Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 387 (2021) 114178

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Enabling convergence of the iterated penalty Picard iteration with $O(1)$ penalty parameter for incompressible Navier–Stokes via Anderson acceleration

Leo G. Rebholz[a,*,1], Duygu Vargun[a,1], Mengying Xiao[b]

[a] *Department of Mathematical Sciences, Clemson University, Clemson, SC 29634, USA*
[b] *Department of Mathematics and Statistics, University of West Florida, Pensacola, FL 32514, USA*

Received 19 May 2021; received in revised form 31 August 2021; accepted 6 September 2021
Available online 29 September 2021

Abstract

This paper considers an enhancement of the classical iterated penalty Picard (IPP) method for the incompressible Navier–Stokes equations, where we restrict our attention to $O(1)$ penalty parameter, and Anderson acceleration (AA) is used to significantly improve its convergence properties. After showing the fixed point operator associated with the IPP iteration is Lipschitz continuous and Lipschitz continuously (Frechet) differentiable, we apply a recently developed general theory for AA to conclude that IPP enhanced with AA improves its linear convergence rate by the gain factor associated with the underlying AA optimization problem. Results for several challenging numerical tests are given and show that IPP with penalty parameter 1 and enhanced with AA is a very effective solver.

Keywords: Navier–Stokes equations; Anderson acceleration; Iterated penalty method; Finite element method

1. Introduction

We consider solvers for the incompressible Navier–Stokes equations (NSE), which are given by

$$u_t + u \cdot \nabla u + \nabla p - \nu \Delta u = f, \tag{1}$$

$$\nabla \cdot u = 0, \tag{2}$$

where u and p are the unknown velocity and pressure, ν is the kinematic viscosity which is inversely proportional to the Reynolds number Re, and f is a known function representing external forcing. For simplicity we assume no-slip boundary conditions and a steady flow ($u_t = 0$) as well as small data so as to be consistent with steady flow, but our analysis and results can be extended to other common boundary conditions and temporarily discretized transient flows with only minor modifications. Due to the wide applicability of (1)–(2) across science and engineering, many nonlinear solvers already exist for it [1], with the most popular being Picard and Newton iterations [2]. Newton's

[*] Corresponding author.
E-mail addresses: rebholz@clemson.edu (L.G. Rebholz), dvargun@clemson.edu (D. Vargun), mxiao@uwf.edu (M. Xiao).

[1] This author was partially supported by NSF Grant DMS 2011490.

https://doi.org/10.1016/j.cma.2021.114178
0045-7825/© 2021 Elsevier B.V. All rights reserved.

iteration converges quadratically once near a root, but requires a good initial guess, especially for higher Re [2]. The Picard iteration for the NSE is linearly convergent, but also globally convergent and is much more robust for higher Re [2,3].

Herein, we consider Anderson acceleration (AA) of the iterated penalty Picard (IPP) iteration. The IPP iteration is generally more efficient than Picard for a single iteration since the linear solve is easier/cheaper, but compared to Picard it can be less robust and require more iterations if the penalty parameter is not chosen correctly. The IPP scheme for the NSE is given in [4] as: Given u_k, p_k, solve for u_{k+1}, p_{k+1} from

$$u_k \cdot \nabla u_{k+1} + \nabla p_{k+1} - \nu \Delta u_{k+1} = f, \tag{3}$$

$$\epsilon p_{k+1} + \nabla \cdot u_{k+1} = \epsilon p_k, \tag{4}$$

where $\epsilon > 0$ is a penalty parameter, generally taken small. The system (3)–(4) is equivalent to the velocity-only system

$$u_k \cdot \nabla u_{k+1} - \epsilon^{-1} \nabla(\nabla \cdot u_{k+1}) - \nu \Delta u_{k+1} = f + \epsilon^{-1} \sum_{j=0}^{k} \nabla(\nabla \cdot u_j), \tag{5}$$

which is used in [5–8], and the pressure can be expressed in terms of velocities, i.e. $p_{k+1} = -\epsilon^{-1} \sum_{j=0}^{k+1} \nabla \cdot u_j$. There are several advantages to using IPP, including allowing for circumventing the inf–sup condition on the velocity and pressure spaces [4], Scott–Vogelius elements can be (implicitly) used without any mesh restriction and will produce a pointwise divergence free solution (along with many advantages this brings, see e.g. [9]), and linear system solves will be easier to perform. Moreover, Codina showed in [4] that a discretization of (3)–(4) converges linearly under a small data condition and sufficiently small ϵ, and has a better convergence rate if the penalty parameter ϵ is chosen sufficiently small.

Unfortunately, with small ϵ the advantages of using (3)–(4) diminish since the same nonsymmetric saddle point system of the usual Picard iteration is recovered as $\epsilon \to 0$, and thus $\epsilon \ll 1$ can lead to linear systems that most common preconditioned iterative linear solvers will have difficulty resolving [10,11]. Hence even though the IPP is theoretically effective when ϵ is small, its use has largely died out over the past few decades since small ϵ leads to the need for direct linear solvers, but direct linear solvers are not effective on most large scale problems of modern interest. Hence, in an effort to show IPP (properly enhanced with AA) can still be a very competitive solver on any size problem, we completely avoid the notion of small ϵ and in our numerical tests use only $\epsilon = 1$, where preconditioned iterative methods found success on linear systems resembling (5) [11–14].

This paper presents an analytical and numerical study of AA applied to IPP, without assuming small ϵ. AA has recently been used to improve convergence and robustness of solvers for a wide range of problems including various types of flow problems [15–17], geometry optimization [18], radiation diffusion and nuclear physics [19,20], machine learning [21], molecular interaction [22], computing nearest correlation matrices [23], and many others e.g. [15,24–28]. In particular, AA was used in [16] to make the Picard iteration for (1)–(2) more robust with respect to Re and to converge significantly faster. Hence it is a natural and important next step to consider AA applied to IPP, which is a classical NSE solver but is not always effective when $\epsilon < 1$ due to linear solver difficulties. Herein we formulate IPP equipped with a finite element discretization as a fixed point iteration $(u_{k+1}, p_{k+1}) = G(u_k, p_k)$, where G is a solution operator to discrete linear system. We then prove that G is continuously (Frechet) differentiable, allowing us to invoke the AA theory from [3], which implies AA will improve the linear convergence rate of the iteration by a factor (less than 1) representing the gain of the underlying AA optimization problem. Results of several numerical tests are also presented, which shows IPP using $\epsilon = 1$ and enhanced with AA can be a very effective solver for the NSE.

This paper is arranged as follows: In Section 2, we provide notation and mathematical preliminaries on the finite element discretizations and AA. In Section 3, we present the IPP method and prove associated fixed point solution operator properties. In Section 4, we give the Anderson accelerated IPP scheme and present a convergence result. In Section 5, we report on the results of several numerical tests, which demonstrate a significant (and sometimes dramatic) positive impact on the convergence.

2. Notation and preliminaries

We consider a domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ that is open, connected, and with Lipschitz boundary $\partial\Omega$. The $L^2(\Omega)$ norm and inner product will be denoted by $\|\cdot\|$ and $(\cdot, \cdot)$. Throughout this paper, it is understood by context whether a particular space is scalar or vector valued, and so we do not distinguish notation.

The natural function spaces for velocity and pressure in this setting are given by

$$\begin{aligned} X &:= H_0^1(\Omega) = \{v \in L^2(\Omega) \mid \nabla v \in L^2(\Omega), v|_{\partial\Omega} = 0\}, \\ Q &:= L_0^2(\Omega) = \{q \in L^2(\Omega) \mid \int_\Omega q\, dx = 0\}. \end{aligned}$$

In the space X, the Poincaré inequality holds [29]: there exists a constant $C_P > 0$ depending only on Ω such that for any $\phi \in X$,

$$\|\phi\| \leq C_P \|\nabla\phi\|.$$

The dual space of X will be denoted by X', with norm $\|\cdot\|_{-1}$. We define the skew-symmetric trilinear operator $b^* : X \times X \times X \to \mathbb{R}$ by

$$b^*(u, v, w) := \frac{1}{2}(u \cdot \nabla v, w) - \frac{1}{2}(u \cdot \nabla w, v),$$

which satisfies

$$b^*(u, v, w) \leq M \|\nabla u\| \|\nabla v\| \|\nabla w\|, \tag{6}$$

for any $u, v, w \in X$, where M is a constant depending on $|\Omega|$ only, see [29].

In our analysis, the following natural norm on $X^* := (X, Q)$ arises

$$\|(v, q)\|_{X^*} := \sqrt{\nu\|\nabla v\|^2 + \epsilon\|q\|^2}. \tag{7}$$

The FEM formulation of the steady NSE is given as follows: Find $(u, p) \in (X_h, Q_h)$ such that

$$\begin{aligned} \nu(\nabla u, \nabla v) + b^*(u, u, v) - (p, \nabla \cdot v) &= (f, v), \\ (q, \nabla \cdot u) &= 0, \end{aligned} \tag{8}$$

for all $(v, q) \in (X_h, Q_h)$. It is known that system (8) has solutions for any data, and those solutions are unique if the small data condition $\kappa := \nu^{-2} M \|f\|_{-1} < 1$ is satisfied. Moreover, all solutions to (8) are bounded by $\|\nabla u\| \leq \nu^{-1}\|f\|_{-1}$.

Assumption 2.1. We will assume in our analysis that $\kappa < 1$ so that (8) is well-posed, and that $\kappa + \varepsilon^{1/2}\nu^{-3/2} M\|p\| < 1$.

2.1. Discretization preliminaries

We denote with τ_h a conforming, shape-regular, and simplicial triangulation of Ω with h denoting the maximum element diameter of τ_h. We represent the space of degree k globally continuous piecewise polynomials on τ_h by $P_k(\tau_h)$, and $P_k^{disc}(\tau_h)$ the space of degree k piecewise polynomials that can be discontinuous across elements.

We choose the discrete velocity space by $X_h = X \cap P_k(\tau_h)$ and the pressure space $Q_h = \nabla \cdot X_h \subseteq Q$. With this choice of spaces, the discrete versions of (3)–(5) are equivalent, although in our computations we use only (5) and so the pressure space is never explicitly used. As discussed in [7], pressure recovery via the L^2 projection of $-\epsilon^{-1}\sum_{j=0}^{\infty} \nabla \cdot u_j$ into $Q \cap P_{k-1}(\tau_h)$ will yield a continuous and optimally accurate pressure. Under certain mesh structures, the (X_h, Q_h) pair will satisfy the discrete inf–sup condition [21,30–33]. While inf–sup is important for small ϵ in the IPP, our focus is on $\epsilon = 1$ and so this compatibility condition is not necessary for our analysis to hold.

2.2. Anderson acceleration

Anderson acceleration is an extrapolation method used to improve convergence of fixed-point iterations. Following [16,24,34], it may be stated as follows, where Y is a normed vector space and $g : Y \to Y$.

Algorithm 2.2 (*Anderson Iteration*)**.** Anderson acceleration with depth m and damping factors β_k.
Step 0: Choose $x_0 \in Y$.
Step 1: Find $w_1 \in Y$ such that $w_1 = g(x_0) - x_0$. Set $x_1 = x_0 + w_1$.
Step k: For $k = 2, 3, \dots$ Set $m_k = \min\{k-1, m\}$.
[a.] Find $w_k = g(x_{k-1}) - x_{k-1}$.
[b.] Solve the minimization problem for the Anderson coefficients $\{\alpha_j^k\}_{j=1}^{m_k}$

$$\min \left\| \left(1 - \sum_{j=1}^{m_k} \alpha_j^k\right) w_k + \sum_{j=1}^{m_k} \alpha_j^k w_{k-j} \right\|_Y . \tag{9}$$

[c.] For damping factor $0 < \beta_k \le 1$, set

$$x_k = (1 - \sum_{j=1}^{m_k} \alpha_j^k) x_{k-1} + \sum_{j=1}^{m_k} \alpha_j^k x_{j-1} + \beta_k \left((1 - \sum_{j=1}^{m_k} \alpha_j^k) w_k + \sum_{j=1}^{m_k} \alpha_j^k w_{k-j} \right), \tag{10}$$

where $w_j = g(x_{j-1}) - x_{j-1}$ is the nonlinear residual (and also sometimes referred to as the update step).

Note that depth $m = 0$ returns the original fixed-point iteration. We define the optimization gain factor θ_k by

$$\theta_k = \frac{\left\| \left(1 - \sum_{j=1}^{m_k} \alpha_j^k\right) w_k + \sum_{j=1}^{m_k} \alpha_j^k w_{k-j} \right\|_Y}{\|w_k\|_Y}, \tag{11}$$

representing the ratio gain of the minimization problem (9) using m_k compared to the $m = 0$ (usual fixed point iteration) case. The gain factor θ_k plays a critical role in the general AA convergence theory [3,35] that reveals how AA improves convergence: specifically, the acceleration reduces the first-order residual term by a factor of θ_k, but introduces higher-order terms into the residual expansion.

The next two assumptions give sufficient conditions on the fixed point operator g for the theory of [3] to be applied.

Assumption 2.3. Assume $g \in C^1(Y)$ has a fixed point x^* in Y, and there are positive constants C_0 and C_1 with

1. $\|g'(x)\|_Y \le C_0$ for all $x \in Y$, and
2. $\|g'(x) - g'(y)\|_Y \le C_1 \|x - y\|_Y$ for all $x, y \in Y$.

Assumption 2.4. Assume there is a constant $\sigma > 0$ for which the differences between consecutive residuals and iterates satisfy

$$\|w_{k+1} - w_k\|_Y \ge \sigma \|x_k - x_{k-1}\|_Y, \quad k \ge 1. \tag{12}$$

Assumption 2.4 is satisfied, for example, if g is contractive (i.e. if $C_0 < 1$ in Assumption 2.3). Other ways that the assumption is satisfied are discussed in [3]. Under Assumptions 2.3 and 2.4, the following result from [3] produces a bound on the residual $\|w_{k+1}\|$ in terms of the previous residuals.

Theorem 2.5 (*Pollock et al. [17]*)**.** *Let Assumptions 2.3 and 2.4 hold, and suppose the direction sines between each column j of matrix*

$$F_j = \begin{pmatrix} (w_j - w_{j-1}) & (w_{j-1} - w_{j-2}) & \cdots & (w_{j-m_j+1} - w_{j-m_j}) \end{pmatrix} = (f_{j,i}) \tag{13}$$

and the subspace spanned by the preceding columns satisfies $|\sin(f_{j,i}, span\{f_{j,1}, \dots, f_{j,i-1}\})| \ge c_s > 0$, for $j = 1, \dots, m_k$. Then the residual $w_{k+1} = g(x_k) - x_k$ from Algorithm 2.2 (depth m) satisfies the bound

$$\|w_{k+1}\|_Y \le \|w_k\|_Y \Bigg(\theta_k((1-\beta_k) + C_0\beta_k) + \frac{CC_1\sqrt{1-\theta_k^2}}{2} \Bigg(\|w_k\|_Y h(\theta_k)$$

$$+2\sum_{n=k-m_k+1}^{k-1}(k-n)\,\|w_n\|_Y\,h(\theta_n)+m_k\,\left\|w_{k-m_k}\right\|_Y h(\theta_{k-m_k})\Bigg)\Bigg), \tag{14}$$

where each $h(\theta_j)\le C\sqrt{1-\theta_j^2}+\beta_j\theta_j$, and C depends on c_s and the implied upper bound on the direction cosines.

The estimate (14) shows how the relative contributions from the lower and higher order terms are determined by the gain factor θ_k: the lower order terms are scaled by θ_k and the higher-order terms by $\sqrt{1-\theta_k^2}$. The estimate reveals that while larger choices of m generally provide lower θ_k's which reduces the lower order contributions to the residual, it also incurs a cost of both increased accumulation and weight of higher order terms. If recent residuals are small then greater algorithmic depths m may be advantageous, but if not, large m may slow or prevent convergence. As discussed in [3], this suggests that depth selection strategies that use small m early in the iteration and large m later may be advantageous in some settings.

This result supposes the sufficient linear independence of the columns of each matrix F_j given by (13). As discussed in [3], this assumption can be both verified and ensured, so long as the optimization problem is solved in a norm induced by an inner-product. One can safeguard by sufficiently reducing m or by removing columns of F_j where the desired inequality fails to hold, as demonstrated in [3].

3. The iterated penalty picard method and associated solution operator properties

This section presents some properties of the IPP iteration and its associated fixed point function.

3.1. Iterated penalty picard method

This subsection studies some properties of IPP method. We begin by defining its associated fixed point operator.

Definition 3.1. We define a mapping $G:(X_h,Q_h)\to(X_h,Q_h),\ G(u,p)=(G_1(u,p),G_2(u,p))$ such that for any $(v,q)\in(X_h,Q_h)$

$$\begin{aligned}\nu(\nabla G_1(u,p),\nabla v)+b^*(u,G_1(u,p),v)-(G_2(u,p),\nabla\cdot v)&=(f,v),\\ \varepsilon(G_2(u,p),q)+(\nabla\cdot G_1(u,p),q)&=\varepsilon(p,q).\end{aligned} \tag{15}$$

Thus the IPP method for solving steady NSE can be rewritten now as follows.

Algorithm 3.2. The iterated penalty method for solving steady NSE is

Step 0 Guess $(u_0,p_0)\in(X_h,Q_h)$.
Step k Find $(u_{k+1},p_{k+1})=G(u_k,p_k)$.

We now show that G is well-defined, and will then prove smoothness properties for it.

Lemma 3.3. *The operator G is well defined. Moreover,*

$$\|\nabla G_1(u,p)\|\le\nu^{-1}\|f\|_{-1}+\sqrt{\frac{\epsilon}{\nu}}\|p\|, \tag{16}$$

for any $(u,p)\in(X_h,Q_h)$.

Proof. Given f,u,p, assume $(u_1,p_1),(u_2,p_2)\in(X_h,Q_h)$ are solutions to (15). Subtracting these two systems and letting $e_u=u_1-u_2$ and $e_p=p_1-p_2$ produces

$$\begin{aligned}\nu(\nabla e_u,\nabla v)+b^*(u,e_u,v)-(e_p,\nabla\cdot v)&=0,\\ \varepsilon(e_p,q)+(\nabla\cdot e_u,q)&=0.\end{aligned}$$

Setting $v=e_u$ and $q=e_p$, and adding these equations gives

$$\nu\|\nabla e_u\|^2+\varepsilon\|e_p\|^2=0,$$

which is satisfied if $e_u = e_p = 0$ implying the solution of (15) is unique. Because (15) is linear and finite dimensional, solutions must exist uniquely. Choosing $v = G_1(u, p)$ and $q = G_2(u, p)$ in (15) produces

$$\|G(u, p)\|^2_{X^*} = \nu\|\nabla G_1(u, p)\|^2 + \epsilon\|G_2(u, p)\|^2 \leq \epsilon\|p\|^2 + \nu^{-1}\|f\|^2_{-1},$$

thanks to Cauchy–Schwarz and Young's inequalities. This shows the solution $G(u, p)$ is bounded continuously by the data, proving (15) is well-posed and thus G is well-defined. Additionally, dropping the term $\|G_2(u, p)\|^2$ and taking square root yields (16). □

Lemma 3.4. *Under Assumption 2.1, let (u, p) be the solution of (8) and (u_k, p_k) be kth iteration from Algorithm 3.2. Then we have for any $\varepsilon > 0$,*

$$\lim_{k\to\infty} \|u_k - u\| = \lim_{k\to\infty} \|p_k - p\| = 0.$$

Proof. Subtracting Eqs. (8) from (15) with (u_{k+1}, p_{k+1}) gives

$$\nu(\nabla(u_{k+1} - u), \nabla v) + b^*(u_k, u_{k+1} - u, v) + b^*(u_k - u, u, v) - (p_{k+1} - p, \nabla \cdot v) = 0, \tag{17}$$

$$\epsilon(p_{k+1} - p, q) + (\nabla \cdot (u_{k+1} - u), q) = \epsilon(p_k - p, q). \tag{18}$$

Adding these equations together and setting $v = u_{k+1} - u, q = p_{k+1} - p$ produces

$$\nu\|\nabla(u_{k+1} - u)\|^2 + \epsilon\|p_{k+1} - p\|^2 \leq M\|\nabla(u_k - u)\|\|\nabla u\|\|\nabla(u_{k+1} - u)\| + \epsilon\|p_{k+1} - p\|\|p_k - p\|,$$

thanks to (6) and Cauchy–Schwarz inequality. Then, using $\|\nabla u\| \leq \nu^{-1}\|f\|_{-1}$, and Young's inequality gives

$$\nu\|\nabla(u_{k+1} - u)\|^2 + \epsilon\|p_{k+1} - p\|^2 \leq \nu\kappa^2\|\nabla(u_k - u)\|^2 + \epsilon\|p_k - p\|^2, \tag{19}$$

where $\kappa := M\nu^{-2}\|f\|_{-1}$. Denoting $U_k = \|\nabla(u - u_k)\|^2$, $P_k = \|p - p_k\|^2$, (19) produces

$$\begin{aligned}
\varepsilon P_{k+1} + \nu U_{k+1} &\leq \varepsilon P_k + \kappa^2 \nu U_k \\
&= \varepsilon P_k + \nu U_k - (1 - \kappa^2)\nu U_k \\
&\leq \varepsilon P_{k-1} + \nu U_{k-1} - (1 - \kappa^2)\nu\,(U_k + U_{k-1}) \\
&\leq \cdots \\
&\leq \varepsilon P_0 + \nu U_0 - (1 - \kappa^2)\nu \sum_{j=0}^{k} U_j,
\end{aligned}$$

and thus

$$\varepsilon P_{k+1} + \nu U_{k+1} + (1 - \kappa^2)\nu \sum_{j=0}^{k} U_j \leq \varepsilon P_0 + \nu U_0.$$

Thanks to Assumption 2.1 and dropping P_{k+1} on the left hand side gives

$$\sum_{j=0}^{k+1} U_j \leq \frac{\varepsilon P_0 + \nu U_0}{(1 - \kappa^2)\nu}.$$

Hence, as $k \to \infty$, $\sum_{j=0}^{\infty} U_j$ converges, which implies $\lim_{k\to\infty} U_k = \lim_{k\to\infty} \|\nabla(u_k - u)\| = 0$. Note that, the system (3)–(4) is equivalent to the velocity-only system (5). Thus the pressure can be represented by velocities as $p_k = -\varepsilon^{-1} \sum_{j=0}^{k} \nabla \cdot u_j$, which gives convergence of the sequence $\{p_k\}$ since the velocity converges (in H^1, using norm equivalence due to finite dimensionality of the systems). Thus by (19) and $\lim_{k\to\infty} \|\nabla(u_k - u)\| = 0$, we get $\lim_{k\to\infty} \|p_k - p\| = 0$. □

Lemma 3.4 shows us that Algorithm 3.2 converges when the small data condition $\kappa < 1$ is satisfied. However, it tells us nothing when $\kappa \geq 1$. With Anderson acceleration, we can discuss the convergence behavior $\kappa \geq 1$, see Theorem 4.2. Next, we show the solution operator G is Lipschitz continuous and Fréchet differentiable, after making an assumption on the data.

Lemma 3.5. *For any given* $(l, y),\ (w, z) \in (X_h, Q_h)$*, we have*

$$\|G(l, y) - G(w, z)\|_{X^*} \le C_L \|(l, y) - (w, z)\|_{X^*}, \tag{20}$$

where $C_L = \min\{\kappa + \sqrt{\epsilon/\nu^3} M\|y\|, \kappa + \sqrt{\epsilon/\nu^3} M\|z\|\}$.

Remark 3.6. A global bound on C_L does not appear possible without assuming input to G is bounded. However, combining Lemma 3.5 with Assumption 2.1 implies G is Lipschitz continuous with constant $C_L < 1$ in a neighborhood $\mathcal{B}_\delta$ around the fixed point (u, p). One can similarly take larger neighborhoods where $C_L < K$ for some chosen constant K.

Remark 3.7. Eq. (20) tells us that with a good initial guess, Algorithm 3.2 converges linearly with rate C_L, which may be larger than the usual Picard method's rate of κ [2]. This is not surprising, since (until now) IPP would never be used with small ϵ. In [4], for example, a smaller C_L is found for IPP, but small ϵ is assumed as is an inf–sup compatibility condition on the discrete spaces. We will show in Section 4 that when IPP is enhanced with AA, the effective linear convergence rate will be much smaller than C_L even when $\epsilon = 1$ and without an assumption of an inf–sup condition, and the resulting solver is demonstrated to be very effective in Section 5.

Proof. From (15) with (l, y) and (w, z) by adding and subtracting $b^*(l, G_1(w, z), v)$ to the first difference, we obtain

$$\begin{aligned}\nu(\nabla(G_1(l, y) - G_1(w, z)), \nabla v) + b^*(l, G_1(l, y) - G_1(w, z), v) + b^*(l - w, G_1(w, z), v)\\ -(G_2(l, y) - G_2(w, z), \nabla \cdot v) = 0,\end{aligned} \tag{21}$$

$$\varepsilon(G_2(l, y) - G_2(w, z), q) + (q, \nabla \cdot (G_1(l, y) - G_1(w, z))) = \varepsilon(y - z, q). \tag{22}$$

Adding these equations and choosing $v = G_1(l, y) - G_1(w, z)$ and $q = G_2(l, y) - G_2(w, z)$ yields

$$\begin{aligned}&\nu\|\nabla(G_1(l, y) - G_1(w, z))\|^2 + \varepsilon\|G_2(l, y) - G_2(w, z)\|^2\\ &= \varepsilon(y - z, G_2(l, y) - G_2(w, z)) - b^*(l - w, G_1(w, z), G_1(l, y) - G_1(w, z))\\ &\le \epsilon\|y - z\|\|G_2(l, y) - G_2(w, z)\| + M\|\nabla(l - w)\|\|\nabla(G_1(w, z))\|\|\nabla(G_1(l, y) - G_1(w, z))\|,\end{aligned}$$

thanks to Cauchy–Schwarz and (6). Applying Young's inequality provides

$$\nu\|\nabla(G_1(l, y) - G_1(w, z))\|^2 + \varepsilon\|G_2(l, y) - G_2(w, z)\|^2 \le \varepsilon\|y - z\|^2 + \nu^{-1}M^2\|\nabla G_1(w, z)\|^2\|\nabla(l - w)\|^2,$$

or, in case of adding and subtracting $b^*(w, G_1(l, y), v)$ to the first difference to get (21), gives

$$\nu\|\nabla(G_1(l, y) - G_1(w, z))\|^2 + \varepsilon\|G_2(l, y) - G_2(w, z)\|^2 \le \varepsilon\|y - z\|^2 + \nu^{-1}M^2\|\nabla G_1(l, y)\|^2\|\nabla(l - w)\|^2,$$

which reduces to (20) due to (7) and (16). □

Next, we define an operator G' and then show that G' is the Fréchet derivative of operator G.

Definition 3.8. Given $(w, z) \in (X_h, Q_h) \cap \mathcal{B}_\delta$, define an operator $G'(w, z; \cdot, \cdot) : (X_h, Q_h) \to (X_h, Q_h)$ by

$$G'(w, z; h, s) := (G_1'(w, z; h, s), G_2'(w, z; h, s))$$

satisfying for all $(h, s) \in (X_h, Q_h)$

$$\begin{aligned}\nu(\nabla G_1'(w, z; h, s), \nabla v) + b^*(h, G_1(w, z), v) + b^*(w, G_1'(w, z; h, s), v) - (G_2'(w, z; h, s), \nabla \cdot v) &= 0,\\ \varepsilon(G_2'(w, z; h, s), q) + (q, \nabla \cdot G_1'(w, z; h, s)) &= \varepsilon(s, q).\end{aligned} \tag{23}$$

Lemma 3.9. G' *is well-defined and is the Fréchet derivative of operator* G *satisfying*

$$\|G'(w, z; h, s)\|_{X^*} \le C_L\|(h, s)\|_{X^*}. \tag{24}$$

and

$$\|G'(w + h, z + s; l, y) - G'(w, z; l, y)\|_{X^*} \le \hat{C}_L\|(l, y)\|_{X^*}\|(h, s)\|_{X^*} \tag{25}$$

where C_L *is defined in* Lemma 3.5 *and* $\hat{C}_L = \sqrt{10}\nu^{-3/2}MC_L$.

Proof. The proof consists of three parts. First, we show that G' is well-defined and (24) holds. Adding equations in (23) and setting $v = G_1'(w,z;h,s)$ and $q = G_2'(w,z;h,s)$ produces

$$\nu\|\nabla G_1'(w,z;h,s)\|^2 + \varepsilon\|G_2'(w,z;h,s)\|^2 = \varepsilon(s, G_2'(w,z;h,s)) \\ - b^*(h, G_1(w,z), G_1'(w,z;h,s)).$$

Applying Cauchy–Schwarz, (6) and Young's inequalities gives

$$\begin{aligned}\nu\|\nabla G_1'(w,z;h,s)\|^2 + \varepsilon\|G_2'(w,z;h,s)\|^2 &\le \varepsilon\|s\|^2 + \nu^{-1}M^2\|\nabla h\|^2\|\nabla G_1(w,z)\|^2 \\ &\le \varepsilon\|s\|^2 + \nu^{-1}M^2(\nu^{-1}\|f\|_{-1} + \sqrt{\frac{\epsilon}{\nu}}\|z\|)^2\|\nabla h\|^2 \\ &\le C_L^2(\varepsilon\|s\|^2 + \nu\|\nabla h\|^2),\end{aligned}$$

thanks to (16), which leads to (24). Since the system (23) is linear and finite dimensional, (24) is sufficient to conclude that (23) is well-posed.

The second part shows that G' is the Fréchet derivative of G. Denote $\eta_1 = G_1(w+h, z+s) - G_1(w,z) - G_1'(w,z;h,s)$, $\eta_2 = G_2(w+h, z+s) - G_2(w,z) - G_2'(w,z;h,s)$. Subtracting the sum of (15) and (23) from Eq. (15) with $(w+h, z+s)$ yields

$$\nu(\nabla\eta_1, \nabla v) + b^*(w, \eta_1, v) + b^*(h, G_1(w+h, z+s) - G_1(w,z), v) - (\eta_2, \nabla\cdot v) = 0, \\ \epsilon(\eta_2, q) + (\nabla\cdot\eta_1, q) = 0.$$

Setting $v = \eta_1,\ q = \eta_2$ produces

$$\begin{aligned}\|G(w+h, z+s) - G(w,z) - G'(w,z;h,s)\|_X^2 &\le \nu^{-1}M^2\|\nabla h\|^2\|\nabla(G_1(w+h,z+s) - G_1(w,z))\|^2 \\ &\le \nu^{-3}M^2C_L^2\|(h,s)\|_{X^*}^4,\end{aligned}$$

thanks to Young's inequality and (20). Thus we have verified that G' is the Fréchet derivative of G.

Lastly, we show G' is Lipschitz continuous over (X_h, Q_h). For $(w,z), (h,s), (l,y) \in (X_h, Q_h)$, letting $e_1 := G_1'(w+h, z+s; l, y) - G_1'(w,z;l,y)$, $e_2 := G_2'(w+h, z+s; l, y) - G_2'(w,z;l,y)$, and then subtracting (15) with $G'(w,z;l,y)$ from (15) with $G'(w+h, z+s; l, y)$ yields

$$\nu(\nabla e_1, \nabla v) + b^*(l, G_1(w+h, z+s) - G_1(w,z), v) + b^*(h, G_1'(w,z;l,y), v) \\ + b^*(w+h, e_1, v) - (e_2, \nabla\cdot v) = 0, \\ \epsilon(e_2, q) + (\nabla\cdot e_1, q) = 0.$$

Adding these equations and setting $v = e_1, q = e_2$ eliminates the fourth term and gives us

$$\begin{aligned}\nu\|e_1\|^2 + \varepsilon\|e_2\|^2 &= -b^*(l, G_1(w+h, z+s) - G_1(w,z), e_1) - b^*(h, G_1'(w,z;l,y), e_1) \\ &\le M\|\nabla w\|\|\nabla(G_1(w+h,z+s) - G_1(w,z))\|\|\nabla e_1\| + M\|\nabla h\|\|\nabla G_1'(w,z;l,y)\|\|\nabla e_1\|,\end{aligned}$$

thanks to (23). Now, applying the Young's inequality and (24), we get

$$\|(e_1, e_2)\|_{X^*}^2 \le 10\nu^{-3}M^2C_L^2\|(l,y)\|_{X^*}^2\|(h,s)\|_{X^*}^2,$$

which implies to (25) after taking square roots on both sides. This finishes the proof. □

Lastly, we show G satisfies Assumption 2.4.

4. The Anderson accelerated iterated penalty picard scheme and its convergence

Now we present the Anderson accelerated iterated penalty Picard (AAIPP) algorithm and its convergence properties. Here, we continue the notation from Section 3 that G is the IPP solution operator for a given set of problem data.

Algorithm 4.1 (*AAIPP*)**.** The AAIPP method with depth m for solving the steady NSE is given by:

```
Step 0 Guess $(u_0, p_0) \in (X_h, Q_h)$.
Step 1 Compute $(\tilde{u}_1, \tilde{p}_1) = G(u_0, p_0)$
       Set residual $(w_1, z_1) = (\tilde{u}_1 - u_0, \tilde{p}_1 - p_0)$ and $(u_1, p_1) = (\tilde{u}_1, \tilde{p}_1)$.
Step $k$ For $k = 2, 3, \dots$ set $m_k = \min\{k, m\}$,
    (a) Find $(\tilde{u}_k, \tilde{p}_k) = G(u_{k-1}, p_{k-1})$ and set $(w_k, z_k) = (\tilde{u}_k - u_{k-1}, \tilde{p}_k - p_{k-1})$.
    (b) Find $\{\alpha_j^k\}_{j=0}^{m_k}$ minimizing
        $$\min_{\sum_{j=0}^{m_k} \alpha_j^k = 1} \left\| \sum_{j=0}^{m_k} \alpha_j^k (w_{k-j}, z_{k-j}) \right\|_{X^*}. \qquad (26)$$
    (c) Update $(u_k, p_k) = (1-\beta_k)\left(\sum_{j=0}^{m_k} \alpha_j^k (u_j, p_j)\right) + \beta_k \left(\sum_{j=0}^{m_k} \alpha_j^k (\tilde{u}_j, \tilde{p}_j)\right)$ where $0 < \beta_k \le 1$ is the damping factor.
```

For any step k with $\alpha_m^k = 0$, one should decrease m and repeat Step k, to avoid potential cyclic behavior. Under the assumption that $\alpha_m^k \neq 0$ and together with Lemmas 3.5, 3.9 and assuming either $C_L < 1$ (see Remark 3.6 for how this can be achieved) or that if $C_L \geq 1$ then generally there exists a $\sigma > 0$ satisfying (12) due to there being only small finite number of steps being involved, then the assumptions required to invoke Theorem 2.5 are satisfied, which establishes the following convergence theory for AAIPP.

Theorem 4.2. *For any step $k > m$ with $\alpha_m^k \neq 0$, the following bound holds for the AAIPP residual:*

$$\|(w_{k+1}, z_{k+1})\|_{X^*} \leq \theta_k(1 - \beta_k + \beta_k C_L)\|(w_k, z_k)\|_{X^*} + C\sqrt{1 - \theta_k^2}\|(w_k, z_k)\|_{X^*} \sum_{j=1}^{m} \|(w_{k-j+1}, z_{k-j+1})\|_{X^*},$$

for the residual (w_k, z_k) from Algorithm 4.1, where θ_k is the gain from the optimization problem, C_L is the Lipschitz constant of G defined in Lemma 3.5, and C depending on θ_k, β_k, C_L.

This theorem tells us that Algorithm 4.1 converges linearly with rate $\theta_k(1 - \beta_k + \beta_k C_L) < 1$, which improves on Algorithm 3.2 due to the scaling θ_k and the damping factor β_k.

5. Numerical tests

We now test AAIPP on the benchmark problems of 2D driven cavity, 3D driven cavity, and (time dependent) Kelvin–Helmholtz instability. For all tests, the penalty parameter is chosen as $\epsilon = 1$, and we use the velocity only formulation (5) for the IPP/AAIPP iteration. The X^*-norm is used for the optimization problems in the AA process.

For the driven cavity problems, AA provides a clear positive impact, reducing total iterations and enabling convergence at much higher Re than IPP without AA. For Kelvin–Helmholtz, AA significantly reduces the number of iterations needed at each time step. Overall, our results show that AAIPP with $\epsilon = 1$ is an effective solver. In all of our tests, we use a direct linear solver (i.e. MATLAB's backslash) for the linear system solves of IPP/AAIPP as the problem sizes are such that direct solvers are more efficient. Since the velocity-only formulation is used, convergence is measured in the L^2 norm instead of the X^*-norm, which requires the pressure. For the problems we consider (up to 1.3 million degree of freedom in 3D) this remains a very robust and efficient linear solver. In all of our tests, the cost of applying AA was negligible compared with the linear solve needed at each iteration, generally at least two orders of magnitude less.

5.1. 2D driven cavity

We first test AAIPP for the steady NSE on a lid-driven cavity problem. The domain of the problem is the unit square $\Omega = (0, 1)^2$ and we impose Dirichlet boundary conditions by $u|_{y=1} = (1, 0)^T$ and $u = 0$ on the other three sides. The discretization uses P_2 elements on barycenter-refinement of uniform triangular mesh. The initial guess u_0 is zero in the interior of the domain and it satisfies boundary conditions. We perform AAIPP with $m = 0$ (no acceleration), $1, 2, 5$ and 10 with varying Re, all with no relaxation ($\beta = 1$). In the following tests, convergence was declared if the velocity residual fell below 10^{-8}.

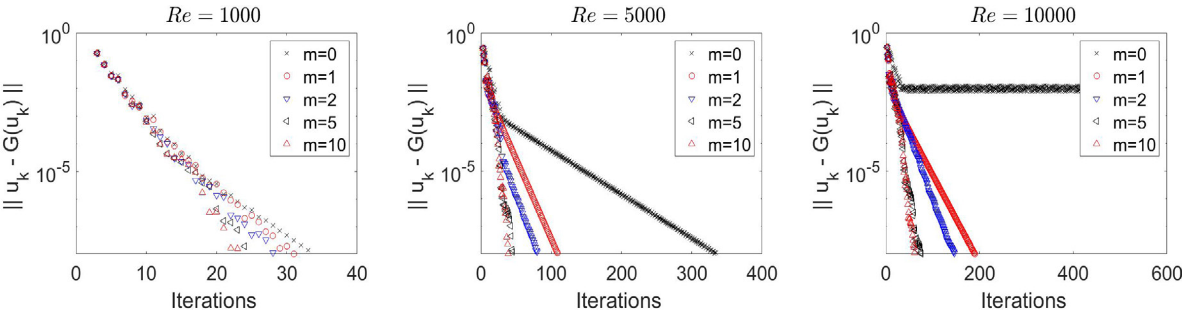

Fig. 1. Shown above is convergence of AAIPP with varying Re and m, using mesh width $h = 1/128$.

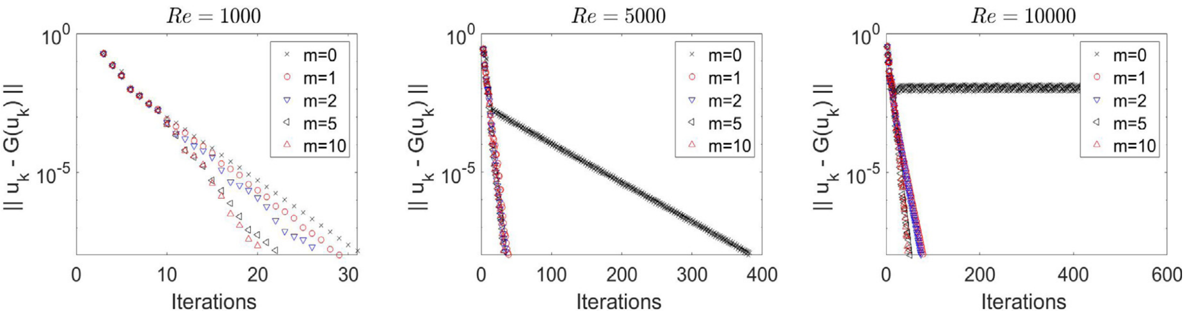

Fig. 2. Shown above is convergence of AAPicard with varying Re and m, using mesh width $h = 1/128$.

Convergence results for AAIPP using $Re = 1000, 5000$ and 10000 with varying m when $h = 1/128$ are shown in Fig. 1. It is observed that as m increases, AA improves convergence. In particular, as Re increases, AA is observed to provide a very significant improvement; for $Re = 10000$, the iteration with $m = 0$ (i.e. IPP iteration) fails but with AA convergence is achieved quickly.

In Fig. 2, convergence results by Anderson accelerated Picard (AAPicard) iteration are shown (i.e. no iterated penalty, and solve the nonsymmetric saddle point linear system at each iteration). The same mesh is used, and here with (P_2, P_1^{disc}) Scott–Vogelius elements. Convergence behavior for AAPicard is observed to be overall similar to that of AAIPP with $\epsilon = 1$, with the exception that for lower $m = 1, 2$ AAPicard performs slightly better than AAIPP in terms of total number of iterations. We note that similar convergence behavior between AAIPP and AAPicard means that AAIPP is overall much more efficient. This is because each iteration of AAPicard requires a saddle point system solve for velocity and pressure that needs to resolve a Schur complement, which is significantly more expensive that AAIPP's one linear solve for velocity only when $\epsilon = O(1)$. Quantifying this advantage requires specifying linear solvers and preconditioners (of which there are many possible choices), but our experience is that it is generally fair to say that most solvers (e.g. [12,36,37]) for the saddle point system of AAPicard would do at least several 'inner' linear solves (as part of its solve process) that are of the same difficulty as AAIPP's one linear solve per iteration.

Since each iteration of AAIPP is significantly cheaper than each iteration of AAPicard (generally speaking, since AAPicard must solve a more difficult linear system), these results show AAIPP with $\epsilon = 1$ performs very well.

5.2. 3D Driven cavity

We next test AAIPP on the 3D lid driven cavity. In this problem, the domain is the unit cube, there is no forcing ($f = 0$), and homogeneous Dirichlet boundary conditions are enforced on all walls and $u = \langle 1, 0, 0\rangle$ on the moving lid. The initial iterate u_0 is zero in the interior of the domain but satisfies the boundary conditions. We compute with P_3 elements on Alfeld split tetrahedral meshes with 796,722 and 1,312,470 total degrees of freedom (dof) that are weighted towards the boundary by using a Chebychev grid before tetrahedralizing. We test AAIPP with varying Re, m, and relaxation parameter β.

Table 1
Shown above is the number of AAIPP iterations required for convergence in the 3D driven-cavity tests, with varying dof, Re, damping factor β and depth.

			Iterations for convergence					
dof	Re	β	$m = 0$	$m = 1$	$m = 2$	$m = 5$	$m = 10$	$m = k - 1$
796,722	100	1	> 300	> 300	138	98	92	75
796,722	400	1	> 300	> 300	260	120	91	62
796,722	1000	1	> 300	> 300	> 300	> 300	126	77
796,722	1000	0.5	> 300	> 300	> 300	198	114	68
796,722	1000	0.2	> 300	> 300	> 300	> 300	177	87
1,312,470	1000	1	> 300	> 300	> 300	> 300	140	83
1,312,470	1000	0.5	> 300	> 300	> 300	203	109	71
1,312,470	1000	0.2	> 300	> 300	> 300	> 300	174	88

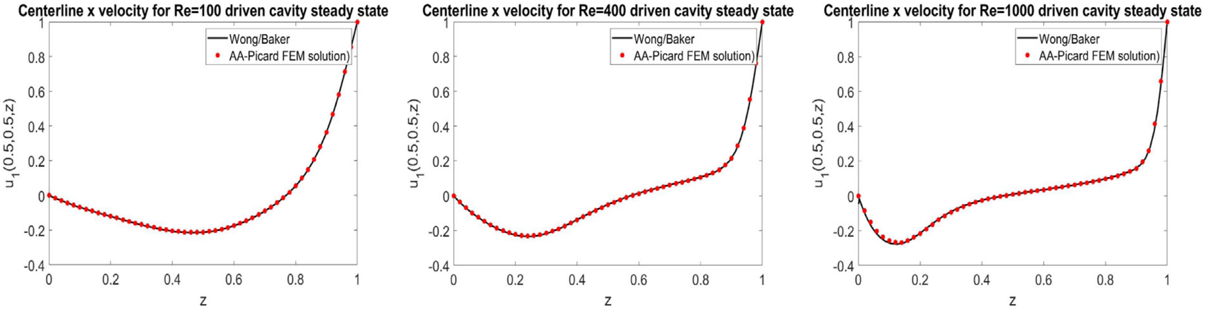

Fig. 3. Shown above is the centerline x-velocity plots for the 3D driven cavity simulations at $Re = 100, 400, 1000$, using AAIPP with $m = 10$, $\beta = 1$ and 796,722 dof.

Fig. 4 shows a visualization of the computed solutions by AAIPP with $m = 10$ for $Re = 100, 400$ and 1000 on the 796,722 total dof, which are in good qualitative agreement with reference results of Wong and Baker [38]. In Fig. 3, we compare centerline x-velocities for varying $Re = 100, 400$ and 1000 on the same mesh with reference data of Wong and Baker [38] and obtain excellent agreement.

In Table 1, the number of AAIPP iterations required for reducing the velocity residual to fall below 10^{-8}, for varying dof, Re, relaxation parameter β and depth, within 300 iterations. In each cases, we observe that as m increases, the number of iterations decreases, and the maximum $m = k - 1$ is observed to be the best choice in all cases. Also, the relaxation parameter β is observed to give improved results for larger Re. Thus, AAIPP with properly chosen depth and relaxation can significantly improve the ability of the iteration to converge.

We also tested AAIPP with $Re = 1500, 2000, 2500$ which would not converge within 1000 iterations without AA. Table 2 shows that with sufficiently large m and properly chosen relaxation parameter, AAIPP converges for even higher Re. This is especially interesting since the bifurcation point where this problem becomes time dependent is around $Re \approx 2000$ [39], and so here AAIPP is finding steady solutions in the time dependent regime.

5.3. Kelvin–Helmholtz instability

For our last test we consider a benchmark problem from [40] for 2D Kelvin–Helmholtz instability. This test is time dependent, and we apply the IPP/AAIPP method at each time step to solve the nonlinear problem. The domain is the unit square, with periodic boundary conditions at $x = 0, 1$. At $y = 0, 1$, the no penetration boundary condition $u \cdot n = 0$ is strongly enforced, along with a natural weak enforcement of the free-slip condition $(-\nu\nabla u \cdot n) \times n = 0$. The initial condition is given by

$$u_0(x, y) = \begin{pmatrix} \tanh(28(2y - 1)) \\ 0 \end{pmatrix} + 10^{-3} \begin{pmatrix} \partial_y \psi(x, y) \\ -\partial_x \psi(x, y) \end{pmatrix},$$

Table 2
Shown above is the number of AAIPP iterations required for convergence in the 3D driven-cavity tests, with varying dof, Re, damping factor β and depth of AA iterations.

Re	dof	m	β	k
1500	1,312,470	$k-1$	0.5	102
1500	1,312,470	$k-1$	0.3	111
2000	1,312,470	$k-1$	0.5	154
2000	1,312,470	$k-1$	0.3	162
2500	1,312,470	100	0.5	381
2500	1,312,470	100	0.3	> 1000
2500	1,312,470	$k-1$	0.5	371

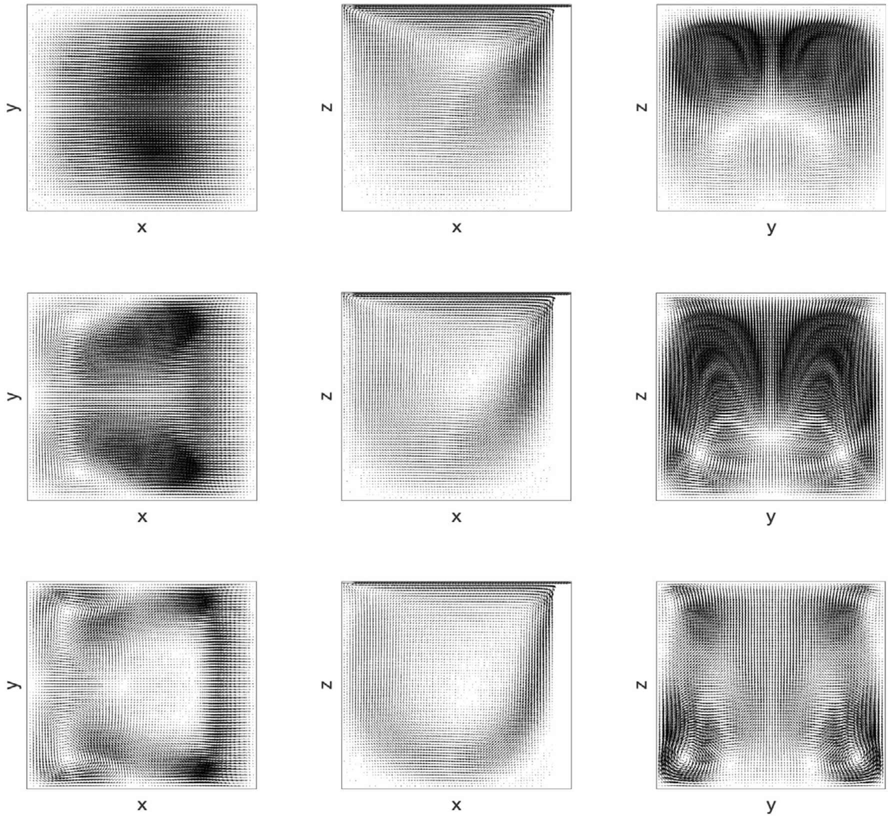

Fig. 4. Shown above are the midsliceplane plots for the 3D driven cavity simulations at $Re = 100, 400$ and 1000 by AAIPP with $m = 10$, $\beta = 1$ and 796,722 dof.

where $\frac{1}{28}$ is the initial vorticity thickness, 10^{-3} is a noise/scaling factor, and

$$\psi(x, y) = \exp\left(-28^2(y-0.5)^2\right)\left(\cos(8\pi x) + \cos(20\pi x)\right).$$

The Reynolds number is defined by $Re = \frac{1}{28\nu}$, and ν is defined by selecting Re. Solutions are computed for both $Re = 100$ and $Re = 1000$, up to end time $T = 5$.

Define $X = \{v \in H^1(\Omega),\ v(0, y) = v(1, y),\ v \cdot n = 0 \text{ at } y = 0, 1\}$, and take $V = \{v \in X,\ \|\nabla \cdot v\| = 0\}$ and $X_h = P_2(\tau_h) \cap X$. The problem now becomes at each time step: Find $u^{n+1} \in X_h \cap V$ satisfying

$$\frac{1}{2\Delta t}(3u^{n+1}, v) + (u^{n+1} \cdot \nabla u^{n+1}, v) + \nu(\nabla u^{n+1}, \nabla v) = (f, v) + \frac{1}{2\Delta t}(4u^n - u^{n-1}, v)\ \forall v \in X_h \cap V.$$

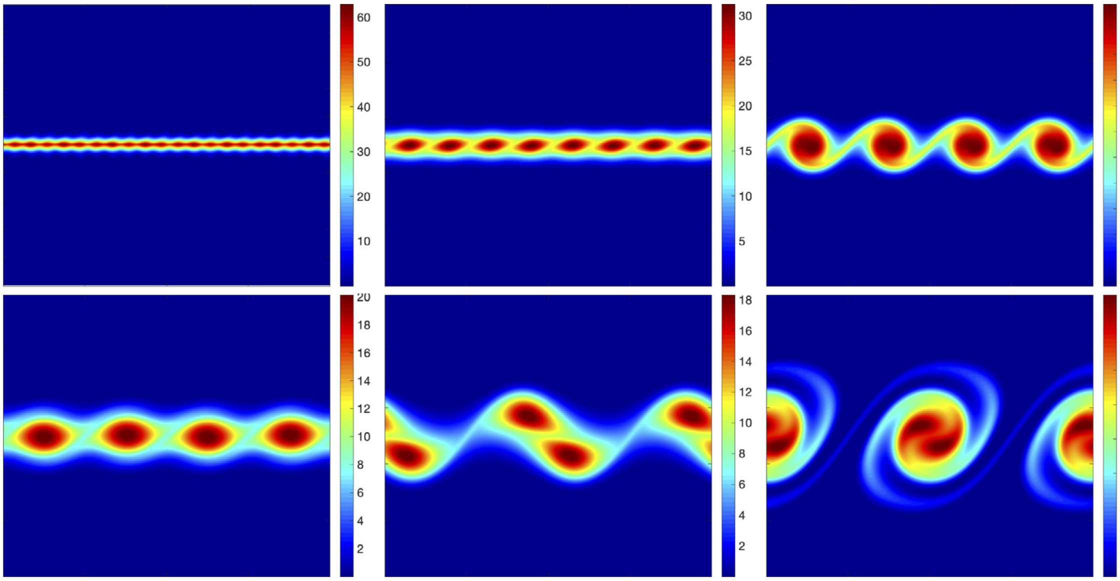

Fig. 5. Shown above are $Re = 100$ absolute vorticity contours for IPP solution, at times $t = 0$, 1, 2, 3, 4, 5 and 6 (left to right, top to bottom).

The IPP iteration to find each u^{n+1} is thus analogous to what is used for solving the steady NSE above including pressure recovery, but now with the time derivative terms and using the previous time step solution as the initial guess.

For accuracy comparison, we also give results using the standard (nonlinear) BDF2 mixed formulation using skew-symmetry, which we will refer to as the SKEW formulation: Find $u^{n+1} \in X_h$ and $p_h^{n+1} \in Q_h = P_1(\tau_h) \cap L_0^2(\Omega)$ satisfying

$$\frac{1}{2\Delta t}(u^{n+1}, v) + b^*(u^{n+1}, u^{n+1}, v) + (p^{n+1}, \nabla \cdot v) + \nu(\nabla u^{n+1}, \nabla v) = (f, v) + \frac{1}{2\Delta t}(u^n - 2u^{n-1}, v),$$

$$(\nabla \cdot u^{n+1}, q) = 0,$$

for all $(v, q) \in (X_h, Q_h)$. The nonlinear problem for SKEW is resolved using Newton's method, and since Taylor–Hood elements are being used, a large divergence error is expected.

For $Re = 100$, a $h = \frac{1}{128}$ uniform triangular mesh was used, together with a time step of $\Delta t = 0.005$. The tolerance for the nonlinear solver was to reduce the H^1 relative residual to 10^{-6}. Simulations were performed with IPP, AAIPP with $m = 1, 3, k - 1$, and SKEW. The evolution of the flow can be seen in Fig. 5 as absolute vorticity contours from the AAIPP solutions (all IPP and AAIPP solutions were visually indistinguishable), and these match those of the high resolution solution from [40] and solutions from [41]. In addition to their plots of vorticity contours being the same, the IPP and AAIPP solutions yielded the same energy and enstrophy to five significant digits (i.e. they all give the same solution, as expected). Fig. 6 shows the energy, enstrophy and divergence of the IPP/AAIPP/SKEW solutions versus time, along with energy and enstrophy of the high resolution solutions from [40]. We observe that the energy and enstrophy solutions of IPP, AAIPP, and SKEW all match the high resolution reference solutions very well. As expected, the IPP/AAIPP solutions have divergence error around 10^{-5}, which is consistent with a relative residual L^2 stopping criteria of 10^{-8}. SKEW, however, has a large divergence error that is $O(10^{-2})$ despite a rather fine mesh and essentially resolving the flow; since Taylor–Hood elements are used, this large divergence error is not surprising [9].

Fig. 7 shows the number of iterations needed to converge IPP/AAIPP at each time step. We observe that using AAIPP with $m = 1$ offers no real improvement over IPP ($m = 0$) in converging the iteration, however both $m = 3$ and $m = k - 1$ both offer significant improvement. While at early iterations the larger m choices give modest improvement, by $t = 4$ the larger m choices cut the iteration count from 27 to 16 at each time step.

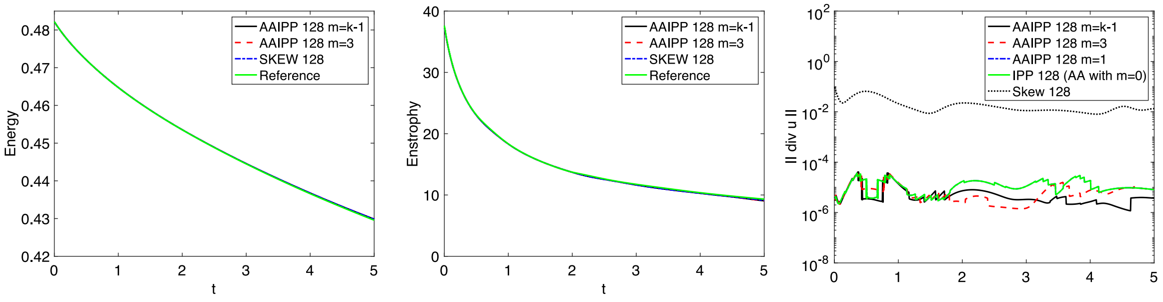

Fig. 6. Shown above are $Re = 100$ energy, enstrophy and divergence error versus time for the IPP/AAIPP, SKEW and reference solutions from [40].

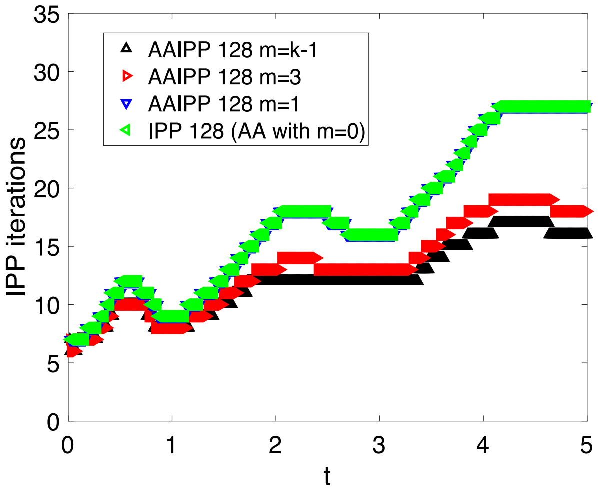

Fig. 7. Shown above are IPP/AAIPP total iteration counts at each time step, for $Re = 100$ and varying m.

For $Re = 1000$, the IPP/AAIPP tests again used $h = \frac{1}{128}$, using the same setup as for the $Re = 100$ case and again solutions are compared with a reference solution from [40] and the SKEW solution (but here the SKEW solution uses a finer mesh with $h = \frac{1}{196}$. As discussed in [41], even the $\frac{1}{196}$ mesh is not fully resolved for this Reynolds number. The time step size was chosen to be $\Delta t = 0.001$ for the IPP/AAIPP and SKEW simulations. IPP/AAIPP vorticity contours are plotted in Fig. 8, and match those from the reference solution qualitatively well (as discussed in [40], the evolution of this flow in time is very sensitive and it is not clear what is the correct behavior in time, even though it is clear how the flow develops spatially and how the eddies combine). Fig. 9 shows the energy, enstrophy and divergence of the computed and reference solutions (the reference solution has divergence on the order of roundoff error, and it is not shown), and we observe that the 1/196 SKEW solution gives the worst predictions of energy, enstrophy and (not surprisingly) divergence, even though IPP/AAIPP uses a significantly coarser mesh. Finally, the impact of AA on the IPP iteration is shown in Fig. 10, where we observe a significant reduction in iterations at each time step, with larger m cutting the total number of iterations by a factor of four. Hence overall, the AAIPP iteration is effective and efficient, and produces accurate divergence-free solutions.

6. Conclusions

In this paper, we studied IPP with penalty parameter $\epsilon = 1$, and showed that while alone it is not an effective solver for the NSE, when used with AA and large m it becomes very effective. We proved the IPP fixed point function satisfies regularity properties which allow the AA theory of [3] to be applied, which shows that AA applied to IPP will scale the linear convergence rate by the ratio gain of the underlying AA optimization problem. We also showed results of three test problems which revealed AAIPP with $\epsilon = 1$ is a very effective solver, without any

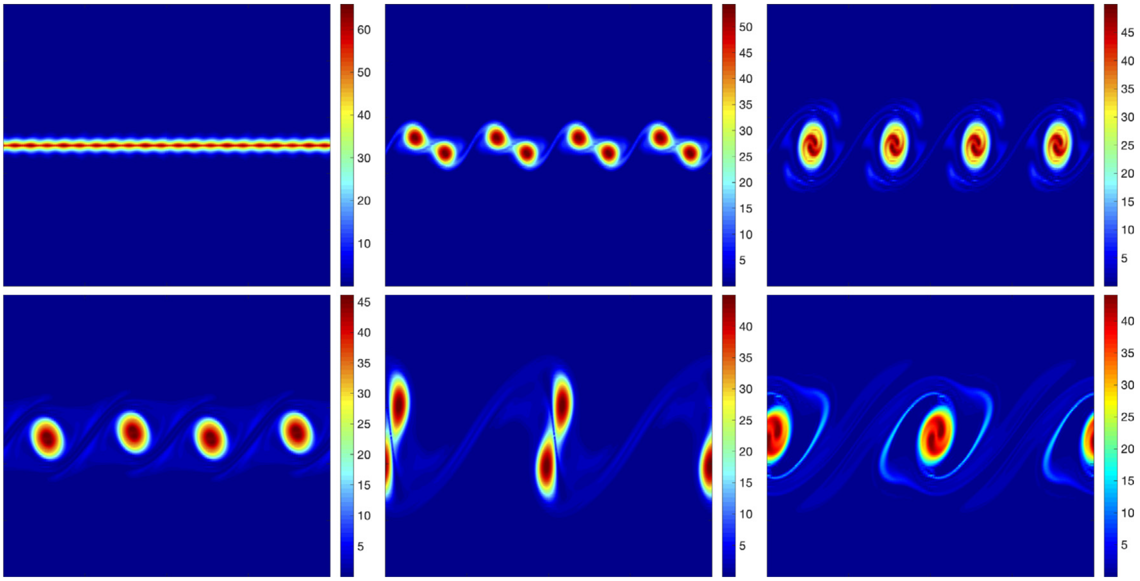

Fig. 8. Shown above are $Re = 1000$ absolute vorticity contours for the iterated penalty solution, at times $t = 0$, 1, 2, 3, 4, 5 and 6 (left to right, top to bottom).

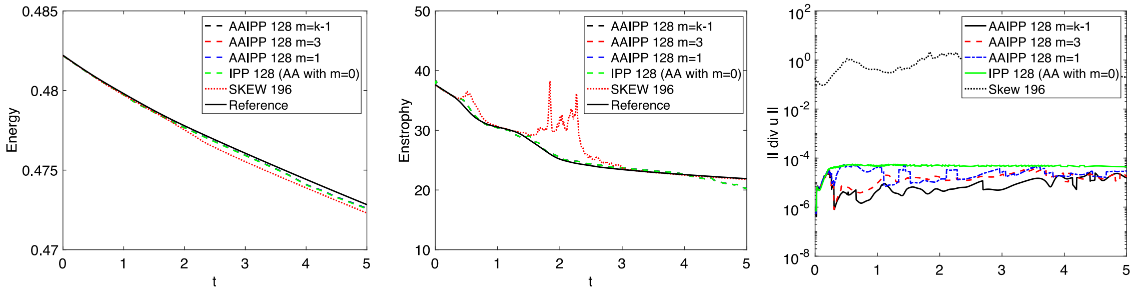

Fig. 9. Shown above are $Re = 1000$ energy, enstrophy and divergence error versus time for the IPP/AAIPP, SKEW and reference solutions from [40].

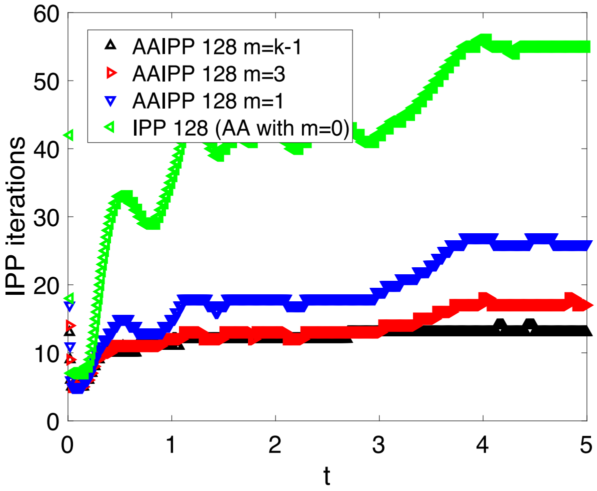

Fig. 10. Shown above are IPP/AAIPP total iteration counts at each time step, for $Re = 1000$ and varying m.

continuation method or pseudo time-stepping. While the classical IPP method is not commonly used for large scale NSE problems due to difficulties with linear solvers when ϵ is small, our results herein suggest it may deserve a second look since using AA allows for the penalty parameter $\epsilon = 1$ to be used, which in turn will allow for effective preconditioned iterative linear solvers to be used such as those in [11,12,14].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] V. John, P. Knobloch, J. Novo, Finite elements for scalar convection-dominated equations and incompressible flow problems-a never ending story? Comput. Vis. Sci. 12 (2018).
[2] V. Girault, P.-A. Raviart, Finite Element Methods for Navier–Stokes Equations: Theory and Algorithms, Springer-Verlag, 1986.
[3] S. Pollock, L. Rebholz, Anderson acceleration for contractive and noncontractive operators, IMA J. Numer. Anal. (2021) in press.
[4] R. Codina, An iterative penalty method for the finite element solution of the stationary Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 110 (1993) 237–262.
[5] M. Gunzburger, Iterative penalty methods for the stokes and Navier–Stokes equations, in: Proceedings from Finite Element Analysis in Fluids Conference, University of Alabama, Huntsville, 1989, pp. 1040–1045.
[6] L.R. Scott, Kinetic energy flow instability with application to Couette flow, 2021, submitted for publication.
[7] H. Morgan, L.R. Scott, Towards a unified finite element method for the Stokes equations, SIAM J. Sci. Comput. 40 (1) (2018) A130–A141.
[8] L. Rebholz, M. Xiao, On reducing the splitting error in Yosida methods for the Navier–Stokes equations with grad-div stabilization, Comput. Methods Appl. Mech. Eng. 294 (2015) 259–277.
[9] V. John, A. Linke, C. Merdon, M. Neilan, L.G. Rebholz, On the divergence constraint in mixed finite element methods for incompressible flows, SIAM Rev. 59 (3) (2017) 492–544.
[10] J. Schöberl, Robust multigrid methods for a parameter dependent problem in primal variables, Numer. Math. 84 (1999) 97–119.
[11] M.A. Olshanskii, E.E. Tyrtyshnikov, Iterative Methods for Linear Systems: Theory and Applications, SIAM, Philadelphia, 2014.
[12] T. Heister, G. Rapin, Efficient augmented Lagrangian-type preconditioning for the Oseen problem using grad-div stabilization, Int. J. Numer. Methods Fluids 71 (2013) 118–134.
[13] B. Cousins, S. Le Borne, A. Linke, L. Rebholz, Z. Wang, Efficient linear solvers for incompressible flow simulations using Scott–Vogelius finite elements, Numer. Methods Partial Differential Equations 29 (2013) 1217–1237.
[14] S. Börm, S. Le Borne, $\mathcal{H}$-LU Factorization in preconditioners for augmented Lagrangian and grad-div stabilized saddle point systems, Int. J. Numer. Methods Fluids 68 (2012) 83–98.
[15] P.A. Lott, H.F. Walker, C.S. Woodward, U.M. Yang, An accelerated Picard method for nonlinear systems related to variably saturated flow, Adv. Water Resour. 38 (2012) 92–101.
[16] S. Pollock, L. Rebholz, M. Xiao, Anderson-accelerated convergence of Picard iterations for incompressible Navier-Stokes equations, SIAM J. Numer. Anal. 57 (2019) 615–637.
[17] S. Pollock, L. Rebholz, M. Xiao, Acceleration of nonlinear solvers for natural convection problems, J. Numer. Math. (2021) in press.
[18] Y. Peng, B. Deng, J. Zhang, F. Geng, W. Qin, L. Liu, Anderson acceleration for geometry optimization and physics simulation, ACM Trans. Graph. 42 (2018) 1–14.
[19] H. An, X. Jia, H.F. Walker, Anderson acceleration and application to the three-temperature energy equations, J. Comput. Phys. 347 (2017) 1–19.
[20] A. Toth, C.T. Kelley, S. Slattery, S. Hamilton, K. Clarno, R. Pawlowski, Analysis of Anderson acceleration on a simplified neutronics/thermal hydraulics system, in: Proceedings of the ANS MC2015 Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, ANS MC2015 CD, 2015, pp. 1–12.
[21] J. Guzman, L.R. Scott, The Scott–Vogelius finite elements revisited, Math. Comput. 88 (316) (2019) 515–529.
[22] P. Stasiak, M.W. Matsen, Efficiency of pseudo-spectral algorithms with anderson mixing for the SCFT of periodic block-copolymer phases, Eur. Phys. J. E 34 (110) (2011) 1–9.
[23] N. Higham, N. Strabic, Anderson acceleration of the alternating projections method for computing the nearest correlation matrix, Numer. Algorithms 72 (2016) 1021–1042.
[24] H.F. Walker, P. Ni, Anderson acceleration for fixed-point iterations, SIAM J. Numer. Anal. 49 (4) (2011) 1715–1735.
[25] C.T. Kelley, Numerical methods for nonlinear equations, Acta Numer. 27 (2018) 207–287.
[26] J. Loffeld, C. Woodward, Considerations on the implementation and use of Anderson acceleration on distributed memory and GPU-based parallel computers, Adv. Math. Sci. (2016) 417–436.
[27] A. Fu, J. Zhang, S. Boyd, Anderson accelerated Douglas-Rachford splitting, SIAM J. Sci. Comput. 42 (6) (2020) A3560–A3583.
[28] D. Wicht, M. Schneider, T. Bohlke, Anderson-accelerated polarization schemes for fast Fourier transform-based computational homogenization, Int. J. Numer. Methods Eng. 122 (9) (2021) 2287–2311.
[29] W. Layton, An Introduction to the numerical Analysis of viscous incompressible flows, SIAM, Philadelphia, 2008.
[30] S. Zhang, Quadratic divergence-free finite elements on Powell–Sabin tetrahedral grids, Calcolo 48 (3) (2011) 211–244.

[31] S. Zhang, Divergence-free finite elements on tetrahedral grids for $k \geq 6$, Math. Comput. 80 (274) (2011) 669–695.

[32] M. Neilan, D. Sap, Stokes elements on cubic meshes yielding divergence-free approximations, Calcolo 53 (3) (2016) 263–283.

[33] D. Arnold, J. Qin, Quadratic velocity/linear pressure Stokes elements, in: R. Vichnevetsky, D. Knight, G. Richter (Eds.), Advances in Computer Methods for Partial Differential Equations VII, IMACS, 1992, pp. 28–34.

[34] A. Toth, C.T. Kelley, Convergence analysis for Anderson acceleration, SIAM J. Numer. Anal. 53 (2) (2015) 805–819.

[35] C. Evans, S. Pollock, L. Rebholz, M. Xiao, A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically), SIAM J. Numer. Anal. 58 (2020) 788–810.

[36] M. Benzi, M. Olshanskii, An augmented Lagrangian-based approach to the Oseen problem, SIAM J. Sci. Comput. 28 (2006) 2095–2113.

[37] Steffen Börm, Sabine Le Borne, $\mathcal{H}$-LU factorization in preconditioners for augmented Lagrangian and grad-div stabilized saddle point systems, Int. J. Numer. Methods Fluids 68 (1) (2012) 83–98.

[38] K.L. Wong, A.J. Baker, A 3D incompressible Navier–Stokes velocity–vorticity weak form finite element algorithm, Int. J. Numer. Methods Fluids 38 (2) (2002) 99–123.

[39] S.-H. Chiu, T.-W. Pan, J. He, A. Guo, R. Glowinski, A Numerical Study of the Transition to Oscillatory Flow in 3D Lid-Driven Cubic Cavity Flows, 2016.

[40] P. Schroeder, V. John, P. Lederer, C. Lehrenfeld, G. Lube, J. Schoberl, On reference solutions and the sensitivity of the 2D Kelvin-Helmholtz instability problem, Comput. Math. Appl. 77 (4) (2019) 1010–1028.

[41] M. Olshanskii, L. Rebholz, Longer time accuracy for incompressible Navier–Stokes simulations with the EMAC formulation, Comput. Methods Appl. Mech. Eng. 372 (113369) (2020) 1–17.