Received: 11 May 2018 | Revised: 3 October 2018 | Accepted: 5 October 2018
DOI: 10.1002/num.22326

RESEARCH ARTICLE

WILEY

Effective Chorin–Temam algebraic splitting schemes for the steady Navier–stokes equations

Alex Viguerie[1] | **Mengying Xiao**[2]

[1]Department of Civil Engineering and Architecture, University of Pavia, Pavia, Italy
[2]Department of Mathematical Sciences, Clemson University, Clemson, South Carolina,

Correspondence
Alex Viguerie, Department of Civil Engineering and Architecture, University of Pavia, Pavia 27100, Italy.
Email: alexander.viguerie@unipv.it

Funding information
The authors (NS) acknowledges the support provided by the Directorate for Mathematical and Physical Sciences, 1522191. NSF, DMS1522191.

This paper continues some recent work on the numerical solution of the steady incompressible Navier–Stokes equations. We present a new method, similar to the one presented in Rebholz et al., but with superior convergence and numerical properties. The method is efficient as it allows one to solve the same symmetric positive-definite system for the pressure at each iteration, allowing for the simple preconditioning and the reuse of preconditioners. We also demonstrate how one can replace the Schur complement system with a diagonal matrix inversion while maintaining accuracy and convergence, at a small fraction of the numerical cost. Convergence is analyzed for Newton and Picard-type algorithms, as well as for the Schur complement approximation.

KEYWORDS
algebraic splitting, finite element methods, grad-div stabilization, Navier-Stokes equations

1 | INTRODUCTION

In this work we consider efficient nonlinear iteration schemes to solve the incompressible steady Navier–Stokes equations (NSE), which are given by

$$u \cdot \nabla u + \nabla p - \nu \Delta u = f, \tag{1.1}$$

$$\nabla \cdot u = 0, \tag{1.2}$$

$$u|_{\partial\Omega} = 0, \tag{1.3}$$

where u and p represent velocity and pressure, respectively, f is a forcing, and ν is the viscosity. We will assume homogeneous Dirichlet boundary conditions in our analysis for simplicity, but

†Mengying Xiao was partially supported by NSF grant DMS1522191.

in our numerical tests we will use both nonhomogeneous Dirichlet and zero-traction boundary conditions.

The nonlinear nature of the problem requires the use of some type of iterative scheme, such as the standard Picard iteration: given an initial guess u^0, for $k = 1, 2, \ldots$ find (u^k, p^k) such that:

$$u^{k-1} \cdot \nabla u^k + \nabla p^k - \nu \Delta u^k = f, \tag{1.4}$$

$$\nabla \cdot u^k = 0, \tag{1.5}$$

$$u|_{\partial\Omega}^k = 0. \tag{1.6}$$

After discretization (using e.g. the finite element method), the above linear system at iteration k arises in the following form:

$$\begin{bmatrix} \nu K + C(u^{k-1}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u^k \\ p^k \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \tag{1.7}$$

where K corresponds to the discretization of the diffusive term, $C(u^{k-1})$ to the convective term, and B to the discrete divergence operator (with its adjoint B^T corresponding to the discrete gradient accordingly).

This saddle point system can be difficult to solve, especially when the nonsymmetric matrix $C(u^{k-1})$ dominates the (1,1) block. The above system admits the following block-LU decomposition (denoting $A := \nu K + C(u^{k-1})$ for the sake of notation):

$$\begin{bmatrix} A & 0 \\ B & -BA^{-1}B^T \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u^k \\ p^k \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{1.8}$$

Solving the block-LU system is often impractical, as the Schur complement $S := BA^{-1}B^T$ is difficult to precondition and computing its action requires the solution of a system in A, which can be difficult, especially for convection-dominated problems. Much work has been done on developing effective preconditioners for the Schur complement [1, 2].

A popular approach in the unsteady (time-dependent) problem is to instead solve an approximate version of (1.8) [3–6] as follows:

$$\begin{bmatrix} A & 0 \\ B & -BH_1B^T \end{bmatrix} \begin{bmatrix} I & H_2B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u^k \\ p^k \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{1.9}$$

where $H_1 \approx A^{-1}$ and $H_2 \approx A^{-1}$. In these schemes one chooses H_1 and H_2 such that the approximated Schur complement has favorable numerical properties (e.g. it may be symmetric positive-definite (SPD)). In many instances, the Schur complement remains the same at each iteration, greatly simplifying preconditioning. The *Algebraic Chorin–Temam* scheme corresponds to using the velocity mass matrix for H_1 and H_2, while for the *Yosida method* one approximates H_1 with the velocity mass matrix but lets $H_2 = A^{-1}$ [3–5]. For this reason, we will hereafter refer to approximate LU factorizations in which $H_1 = H_2$ as *Chorin–Temam*-type methods and $H_1 \neq H_2$ as *Yosida-type* methods, even if the discrete operators corresponding to our approximations are not the same.

The extension of these methods to the steady setting is not immediate, as they require the presence of a time derivative term. Recently, however, schemes have been developed specifically for the steady problem that retain many of these advantages [7–9]. In this work, we present a direct follow-up to the schemes studied in [7]. The schemes herein are somewhat less computationally intensive and demonstrate superior stability and convergence properties when compared to those of [7], while retaining all of those methods' advantages. We also present a strategy to circumvent the Schur complement solves entirely, replacing it with the inversion of a diagonal mass matrix. We analyze these methods in detail and demonstrate their effectiveness with numerical results.

1.1 | Mathematical preliminaries

We consider a domain $\Omega \subset \mathbb{R}^d$ $(d=2,3)$ that is open, connected, and with Lipschitz boundary $\partial\Omega$. Denote the L^2-norm and inner product as $\|\cdot\|$ and $(\cdot,\cdot)$, and $L_0^2(\Omega)$ is the zero mean subspace of $L^2(\Omega)$. 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 space for velocity and pressure of Stokes and NSE are

$$X := H_0^1(\Omega) = \{v \in H^1(\Omega), v = 0 \text{ on } \partial\Omega\}, \qquad Q := L_0^2(\Omega) = \left\{ q \in L^2(\Omega), \int_\Omega q \ dx = 0 \right\}.$$

In the space X, the Poincare inequality is known to hold: there exists $\lambda > 0$, dependent only on the size of Ω, such that for every $v \in X$,

$$\|v\| \le \lambda \|\nabla v\|.$$

The dual spaces of X will be denoted by X', with norm $\|\cdot\|_{-1}$.

Let τ_h be a conforming, shape-regular, and simplicial triangulation of Ω with h_T denoting the maximum element diameter. We denote with P_k the space of degree k globally continuous piecewise polynomials with respect to τ_h, and P_k^{disc} the space of degree k piecewise polynomials that can be discontinuous across elements.

Throughout the paper, we consider only discrete velocity–pressure spaces $(X_h, Q_h) \subset (X, Q)$ that satisfy the Ladyzhenskaya–Babuska–Brezzi condition: there exists a constant β satisfying

$$\inf_{q \in Q_h} \sup_{v \in X_h} \frac{(\nabla \cdot v, q)}{\|q\| \|\nabla v\|} \ge \beta > 0. \tag{1.10}$$

where β is independent of h. Common examples of such elements are (P_2, P_1) Taylor–Hood (TH) elements, and (P_k, P_{k-1}^{disc}) Scott–Vogelius (SV) elements on meshes with particular structure [10, 11], and [12, 13]. Define the discretely divergence free velocity space by

$$V_h := \{v \in X_h, \ (\nabla \cdot vq) = 0 \ \ \forall q \in Q_h\}.$$

Define the skew-symmetric, trilinear operator $b^* : X \times X \times X \to \mathbb{R}$ by

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

and recall, from for example. [14], that there exists M depending only on Ω such that

$$| b^*(u, v, w) | \le M \|\nabla u\| \|\nabla v\| \|\nabla w\|, \tag{1.11}$$

for every $u, v, w \in X$.

1.2 | Discrete steady NSE

The discrete steady NSE are given by: find $(u, p) \in (X_h, Q_h)$ satisfying for all $(v, q) \in (X_h, Q_h)$,

$$\xi(\nabla \cdot u, \nabla \cdot v) + b^*(u, u, v) - (p, \nabla \cdot v) + \nu(\nabla u, \nabla v) = (f, v), \tag{1.12}$$

$$(\nabla \cdot u, q) = 0. \tag{1.13}$$

The parameter $\xi \ge 0$ is the grad-div stabilization parameter. We will use $\xi \sim \mathcal{O}(1)$ in this paper, which is known to give good results for $\mathbb{P}^2/\mathbb{P}^1$ TH finite elements (which we use for several numerical simulations in this work) [15].

Recall that if

$$\alpha := M\nu^{-2}\|f\|_{-1} < 1, \tag{1.14}$$

then (1.12) and (1.13) is well-posed and

$$\|\nabla u\| \leq \nu^{-1}\|f\|_{-1}. \tag{1.15}$$

The classical Picard and Newton algorithms for solving the nonlinear problem are given below. In general, the linear systems arising from both schemes are of the form (1.7) and thus are difficult to precondition and solve, motivating the need for alternative schemes presented below.

Algorithm 1.1 *Picard iteration for steady Navier–Stokes:*
Step 1: Guess $u_0 \in X_h$.
Step k: Find $(u_k, p_k) \in (X_h, Q_h)$ satisfying for all $(v, q) \in (X_h, Q_h)$,

$$\begin{aligned} b^*(u_{k-1}, u_k, v) - (p_k, \nabla \cdot v) + \nu(\nabla u_k, \nabla v) + \xi(\nabla \cdot u_k, \nabla \cdot v) &= (f, v), \\ (\nabla \cdot u_k, q) &= 0. \end{aligned}$$

Algorithm 1.2 Newton iteration for steady Navier–Stokes:
Step 1: Guess $u_0 \in X_h$.
Step k: Find $(u_k, p_k) \in (X_h, Q_h)$ satisfying for all $(v, q) \in (X_h, Q_h)$,

$$\begin{aligned} b^*(u_{k-1}, u_k, v) + b^*(u_k, u_{k-1}, v) - b^*(u_{k-1}, u_{k-1}, v) & \\ -(p_k, \nabla \cdot v) + \nu(\nabla u_k, \nabla v) + \xi(\nabla \cdot u_k, \nabla \cdot v) &= (f, v), \\ (\nabla \cdot u_k, q) &= 0. \end{aligned}$$

Newton is known to converge quadratically (under a small data condition), though its performance is very sensitive to the quality of initial guess and may fail [14]. Picard converges linearly but it is significantly less sensitive to the initial guess and is preferred in many settings for this reason. It is not uncommon to combine the iteration types, running several iterations of Picard before switching to Newton, thus taking advantage of both Newton's superior convergence rate and Picard's superior reliability [16].

In [7], the following alternative scheme was presented and shown to be convergent for $\xi \geq \nu$:

Algorithm 1.3 The grad-div stabilized Picard–Yosida iteration for the steady Navier–Stokes:
Step 1: Guess $u_0 \in X_h$ and $p_0 \in Q_h$.
Step k consists of the following three steps:
k.1 Find $z_k \in X_h$ satisfying for all $v \in X_h$,

$$\xi(\nabla \cdot z_k, \nabla \cdot v) + b^*(u_{k-1}, z_k, v) + \nu(\nabla z_k, \nabla v) = (f, v) + (p_{k-1}, \nabla \cdot v).$$

k.2 Find $(w_k, \delta_k^p) \in (X_h, Q_h)$ satisfying for all $(v, q) \in (X_h, Q_h)$,

$$\begin{aligned} \xi(\nabla \cdot w_k, \nabla \cdot v) - (\delta_k^p, \nabla \cdot v) + \nu(\nabla w_k, \nabla v) &= 0, \\ (\nabla \cdot w_k, q) &= -(\nabla \cdot z_k, q). \end{aligned}$$

k.3 Set $p_k := p_{k-1} + \delta_k^p$ and then find $u_k \in X_h$ satisfying for all $v \in X_h$,

$$\xi(\nabla \cdot u_k, \nabla \cdot v) + b^*(u_{k-1}, u_k, v) + \nu(\nabla u_k, \nabla v) = (f, v) + (p_k, \nabla \cdot v).$$

This is equivalent to solving the following block-LU system [7]:

$$\begin{bmatrix} A + D & 0 \\ B & -B(\nu K + D)^{-1}B^T \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ \delta_p^k \end{bmatrix} = \begin{bmatrix} f - p^{k-1} \\ 0 \end{bmatrix}, \tag{1.16}$$

and setting $p^k = \delta_p^k + p^{k-1}$, with D corresponding to matrix arising from the grad-div stabilization term. This scheme requires the solution of two systems in A and the solution of the approximate Schur complement $B(\nu K + D)^{-1}B^T$.

The advantage of this approximation is that Schur complement is easy to precondition and solve and remains the same at each nonlinear iteration. The diagonal of the pressure mass matrix gives an optimal preconditioner for the Schur complement, ensuring that the required number of iterations to solve the system remains small, and this preconditioner need only be assembled once. Moreover, both the approximate Schur complement and the matrix $K + D$ are SPD, allowing one to use conjugate-gradient iterations for both inner and outer solves. Although this inexact LU decomposition breaks the continuity equation, we add a grad-div stabilization term in the momentum equation to force the incompressibility of the velocity field. It was shown in [7] that this scheme does not significantly increase the required number of nonlinear iterations while reducing the cost of each iteration, resulting in substantial savings.

The Newton-type variant can be obtained by replacing $b^*(u_{k-1}, z_k, v)$ with $b^*(u_{k-1}, z_k, v) + b^*(z_k, u_{k-1}, v) - b^*(u_{k-1}, u_{k-1}, v)$ at step $k1$ and a similar substitution for $k3$ (replacing z_k with u_k). In this work, we present similar schemes to Algorithm 1.3 and analyze and test them.

2 | GRAD-DIV ALGEBRAIC CHORIN–TEMAM PICARD ITERATION

It was shown in [9] that *Algebraic Chorin–Temam*-type schemes have superior stability and convergence, compared to the Yosida-type schemes (such as Algorithm 1.3 for the steady problem) due to favorable spectral properties. We therefore define the following scheme which can be regarded as the Algebraic Chorin–Temam analogue of Algorithm 1.3:

Algorithm 2.1 The grad-div stabilized Algebraic Chorin–Temam Picard iteration for the steady Navier–Stokes is defined by:
Step 1: Guess $u_0 \in X_h$ and $p_0 \in Q_h$.
Step k consists of the following four steps:
k.1 Find $z_k \in X_h$ satisfying for all $v \in X_h$,

$$\xi(\nabla \cdot z_k, \nabla \cdot v) + b^*(u_{k-1}, z_k, v) + \nu(\nabla z_k, \nabla v) = (f, v) + (p_{k-1}, \nabla \cdot v).$$

k.2 Find $(w_k, \delta_k^p) \in (X_h, Q_h)$ satisfying for all $(v, q) \in (X_h, Q_h)$,

$$\xi(\nabla \cdot w_k, \nabla \cdot v) - (\delta_k^p, \nabla \cdot v) + \nu(\nabla w_k, \nabla v) = 0,$$
$$(\nabla \cdot w_k, q) = -(\nabla \cdot z_k, q).$$

k.3 Find $u_k \in X_h$ satisfying for all $v \in X_h$,

$$\xi(\nabla \cdot u_k, \nabla \cdot v) + \nu(\nabla u_k, \nabla v) = (\delta_k^p, \nabla \cdot v) + \xi(\nabla \cdot z_k, \nabla \cdot v) + \nu(\nabla z_k, \nabla v).$$

k.4 Set $p_k := p_{k-1} + \delta_k^p$.

This yields the following discrete formulation:

$$(\nu K + C(\hat{u}_{k-1}) + \xi D)\hat{z}_k = \hat{f} + B^T \hat{p}_{k-1}, \tag{2.1}$$

$$B(\nu K + \xi D)^{-1} B^T \hat{\delta}_k^p = -B\hat{z}_k, \tag{2.2}$$

$$(\nu K + \xi D)\hat{u}_k = (\nu K + \xi D)\hat{z}_k + B^T \hat{\delta}_k^p, \tag{2.3}$$

$$\hat{p}_k = \hat{\delta}_k^p + \hat{p}_{k-1}, \tag{2.4}$$

where D is the contribution of the grad-div stabilization term. This is equivalent to solving the following block-LU system:

$$\begin{bmatrix} A+D & 0 \\ B & -B(\nu K+D)^{-1}B^T \end{bmatrix} \begin{bmatrix} I & (\nu K+D)^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ \delta_p^k \end{bmatrix} = \begin{bmatrix} f-p^{k-1} \\ 0 \end{bmatrix}, \tag{2.5}$$

and setting $p^k = \delta_p^k + p^{k-1}$. Our numerical tests (presented later) confirm faster convergence of Algorithm 2.1 compared to Algorithm 1.3. This scheme is also slightly less numerically demanding, as we require only one solve of $(\nu K + C(\hat{u}_{k-1}) + \xi D)$ compared to two for the Yosida scheme. As before, the Schur complement does not change over iterations and can be optimally preconditioned with the diagonal of the scaled pressure mass matrix, which need only be assembled once. Following our naming convention, as $H_1 = H_2$ in (2.5), we refer to this as a *Chorin–Temam* type method.

Remark To obtain the Newton version of the algorithm (presented later), we replace step k.1 in Algorithm 2.1 with: find $z_k \in X_h$ satisfying for all $v \in X_h$,

$$\xi(\nabla \cdot z_k, \nabla \cdot v) + b^*(z_k, u_{k-1}, v) + b^*(u_{k-1}, z_k, v) + \nu(\nabla z_k, \nabla v) = b^*(u_{k-1}, u_{k-1}, v) + (f, v) + (p_{k-1}, \nabla \cdot v).$$

2.1 | Convergence

We now prove the convergence of Algorithm 2.1. The proof requires a small data assumption, a sufficiently close initial guess, and the grad-div parameter $\xi \geq \nu$. In practice, we typically have $\xi = \mathcal{O}(1)$ for optimal accuracy, with $\xi = 1$ being the most common choice.

Theorem 2.1 *Assume the initial guess is good enough, in the sense that* $\nu\|\nabla(u-u_0)\|^2 + \xi^{-1}\|p-p_0\|^2 \leq \|\nabla u\|^2$ *and* $\xi \geq \nu$, *where (u, p) is the solution of (1.12) and (1.13) and* $(u_0, p_0) \in (X_h, Q_h)$ *to be the initial guess. Letting* $(u_k, p_k) \in (X_h, Q_h)$ *be the Step k solution of Algorithm 2.1, then the algorithm converges linearly to (u, p), provided the condition:*

$$\alpha < \min\{\nu(8\beta^{-2}(2\nu + 2\nu^{1/2} + 3) + 2(1+\nu^{1/2})^2)^{-1}, (16\beta^{-2}+3)^{-1}, 1\}$$

is satisfied.

Proof. Denote $e_k^u := u - u_k$ and $e_k^z := u - z_k$. Our proof will assume $\nu\|\nabla e_{k-1}^u\|^2 + \xi^{-1}\|p-p_{k-1}\|^2 \leq \|\nabla u\|^2$, and by proving that the sequence defined by $\nu\|\nabla e_{k-1}^u\|^2 + \xi^{-1}\|p-p_{k-1}\|^2$ is decreasing, this will imply the condition at the next iteration.

Subtracting step k.1 from the unique steady solution Equation (1.12), we obtain for all $v \in X_h$,

$$\xi(\nabla \cdot e_k^z, \nabla \cdot v) + \nu(\nabla e_k^z, \nabla v) = (p - p_{k-1}, \nabla \cdot v) - b^*(e_{k-1}^u, u, v) - b^*(u_{k-1}, e_k^z, v).$$

Choosing $v = e_k^z$ vanishes the last nonlinear term, and provides the bound

$$\xi\|\nabla \cdot e_k^z\|^2 + \nu\|\nabla e_k^z\|^2 \leq \xi^{-1}\|p-p_{k-1}\|^2 + \nu\alpha^2\|\nabla e_{k-1}^u\|^2, \tag{2.6}$$

thanks to the definition of α, the bound on the true solution u, Young's inequalities and (1.11).

Next, we bound $\|p - p_k\|$. Begin by adding steps 1 and 2, which gives for all $v \in X_h$,

$$\xi(\nabla \cdot (w_k + z_k), \nabla \cdot v) + \nu(\nabla(w_k + z_k), \nabla v) = (p_k, \nabla \cdot v) - b^*(u_{k-1}, z_k, v) + (f, v),$$

and we note that $(w_k + z_k) \in V_h$. Subtracting the unique steady solution Equation (1.12) from this, we obtain the error equation

$$\begin{aligned}&\xi(\nabla \cdot (w_k + z_k - u), \nabla \cdot v) + \nu(\nabla(w_k + z_k - u), \nabla v)\\ &\qquad = (p_k - p, \nabla \cdot v) - b^*(e^u_{k-1}, e^z_k, v) - b^*(u, e^z_k, v) - b^*(e^u_{k-1}, u, v) \ \ \forall v \in X_h. \end{aligned} \tag{2.7}$$

Choosing $v = (w_k + z_k - u) \in V_h$ vanishes the pressure term, and yields the bound

$$\begin{aligned}&\xi\|\nabla \cdot (w_k + z_k - u)\|^2 + \nu\|\nabla(w_k + z_k - u)\|^2\\ &\qquad \le 2M^2\nu^{-1}\|\nabla e^u_{k-1}\|^2\|\nabla e^z_k\|^2 + \nu\alpha^2\|\nabla e^z_k\|^2 + \nu\alpha^2\|\nabla e^u_{k-1}\|^2,\end{aligned} \tag{2.8}$$

thanks to (1.11), Young's inequality, and the bound on u. Using the assumption that $\nu\|\nabla e^u_{k-1}\|^2 \le \|\nabla u\|^2$, this reduces to

$$\xi\|\nabla \cdot (w_k + z_k - u)\|^2 + \nu\|\nabla(w_k + z_k - u)\|^2 \le (\nu + 2)\alpha^2\|\nabla e^z_k\|^2 + \nu\alpha^2\|\nabla e^u_{k-1}\|^2. \tag{2.9}$$

We now use this bound to bound the pressure error, after applying inf-sup to (2.8) to find

$$\begin{aligned}\beta\|p - p_k\| &\le \xi\|\nabla \cdot (w_k + z_k - u)\| + \nu\|\nabla(w_k + z_k - u)\|\\ &\quad + M\|\nabla u\|\|\nabla e^z_k\| + M\|\nabla u\|\|\nabla e^u_{k-1}\| + M\|\nabla e^u_{k-1}\|\|\nabla e^z_k\|\\ &\le \xi\|\nabla \cdot (w_k + z_k - u)\| + \nu\|\nabla(w_k + z_k - u)\| + \alpha(\nu + \nu^{1/2})\|\nabla e^z_k\| + \alpha\nu\|\nabla e^u_{k-1}\|.\end{aligned}$$

Squaring both sides, using that $\xi \ge \nu$, and reducing yields

$$\begin{aligned}\beta^2\|p - p_k\|^2 &\le 4\xi(\xi\|\nabla \cdot (w_k + z_k - u)\|^2 + \nu\|\nabla(w_k + z_k - u)\|^2)\\ &\quad + 4\xi((\nu^{1/2} + 1)^2\alpha^2\|\nabla e^z_k\|^2 + \alpha^2\nu\|\nabla e^u_{k-1}\|^2).\end{aligned}$$

Using the bound (2.11) and multiplying both sides by ξ^{-1} reduces this estimate to

$$\xi^{-1}\|p - p_k\|^2 \le 4\beta^{-2}((2\nu + 2\nu^{1/2} + 3)\alpha^2\|\nabla e^z_k\|^2 + 2\nu\alpha^2\|\nabla e^u_{k-1}\|^2). \tag{2.10}$$

Next, we use (2.10) and (2.6) to bound $\|\nabla e^u_k\|$. Adding step 1 and step 3, and then subtracting this from (1.12) obtains

$$\xi(\nabla \cdot e^u_k, \nabla \cdot v) + \nu(\nabla e^u_k, \nabla v) = (p - p_k, \nabla \cdot v) - b^*(e^u_{k-1}, u, v) - b^*(u, e^z_k, v) - b^*(e^u_{k-1}, e^z_k, v).$$

Letting $v = e^u_k$ and applying Cauchy-Schwarz inequality, (1.11) and assumption $\|\nabla e^u_{k-1}\|^2 \le \nu^{-1}\|\nabla u\|^2$ produces

$$\xi\|\nabla \cdot e^u_k\|^2 + \nu\|\nabla e^u_k\|^2 \le \|p - p_k\|\|\nabla \cdot e^u_k\| + (\nu\alpha\|\nabla e^u_{k-1}\| + (\nu + \nu^{1/2})\alpha\|\nabla e^z_k\|)\|\nabla e^u_k\|.$$

Applying Young's inequality yields

$$\xi\|\nabla \cdot e^u_k\|^2 + \nu\|\nabla e^u_k\|^2 \le \xi^{-1}\|p - p_k\|^2 + 2\nu\alpha^2\|\nabla e^u_{k-1}\|^2 + 2(1 + \nu^{1/2})^2\alpha^2\|\nabla e^z_k\|^2. \tag{2.11}$$

Adding this to (2.10) and combining with (2.6) and (2.10), we obtain

$$\begin{aligned}&\xi^{-1}\|p - p_k\|^2 + \nu\|\nabla e^u_k\|^2\\ &\quad \le (2(1 + \nu^{1/2})^2 + 8\beta^{-2}(2\nu + 2\nu^{1/2} + 3))\alpha^2\nu^{-1}\xi^{-1}\|p - p_{k-1}\|^2\\ &\quad\quad + ((2(1 + \nu^{1/2})^2 + 8\beta^{-2}(2\nu + 2\nu^{1/2} + 3))\alpha^2\nu^{-1} + 2 + 16\beta^{-2})\alpha^2\nu\|\nabla e^u_{k-1}\|^2.\end{aligned}$$

Applying the small data condition $\alpha < \min\{\nu(8\beta^{-2}(2\nu + 2\nu^{1/2} + 3) + 2(1 + \nu^{1/2})^2)^{-1}, (16\beta^{-2} + 3)^{-1}, 1\}$, we find that

$$\begin{aligned}\xi^{-1}\|p - p_k\|^2 + \nu\|\nabla e^u_k\|^2 &\le \alpha\xi^{-1}\|p - p_{k-1}\|^2 + (\alpha + 16\beta^{-2} + 2)\alpha^2\nu\|\nabla e^u_{k-1}\|^2\\ &\le \alpha(\xi^{-1}\|p - p_{k-1}\|^2 + \nu\|\nabla e^u_{k-1}\|^2).\end{aligned}$$

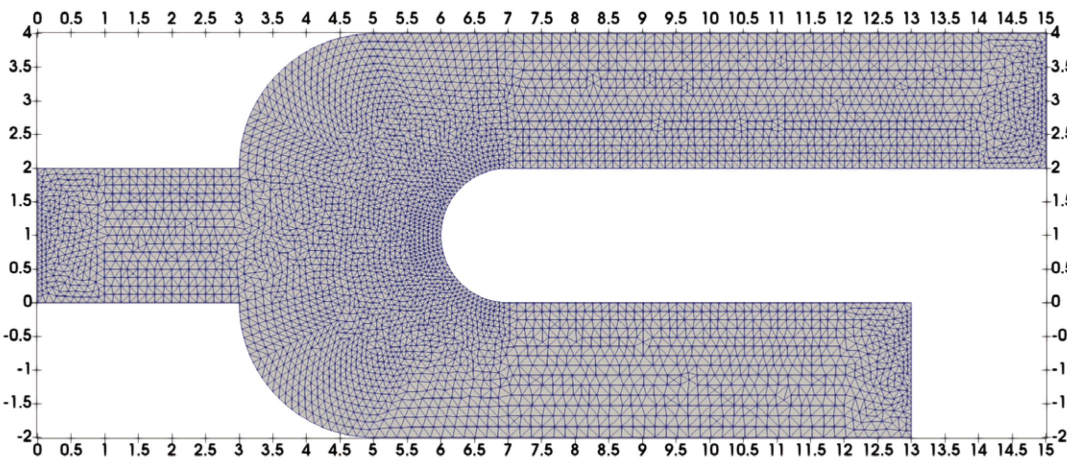

FIGURE 1 Bifurcation geometry [Color figure can be viewed at wileyonlinelibrary.com]

We have thus proven that $\xi^{-1}\|p - p_k\|^2 + \nu\|\nabla e_k^u\|^2$ is a contractive sequence in k, and hence it converges. Since the solution of the (finite dimensional) problem (1.12) and (1.13) is unique and bounded by the data, we have that the limit of the incremental Picard–Yosida iteration converges linearly to the solution of (1.12) and (1.13). ■

3 | NUMERICAL TESTS FOR PICARD–CHORIN–TEMAM SCHEMES

In this section, we present two numerical tests: a two-dimensional (2D) flow in a bifurcated domain and three-dimensional (3D) flow in a coronary artery. Here we want to show several things: that our incremental method converges linearly to the solution of discrete steady Navier–Stokes system (1.12) and (1.13), that the convergence rate is similar to standard Picard, and that the solution method outperforms or is competitive with standard Picard in terms of efficiency.

3.1 | 2D bifurcation flow

We first test our proposed Algorithm 2.1 by solving the 2D steady Navier–Stokes problem in the same bifurcated domain shown in Figure 1. For the discretization, we employ TH $\mathbb{P}^2/\mathbb{P}^1$ elements on a fine mesh of 8,988 elements ($h = .05$), leading to 41,589 total degrees of freedom (DOF). We prescribe a parabolic inflow profile with peak velocity 2.0 and assign traction-free boundary conditions at both outflows. We solve for $\nu = .0133$ and $\nu = .0067$ for $\xi = 1$ and $\xi = 2$ using the software FreeFEM++ on a 2017 MacBook Pro.

We compare the solutions computed by Algorithm 1.3 (iPY) and Algorithm 2.1 (GISACT) to the reference solution from the standard Picard iterations up to a very high level of accuracy (1e-12). To ensure the best possible accuracy, we solved the full saddle-point system with UMFPACK at each iteration to compute the reference solution. For iPY and GISACT, we solve both velocity systems with UMFPACK (step k.1 and step k.2), and the Schur complement with CG preconditioned by the lumped pressure mass-matrix for outer solve and UMFPACK for inner solve.

For the purposes of comparison, we also computed the solution for the same problem configuration with a standard Picard scheme (1.1). At each iteration we solved the full saddle-point system with

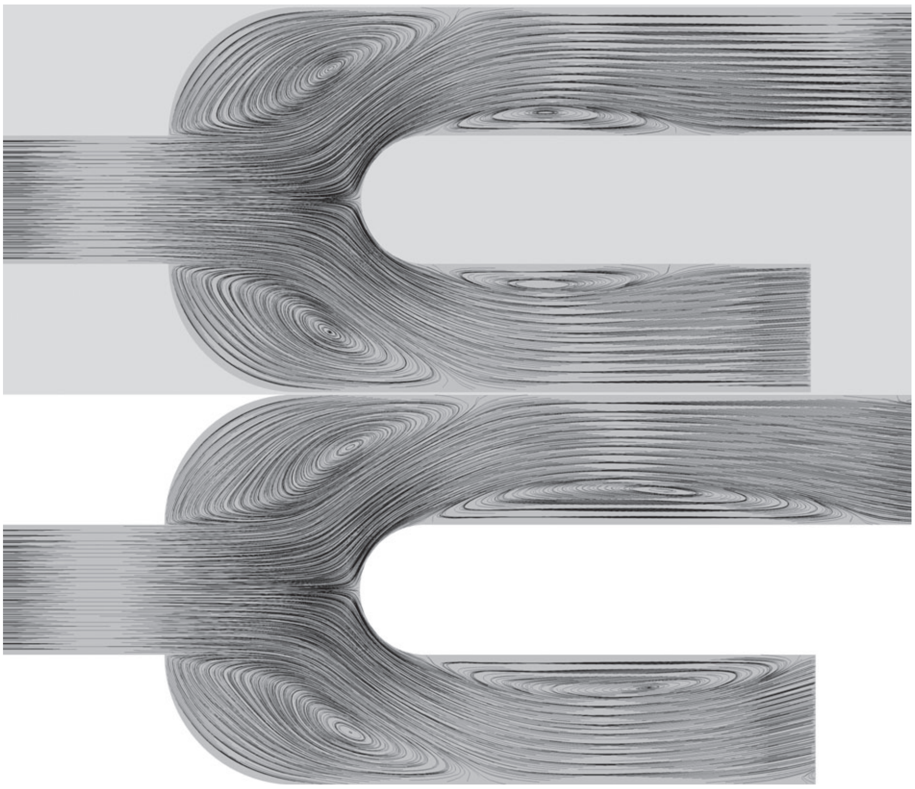

FIGURE 2 Bifurcation test streamlines; $\nu = .0133$ (top) and $\nu = .0067$ (bottom)

GMRES preconditioned with the following block-triangular preconditioner found in [17]:

$$P^{-1} = \begin{bmatrix} A_\xi & B^T \\ 0 & -(\nu + \xi)^{-1} M_p \end{bmatrix}^{-1} \tag{3.1}$$

where M_p is the lumped pressure mass matrix and A_ξ the velocity block with grad-div stabilization. M_p is simply a diagonal matrix inversion and we solve A_ξ with UMFPACK. We set the outer solve tolerance to 1e-6.

For $\nu = .0133$ (Figure 3), we observe a similar rate of convergence for standard Picard and GISACT. iPY converges more slowly, with $\xi = 1$ being noncompetitive and $\xi = 2$ converging faster but still more slowly than standard Picard or GISACT. For $\nu = .0067$ (Figure 4), we see that iPY with $\xi = 1$ is again not competitive, converging very slowly. However, GISACT with both values of ξ and iPY with $\xi = 2$ are both comparable or slightly superior to standard Picard. Although we varied the value of ξ for standard Picard as well, we found that it did not impact the convergence rate significantly in this case. For the sake of clarity, we display only the convergence of standard Picard with $\xi = 2$ in Figures 3 and 4.

We note that in the figures above that although the convergence is rate of iPY and GISACT shows a linear trend, it is not always monotonic and some oscillations may be present. This phenomenon appears more pronounced for iPY but we observe it for GISACT as well. Increasing ξ seems to reduce its effect. Note that standard Picard does not show this behavior. Although we are not certain as to what causes this, the fact that increasing ξ reduces its effect suggests it may be related to mass conservation. This would also explain why it affects iPY more than GISACT, as GISACT globally mass-conservative while iPY is not. This phenomenon requires further investigation to understand properly.

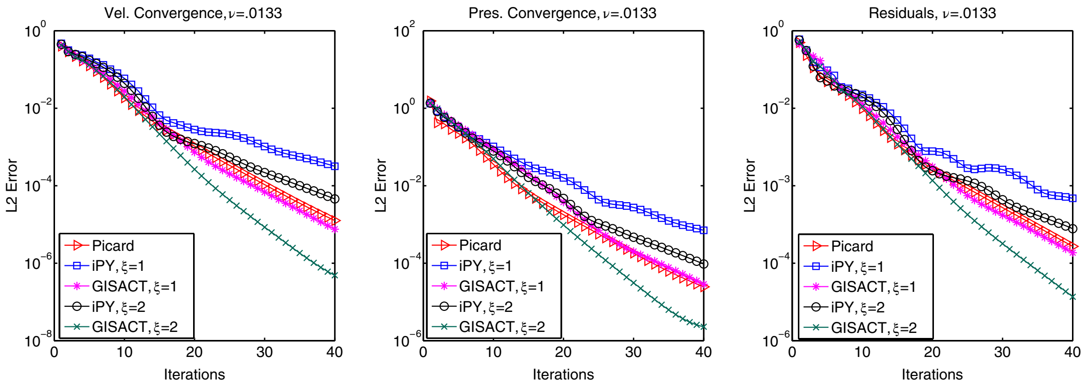

FIGURE 3 Comparison of convergence for test case 1, $\nu = .0133$ [Color figure can be viewed at wileyonlinelibrary.com]

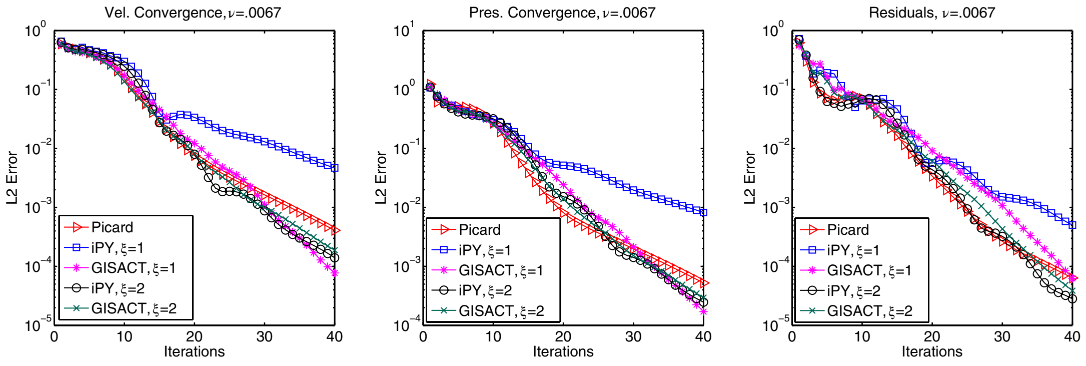

FIGURE 4 Comparison of convergence for test case 1, $\nu = .0067$ [Color figure can be viewed at wileyonlinelibrary.com]

In Table 1 we provide information regarding the numerical cost of each nonlinear iteration. Here, outer Krylov iterations refers to the number of outer GMRES iterations for standard Picard and the number of outer preconditioned conjugate-gradient (PCG) iterations for the approximate Schur complement solve for iPY and GISACT. The average solve time is the time required to solve the systems at each nonlinear iteration, and does not factor in the assembly costs (similar across all algorithms and therefore excluded to more clearly show the differences between methods). For all three methods, the number of outer iterations does not appear to change significantly with ν. Standard Picard shows a mild but observable dependence on ξ for its required outer iterations, but both iPY and GISACT appear less sensitive in this regard. Overall, the splitting schemes require fewer outer iterations per solve, resulting in clear computational savings at each step.

Overall, from this test we conclude that the convergence of GISACT is roughly the same, or slightly faster, than standard Picard for both values of ξ. iPY is somewhat slower than both Picard and GISACT for $\xi = 1$, but performs comparably for $\xi = 2$. Changing ξ appears to affect the convergence of GISACT somewhat, but on the whole it is less sensitive to ξ than iPY. Like the convergence oscillations discussed earlier, we believe that this reduced sensitivity to ξ for GISACT may be related to mass conservation. As GISACT satisfies the continuity equation, it is mass conservative by construction; however since iPY does not satisfy the continuity equation, the enforcement of mass conservation comes from the grad-div penalization and therefore depends on ξ. This behavior may also be caused by the small data assumption no longer holding. In terms of cost per iteration, we find both iPY and GISACT to be cheaper than standard Picard, resulting in around 50% savings in solve time.

TABLE 1 Iteration statistics for 2D bifurcation test

	GISACT		iPY		Standard Picard	
ξ	Outer Krylov iter.	Avg. solve time	Outer Krylov iter.	Avg. solve time	Outer Krylov iter.	Avg. solve time
2D bifurcation: $\nu = .0133$ ($Re = 200$)						
1.0	11	.600 s	11	.594 s	30	1.346 s
2.0	10	.549 s	10	.552 s	25	1.134 s
2D bifurcation: $\nu = .00667$ ($Re = 400$)						
1.0	11	.603 s	11	.595 s	30	1.409 s
2.0	11	.598 s	11	.593 s	25	1.146 s

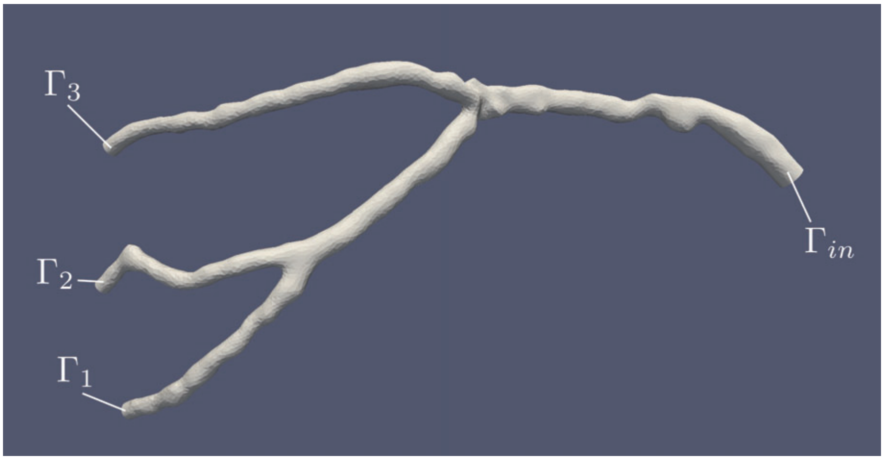

FIGURE 5 The geometry for test case 3.2. Γ_{in} is the inlet and Γ_1, Γ_2, and Γ_3 are outlets. Our variable of interest is the pressure value at Γ_3 [Color figure can be viewed at wileyonlinelibrary.com]

3.2 | 3D test case: flow in a coronary artery

We next test our method on a reconstructed coronary artery taken from the left circumflex branch of an actual patient. The goal of this simulation is to compute the pressure distal to the stenosis at the outlet indicated as Γ_3 in Figure 5. The input data is the pressure at the inlet Γ_{in}, measured to be 92 mmHg. The reconstructed mesh consists of 50,165 tetrahedra and is pictured in Figure 5. We use iterative linear solvers to solve each linear system in this test, as problems of this type are often too large to employ a direct solver. This test is intended to demonstrate the method's applicability for realistic problems of practical interest and to show that it remains effective when one employs iterative solvers instead of direct approaches.

We run our simulation with $\mathbb{P}^2/\mathbb{P}^1$ TH elements on a moderately fine mesh with 50,165 tetrahedra. We set the kinematic viscosity $\nu = .033\,\text{cm}^2/\text{s}$,[2] based on the values found in [18]. At the inflow,

[2]Note that the true physical parameters are actually a fluid density of 1.06 g/cm^3 and dynamic viscosity of .035 g/cm-s. From a simulation standpoint this does not make a practical difference; however one must multiply the pressure by the density to recover the correct units. Alternatively, one may avoid the need for scaling by setting $\nu = .035$ and multiplying the trilinear form b^* by $\rho = 1.06$.

TABLE 2 Iteration statistics for 3D coronary flow test

GISACT				Standard Picard		
Nonlinear iter.	Avg. outer SC iter.	Avg. inner SC iter.	Avg. solve time	Nonlinear iter.	Avg. outer GMRES iter.	Avg. solve time
34	51	10	70.1 s	28	1816	216.7 s

we prescribe a parabolic Poiseulle profile with a flow rate of 1.75 mL/s, estimated based on literature values for coronary arteries [19, 20], resulting in a Reynolds number of approximately 250. As our parameter of interest is the outlet pressure, we assign the outlet boundary conditions using the minimization method found in [21]. We set $\xi = 2$ and end our computation when the difference in L^2 norm of velocity between consecutive iterations falls below 1e-3.

At each step, we solve the initial velocity step using GMRES with an ILU(1) preconditioner and a stopping tolerance of 1e-7 and the velocity correction step with Jacobi-preconditioned CG. To solve the Schur complement, we use PCG preconditioned with the lumped pressure mass matrix for the outer solves and CG preconditioned with ILU(1) for the inner solves. We used a stopping tolerance of 1e-3 for the inner solves and 1e-7 for the outer solve. All computations were performed on a 2017 MacBook pro using the finite element software package FEniCS, with the linear systems solved using the PETSc linear algebra library [22].

For the purposes of comparison, we also computed the solution for the same problem configuration with a standard Picard scheme (1.1). At each iteration we solved the full saddle-point system with GMRES using a preconditioner similar to (3.1); however instead of solving A_ξ directly, we now approximate with ILU(2). While replacing A_ξ with an approximation increased the number of outer iterations, the cheaper cost of each iteration made this approach faster than solving A_ξ iteratively (even for high stopping tolerances).

We found that our method performed favorably in comparison to standard Picard. We report performance statistics in Table 2. Though GISACT required slightly more nonlinear iterations to reach convergence (28 compared to 35), this is offset by the reduction in cost of each iteration. One GISACT iteration took an average of 70.1 s compared to 216.7 s for standard Picard. This resulted in GISACT requiring only 40% as much time as the comparison.

In terms of accuracy, the solutions were identical and each computed a distal pressure of 82 mmHg, in good agreement with the measured value of 84 mmHg. We note that 2 mmHg is well within expected variability for blood pressure across different measurement times and methods from the available medical literature [23–25]. We show the pressure gradients of the computed solutions in Figure 6.

4 | GRAD-DIV ALGEBRAIC CHORIN–TEMAM NEWTON ITERATION

We now present and study a higher order of Algebraic Chorin–Temam iteration for steady Navier–Stokes system, which is defined as follows.

Algorithm 4.1 The higher order algebraic Chorin–Temam iteration for the steady Navier–Stokes is given by*:*
Step 1: Guess $u_0 \in X_h$, $p_0 \in Q_h$.
Step k consists of the following four steps:
k.1 Find $z_k \in X_h$ satisfying for all $v \in X_h$,

$$\xi(\nabla \cdot z_k, \nabla \cdot v) + b^*(u_{k-1}, z_k, v) + b^*(z_k, u_{k-1}, v) + \nu(\nabla z_k, \nabla v) = (f, v) + (p_{k-1}, \nabla \cdot v) + b^*(u_{k-1}, u_{k-1}, v).$$

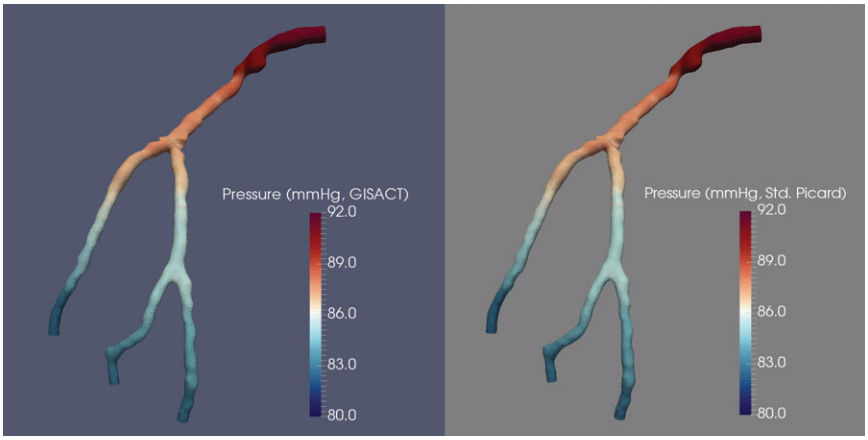

FIGURE 6 Computed pressure profiles for the 3D coronary flow test; GISACT solution (left) and standard Picard (right) [Color figure can be viewed at wileyonlinelibrary.com]

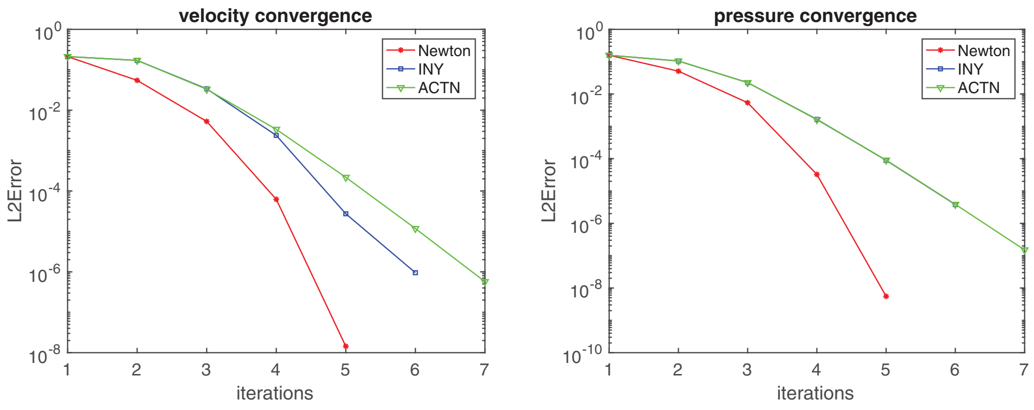

FIGURE 7 Shown above are the velocity and pressure convergence of three methods (Newton's iteration (red), incremental Newton–Yosida (blue), algebraic Chorin–Temam Newton (green)) [Color figure can be viewed at wileyonlinelibrary.com]

k.2 Find $(w_k, \delta_k^p) \in (X_h, Q_h)$ satisfying for all (v, q) ∈ (X$_h$, Q$_h$),

$$\xi(\nabla \cdot w_k, \nabla \cdot v) - (\delta_k^p, \nabla \cdot v) + \nu(\nabla w_k, \nabla v) = 0,$$
$$(\nabla \cdot w_k, q) = -(\nabla \cdot z_k, q).$$

k.3 Find u$_k$ ∈ X$_h$ satisfying for all v ∈ X$_h$,

$$\xi(\nabla \cdot u_k, \nabla \cdot v) + \nu(\nabla u_k, \nabla v) = (\delta_k^p, \nabla \cdot v) + \xi(\nabla \cdot z_k, \nabla \cdot v) + \nu(\nabla z_k, \nabla v),$$

k.4 Set $p_k = p_{k-1} + \delta_k^p$.

The standard Newton method for steady NSE, converges and has unique solution under the condition $\widetilde{\alpha} = (1+\varepsilon)M\nu^{-1}\|f\|_{-1} < 1$, see [7]. Here we state a theorem that the Algorithm 4.1 converges to the solution of (1.12) and (1.13) under a more restrictive condition.

Theorem 4.1 *Let ε > 0 and define*

$$\widetilde{\alpha} < \min\{1,\ (9+10\beta^{-2})^{-1},\ \nu^{-1}(1+\varepsilon)^2\|f\|_{-1}^2(2(1+12\beta^{-2}) + (8+80\beta^{-2})^{-1})^{-1}\}.$$

Denote by (u, p) *the solution of system (1.12) and (1.13),* (u$_0$, p$_0$) ∈ (X$_h$, Q$_h$) *the initial guess of Algorithm 4.1, and* (u$_k$, p$_k$) *the step* k *solution. Then if* $\xi \geq \nu$ *and* $\nu\|\nabla(\mathrm{u}-\mathrm{u}_0)\|^2+\xi^{-1}\|\mathrm{p}-\mathrm{p}_0\|^2 \leq \|\nabla \mathrm{u}\|^2$, *the sequence* (u$_k$, p$_k$) *converges to* (u, p).

Remark 4.2 Even though Newton's method converges quadratically, we would not expect Algorithm 4.1 to converge quadratically, since approximations are being made. From the proof, in particular (4.10), observe that if the pressure terms are small, then quadratic convergence of the velocity is recovered.

Proof. We begin the proof by giving one assumption that the sequence $\{u-u_k\}$ is bounded by $\min\{\nu^{-1/2}, \varepsilon\|\nabla u\|\}$ for all $k\in\mathbb{N}$, where ε is the same constant used in the definition of $\widetilde{\alpha}$. Hence for all k, we have

$$\|\nabla u_{k-1}\| \leq \|\nabla(u-u_{k-1})\| + \|\nabla u\| \leq (1+\varepsilon)\nu^{-1}\|f\|_{-1}. \tag{4.1}$$

by using the triangle inequality, and the upper bound of true solution u.

Also, we assume that

$$\nu\|\nabla(u-u_{k-1})\|^2+\xi^{-1}\|p-p_{k-1}\|^2 \leq \|\nabla u\|^2, \tag{4.2}$$

and by proving that the sequence defined by $\nu\|\nabla(u-u_{k-1})\|^2+\xi^{-1}\|p-p_{k-1}\|^2$ is decreasing, this will imply the condition at the next iteration.

Denote $e^u_k := u-u_k$ and $e^z_k := u-z_k$. Subtracting step k.1 of Algorithm 4.1 from the unique steady solution Equation (1.12), we obtain for all $v\in X_h$,

$$\begin{aligned}&\xi(\nabla\cdot e^z_k, \nabla\cdot v)+\nu(\nabla e^z_k, \nabla v)\\ &= (p-p_{k-1}, \nabla\cdot v) - b^*(e^u_{k-1}, e^u_{k-1}, v) - b^*(e^z_k, u_{k-1}, v) - b^*(u_{k-1}, e^z_k, v).\end{aligned}$$

Choosing $v=e^z_k$ vanishes the last term, we get

$$\xi\|\nabla\cdot e^z_k\|^2+\nu(1-\widetilde{\alpha})\|\nabla e^z_k\|^2 \leq \xi^{-1}\|p-p_{k-1}\|^2+\frac{M^2}{\nu(1-\widetilde{\alpha})}\|\nabla e^u_{k-1}\|^4, \tag{4.3}$$

thanks to Young's inequality, (4.1), and the definition of $\widetilde{\alpha}$.

Next, we give a bound of $\|p-p_k\|$. Begin by adding step k.1 and step k.2, and subtracting it from the unique steady solution Equation (1.12). We then obtain the error equation for all $v\in X_h$,

$$\begin{aligned}&\xi(\nabla\cdot(z_k+w_k-u), \nabla\cdot v)+\nu(\nabla(z_k+w_k-u), \nabla v)\\ &= (p_k-p, \nabla\cdot v)+b^*(e^u_{k-1}, e^u_{k-1}, v)+b^*(u_{k-1}, e^z_k, v)+b^*(e^z_k, u_{k-1}, v).\end{aligned} \tag{4.4}$$

Choosing $v=z_k+w_k-u\in V_h$ vanishes the pressure term and yields the bound

$$\xi\|\nabla\cdot(z_k+w_k-u)\|^2+\nu\|\nabla(z_k+w_k-u)\|^2 \leq 8\nu\widetilde{\alpha}^2\|\nabla e^z_k\|^2+2M^2\nu^{-1}\|\nabla e^u_{k-1}\|^4. \tag{4.5}$$

thanks to Young's inequality, (4.1), and the definition of α.

Applying the inf-sup condition to (5.19) gives

$$\beta\|p_k-p\| \leq \xi\|\nabla\cdot(z_k+w_k-u)\|+\nu\|\nabla(z_k+w_k-u)\|+M\|\nabla e^u_{k-1}\|^2+2\nu\widetilde{\alpha}\|\nabla e^z_k\|.$$

Squaring both sides, using that $\xi\geq\nu$, and reducing yields

$$\begin{aligned}\beta^2\|p_k-p\|^2 \leq 4\xi(&\xi\|\nabla\cdot(z_k+w_k-u)\|^2+\nu\|\nabla(z_k+w_k-u)\|^2\\ &+\nu^{-1}M^2\|\nabla e^u_{k-1}\|^4+2\nu\widetilde{\alpha}^2\|\nabla e^z_k\|^2).\end{aligned}$$

Applying the bound (5.20) reduces this estimate to

$$\|p - p_k\|^2 \le 4\xi\beta^{-2}(10\nu\widetilde{\alpha}^2\|\nabla e_k^z\|^2 + 3\nu^{-1}M^2\|\nabla e_{k-1}^u\|^4). \tag{4.6}$$

Combining (5.17) and (4.6) and multiplying both sides by ξ^{-1} produces

$$\xi^{-1}\|p - p_k\|^2 \le 4\beta^{-2}\left(\frac{10\widetilde{\alpha}^2}{1-\widetilde{\alpha}}\xi^{-1}\|p - p_{k-1}\|^2 + \nu^{-1}M^2\left(3 + \frac{10\widetilde{\alpha}^2}{(1-\widetilde{\alpha})^2}\right)\|\nabla e_{k-1}^u\|^4\right). \tag{4.7}$$

Now we are going to use (4.7) and (5.17) to bound $\|e_k^u\|$. Adding step k.3 and step k.1 and then subtracting from (1.12), we have

$$\begin{aligned}\xi(\nabla\cdot e_k^u, \nabla\cdot v) + \nu(\nabla e_k^u, \nabla v) = (p - p_k, \nabla\cdot v) - b^*(e_{k-1}^u, e_{k-1}^u, v)\\ - b^*(e_k^z, u_{k-1}, v) - b^*(u_{k-1}, e_k^z, v).\end{aligned} \tag{4.8}$$

Choosing $v = e_k^u$ yields

$$\xi\|\nabla\cdot e_k^u\|^2 + \nu\|\nabla e_k^u\|^2 \le \xi^{-1}\|p - p_k\|^2 + 2\nu^{-1}M^2\|\nabla e_{k-1}^u\|^4 + 8\widetilde{\alpha}^2\nu\|\nabla e_k^z\|^2, \tag{4.9}$$

thanks to Young's inequality, (4.1) and the definition of $\widetilde{\alpha}$.

Adding this bound with (4.7) gives

$$\begin{aligned}\xi^{-1}\|p - p_k\|^2 + \nu\|\nabla e_k^u\|^2 \le{}& \frac{8\widetilde{\alpha}^2}{1-\widetilde{\alpha}}(10\beta^{-2}+1)\xi^{-1}\|p - p_{k-1}\|^2\\ &+ \left(2\nu^{-2}M^2(12\beta^{-2}+1) + \frac{8\widetilde{\alpha}^2M^2}{\nu^2(1-\widetilde{\alpha})^2}(10\beta^{-2}+1)\right)\nu\|\nabla e_{k-1}^u\|^4.\end{aligned}$$

Using the assumptions $\widetilde{\alpha} < (9 + 10\beta^{-2})^{-1}$ and $\widetilde{\alpha} < \nu^{-1}(1+\varepsilon)^2\|f\|_{-1}^2$ $(2(1+12\beta^{-2}) + (8+80\beta^{-2})^{-1})^{-1}$,

$$\begin{aligned}\xi^{-1}\|p - p_k\|^2 + \nu\|\nabla e_k^u\|^2 &\le \widetilde{\alpha}\xi^{-1}\|p - p_{k-1}\|^2 + \left(2(12\beta^{-2}+1) + \frac{\widetilde{\alpha}}{1-\widetilde{\alpha}}\right)\frac{M^2}{\nu^3}\nu^2\|\nabla e_{k-1}^u\|^4\\ &\le \widetilde{\alpha}\xi^{-1}\|p - p_{k-1}\|^2\\ &\quad + (2(12\beta^{-2}+1) + (8+80\beta^{-2})^{-1})\frac{\widetilde{\alpha}^2\nu}{\|f\|_{-1}^2(1+\varepsilon)^2}\nu^2\|\nabla e_{k-1}^u\|^4\\ &\le \widetilde{\alpha}(\xi^{-1}\|p - p_{k-1}\|^2 + \nu^2\|\nabla e_{k-1}^u\|^4).\end{aligned} \tag{4.10}$$

By (4.2), we then have $\nu\|\nabla(u - u_k)\|^2 \le 1$. We have therefore proved that $\nu\|\nabla(u - u_{k-1})\|^2 + \xi^{-1}\|p - p_{k-1}\|^2$ is a contractive sequence in k, and thus converges. Since the solution of the problem (1.12) and (1.13) is unique and bounded by the data, we have that the limit of Algorithm 4.1 converges to the solution of (1.12) and (1.13). ■

4.1 | Numerical test: 3D lid driven cavity

We now compare the convergence of three algorithms: the usual Newton iterations, the incremental Newton–Yosida (INY, proposed in [7]), and our Algorithm 4.1 (ACTN). We test using the 3D lid driven cavity problem on a barycenter mesh with 413,748 DOF, with SV elements (P_3, P_2^{dc}), for a Reynolds number of 100 and a grad-div stabilization parameter $\xi = 1$.

Although Newton's iteration converges slightly faster than INY and ACTN, the computational cost of INY and ACTN is much lower. From Table 3, both INY and ACTN methods have an average linear solve time of around 20 s while standard Newton requires 2,000 s. Furthermore, standard Newton fails when using a fine mesh (e.g. barycenter mesh with 1,593,444 DOF) due to memory limits. However, the ACTN does not have this problem.

TABLE 3 Iteration times for 3D lid-driven cavity test

Method	Newton	INY	ACTN
Avg. solve time	$1.7623e+3$	$1.7291e+01$	$1.6395e+01$

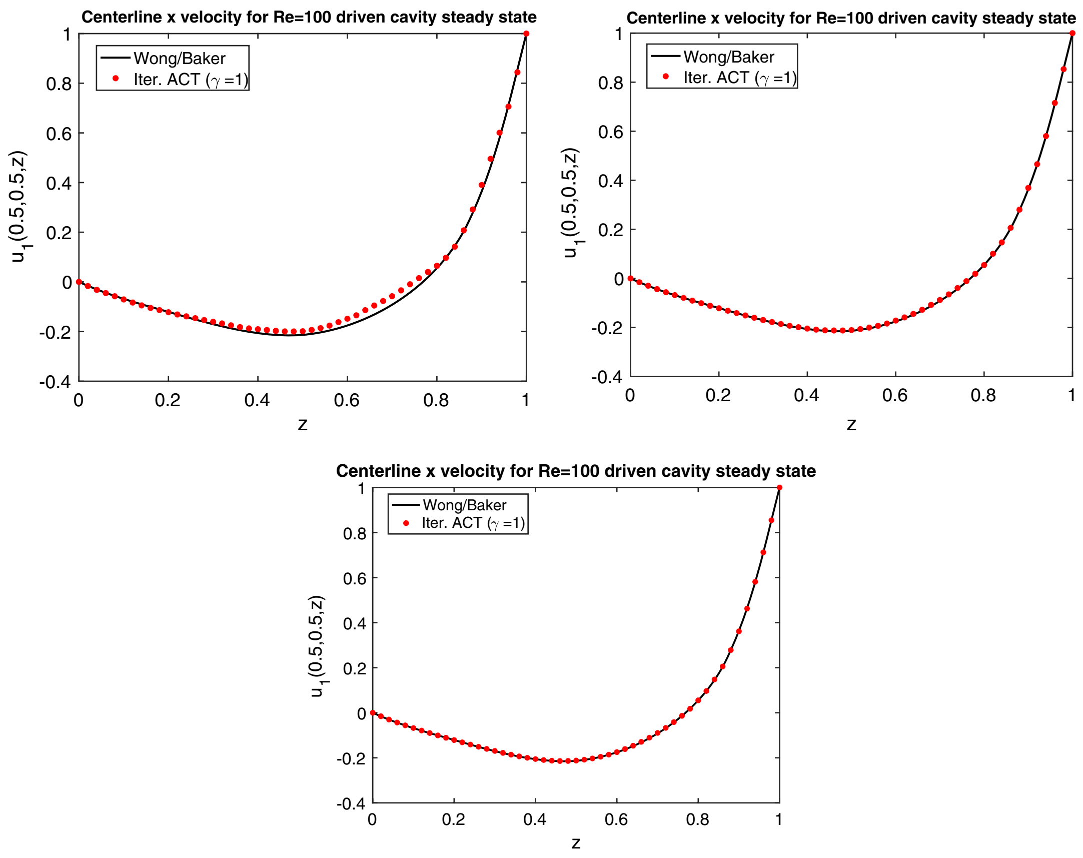

FIGURE 8 Shown above is the centerline x-velocity for the algebraic Chorin–Temam Newton solution with $Re=100$, $\xi=1$, found using Scott–Vogelius elements with three different mesh levels: 33540 DOF (top-left), 413,748 DOF (top-right), 1,593,444 DOF (bottom) [Color figure can be viewed at wileyonlinelibrary.com]

Figure 8 shows the centerline x-velocity of computed solution (red) and the reference solution (black) from [26] on different mesh levels (DOF = 33,540, 413,748, 1,593,444). For a coarse mesh, the solution from Algorithm 4.1 is slightly off in the middle. This problem is fixed when using a finer mesh. Solutions on the moderate and fine mesh levels align with the literature results very well. Figure 9 are the centerplane slices of the velocity field of ACTN solution on a mesh with 413,748 DOF using SV elements. It matches the results from [26] well.

5 | APPROXIMATION OF THE SCHUR COMPLEMENT

We can improve the numerical efficiency of the scheme in Algorithm 2.1 by several orders of magnitude if we use an approximated Schur Complement, as we will show below. As seen in [17], at the

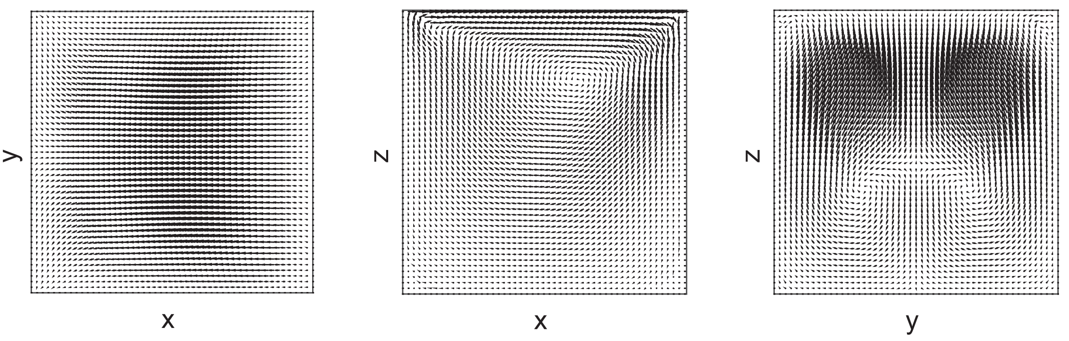

FIGURE 9 Shown above are centerplane slices of the velocity field for the algebraic Chorin–Temann Newton solution with $Re = 100$, $\xi = 1$, found using Scott–Vogelius elements and 413,748 total degrees of freedom. These plots are in good agreement with those found in the literature [26]

discrete level, the matrix ξD has the following form, where M_p denotes the pressure-mass matrix and R is some remainder:

$$\xi D = \xi B^T M_p^{-1} B + \xi h R. \tag{5.1}$$

We can therefore regard ξD as a very close approximation to $\xi B^T M_p^{-1} B$ when h is small enough. We will make use of the following lemma:

Lemma 5.1 *$B(\nu K + \xi D)^{-1} B^T$ can be expressed as $B(\nu K + \xi B^T M_p B)^{-1} B^T + \mathcal{O}(h)$.*

Proof. Begin by recalling the following formula for two invertible matrices X and Y (see for example, . [27]):

$$(X + Y)^{-1} = X^{-1} - X^{-1} Y (I + X^{-1} Y)^{-1} X^{-1} \tag{5.2}$$

By (5.1):

$$(\nu K + \xi D)^{-1} = (\nu K + \xi B^T M_p^{-1} B + \xi h R)^{-1} \tag{5.3}$$

Applying formula (5.2) with $X = \nu K + \xi B^T M_p^{-1} B$ and $Y = \xi\ hR$:

$$\begin{aligned}(\nu K + \xi D)^{-1} &= (\nu K + \xi B^T M_p^{-1} B)^{-1} - h\xi(\nu K + \xi B^T M_p^{-1} B)^{-1} R (I + h\xi(\nu K + \xi B^T M_p^{-1} B)^{-1} R)^{-1} \\ &\quad \times (\nu K + \xi B^T M_p^{-1} B)^{-1}\end{aligned} \tag{5.4}$$

Call the second term in (5.4) H. We will analyze the growth order of H with respect to the parameters ξ, ν, and h. Clearly:

$$(\nu K + \xi B M_p^{-1} B)^{-1} \sim \mathcal{O}((\nu + \xi)^{-1}) \tag{5.5}$$

From (5.5) we then have:

$$\begin{aligned} H &\sim \mathcal{O}(h\xi(\nu + \xi)^{-1}(1 + h\xi(\nu + \xi)^{-1})^{-1}(\nu + \xi)^{-1}) \\ &\sim \mathcal{O}\left(\frac{h\xi}{(\nu + \xi)^2}\left(1 + \frac{h\xi}{\nu + \xi}\right)^{-1}\right) \\ &\sim \mathcal{O}\left(\frac{h\xi}{(\nu + \xi)^2}\frac{\nu + \xi}{\nu + \xi + h\xi}\right) \\ &\sim \mathcal{O}\left(\frac{h\xi}{(\nu + \xi)(\nu + \xi + h\xi)}\right) \end{aligned} \tag{5.6}$$

We recall now that $0 < \nu < \xi$ by hypothesis and therefore we assume:

$$\mathcal{O}((\nu+\xi)^{-1}) \sim \mathcal{O}(\xi^{-1}) \tag{5.7}$$

And so the last line of (5.6) becomes:

$$\begin{aligned} H &\sim \mathcal{O}\left(\frac{h\xi}{\xi^2 + h\xi^2}\right) \\ &\sim \mathcal{O}\left(\frac{h}{(1+h)\xi}\right) \\ &\sim \mathcal{O}(h/\xi) \end{aligned} \tag{5.8}$$

Left and right multiply (5.4) by B and B^T, respectively to obtain:

$$B(\nu K + \xi B^T M_p^{-1} B)^{-1} B^T + BHB^T \tag{5.9}$$

which together with (5.8(gives:

$$B(\nu K + \xi B^T M_p^{-1} B)^{-1} B^T + \mathcal{O}(h/\xi) \tag{5.10}$$

completing the proof. This suggests that we must have $\xi \gg h$ for $B(\nu K + \xi B^T M_p^{-1} B)^{-1} B^T$ to be an effective approximation of $B(\nu K + \xi D)^{-1} B^T$; for $\xi \sim \mathcal{O}(h)$, this approximation becomes less reliable as the other term is no longer small. However, throughout this work we take $\xi \sim \mathcal{O}(1)$, in which case this is not an issue. ■

We will also use this identity (see [1]):

$$(B(\nu K + \xi B^T M_p^{-1} B)^{-1} B^T)^{-1} = \nu (BK^{-1}B^T)^{-1} + \xi M_p^{-1}. \tag{5.11}$$

It is well-known ([1, 2]) that the matrix $BK^{-1}B^T$ is spectrally equivalent to M_p. Based on this fact and the preceding lemma, we propose to use the following approximation for the Schur complement:

$$(B(\nu K + \xi D)^{-1} B^T)^{-1} \approx (\nu + \xi) M_p^{-1}, \tag{5.12}$$

where M_p is lumped. This approximation replaces the inverse of the Schur complement with a diagonal matrix, which can be solved easily using direct solves even for very large system. Hence the systems (2.1)–(2.4) become:

$$(\nu K + C(\hat{u}_{k-1}) + \xi D)\hat{z}_k = \hat{f} + B^T \hat{p}_{k-1}, \tag{5.13}$$

$$(\nu + \xi)^{-1} M_p \hat{\delta}_k^p = -B\hat{z}_k, \tag{5.14}$$

$$(\nu K + \xi D)\hat{u}_k = (\nu K + \xi D)\hat{z}_k + B^T \hat{\delta}_k^p, \tag{5.15}$$

$$\hat{p}_k = \hat{\delta}_k^p + \hat{p}_{k-1}. \tag{5.16}$$

it works for both Picard's Algorithm 2.1 and Newton's Algorithm 4.1 with different $C(\hat{u}_{k-1})$.

Before we continue, let's first take a closely look at the new system by converting it back to finite element setup. (5.13)–(5.16) is equivalent to: find $z_k \in X_h$ such that for any $v \in X_h$

$$\nu(\nabla z_k, \nabla v) + \xi(\nabla \cdot z_k, \nabla \cdot v) + b^*(u_{k-1}, z_k, v) = (f, v) + (p_{k-1}, \nabla \cdot v), \tag{5.17}$$

then find $(u_k, p_k) \in (X_h, Q_h)$ satisfying for all $(v, q) \in (X_h, Q_h)$

$$(\nu + \xi)^{-1}(\delta_k^p, q) = -(\nabla \cdot z, q), \tag{5.18}$$

$$\nu(\nabla u_k, \nabla v) + \xi(\nabla \cdot u_k, \nabla \cdot v) = \nu(\nabla z_k, \nabla v) + \xi(\nabla \cdot z_k, \nabla \cdot v) + (\delta_k^p, \nabla \cdot v), \tag{5.19}$$

$$p_k = \delta_k^p + p_{k-1}. \tag{5.20}$$

Equations (5.17)–(5.20) are essentially the penalty projection methods shown in [28], suitably adapted for steady problems. Both have the same intermediate velocity solved by (5.17). As the projection step in [28] is heavily dependent on timestep, in our work we must adjust the projection step in [28] as (5.18) and (5.19) so that u_k is divergence-free. Finally, we recover the pressure by (5.20).

In the following sections, we will demonstrate through algebraic arguments to show that this penalty-projection method gives a good approximation. Using this approach, computational costs are reduced by several orders of magnitude. Our numerical tests confirm this efficiency and this method has the same accuracy as the direct one, though it does generally require more iterations than solving the full system. Numerical tests and a rigorous error analysis will follow.

5.1 | Analysis

In this section we rigorously analyze the approximation error incurred by the use of (5.12). Let $\widetilde{S} := B(\nu K + \xi B^T M_p^{-1} B)^{-1} B^T$, $S := BK^{-1}B^T$, and recall that $B(K + \xi D)^{-1} B^T = \widetilde{S} + \mathcal{O}(h/\xi)$.

We will proceed as follows: we will first show that $\|I - (\nu + \xi) M_p^{-1} \widetilde{S}\|_N$ is bounded continuously by ξ in an appropriate norm $\|\cdot\|_N$. That result combined with (5.1) will imply that the approximation error is controlled by the mesh level h and the user controlled parameter ξ.

Recall that S and M_p are spectrally equivalent (see e.g. [1, 2]) implying that

$$0 < d_{\min} < \sigma(S^{-1} M_p) < d_{\max} < \infty \tag{5.21}$$

where $d_{\min}$ and $d_{\max}$ are independent of the mesh size h. Fix $\varepsilon > 0$. As the spectral radius of $\rho(S^{-1}M_p) < d_{\max}$ there exists a matrix norm $\|\cdot\|_N$ such that:

$$\|S^{-1} M_p\|_N < \rho(S^{-1} M_p) + \varepsilon < d_{\max} + \varepsilon < 2 d_{\max} \tag{5.22}$$

Theorem 5.1 *$\|I - (\nu + \xi) M_p^{-1} \widetilde{S}\|_N$ is bounded by ν/ξ independently of h.*

Proof. Observe that:

$$I - (\nu + \xi) M_p^{-1} \widetilde{S} = I - \left(\widetilde{S}^{-1} \frac{1}{\nu + \xi} M_p \right)^{-1} \tag{5.23}$$

Applying (5.11):

$$I - (\nu + \xi) M_p^{-1} \widetilde{S} = I - \left(\frac{\nu}{\nu + \xi} S^{-1} M_p + \frac{\xi}{\nu + \xi} I \right)^{-1} \tag{5.24}$$

$$= I - \left(\frac{\xi}{\nu + \xi} \left(I + \frac{\nu(\nu + \xi)}{(\nu + \xi)\xi} S^{-1} M_p \right) \right)^{-1} \tag{5.25}$$

$$= I - \frac{\nu + \xi}{\xi} \left(I + \frac{\nu}{\xi} S^{-1} M_p \right)^{-1} \tag{5.26}$$

Let ξ be such that:

$$\left\| \frac{\nu}{\xi} S^{-1} M_p \right\|_N \le \frac{1}{2} \tag{5.27}$$

with $\|\cdot\|_N$ defined as (5.22). Note that the choice of 1/2 here is simply for convenience and one could use any value less than one without loss of generality. By (5.27) the Neumann series converges:

$$\left(I + \frac{\nu}{\xi} S^{-1} M_p \right)^{-1} = \sum_{j=0}^{\infty} \left(-\frac{\nu}{\xi} S^{-1} M_p \right)^j \tag{5.28}$$

And therefore:

$$I-(\nu+\xi)M_p^{-1}\widetilde{S}=I-\frac{\nu+\xi}{\xi}\sum_{j=0}^{\infty}\left(-\frac{\nu}{\xi}S^{-1}M_p\right)^j \tag{5.29}$$

$$=-\frac{\nu}{\xi}I-\sum_{j=1}^{\infty}\left(-\frac{\nu}{\xi}S^{-1}M_p\right)^j \tag{5.30}$$

$$=-\frac{\nu}{\xi}I-\frac{\nu}{\xi}S^{-1}M_p\sum_{j=0}^{\infty}(-1)^{j+1}\left(\frac{\nu}{\xi}S^{-1}M_p\right)^j \tag{5.31}$$

We then take norms using the norm defined in (5.22):

$$\|I-(\nu+\xi)M_p^{-1}\widetilde{S}\|_N=\left\|-\frac{\nu}{\xi}I-\frac{\nu}{\xi}S^{-1}M_p\sum_{j=0}^{\infty}(-1)^{j+1}\left(\frac{\nu}{\xi}S^{-1}M_p\right)^j\right\|_N \tag{5.32}$$

$$\leq\frac{\nu}{\xi}+\frac{\nu}{\xi}\|S^{-1}M_p\|_N\sum_{j=0}^{\infty}\left\|\frac{\nu}{\xi}S^{-1}M_p\right\|_N^j \tag{5.33}$$

$$\leq\frac{\nu}{\xi}(1+4d_{\max}) \tag{5.34}$$

Noting that d_{max} is independent of h completes the proof. ■

This then immediately implies our main result:

Theorem 5.2 *The splitting error* $\|I-(\nu+\xi)M_p^{-1}B(\nu K+\xi D)^{-1}B^T\|$ *incurred by the approximation (5.12) is bounded by* ν/ξ *and* h$/\xi$ *and tends to zero as* $\xi\to\infty$.

Proof. Applying the previous theorem and Lemma 5.1 gives:

$$\begin{aligned}&\|I-(\nu+\xi)M_p^{-1}B(\nu K+\xi D)^{-1}B^T\|\\&\leq\|I-(\nu+\xi)M_p^{-1}\widetilde{S}\|+\|(\nu+\xi)M_p^{-1}\mathcal{O}(h/\xi)\|\\&\leq\frac{\nu}{\xi}C_1+\frac{h}{\xi}C_2\end{aligned} \tag{5.35}$$

■

Theorem 5.3 *The local splitting error at an iteration incurred by replacing* $(B(\nu K+\xi D)^{-1}B^T)^{-1}$ *with* $(\nu+\xi)M_p^{-1}$ *at an iteration k in the discrete version of (2.1) is bounded by* ν/ξ *and h.*

Proof. By direct inspection we may find that one step of the discrete problem given by (2.1) is equivalent to solving the following system in the block matrix A_F (letting $\widetilde{K}=(\nu K+\xi D)$ and $\widetilde{S}_F=B\widetilde{K}^{-1}B^T$ for the sake of notation):

$$A_F=\begin{bmatrix}A & A\widetilde{K}^{-1}B^T\\ B & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{u}^k\\ \delta_p^k\end{bmatrix}=\begin{bmatrix}f+B^Tp^{k-1}\\ 0\end{bmatrix} \tag{5.36}$$

Letting $\widetilde{M}_p=(\nu+\xi)M_p$, the approximate inverse of A_F using (5.12) is given by:

$$A_{apx}^{-1}=\begin{bmatrix}(I-\widetilde{K}^{-1}B^T\widetilde{M}_p^{-1}B)A^{-1} & \widetilde{K}^{-1}B^T\widetilde{M}_p^{-1}\\ \widetilde{M}_p^{-1}BA^{-1} & -\widetilde{M}_p^{-1}\end{bmatrix} \tag{5.37}$$

Let $\boldsymbol{v}_F$ be the solution vector computed by solving the system (5.36) and $\boldsymbol{v}_{apx}$ the approximated solution computed by applying (5.37) to the same right hand side $\boldsymbol{b}$ given in (5.36).

Then we define the splitting error vector $\boldsymbol{e}_s = [e_u, e_p]^T$ as:

$$\boldsymbol{e}_s = \mathrm{v}_F - \mathrm{v}_{apx} = \mathrm{v}_F - A_{apx}^{-1}\boldsymbol{b} = \mathrm{v}_F - A_{apx}^{-1}(A_F\mathrm{v}_F) = (I - A_{apx}^{-1}A_F)\mathrm{v}_F \tag{5.38}$$

Expanding this expression:

$$(I - A_{apx}^{-1}A_F)\mathrm{v}_F = \begin{bmatrix} 0 & \widetilde{K}^{-1}B^T(I - \widetilde{M}_p^{-1}\widetilde{S}_F) \\ 0 & I - \widetilde{M}_p^{-1}\widetilde{S}_F \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_F \\ \boldsymbol{\delta}_{p,F} \end{bmatrix} \tag{5.39}$$

Note that neither the momentum nor mass equation is satisfied, unlike the full solution which conserves mass. By our bounds from part 1,

$$||\boldsymbol{e}_s|| \le \left(\frac{\nu}{\xi}C_1 + \frac{h}{\xi}C_2\right) ||\boldsymbol{\delta}_{p,F}|| \tag{5.40}$$

With a bound on the local splitting error, we can now prove a bound for the global splitting error. ■

Theorem 5.4 *Let $\boldsymbol{v}_{apx} = [\boldsymbol{u}_{apx}, p_{apx}]$ be the solution obtained from Picard iteration given by approximating the Schur complement solve with (5.12) and $\boldsymbol{v}_F = [\boldsymbol{u}_F, p_F]$ be the solution obtained from (2.1). Then:*

$$||v_{apx} - v_{ex}|| \le C_1\frac{\nu}{\xi} + C_2\frac{h}{\xi}$$

for suitable ξ and h.

Proof. We will proceed by induction. Starting with the same initial guess $\boldsymbol{v}^0 = (\boldsymbol{u}_0, p_0)$, the local splitting error derived in the previous theorem implies:

$$||\boldsymbol{e}_s^1|| < \left(\frac{\nu}{\xi}C_1 + \frac{h}{\xi}C_2\right) ||\boldsymbol{\delta}_{p,F}^1||$$

Define $\varepsilon := \frac{\nu}{\xi}C_1 + \frac{h}{\xi}C_2$. The above bound implies the existence of a vector $\boldsymbol{\eta}$ such that $\mathrm{v}_{apx}^1 := \mathrm{v}_F^1 + \widetilde{\varepsilon}\boldsymbol{\eta}$. Similarly, there exists a matrix Δ such that $A_{apx}^2 := A_F^2 + \widetilde{\varepsilon}\Delta$. Note that the $\widetilde{\varepsilon}$ are both $\mathcal{O}(\varepsilon)$ but possibly distinct.

Now assume that $||\boldsymbol{e}_s^{k-1}|| \sim \mathcal{O}(\varepsilon)$ and that $\boldsymbol{v}_{F,\,k-1}$ is bounded. Then we again have $\mathrm{v}_{apx}^{k-1} := \mathrm{v}_F^{k-1} + \widetilde{\varepsilon}\boldsymbol{\eta}$ for some $\boldsymbol{\eta}$ and $A_{apx}^k := A_F^k + \widetilde{\varepsilon}\Delta$ for some Δ. We seek to show that $||\boldsymbol{e}_s^k|| \sim \mathcal{O}(\varepsilon)$ (note that the base case holds for iteration $||\boldsymbol{e}_s^1||$). This establishes that v_{apx}^k is always in an ε-neighborhood of v_F^k for each k, implying that as $k \to \infty$, $\boldsymbol{v}_{apx}$ converges to a solution within an ε neighborhood of $\boldsymbol{v}_F$.

$$A_F^k\boldsymbol{v}_F^k = \boldsymbol{r}_F^{k-1} \tag{5.41}$$

$$(A_F^k + \widetilde{\varepsilon}\Delta)\boldsymbol{v}_{apx}^k = \boldsymbol{r}_F^{k-1} + \widetilde{\varepsilon}\widetilde{\boldsymbol{\eta}} \tag{5.42}$$

As the right-hand side depends on the solution at the last iteration. Then:

$$||\boldsymbol{v}_F^k - \boldsymbol{v}_{apx}^k|| = ||((A_F^k)^{-1} - (A_F^k + \widetilde{\varepsilon}\Delta)^{-1})\boldsymbol{r}_F^{k-1} - (A_F^k + \widetilde{\varepsilon}\Delta)^{-1}\widetilde{\varepsilon}\widetilde{\boldsymbol{\eta}}|| \tag{5.43}$$

$$\le ||(A_F^k)^{-1} - (A_F^k + \widetilde{\varepsilon}\Delta)^{-1}||\,||\boldsymbol{r}_F^{k-1}|| + \widetilde{\varepsilon}||(A_F^k + \widetilde{\varepsilon}\Delta)^{-1}||\,||\boldsymbol{\eta}|| \tag{5.44}$$

$$\le \widetilde{\varepsilon}(||\boldsymbol{r}_F^{k-1}|| + ||(A_F^k + \widetilde{\varepsilon}\Delta)^{-1}||\,||\boldsymbol{\eta}||) \tag{5.45}$$

where the last line follows from the continuity of matrix inverses. This completes the proof. ■

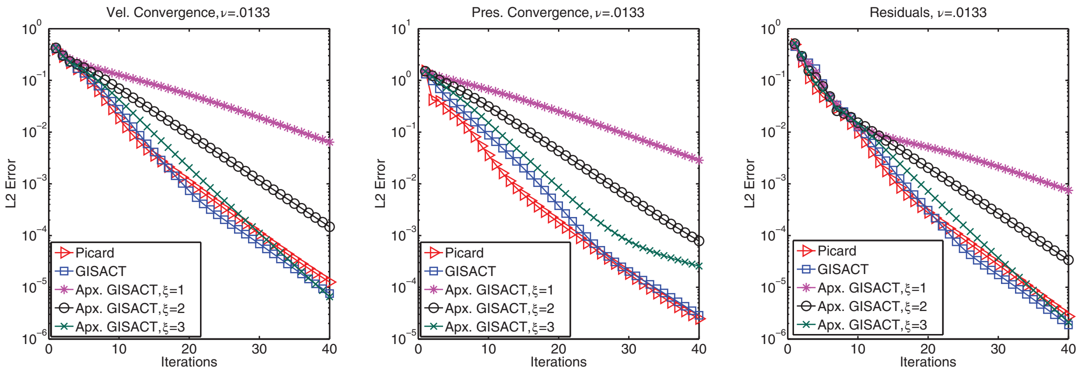

FIGURE 10 Bifurcation test case, $\nu = .0133$ [Color figure can be viewed at wileyonlinelibrary.com]

Note that strictly speaking Theorem 5.4 does not imply the convergence of $(\boldsymbol{u}_{apx}, p_{apx})$ to $(\boldsymbol{u}_F, p_F)$, but only convergence to within a certain neighborhood. The (true) convergence of $(\boldsymbol{u}_F, p_F)$ to the solution $(\boldsymbol{u}, p)$ of (2.1) then implies that $(\boldsymbol{u}_{apx}, p_{apx})$ converges to within a certain neighborhood of $(\boldsymbol{u}, p)$ as well. This neighborhood can be made arbitrarily small for large ξ; however for any fixed ξ this only establishes convergence up to a fixed level of accuracy.

This theorem shows the practical limitations of this scheme. As stated previously, large ξ can make the problem difficult to solve. In practice, with $\xi \sim \mathcal{O}(1)$, the bound predicted by Theorem 5.4 may be small due to low values of ν and h; however for problems with higher viscosity or relatively coarse meshes, it is possible that $\xi \sim \mathcal{O}(1)$ is not sufficiently large to make the splitting error suitably small, and taking ξ extremely large is not a viable option in general. While the substantial numerical savings offered by this approach make it potentially worthwhile, we acknowledge it has significant limitations and may not be applicable for some problems. Nonetheless, in our tests we still found that it performed quite well. Although we did observe some evidence of limiting behavior likely arising from the global splitting error, the effect was small.

5.2 | Numerical test: bifurcation flow

We repeat the same 2D bifurcation test previously used to compare the solutions computed by Algorithm 2.1 (GISACT), and Algorithm 2.1 using the approximated Schur complement (5.12) (Apx. GISACT) to the reference solution from the standard Picard iterations up to a very high level of accuracy (1e-12). For the reference case, we solved the full saddle-point system with UMFPACK at each iteration. For ACT, we solve both velocity systems with UMFPACK (step k.1 and step k.2), and the Schur complement with CG preconditioned by the lumped pressure mass-matrix for outer solve and UMFPACK for inner solve. For Apx. GISACT, we use UMFPACK for the velocity solves (step k.1 and step k.2), and solve the Schur complement system by multiplying a diagonal matrix. For our comparison Picard method, we use the same solver configuration as used in Section 3.1. We will test the method for $\xi = 1, 2, 3$.

We display the L^2 norm of velocity and pressure convergence, as well as the nonlinear residuals, in Figures 10 and 11. Note here that $\xi = 2$ for the plotted Picard and GISACT comparisons.

For $\nu = .0133$ (Figure 10), the approximated Schur complement solution converges more slowly than GISACT with $\xi = 1$, as might be expected, but we clearly observe monotonic convergence to the desired solution. For $\xi = 2$ the convergence is still slower than full GISACT, but the gap narrows considerably, while for $\xi = 3$ the convergence is the same as standard Picard and full GISACT in the

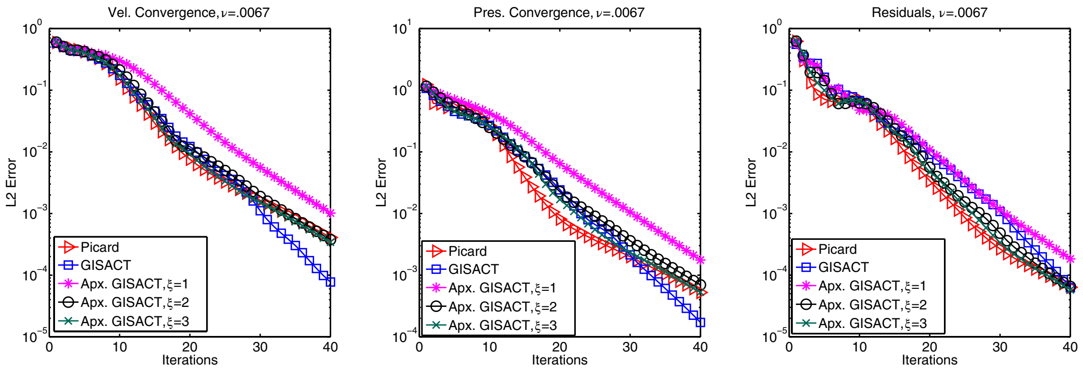

FIGURE 11 Bifurcation test case, $\nu = .0067$ [Color figure can be viewed at wileyonlinelibrary.com]

velocity space. This is expected based on our bound (5.35), which suggests that large ξ implies a better approximation. In the pressure space for $\xi = 3$, we see the convergence start comparably to GISACT and Picard, but then flatten out. Recall from Theorem 5.4 that the approximate Schur complement scheme converges only to within a certain neighborhood, the size of which depends on ξ; we suspect this flattening is caused by the solution reaching the threshold of convergence. Unsurprisingly, the average solve time across all cases was .091 s, a small fraction of the times shown in Table 1.

For $\nu = .0067$ (Figure 11), we notice that Apx. GISACT is still slower than GISACT for $\xi = 1$, however the difference is much less pronounced than for the case $\nu = .0133$. For $\xi = 2$ and $\xi = 3$, we see nearly identical convergence behavior for GISACT and Apx. GISACT. This is consistent with our expectation based on Theorem 5.4, as the quality of the approximation depends on ν/ξ and therefore as we decrease ν or increase ξ the approximated scheme should perform more similarly to GISACT.

As expected, the convergence of the approximate GISACT depends both on the grad-div parameter ξ and the viscosity parameter ν. For cases with higher ν, it appears one must use a relatively high value of ξ in order for the approximate scheme to converge at a similar rate; however as ν decreases the schemes behave more similarly, to where one must use a higher value of ξ anyway in order for GISACT to converge rapidly (as seen in Section 3.1). In these instances, it appears the approximate scheme offers a similar convergence rate at a fraction of the cost. Although we only expect the approximate scheme to converge up to a certain level of accuracy, in this test this we found $\xi \sim \mathcal{O}(1)$ provided a small enough threshold for us to obtain accurate solutions.

5.3 | Numerical test: Newton-type version

We now compare the convergence of the Newton and Picard formulations of our scheme and the effect of the approximate Schur complement. We run the same bifurcation test with $\nu = .0133$ as in Section 5.2. We again compare the standard GISACTN scheme with the approximate GISACTN scheme for $\xi = 1, 5, 10$. We plot the results in Figure 12.

The first thing we note is that the approximate GISACTN algorithm does not converge as quickly as standard GISACTN for any value of ξ. For $\xi = 1$, approximate GISACTN does not seem to converge at all. We clearly see in the pressure plot that the approximate version GISACTN with $\xi = 10$ reaches its neighborhood of convergence around 10 iterations, after which it does not converge further. This is the expected behavior based on Theorem 5.4. Interestingly, in this test this does not seem to happen in the velocity space.

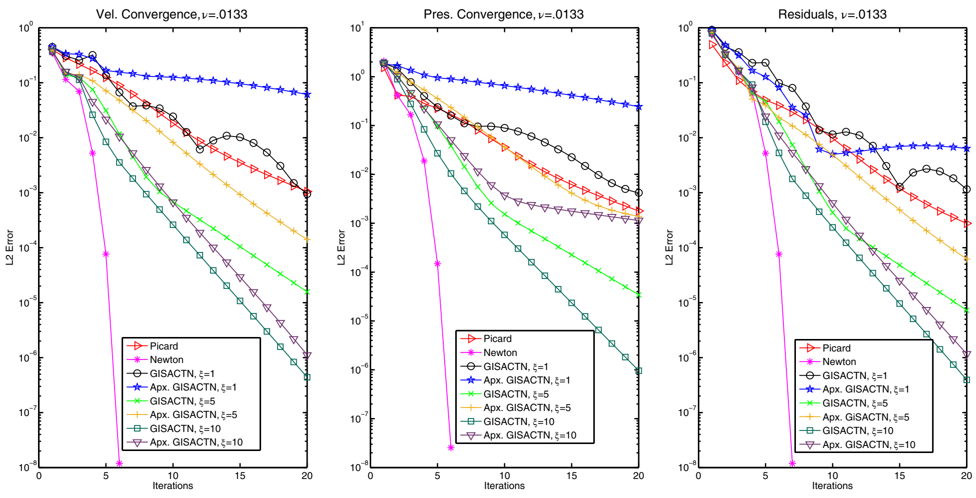

FIGURE 12 Bifurcation test, GISACTN and approximate GISACTN [Color figure can be viewed at wileyonlinelibrary.com]

6 | CONCLUSIONS

We have developed new efficient solvers for the steady incompressible NSE. These new solvers are closely related to the recently developed Yosida-type methods but demonstrate faster convergence and greater robustness for high Reynolds numbers. Like the aforementioned Yosida methods, the Schur complement system is easy to precondition and does not change, greatly reducing computational costs. We have also shown that one may eliminate the Schur complement system and replace it with a diagonal matrix inversion; though this slows convergence, it appears to be robust to increasing *Re* and the numerical savings may nonetheless make this formulation practical in many settings.

For future work, one may wish to further study the convergence and robustness properties of these with analytical or numerical tools to determine how they are affected by changes in ξ or the Reynolds number. This would be especially useful for the approximate Schur complement version of the scheme. Determining an optimal value of ξ that gives rapid convergence for this approximation while not harming the problem conditioning too much could yield further improvements. It may also be worthwhile to see if the convergence can be improved through the use of nonlinear acceleration techniques, such as Anderson acceleration.

ORCID

Alex Viguerie http://orcid.org/0000-0002-8556-3499

REFERENCES

[1] M. Benzi, M. Olshanskii, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput. vol. 28 (2006) pp. 2095–2113.
[2] H. Elman, D. Silvester, A. Wathen, "*Finite elements and fast iterative solvers with applications in incompressible fluid dynamics*," in *Numerical mathematics and scientific computation*, Oxford University Press, Oxford, 2014.
[3] P. Gervasio, F. Saleri, A. Veneziani, *Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations*, J. Comput. Phys. vol. 214 (2006) pp. 347–365.
[4] A. Quarteroni, F. Saleri, A. Veneziani, *Analysis of the Yosida method for incompressible Navier–Stokes equations*, J. Math. Pures Appl. vol. 78 (1999) pp. 473–503.
[5] A. Quarteroni, F. Saleri, A. Veneziani, *Factorization methods for the numerical approximation of Navier–Stokes equations*, Comput. Methods Appl. Mech. Eng. vol. 188 (2000) pp. 505–526.
[6] L. Rebholz, M. Xiao, *Improved accuracy in algebraic splitting methods for Navier–Stokes equations*, SIAM J. Sci. Comput. vol. 39 (2017) pp. A1489–A1513.

[7] L. Rebholz, A. Viguerie, M. Xiao, Incremental Picard–Yosida and Newton–Yosida iterations for the efficient numerical solution of steady Navier–Stokes equations, in preparation.

[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. vol. 294 (2015) pp. 259–277.

[9] A. Viguerie, A. Veneziani, *Algebraic splitting methods for the steady incompressible Navier–Stokes equations at moderate Reynolds numbers*, Comput. Methods Appl. Mech. Eng. vol. 330 (2018) pp. 271–291.

[10] D. Arnold, J. Qin, “*Quadratic velocity/linear pressure Stokes elements*,” in *Advances in Computer Methods for Partial Differential Equations VII*, R. Vichnevetsky, D. Knight, G. Richter (Editors), IMACS, New Brunswick, 1992, pp. 28–34.

[11] S. Zhang, *A new family of stable mixed finite elements for the 3d Stokes equations*, Math. Comput. vol. 74 (2005) pp. 543–554.

[12] R. Falk, M. Neilan, *Stokes complexes and the construction of stable finite elements with pointwise mass conservation*, SIAM J. Numer. Anal. vol. 51 (2013) pp. 1308–1326.

[13] J. Guzmán, M. Neilan, *Conforming and divergence-free Stokes elements on general triangulations*, Math. Comput. vol. 83 (2014) pp. 15–36.

[14] V. Girault, P.-A. Raviart, *Finite element methods for Navier–Stokes equations: theory and algorithms*, Springer-Verlag, Berlin, 1986.

[15] V. John et al., *On the divergence constraint in mixed finite element methods for incompressible flows*, SIAM Rev. vol. 59 (2017) pp. 492–544.

[16] H. Vande Sande et al., “*Accelerating nonlinear time-harmonic problems by a hybrid Picard–Newton approach*,” in *Proceedings: 10th International IGTE Symposium on Numerical Field Calculation in Electrical Engineering: September 16–18 2002, Graz, Austria*, Verlag d. Techn. Univ. Graz, Graz, 2002, pp. 342–347.

[17] T. Heister, G. Rapin, *Efficient augmented Lagrangian-type preconditioning for the Oseen problem using grad-div stabilization*, Int. J. Numer. Methods Fluids vol. 71 (2013) pp. 118–134.

[18] M. A. Castro, J. R. Cebral, C. M. Putman, *Computational fluid dynamics modeling of intracranial aneurysms: Effects of parent artery segmentation on intra-aneurysmal hemodynamics*, Am. J. Neuroradiol. vol. 27 (2006) pp. 1703–1709.

[19] W. W. Nichols, M. F. O’Rourke, *McDonald’s blood flow in arteries*, Oxford University Press, Oxford, 2005.

[20] P. Spiller et al., *Measurement of systolic and diastolic flow rates in the coronary artery system by X-ray densitometry*, Circulation vol. 68 (1983) pp. 337–347.

[21] A. Viguerie, *Efficient, stable, and reliable solvers for the steady incompressible Navier-Stokes equations in computational hemodynamics*. Ph.D., Emory University, 2018.

[22] M. S. Alnæs et al., *The FEniCS project version 1.5*, Arch. Numer. Softw. vol. 3 (2015), Online publication. https://journals.ub.uni-heidelberg.de/index.php/ans/article/view/20553.

[23] M. E. Lacruz et al., *Short term blood pressure variability- variation between arm side, body position, and successive measurements: a population-based cohort study*, BMC Cardiovasc. Disord. vol. 17 (2017) p. 31.

[24] V. M. Musini, J. M. Wright, *Factors affecting blood pressure variability: lessons learned from two systematic reviews of randomized controlled trials*, PLoS One vol. 4 (2009) pp. S19–S25

[25] G. Pannarale et al., *Bias and variabiity in blood pressure measurement with ambulatory recorders*, Hypertension vol. 22 (1993) pp. 591–598.

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

[27] H. V. Henderson, S. R. Searle, *On deriving the inverse of a sum of matrices*, SIAM Rev. vol. 23 (1981) pp. 53–60.

[28] A. Linke et al., *A connection between coupled and penalty projection timestepping schemes with FE spatial discretization for the Navier-stokes equations*, J. Numer. Math. vol. 25 (2018) pp. 229–248.

How to cite this article: Viguerie A, Xiao M. Effective Chorin–Temam algebraic splitting schemes for the steady Navier–stokes equations. *Numer Methods Partial Differential Eq.* 2018;1–25. https://doi.org/10.1002/num.22326