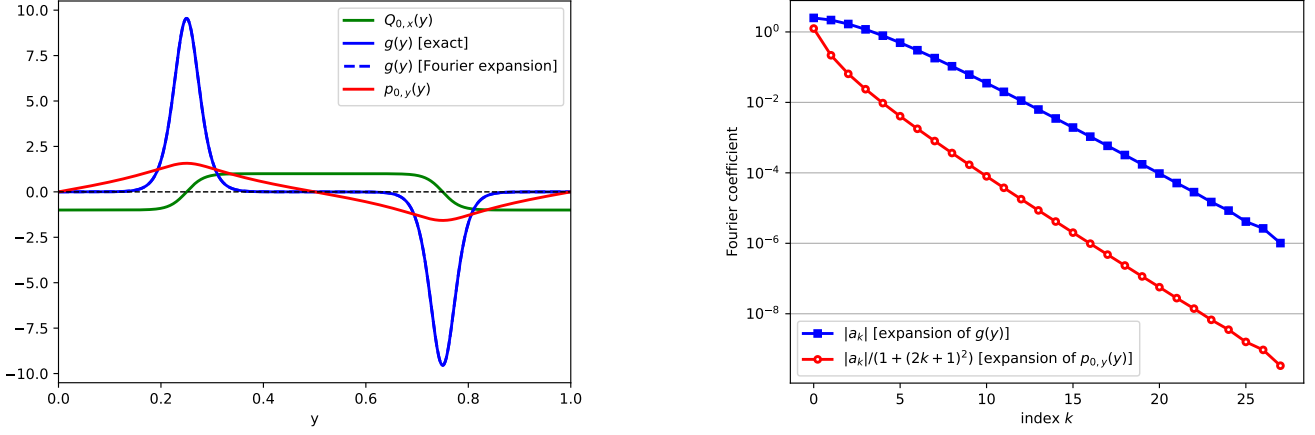


Figure 7: Initial condition for double layer shear flow in [9] (left) and Fourier coefficients of the expansions in Eqs. (66), (67) and (68) (right). The results are obtained by truncation the Fourier series after 28 terms.

Note that while $Q_{0,x}(y)$ is symmetric around the point $y = \frac{1}{2}$, $g(y)$ is antisymmetric around this point, i.e. $g(1 - y) = -g(y)$. To solve Eq. (62) we introduce the variable transformation $z = \pi(2y - 1)$ and two new functions θ , f with $\theta(z) = p_{0,y}(y)$, $f(z) = g(y)$. This results in a second order problem on the periodic interval $[-\pi, +\pi]$:

$$\theta - \theta'' = f(z) \quad (64)$$

with the odd function

$$f(z) = (\pi\rho)^{-1} \cdot \begin{cases} 1 - \tanh^2\left(\frac{\pi+2z}{4\pi\rho}\right) & \text{for } -\pi \leq z \leq 0 \\ -1 + \tanh^2\left(\frac{\pi-2z}{4\pi\rho}\right) & \text{for } 0 < z \leq \pi. \end{cases} \quad (65)$$

To solve Eq. (64) we Fourier-expand the right hand side $f(z)$ as

$$f(z) = \sum_{k=0}^{\infty} a_k \sin((2k+1)z), \quad a_k = \frac{1}{\pi} \int_{-\pi}^{+\pi} f(z) \sin((2k+1)z) dz \quad (66)$$

where only odd integers contribute to the sum since the function f has the additional symmetries $f(-\pi + z) = f(-z)$ for $-\pi \leq z \leq 0$ and $f(\pi - z) = f(z)$ for $0 < z \leq \pi$. This results in

$$\theta(z) = \sum_{k=0}^{\infty} a_k \frac{\sin((2k+1)z)}{1 + (2k+1)^2}. \quad (67)$$

Putting everything together, this gives the following expression for the initial pressure:

$$p_0(x, y) = \delta \cos(2\pi x) \sum_{k=0}^{\infty} a_k \frac{\sin(2\pi(2k+1)(y - \frac{1}{2}))}{1 + (2k+1)^2} \quad (68)$$

where setting $P_0 = 0$ guarantees that $\int_{\Omega} p_0(x, y) dx dy = 0$.

References

- [1] Alexandre Joel Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

- [2] Mattheus P Ueckermann and Pierre FJ Lermusiaux. Hybridizable discontinuous Galerkin projection methods for Navier–Stokes and Boussinesq equations. *Journal of Computational Physics*, 306:390–421, 2016.
- [3] Johnny Guzmán, Chi-Wang Shu, and Filánder A Sequeira. H(div) conforming and DG methods for incompressible Euler’s equations. *IMA Journal of Numerical Analysis*, 37(4):1733–1771, 11 2016.
- [4] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.
- [5] Bernardo Cockburn, Olivier Dubois, Jay Gopalakrishnan, and Shuguang Tan. Multigrid for an hdg method. *IMA Journal of Numerical Analysis*, 34(4):1386–1425, 2014.
- [6] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.
- [7] Philippe G Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.
- [8] Hilary Weller, Sarah-Jane Lock, and Nigel Wood. Runge–Kutta IMEX schemes for the horizontally explicit/vertically implicit (HEVI) solution of wave equations. *Journal of Computational Physics*, 252:365–381, 2013.
- [9] John B Bell, Phillip Colella, and Harland M Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of computational physics*, 85(2):257–283, 1989.