



Balanced Iterative Solvers for Linear Nonsymmetric Systems and Nonlinear Systems with PDE Origins: Efficient Black-Box Stopping Criteria

DOI:

[10.1007/s10915-019-01018-w](https://doi.org/10.1007/s10915-019-01018-w)

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Pranjal, & Silvester, D. (2019). Balanced Iterative Solvers for Linear Nonsymmetric Systems and Nonlinear Systems with PDE Origins: Efficient Black-Box Stopping Criteria. *Journal of Scientific Computing*.
<https://doi.org/10.1007/s10915-019-01018-w>

Published in:

Journal of Scientific Computing

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



Balanced iterative solvers for linear nonsymmetric systems and nonlinear systems with PDE origins: efficient black-box stopping criteria

Pranjal · David Silvester

Received: date / Accepted: date

Abstract This paper discusses the design of efficient algorithms for solving linear nonsymmetric systems and nonlinear systems associated with FEM approximation of elliptic PDEs. The novel feature of the designed linear solvers like GMRES, BICGSTAB(ℓ), TFQMR, and nonlinear solvers like Newton and Picard, is the incorporation of error control in the ‘natural norm’ in combination with an effective a posteriori estimator for the PDE approximation error. This leads to robust black-box stopping criteria in the sense that the iteration is terminated as soon as the algebraic error is insignificant compared to the approximation error. Such a solver is called ‘balanced’ in this paper since the stopping criteria are obtained by balancing (comparing) the algebraic error and the approximation error.

Keywords FEM approximation of PDEs · a posteriori FEM error estimators · nonsymmetric linear systems · iterative solvers · GMRES · BICGSTAB(ℓ) · TFQMR · Newton solvers · preconditioning

PACS 65C20 · 65N30 · 65N15

1 Introduction and problem description

Consider the following boundary value problem: find the solution $u(\vec{x}) : \Omega \rightarrow \mathbb{R}$ such that

$$\mathcal{L}(\vec{x})u(\vec{x}) = f(\vec{x}), \quad \forall \vec{x} \in \Omega, \quad (1a)$$

$$\mathcal{B}(\vec{x})u(\vec{x}) = g(\vec{x}), \quad \forall \vec{x} \in \partial\Omega. \quad (1b)$$

where $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3, \dots$) denotes the spatial domain and $\partial\Omega$ is the spatial boundary. Here \mathcal{L} is the (possibly nonlinear) elliptic PDE operator, \mathcal{B} denotes the

Pranjal
Department of Mathematics, Virginia Polytechnic Institute and State University, USA
E-mail: pranjalprasad21@gmail.com

David Silvester
School of Mathematics, University of Manchester, UK
E-mail: d.silvester@manchester.ac.uk

boundary operator, f is the given source term, g denotes the boundary value, and u the true solution. Equation (1) is discretized in this paper using finite element method (FEM) [2], and the corresponding linear or nonlinear discrete system is solved by a linear or a nonlinear iterative solver, respectively.

1.1 Linear systems

For chosen spatial discretization parameter h , FEM for solving (1) results in solving for a linear system, that is,

$$F_h \mathbf{x}_h = \mathbf{b}_h \iff M_h^{-1} F_h \mathbf{x}_h = M_h^{-1} \mathbf{b}_h, \quad (2)$$

where the matrix M_h is a preconditioner. The matrix F_h and the vector \mathbf{b}_h are known quantities arising from the FEM process while the unknown algebraic solution \mathbf{x}_h denotes the coordinate vector of the FEM solution u_h in the chosen FEM basis.

The numerical solution of (1) essentially involves two types of errors which are: approximation error ($\|u - u_h\|_{\mathcal{E}}$) and algebraic error ($\|u_h - u_h^{(k)}\|_{\mathcal{E}}$). Here $u_h^{(k)}$ denotes the FEM solution formed from the algebraic solution $\mathbf{x}_h^{(k)}$ at the k th step of the chosen iterative solver. Also, $\|\cdot\|_{\mathcal{E}}$ denotes the ‘natural norm’. Wathen [24] has observed that FEM approximation of a PDE endows the problem with a natural norm, which is determined by the specific approximation space. Typically, in FEM setting, the PDE approximation error and the algebraic error are measured in this natural norm.

The approximation error is fixed for chosen spatial discretization parameter. Solving iteratively the corresponding discrete linear(ized) system(s) to a very high accuracy is not desirable. This is because a highly accurate iterative solution may require too many iterations and simply waste computational resources without decreasing the approximation error. On the other hand, if the iterations are stopped too early the iterative solution will not be a good approximation to the exact solution. This paper attempts to address these issues by presenting balanced black-box stopping tests in Krylov solvers [8, sections 11.3–11.4] for solving linear systems with PDE origins. This is an active research field; see [17, 20, 11, 14, 15].

In solving PDEs numerically, the storage requirements and computational flops increase with the size and the number of linear systems. Typically, using a balanced black-box stopping methodology would lead to computational savings (by reducing unnecessary iterations) and would definitely rule out premature stopping of the chosen iterative solver.

1.2 Nonlinear systems

When FEM approximation to solve (1) results in a nonlinear system, nonlinear solvers like Newton solvers are used to solve them. Starting with a given initial guess $u_h^{(0)}$, nonlinear solvers typically construct a sequence of iterates $\{u_h^{(l+1)}\}$, $l = 0, 1, \dots$ satisfying

$$u_h^{(l+1)} = u_h^{(l)} + \delta u_h^{(l)}. \quad (3)$$

The problem of balanced black-box stopping of the nonlinear iterative solver is of interest in the same manner as that for a linear solver. Also, the ‘correction’ term $\delta u_h^{(l)}$ at each nonlinear iterative step l requires solving a linear system for the basis coefficients of $\delta u_h^{(l)}$. Thus, balanced black-box stopping criteria developed in linear solvers can be applied in solving the linear system arising at each nonlinear iterative step along with balanced black-box stopping criteria for the nonlinear solver.

1.3 Paper organization

The structure of this paper is as follows. The problem statement has already been discussed in section 1. The general balanced black-box stopping solution methodology for linear and nonlinear systems is presented in sections 2 and 3 respectively. In section 2, balanced black-box stopping tests are also presented in GMRES [18] and suboptimal Krylov solvers of nonsymmetric linear systems with PDE origins. Using the IFISS [4] toolbox in MATLAB, computational results illustrating the devised balanced black-box methodology are presented and discussed for convection-diffusion equations and Navier–Stokes equations in sections 4 and 5 respectively. A balanced black-box stopping test in (nonlinear) Newton/Picard solver for the Navier–Stokes PDE is also presented in section 5. Conclusions are presented in section 6. Note that C and c denote generic constants throughout. Also, l denotes a nonlinear iterative step and k denotes a linear iterative step throughout this paper.

2 Solution methodology for linear solvers

For a given approximation, the approximation error is fixed. The triangle inequality at (linear solver) iteration k ($k = 0, 1, 2, \dots$) gives the following decomposition of the total error,

$$\begin{array}{rcccl} \|u - u_h^{(k)}\|_{\mathcal{E}} & \leq & \|u - u_h\|_{\mathcal{E}} & + & \|u_h - u_h^{(k)}\|_{\mathcal{E}}. \\ \text{total error} & & \text{approximation error} & & \text{algebraic error} \end{array} \quad (4)$$

The total error at iteration step k is nothing but the approximation error obtained from k th FEM iterate $u_h^{(k)}$ (which in turn is obtained from $\mathbf{x}_h^{(k)}$ whose components are the coefficients in the FEM basis representation of $u_h^{(k)}$).

In FEM setting, the approximation error (and hence total errors) can be measured a priori or a posteriori [23]. A priori approximation error estimation usually requires the PDE solution to satisfy some regularity conditions which may not hold or/and may not be easily verifiable a priori. On the other hand, robust a posteriori approximation error estimation techniques are popular for driving the FEM procedure adaptively [2, chapter 9] and are generally readily available in the sense that

$$c \eta_h \leq \|u - u_h\|_{\mathcal{E}} \leq C \eta_h, \quad \text{with } \frac{C}{c} \sim O(1), \quad (5)$$

where η_h denotes the a posteriori estimate of the approximation error. Also, the algebraic error $\|u_h - u_h^{(k)}\|_{\mathcal{E}}$ can usually be expressed in terms of the iteration error $\mathbf{e}_h^{(k)} := \mathbf{x}_h - \mathbf{x}_h^{(k)}$ norm $\|\mathbf{e}_h^{(k)}\|_{E_h} := \sqrt{(\mathbf{e}_h^{(k)})^T E_h \mathbf{e}_h^{(k)}}$, that is,

$$c\|\mathbf{e}_h^{(k)}\|_{E_h} \leq \|u_h - u_h^{(k)}\|_{\mathcal{E}} \leq C\|\mathbf{e}_h^{(k)}\|_{E_h}, \quad \text{with } \frac{C}{c} \sim O(1). \quad (6)$$

Here E_h is a ‘suitable’ symmetric positive-definite matrix such that $\|\cdot\|_{E_h}$ defines a norm.

Thus, if $\eta_h^{(k)}$, η_h , $\|\mathbf{e}_h^{(k)}\|_{E_h}$ are tight estimates of the total error (at iteration k), the approximation error, and the algebraic error (at iteration k) respectively, then in light of the above discussions (4) becomes

$$\eta_h^{(k)} \simeq \eta_h + \|\mathbf{e}_h^{(k)}\|_{E_h}, \quad k = 0, 1, 2, \dots \quad (7)$$

The equivalence \simeq in (7) follows from (5) and (6).

Remark 1 Notice from (7) that when the contribution of $\|\mathbf{e}_h^{(k)}\|_{E_h}$ to the sum is ‘small’ then the sequence $\{\eta_h^{(k)}\}$ would converge with some accuracy to (unknown but fixed) η_h such that the a posteriori estimate for the total error cannot be significantly reduced further. Thus, the iterative strategy here can be looked upon as constructing a sequence $\{\eta_h^{(k)}\}$ converging to η_h .

Thus, one would stop when $\|\mathbf{e}_h^{(k)}\|_{E_h}$ ‘balances’ η_h in the sum (7), that is, stop at the first iteration k^* such that

$$\|\mathbf{e}_h^{(k^*)}\|_{E_h} \leq \eta_h. \quad (8)$$

At this specific iteration k^* , it follows from (4) that the total error estimate $\eta_h^{(k)}$ will be bounded by twice the unknown approximation error (up to the constants in (5)). Under the assumption that the equivalence (7) represents an equality, a practical stopping test is obtained as in (8) except that the right-hand-side is given by

$$\|\mathbf{e}_h^{(k^*)}\|_{E_h} \leq \theta \eta_h^{(k^*)}, \quad (9)$$

where $0 < \theta \leq 1$. Observe that $\theta = 1/2$ corresponds to equality in (7). Note that the numerical results presented later have been produced with $\theta = 1$.

Generally, the iteration error $\mathbf{e}_h^{(k)}$ is unknown (and hence $\|\mathbf{e}_h^{(k)}\|_{E_h}$ is unknown too) since the exact algebraic solution \mathbf{x}_h is not usually available. Usually some norm $\|\mathbf{r}_h^{(k)}\|_{S_h} := \sqrt{(\mathbf{r}_h^{(k)})^T S_h \mathbf{r}_h^{(k)}}$ (where S_h is a symmetric positive-definite matrix) of the iteration residual $\mathbf{r}_h^{(k)} := \mathbf{b}_h - F_h \mathbf{x}_h^{(k)}$ is readily computable and usually monotonically decreasing with respect to the iteration count k of the chosen solver. Obtaining tractable upper and lower bounds on the $\|\mathbf{e}_h^{(k)}\|_{E_h}$ norm in terms of the surrogate norm $\|\mathbf{r}_h^{(k)}\|_{S_h}$ is the novel feature of the balanced black-box stopping strategy that is presented here. Moreover, the work in this paper states in these bounds, the exact positive constants $\lambda_h \in \mathbb{R}$, $A_h \in \mathbb{R}$ such that

$$\lambda_h \|\mathbf{r}_h^{(k)}\|_{S_h}^2 \leq \|\mathbf{e}_h^{(k)}\|_{E_h}^2 \leq A_h \|\mathbf{r}_h^{(k)}\|_{S_h}^2, \quad k = 0, 1, 2, \dots \quad (10)$$

Equation (10) leads to the following two bounds on $\|\mathbf{e}_h^{(k)}\|_{E_h}$, that is,

$$\frac{\|\mathbf{e}_h^{(k)}\|_{E_h}}{\|\mathbf{e}_h^{(0)}\|_{E_h}} \leq \sqrt{\frac{\Lambda_h}{\lambda_h}} \frac{\|\mathbf{r}_h^{(k)}\|_{S_h}}{\|\mathbf{r}_h^{(0)}\|_{S_h}} \implies \|\mathbf{e}_h^{(k)}\|_{E_h} \leq \frac{\Lambda_h}{\sqrt{\lambda_h}} \|\mathbf{r}_h^{(k)}\|_{S_h}, \quad (11a)$$

$$\|\mathbf{e}_h^{(k)}\|_{E_h} \leq \sqrt{\Lambda_h} \|\mathbf{r}_h^{(k)}\|_{S_h}. \quad (11b)$$

From $\mathbf{r}_h^{(k)} = F_h \mathbf{e}_h^{(k)}$, it follows that $\|\mathbf{e}_h^{(k)}\|_{E_h}^2 = (\mathbf{r}_h^{(k)})^T F_h^{-T} E_h F_h^{-1} \mathbf{r}_h^{(k)}$. Thus, using (10) this implies that obtaining a lower (λ_h) bound and an upper (Λ_h) bound on the quantity $\frac{\|\mathbf{e}_h^{(k)}\|_{E_h}^2}{\|\mathbf{r}_h^{(k)}\|_{S_h}^2} = \frac{(\mathbf{r}_h^{(k)})^T F_h^{-T} E_h F_h^{-1} \mathbf{r}_h^{(k)}}{(\mathbf{r}_h^{(k)})^T S_h \mathbf{r}_h^{(k)}}$ requires calculating the extremal Rayleigh quotients [8, p. 453] of $F_h^{-T} E_h F_h^{-1}$ and S_h . This is equivalent to computing the extremal (outer most) eigenvalues, that is, the smallest eigenvalue and the largest eigenvalue of the symmetric positive-definite generalized eigenvalue problem for $F_h^{-T} E_h F_h^{-1}$ and S_h (or equivalently of E_h and $F_h^T S_h F_h$). This generalized eigenvalue problem can be transformed (theoretically through a Cholesky factorization of S_h) into a symmetric positive-definite algebraic eigenvalue problem (keeping the eigenvalues same) and hence λ_h and Λ_h will both be positive.

In light of (9), the bounds in (11) lead to the following strong and weak balanced black-box stopping tests: stop at the first iteration k^* such that either holds,

$$\frac{\Lambda_h}{\sqrt{\lambda_h}} \|\mathbf{r}_h^{(k^*)}\|_{S_h} \leq \eta_h^{(k^*)}; \quad \sqrt{\Lambda_h} \|\mathbf{r}_h^{(k^*)}\|_{S_h} \leq \eta_h^{(k^*)},$$

or equivalently,

$$\|\mathbf{r}_h^{(k^*)}\|_{S_h} \leq \frac{\sqrt{\lambda_h}}{\Lambda_h} \eta_h^{(k^*)} \text{ (strong stop); } \|\mathbf{r}_h^{(k^*)}\|_{S_h} \leq \frac{1}{\sqrt{\Lambda_h}} \eta_h^{(k^*)} \text{ (weak stop)}. \quad (12)$$

In terms of the number of iterations for convergence, the strong stopping test cannot perform better than the weak stopping test since $\frac{\sqrt{\lambda_h}}{\Lambda_h} = \frac{\sqrt{\lambda_h}}{\sqrt{\Lambda_h} \sqrt{\Lambda_h}} \leq \frac{1}{\sqrt{\Lambda_h}}$. Moreover, the strong stopping test involves an additional overhead of computing the smallest eigenvalue λ_h . Thus, it would be prudent to employ the weak stopping test whenever possible.

Remark 2 A crucial point to note is that if the employed a posteriori error estimator overestimates the approximation error (total errors), it will be better to employ the strong stopping test for the weak stopping test is more likely to cause premature stopping.

Note that the behavior (overestimation or underestimation) of the a posterior error estimator for a given problem would usually be known in advance and hence the choice of the corresponding (strong or weak) stopping test in (12). The strong and the weak stopping tests (12) are black-box in the sense that the quantities in these stopping tests are computed without using any user-specified safety factors or tuning parameters.

Computational logistics of linear solver stopping tests are the following:

- The quantity $\|\mathbf{r}_h^{(k)}\|_{S_h}$ is readily computable in most iterative solvers; for our purpose, S_h is the identity matrix and thus, $\|\mathbf{r}_h^{(k)}\|_{S_h}$ is merely the Euclidean norm of the residual.

- The a posteriori error estimate $\eta_h^{(k)}$ should be computed periodically (say every 10–20 iterations) to reduce its impact on the overall algorithmic cost.
- Unlike symmetric linear systems, where λ_h , Λ_h could be estimated cheaply by exploiting Ritz and harmonic Ritz value relations [17, chapters 2–3] in a preconditioned MINRES [13] solver, such relations are difficult to exploit for nonsymmetric linear solvers. But matrices arising from FEM approximation of PDEs are sparse in general, so, MATLAB `eigs` has been used in this work to compute λ_h , Λ_h . Devising computationally cheaper methods than `eigs` for estimating these eigenvalues for generic sparse eigenvalue problems remains a challenging research question, which is not addressed in this paper.

One can use similar ideas to develop a balanced stopping test in cases where the coefficient matrix is not symmetric. To this end, note that the preconditioned generalized minimum residual method GMRES [18] is a popular method for solving nonsymmetric linear systems. GMRES solver’s readily available and monotonically decreasing (with iteration count k) residual norm $\|\mathbf{r}_h^{(k)}\|_2 := \sqrt{(\mathbf{r}_h^{(k)})^T \mathbf{r}_h^{(k)}}$ (S_h is the identity matrix here) can be used to obtain bounds on the iteration error norm $\|\mathbf{e}_h^{(k)}\|_{E_h} = \sqrt{(\mathbf{r}_h^{(k)})^T F_h^{-T} E_h F_h^{-1} \mathbf{r}_h^{(k)}}$. This involves computing the smallest eigenvalue λ_h and the largest eigenvalue Λ_h of the generalized symmetric positive-definite eigenvalue problem for E_h and $F_h^T F_h$. A popular choice of E_h for nonsymmetric linear systems is the symmetric part of the coefficient matrix F_h , that is, $\frac{F_h^T + F_h}{2}$ provided it is also positive-definite [3, chapter 6].

In light of (12), the strong and weak stopping criteria in preconditioned GMRES for solving nonsymmetric linear systems is: stop at the first iteration k^* such that either holds,

$$\|\mathbf{r}_h^{(k^*)}\|_2 \leq \frac{\sqrt{\lambda_h}}{\Lambda_h} \eta_h^{(k^*)} \text{ (strong stop); } \quad \|\mathbf{r}_h^{(k^*)}\|_2 \leq \frac{1}{\sqrt{\Lambda_h}} \eta_h^{(k^*)} \text{ (weak stop).} \quad (13)$$

Note that (13) are black-box stopping tests as opposed to the devised stopping test in [25, chapter 5]. As opposed to [1], (13) involves a posteriori error bounds and the strong and the weak stopping tests are black-box in the sense that these stopping tests can be applied without using any user-specified safety factors (constants) or tuning parameters. This is in contrast to [11, 14, 15] too, where the application of their stopping tests requires user supplied safety factors and tuning parameters.

It is further proposed in this paper that (13) can be used in suboptimal (but memory efficient over GMRES) solvers like BICGSTAB(ℓ) [21], TFQMR [6] etc., for which little convergence theory exists currently. This is provided that breakdowns in such solvers are handled adequately (see [7]) and these solvers converge at least to the order of the PDE approximation error. Observe that the Euclidean norm of the residual in suboptimal solvers is not monotonically decreasing. However, practitioners still use it as a convergence indicator in such solvers. So, the balanced black-box stopping criteria (13) based on Euclidean residual norm is also employed here in these solvers.

3 Solution methodology for nonlinear solvers

It follows from (3) that at any nonlinear iterative step $l + 1$,

$$\|u_h^{(l+1)} - u_h^{(l)}\|_{\mathcal{E}} = \|\delta u_h^{(l)}\|_{\mathcal{E}}. \quad (14)$$

Note that since norm of difference is greater than or equal to the difference of norms, $\|u_h^{(l+1)} - u_h^{(l)}\|_{\mathcal{E}} \geq \|u_h^{(l+1)} - u\|_{\mathcal{E}} - \|u_h^{(l)} - u\|_{\mathcal{E}}$, therefore (14) becomes

$$\|u_h^{(l+1)} - u\|_{\mathcal{E}} \leq \|u_h^{(l)} - u\|_{\mathcal{E}} + \|\delta u_h^{(l)}\|_{\mathcal{E}}. \quad (15)$$

If $\eta_h^{(l)}$ denotes ‘tight’ a posteriori approximation error estimator at l th nonlinear iterative step such that $c\eta_h^{(l)} \leq \|u - u_h^{(l)}\|_{\mathcal{E}} \leq C\eta_h^{(l)}$, with $\frac{C}{c} \sim O(1)$, then (15) can be rewritten as

$$\eta_h^{(l+1)} \simeq \eta_h^{(l)} + \|\delta u_h^{(l)}\|_{\mathcal{E}}. \quad (16)$$

Note that the sequence $\{\eta_h^{(l)}\}$ ultimately converges to true a posteriori approximation error estimate η_h . So $\forall l \geq \hat{l}$ (say), $\eta_h^{(l)}$ are $\eta_h^{(l+1)}$ are essentially the same. Using this idea, one can stop the nonlinear iteration when the contribution from $\|\delta u_h^{(l)}\|_{\mathcal{E}}$ in (16) is insignificant. Thus, in light of the discussion in section 2, stop the nonlinear iteration at the smallest value l^* such that

$$\|\delta u_h^{(l^*)}\|_{\mathcal{E}} \leq \eta_h^{(l^*+1)}. \quad (17)$$

Computational logistics for nonlinear solver stopping test are the following:

- At the l th nonlinear iteration, $\|\delta u_h^{(l)}\|_{\mathcal{E}}$ is cheaply computable using some matrix-vector products; see for example subsection 5.3.
- The a posteriori error estimate $\eta_h^{(l)}$ should be computed periodically (say every 4–5 iterations) to minimize its impact on the overall algorithmic cost.

An iterative (linear or nonlinear) solver employing the balanced stopping methodology that is presented in this paper will be a black-box solver. In the various Krylov solvers used in this paper, the eigenvalue problem for finding λ_h , Λ_h will be specified explicitly by identifying suitable E_h and S_h . For the remainder of this paper F_h will be assumed to be nonsymmetric. This work has appeared as chapters 4 and 5 in first author’s PhD thesis [17].

4 Convection-diffusion equations

Convection-diffusion equations are used for modelling various phenomena such as the temperature of a fluid moving along a heated wall, the transfer and diffusion of pollutants, etc.; see [5, chapter 18].

Following the notation in [3, p. 234], the steady-state scalar convection-diffusion solution $u(\vec{x}) : \Omega \rightarrow \mathbb{R}$ satisfies

$$-\nabla \cdot \epsilon(\vec{x}) \nabla u(\vec{x}) + \vec{w}(\vec{x}) \cdot \nabla u(\vec{x}) = f(\vec{x}), \quad \forall \vec{x} \in \Omega \subset \mathbb{R}^d \quad (18a)$$

$$u(\vec{x}) = g_D(\vec{x}), \quad \forall \vec{x} \in \partial\Omega_D, \quad (18b)$$

$$\nabla u(\vec{x}) \cdot \vec{n} = g_N(\vec{x}), \quad \forall \vec{x} \in \partial\Omega_N = \partial\Omega \setminus \partial\Omega_D. \quad (18c)$$

Here Ω is the spatial domain, \vec{w} denotes the wind, and $\epsilon := \kappa I$ is the isotropic permeability tensor, $\kappa : \Omega \rightarrow \mathbb{R}$. The quantities f, g_D, g_N are given functions and \vec{n} denotes the normal to boundary $\partial\Omega$, which is the union of the Dirichlet ($\partial\Omega_D$) and the Neumann ($\partial\Omega_N$) spatial boundary. For the simplicity of exposition, the diffusion coefficient $\epsilon > 0$ will be assumed to be independent of the spatial coordinates. Also, it will be assumed that $\nabla \cdot \vec{w} = 0$.

The Galerkin FEM formulation of (18) is to find $u_h \in S_E^h$ such that

$$\epsilon \int_{\Omega} (\nabla u_h \cdot \nabla v_h) + \int_{\Omega} (\vec{w} \cdot \nabla u_h) v_h = \int_{\Omega} f v_h + \epsilon \int_{\partial\Omega_N} g_N v_h, \quad \forall v_h \in S_0^h, \quad (19)$$

where S_E^h and S_0^h are finite dimensional subspaces of H_E^1 and $H_{E_0}^1$ respectively. Here,

$$\begin{aligned} H_E^1(\Omega) &:= \{v \in H^1(\Omega) \mid v = g_D \text{ on } \partial\Omega_D\}, \\ H_{E_0}^1(\Omega) &:= \{v \in H^1(\Omega) \mid v = 0 \text{ on } \partial\Omega_D\}, \\ H^1(\Omega) &:= \{u \in L^2(\Omega) \mid D^\alpha u \in L^2(\Omega), \forall |\alpha| \leq 1\}, \end{aligned}$$

where D^α is distributional derivative of u , $|\alpha| := \sum_{i=1}^d \alpha_i$, $\alpha = (\alpha_1, \dots, \alpha_d)$ is a multiindex, see [12, p. 434].

The FEM formulation (19) gives rise to the following system of linear equations with given coefficient matrix F_h , given right-hand-side vector \mathbf{b}_h , and unknown vector \mathbf{x}_h ,

$$F_h \mathbf{x}_h = \mathbf{b}_h \iff M_h^{-1} F_h \mathbf{x}_h = M_h^{-1} \mathbf{b}_h, \quad (20)$$

where M_h is a preconditioner. The coefficient matrix $F_h = \epsilon A_h + N_h$ for finite elements of any order. The matrix A_h is symmetric and positive-definite provided Dirichlet boundary conditions exist over an interval ($\int_{\partial\Omega_D} \neq 0$), however small. The matrix N_h is a skew-symmetric matrix [3, p. 271]. Thus, F_h is a nonsymmetric matrix that will be assumed to be invertible throughout this paper.

Wathen [24] advocates that a natural norm for a function u in the Sobolev space $H_{E_0}^1$ is the L^2 norm of its gradient, that is, $\|\nabla u\|_{L^2(\Omega)} := \sqrt{\int_{\Omega} (\nabla u_h)^2}$. An alternative popular norm known as the streamline diffusion norm is also discussed in [3, p. 252]. This norm arises when the streamline diffusion method introduced by [10] is used for overcoming the drawbacks associated with the Galerkin discretization.¹ This leads to a slightly different FEM formulation to (19) known as the streamline diffusion FEM formulation [3, p. 252] and the corresponding streamline diffusion norm is $\|u_h\|_{\text{sd}} := (\epsilon \|\nabla u_h\|_{L^2(\Omega)}^2 + \delta \|\vec{w} \cdot \nabla u_h\|_{L^2(\Omega)}^2)^{1/2}$. Here δ denotes the stabilization parameter [3, p. 253]. For convection dominated problems, that is, for large Peclet numbers (small ϵ), the solution u_h is dominated by its behavior along the streamlines, and hence $\|u_h\|_{\text{sd}}$ which involves the streamline derivative $\|\vec{w} \cdot \nabla u_h\|_{L^2(\Omega)}$ is a more meaningful measure than $\|\nabla u_h\|_{L^2(\Omega)}$.

The IFISS toolbox employs streamline diffusion stabilization for solving (18), but measures errors in the L^2 norm of the gradient. The balanced stopping test will be based on this norm. However, the stopping methodology can easily be modified to cater to the streamline diffusion norm.

¹ Galerkin approximation for (18) is inaccurate if the mesh is not fine enough to resolve the layers in the solution and these inaccuracies may also propagate and pollute the approximated solution in regions where the exact solution is well behaved. An alternative way to handle boundary layers is by using Shishkin grids; see [19].

4.1 A posteriori approximation error estimation

The posteriori approximation error estimator η_h for this problem is computed in the IFISS software as $\eta_h := (\sum_{T \in \mathcal{T}_h} \eta_T^2)^{1/2}$, where T denotes an element of the finite element discretization and $\eta_T := \|\nabla e_T\|_{L^2(T)}$ denotes the (local) elemental a posteriori error estimate on the element T . Here $e_T := (u - u_h)_T$ is the finite element solution of a (local) Poisson problem on T , whose computation requires solving for a local 5×5 linear system (see [3, p. 264] for more details). Thus, calculating η_T and hence η_h is computationally quite cheap.

The employed η_h in this paper for the deterministic convection-diffusion equations is reliable but need not always be efficient. It is reliable in the sense that the global upper bound on the true error does not depend on the mesh parameter h and the diffusion parameter ϵ . However, it might not always be possible that the a posteriori error estimate is a lower bound on the local (elemental) approximation error [3, proposition 6.11, p. 265]. According to [3, p. 265], efficiency issue is generic for any local error estimator whenever boundary layers are not resolved by the FEM approximation. Thus, the estimated error is likely to be a gross overestimate of the true error, if the mesh Peclet number [3, p. 253] $\mathcal{P}_h \gg 1$. This overestimation is reduced if the solution is computed using streamline diffusion stabilization (see [3, pp. 265–266]).

To demonstrate that the employed a posteriori error estimator (for streamline diffusion stabilization) is a ‘close’ estimate of the approximation error, some computational results are presented for the test problem described in subsection 4.2. Using a reference solution, the a posteriori error η_h and the actual approximation error $\|\nabla(u_{ref} - u_h)\|_{L^2(\Omega)} := \sqrt{\int_{\Omega} \nabla(u_{ref} - u_h)^2}$ are tabulated in Table 1 for a sequence of uniform grids. Since the exact solution to the model problem is

Table 1 Approximation errors, a posteriori errors, and effectivity indices for convection-diffusion test problem on uniform grids.

h	η_h	$\ \nabla(u_{ref} - u_h)\ _{L^2(\Omega)}$	β_{eff}	$\mathcal{P}_h^{\text{max}}$
1/16	0.6238	0.5911	1.06	3.87
1/32	0.3015	0.2952	1.02	1.97
1/64	0.1487	0.1468	1.01	0.99
1/128	0.0740	0.0716	1.03	0.50
1/256	0.0370	0.0320	1.16	0.25

not available, a reference solution u_{ref} is computed on a fine ($h = 1/512$) spatial 1024×1024 uniform grid. This reference solution is then compared with the computed FEM solution u_h (which is linearly interpolated using MATLAB `interp2` function for compatible comparison with the reference solution) for grids with $h = 1/16, 1/32, 1/64, 1/128, 1/256$. The corresponding effectivity index, that is, $\beta_{\text{eff}} = \frac{\eta_h}{\|\nabla(u_{ref} - u_h)\|_{L^2(\Omega)}}$ is also presented. The column for β_{eff} (even for maximum mesh Peclet number $\mathcal{P}_h^{\text{max}} \gg 1$) in Table 1 indicates that the a posteriori error estimator is a close estimate of the approximation error.

4.2 Experimental results

From Table 1, observe that the computed a posteriori approximation errors do not overestimate the corresponding approximation error appreciably. In light of Remark 2, this implies that the weak stopping test in (13) can be used as the balanced black-box stopping test in GMRES, BICGSTAB(ℓ), and TFQMR for solving (20). Thus, one needs to compute the largest eigenvalue Λ_h of the generalized eigenvalue problem for $\frac{F_h^T + F_h}{2\epsilon}$ and $F_h^T F_h$. Observe that the symmetric positive-definite part of F_h here is $A_h = \frac{F_h^T + F_h}{2\epsilon}$ and not just $\frac{F_h^T + F_h}{2}$ as mentioned in section 2.

The results are presented for GMRES, BICGSTAB(2), and TFQMR together with incomplete LU (ILU) and algebraic multigrid (AMG) preconditioning which are built in IFISS [3, chapter 7]. The choice $\ell = 2$ for BICGSTAB(ℓ) is quite popular and widespread among practitioners; see [3, p. 296]. Roundoff errors might pollute the residual norm computed from short-term recurrences for suboptimal Krylov solvers. In order to avoid these inaccuracies, $\|\mathbf{r}_h^{(k)}\|_2$ is computed after forming the residual $\mathbf{r}_h^{(k)} = \mathbf{b}_h - F_h \mathbf{x}_h^{(k)}$ explicitly.

The convection-diffusion problem (18) is defined on $\Omega = (-1, 1) \times (-1, 1)$ with the source function $f(x_1, x_2) = 0, \forall (x_1, x_2) \in \Omega$. Discontinuous Dirichlet boundary conditions exist leading to the formation of boundary layers near the two (top right and bottom right) corners $x_1 = 1, x_2 = \pm 1$ of the domain and the wind \vec{w} is recirculating [3, p. 240]. This problem can be set up by choosing test problem 4 after running the driver `cd.testproblem` in the IFISS software in MATLAB. Rectangular piecewise bilinear (\mathbf{Q}_1) finite elements are used on a sequence of uniform grids. The viscosity parameter $\epsilon = 1/64$ is fixed and the built in value of the stabilization parameter δ is used; see [3, p. 253]. Note that the employed a posteriori approximation error estimator η_h (and the exact error) for this problem does not converge to zero as $h \rightarrow 0$ [17, p. 98, Table 4.1] (because u is not actually in $H^1(\Omega)$). Here, instead of solving this ill-posed problem, a regularised version is solved (with a smoothed discontinuity in the boundary data). This regularisation ensures (see Table 1, page 9) that the employed a posteriori error estimator η_h (and hence the exact error) converges to zero as $h \rightarrow 0$.

Let \mathbf{x}_h denote the reference algebraic solution (obtained using MATLAB Gaussian elimination) on grid with spatial discretization parameter h . From \mathbf{x}_h , the reference a posteriori error estimate η_h is computed. The starting vector $\mathbf{x}_h^{(0)}$ is generated using the MATLAB function `rand`. Also, let $\eta_h^{(k^*)}$ denote the a posteriori error estimate at the balanced stopping iteration k^* (obtained from the balanced stopping solution $\mathbf{x}_h^{(k^*)}$) and $e_{\eta_h}^* := |\eta_h - \eta_h^{(k^*)}|$.

Table 2 GMRES iteration counts and errors for ILU (left), AMG (right) preconditioning on uniform grids for discrete convection-diffusion system.

h	k_1	k_2	k^*	$e_{\eta_h}^*$
1/16	18	24	10	1.6e-3
1/32	42	54	31	1.2e-4
1/64	112	143	99	1.0e-5
1/128	286	372	284	3.2e-6

h	k_1	k_2	k^*	$e_{\eta_h}^*$
1/16	7	11	3	2.2e-3
1/32	7	12	4	1.2e-4
1/64	8	14	6	2.6e-5
1/128	7	14	7	3.0e-5

To show the effectiveness of the balanced stopping test, the iteration counts k^* needed to satisfy the balanced stopping test have been compared with iteration counts k_1, k_2 needed to satisfy a fixed relative residual $\|\mathbf{r}_h^{(k)}\|_2/\|\mathbf{r}_h^{(0)}\|_2$ reduction tolerance of $1\mathbf{e}-6$ (which is the default tolerance in MATLAB solvers) and $1\mathbf{e}-9$ respectively.² These tolerance values are a realistic user-input tolerance choices in the absence of a balanced stopping test. The user will not know in general the stopping point k^* a priori and is more likely to provide a tighter or coarser tolerance than actually required leading to unnecessary computations or premature stopping respectively.

The values of quantities mentioned above are tabulated in the Tables 2, 3, 4 for each preconditioner on every grid level. The insights from these quantities are quite generic, which are summarized in the following paragraphs.

Table 3 BICGSTAB(2) iteration counts and errors for ILU (left), AMG (right) preconditioning on uniform grids for discrete convection-diffusion system.

h	k_1	k_2	k^*	$e_{\eta_h}^*$
1/16	12	16	5	1.4e-3
1/32	30	38	16	2.1e-6
1/64	84	106	44	1.3e-5
1/128	228	292	140	3.6e-7

h	k_1	k_2	k^*	$e_{\eta_h}^*$
1/16	4	6	2	2.6e-4
1/32	4	8	3	1.4e-5
1/64	6	8	4	4.9e-5
1/128	6	10	5	5.6e-7

The $e_{\eta_h}^*$ columns show that the sequence $\{\eta_h^{(k)}\}$ has converged with a good accuracy to the reference a posteriori error estimate η_h at the balanced stopping iteration.

A comparison of the corresponding columns for ILU iteration counts shows that for the same approximation error, a significant number of iterations is saved by using the balanced stopping test. This would result in savings in iterations (when compared to fixed relative residual reduction tolerance tighter than $1\mathbf{e}-6$ iteration counts) especially if one were to solve the preconditioned linear systems arising from adaptive finite element for the chosen problem parameters. The linear systems that are solved are of size: 1089×1089 , 4225×4225 , 16641×16641 , and 66049×66049 (some of which are large enough for the demonstration of the balanced stopping criteria.)

Table 4 TFQMR iteration counts and errors for ILU (left), AMG (right) preconditioning on uniform grids for discrete convection-diffusion system.

h	k_1	k_2	k^*	$e_{\eta_h}^*$
1/16	29	37	15	1.4e-4
1/32	72	85	50	4.4e-4
1/64	200	231	166	1.7e-4
1/128	574	647	531	9.7e-7

h	k_1	k_2	k^*	$e_{\eta_h}^*$
1/16	8	12	4	1.1e-4
1/32	9	14	5	2.2e-5
1/64	10	18	7	1.5e-5
1/128	9	20	8	1.7e-5

² Note that for BICGSTAB2, k_1, k_2 are the iteration counts for the built in stopping criterion in the IFISS implementation of BICGSTAB2 to tolerances of $1\mathbf{e}-6$, $1\mathbf{e}-9$ respectively. The iteration counts k^* for BICGSTAB2 is obtained by incorporating the weak stopping criterion in this IFISS implementation of BICGSTAB2.

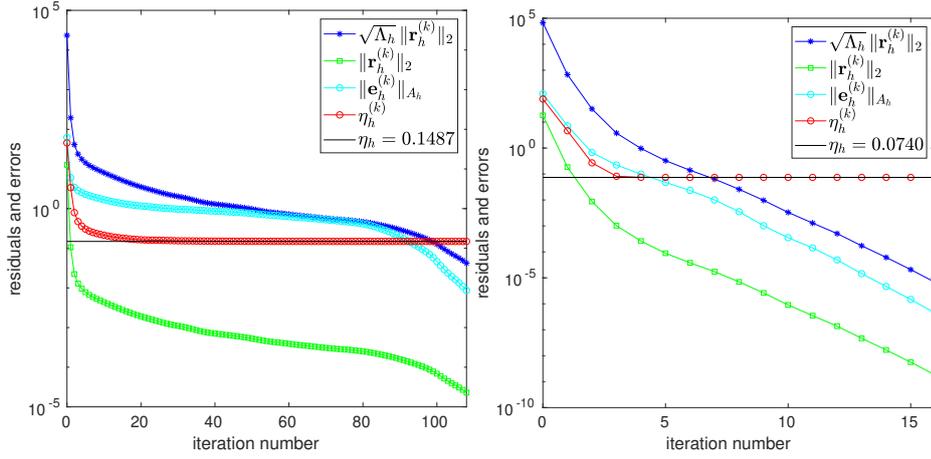


Fig. 1 Errors vs iteration number for convection-diffusion test problem for ILU preconditioned GMRES with $h = 1/64$ (left) and AMG preconditioned GMRES with $h = 1/128$ (right).

In the case of AMG (which is a better preconditioner than ILU), not much savings of iterations is achieved by using the balanced black-box stopping test. However, using the balanced black-box stopping test does ensure that the employed solver has not stopped prematurely.

Using GMRES over BICGSTAB(2) or TFQMR could be memory extensive in terms of storage. However, very little convergence theory exists for suboptimal solvers like BICGSTAB(2), TFQMR etc. The balanced black-box stopping test provides a stopping point for such suboptimal Krylov solvers. Indeed this is crucially dependent on the fact that these suboptimal solvers do not break down prematurely.

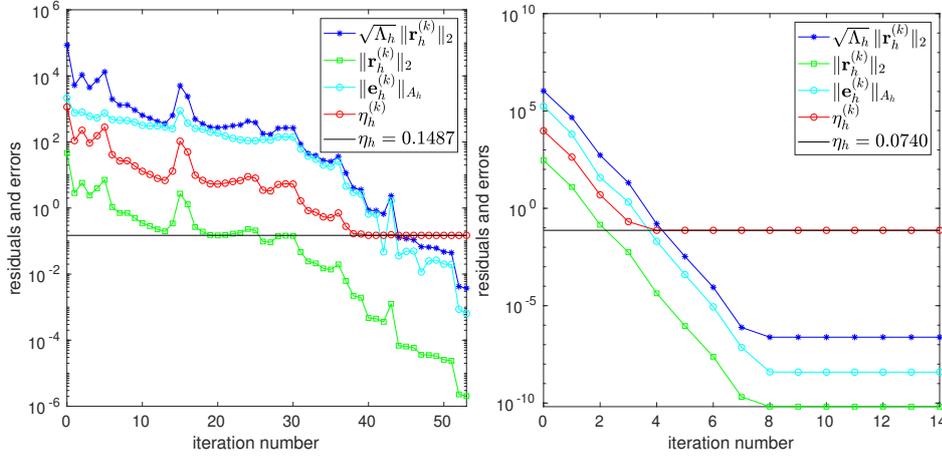


Fig. 2 Errors vs iteration number for convection-diffusion test problem for ILU preconditioned BICGSTAB(2) with $h = 1/64$ (left) and AMG preconditioned BICGSTAB(2) with $h = 1/128$ (right).

In order to gain further insight from the numerical experiments, the evolution of the following quantities— $\eta_h^{(k)}$, $\|\mathbf{e}_h^{(k)}\|_{A_h}$, $\|\mathbf{r}_h^{(k)}\|_2$, and the (weak) algebraic error bound $\sqrt{\Lambda_h}\|\mathbf{r}_h^{(k)}\|_2$ is also plotted. The solver stops when the $\sqrt{\Lambda_h}\|\mathbf{r}_h^{(k)}\|_2$ curve is below the $\eta_h^{(k)}$ curve. Indeed this is the case in all plots of Figures 1, 2, 3. In order to illustrate this convergence, iterations have been continued for nine more steps after stopping in each plot. This also illustrates stopping at the ‘correct’ iteration, that is, at iteration k^* , the sequence $\{\eta_h^{(k)}\}$ has converged to η_h on each plot. Further, the plots show that as soon as $\|\mathbf{e}_h^{(k)}\|_{A_h} \leq \eta_h$, $\{\eta_h^{(k)}\}$ converges to η_h . This is consistent with observations from (8) and Remark 1. Generally, both $\|\mathbf{e}_h^{(k)}\|_{A_h}$, η_h are unknown or difficult to compute. So, a priori knowledge of the balanced stopping step is generally difficult.

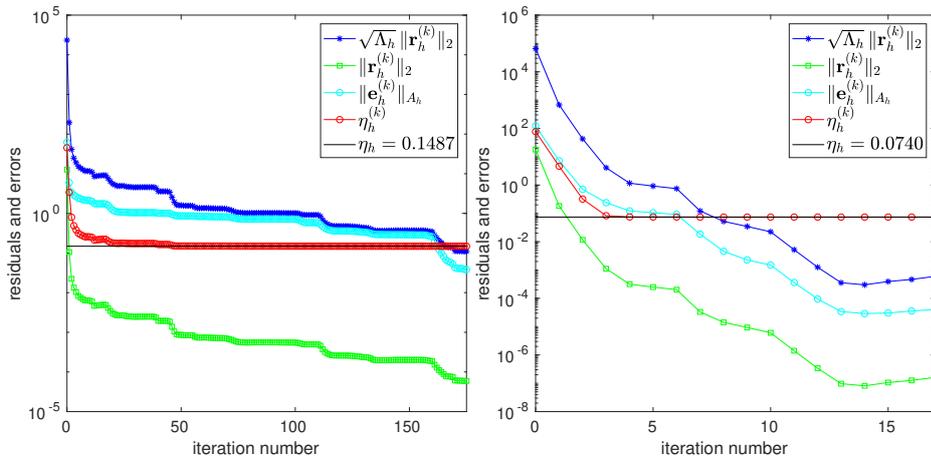


Fig. 3 Errors vs iteration number for convection-diffusion test problem for ILU preconditioned TFQMR with $h = 1/64$ (left) and AMG preconditioned TFQMR with $h = 1/128$ (right).

Before proceeding to the next section, tabulated in Table 5 are the values of Λ_h for various h , along with the corresponding average cputimes $t_{\mathbf{eigs}}$ (in seconds) taken by MATLAB `eigs` to compute them. Also, tabulated are the corresponding average cputimes t_{η_h} (in seconds) taken by the IFISS software in MATLAB to compute η_h once.³ The t_{η_h} column in Table 5 indicates that the overall algorithmic cost can be reduced if $\eta_h^{(k)}$ is computed periodically (say every 10–20 iterations). Note that the $\eta_h^{(k)}$ has been computed at every iteration k in the plots here only for the demonstration of the balanced stopping test. From the values of Λ_h (which become almost 4 times their value on the immediate previous grid), it seems that the timings (here $t_{\mathbf{eigs}}$) for computing Λ_h could be reduced by devising a cheaper method for estimating Λ_h by exploiting properties of the underlying problem. For example, here, one could compute Λ_h on a coarse mesh and use the observation

³ The average cputimes were computed for each h by averaging over cputimes generated from 10 independent runs (of `eigs` and the IFISS script for computing η_h) for each h .

Table 5 Computed Λ_h , average **eigs** cputimes (in seconds), and average cputimes (in seconds) to compute η_h once for various h .

h	Λ_h	t_{eigs}	t_{η_h}
1/16	212787.925	0.043	0.09
1/32	850190.539	0.138	0.26
1/64	3399301.169	0.379	0.78
1/128	13595670.080	1.394	2.71

that Λ_h grows like h^{-2} to estimate Λ_h on finer meshes.⁴ However, as mentioned in section 2, devising generic computationally cheaper methods for estimating Λ_h than using **eigs** (which is currently the state-of-the-art for estimating some eigenvalues of generic sparse eigenvalue problems) remains a challenging research question, which is not addressed in this paper.

5 Navier–Stokes equations

Navier–Stokes equations form the fundamental model of an incompressible Newtonian fluid such as air etc [3, p. 333 ff.]. The steady-state Navier–Stokes solution (\vec{u}, p) is defined on a spatial domain $\Omega \subset \mathbb{R}^d$, ($d = 2, 3$), where the vector valued velocity function $\vec{u}(\vec{x}) : \Omega \rightarrow \mathbb{R}^d$ and the scalar valued pressure function $p(\vec{x}) : \Omega \rightarrow \mathbb{R}$ satisfy

$$-\nu \nabla \cdot \nabla \vec{u}(\vec{x}) + \vec{u}(\vec{x}) \cdot \nabla \vec{u}(\vec{x}) + \nabla p(\vec{x}) = \vec{f}(\vec{x}), \quad \forall \vec{x} \in \Omega, \quad (21a)$$

$$\nabla \cdot \vec{u}(\vec{x}) = 0, \quad \forall \vec{x} \in \Omega, \quad (21b)$$

$$\vec{u}(\vec{x}) = \vec{w}(\vec{x}), \quad \forall \vec{x} \in \partial\Omega_D, \quad (21c)$$

$$\nu \nabla \vec{u}(\vec{x}) \cdot \vec{n} - \vec{n} p(\vec{x}) = \vec{0}, \quad \forall \vec{x} \in \partial\Omega_N. \quad (21d)$$

The functions \vec{f} , \vec{w} are given and $\partial\Omega_D$, $\partial\Omega_N$ are the Dirichlet and Neumann parts respectively of the spatial boundary $\partial\Omega$. Kinematic velocity $\nu > 0$ is given and \vec{n} denotes the outward normal to $\partial\Omega$. The presence of the convective term gives the Navier–Stokes equations a nonlinear behavior.

The mixed FEM formulation of (21) requires to find $\vec{u}_h \in \mathbf{X}_E^1 \subset \mathbf{H}_E^1(\Omega)$, $p_h \in M^h \subset L^2(\Omega)$ such that $\forall \vec{v}_h \in \mathbf{X}_{E_0}^h \subset \mathbf{H}_{E_0}^1(\Omega)$, $\forall q_h \in M^h$,

$$\begin{aligned} \nu \int_{\Omega} \nabla \vec{u}_h : \nabla \vec{v}_h + \int_{\Omega} (\vec{u}_h \cdot \nabla \vec{u}_h) \cdot \vec{v}_h - \int_{\Omega} p_h (\nabla \cdot \vec{v}_h) &= \int_{\Omega} \vec{f} \cdot \vec{v}_h, \\ \int_{\Omega} q_h (\nabla \cdot \vec{u}_h) &= 0, \end{aligned} \quad (22)$$

where $\mathbf{H}_E^1(\Omega)$, $\mathbf{H}_{E_0}^1(\Omega)$ are the vector counterparts of spaces $H_E^1(\Omega)$, $H_{E_0}^1(\Omega)$ (defined in (19)) respectively and $\nabla \vec{u} : \nabla \vec{v}$ denotes componentwise dot product. The solution of (22) involves nonlinear iterations that requires solving a linearized problem at each iterative step.

⁴ This observation is clear from Λ_h values in Table 5, however a rigorous mathematical proof is still under research.

Starting with an initial guess $(\vec{u}_h^{(0)}, p_h^{(0)}) \in \mathbf{X}_E^1 \times M^h$, a sequence of iterates $\{(\vec{u}_h^{(l+1)}, p_h^{(l+1)})\}$, $(l = 0, 1, \dots)$ is constructed satisfying (22) such that [3, pp. 344, 341],

$$\vec{u}_h^{(l+1)} = \vec{u}_h^{(l)} + \delta \vec{u}_h^{(l)}, \quad p_h^{(l+1)} = p_h^{(l)} + \delta p_h^{(l)}. \quad (23)$$

Let $\{\vec{\phi}_j\}_{j=1}^{n_u}$ be a basis for $\mathbf{X}_{E_0}^h$. Then any arbitrary $\delta \vec{u}_h^{(l)} \in \mathbf{X}_{E_0}^h$ can be expressed as $\delta \vec{u}_h^{(l)} = \sum_{j=1}^{n_u} \Delta u_j^{(l)} \vec{\phi}_j$, $\Delta u_j^{(l)} \in \mathbb{R}$. Also, $\{\vec{\phi}_j\}_{j=1}^{n_u}$ can be extended (loosely speaking⁵) to form a basis for \mathbf{X}_E^h , so that any $\vec{u}_h^{(l)} \in \mathbf{X}_E^h$ can be expanded as $\vec{u}_h^{(l)} = \sum_{j=1}^{n_u+n_\partial} u_j^{(l)} \vec{\phi}_j$, $u_j^{(l)} \in \mathbb{R}$, where the term $\sum_{j=n_u+1}^{n_u+n_\partial} u_j^{(l)} \vec{\phi}_j$ interpolates the boundary data on $\partial\Omega_D$.

Similarly, if $\{\psi_k\}_{k=1}^{n_p}$ is a basis for M^h , then any $p_h^{(l)}, \delta p_h^{(l)} \in M^h$ has an expression $p_h^{(l)} = \sum_{k=1}^{n_p} p_k^{(l)} \psi_k$, $\delta p_h^{(l)} = \sum_{k=1}^{n_p} \Delta p_k^{(l)} \psi_k$, $p_k^{(l)}, \Delta p_k^{(l)} \in \mathbb{R}$. Since $\vec{u}_h^{(l)}, p_h^{(l)}$ are known from the previous iterative step, their basis coefficients are known too.

Using (23) along with these basis expansions in (22) together with linearization [3, p. 345] leads to solving for the following discrete Newton system of linear equations at the l th nonlinear iterative step,

$$\begin{bmatrix} \nu \mathbf{A}_h + \mathbf{N}_h^{(l)} & B_h^T \\ B_h & O \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_h^{(l)} \\ \Delta \mathbf{p}_h^{(l)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h^{(l)} \\ \mathbf{g}_h^{(l)} \end{bmatrix}, \quad (24)$$

where $\Delta \mathbf{u}_h^{(l)} = [\Delta u_1^{(l)}, \dots, \Delta u_{n_u}^{(l)}]^T$, $\Delta \mathbf{p}_h^{(l)} = [\Delta p_1^{(l)}, \dots, \Delta p_{n_p}^{(l)}]^T$. For the entries of \mathbf{A}_h , B_h , $\mathbf{N}_h^{(l)}$, $\mathbf{W}_h^{(l)}$, $\mathbf{f}_h^{(l)}$, and $\mathbf{g}_h^{(l)}$, see [3, p. 348]. The symmetric positive-definite matrix \mathbf{A}_h (vector-Laplacian matrix) is the block diagonal matrix with the usual FEM stiffness matrix on its diagonals and $\mathbf{N}_h^{(l)}$ is the vector convection matrix (the scalar versions of both were introduced in (20)). Note that the matrices are denoted in bold face here to emphasize their vector dependence.

Dropping the Newton derivative matrix $\mathbf{W}_h^{(l)}$ in (24) results in the linear system arising from Picard iteration, which is,

$$\begin{bmatrix} \nu \mathbf{A}_h + \mathbf{N}_h^{(l)} & B_h^T \\ B_h & O \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_h^{(l)} \\ \Delta \mathbf{p}_h^{(l)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h^{(l)} \\ \mathbf{g}_h^{(l)} \end{bmatrix}. \quad (25)$$

5.1 A posteriori error estimation

Computation of a posteriori error estimates for the Navier–Stokes mixed FEM formulation entails solving local Poisson problems for each component of velocity [3, p. 352 ff.]. In fact it has been stated in [3, proposition 8.9, p. 354] that a posteriori error estimators for stabilized $\mathbf{Q}_1\text{-}\mathbf{P}_0$ rectangular finite elements are reliable in the sense that the global upper bound on the approximation error does not depend on the parameters of the continuous problem. Thus, results presented in the subsection 5.4 are based on stabilized $\mathbf{Q}_1\text{-}\mathbf{P}_0$ rectangular finite elements.

⁵ \mathbf{X}_E^h is not a vector space unless its elements (which are functions) are zero on the boundary.

5.2 Balanced black-box stopping tests for linear solver

Since, the coefficient matrix in (24) or (25) is nonsymmetric, GMRES, BICGSTAB(ℓ) etc., will be used to solve them. Hence the stopping methodology developed in section 2 can be applied in solving (24) or (25). At the l th nonlinear iteration, the iteration residual at k th step of the linear solver is

$$\mathbf{r}_h^{(l_k)} = [(\mathbf{f}_h^{(l)})^T, (\mathbf{g}_h^{(l)})^T]^T - F_h^{(l)}[(\Delta \mathbf{u}_h^{(l_k)})^T, (\Delta \mathbf{p}_h^{(l_k)})^T]^T.$$

Here, $F_h^{(l)}$ denotes the (Newton or Picard) nonsymmetric coefficient matrix of the (linearized) discrete Navier–Stokes system in (24) or (25) at the l th (nonlinear) iterative step. Proceeding as in section 2, at the l th nonlinear iteration, linear solvers GMRES, BICGSTAB(ℓ) etc., solving linear systems (24) or (25) are stopped at the first iteration l_{k^*} such that either holds,

$$\|\mathbf{r}_h^{(l_{k^*})}\|_2 \leq \frac{\sqrt{\lambda_h^{(l)}}}{\Lambda_h^{(l)}} \eta_h^{(l_{k^*})} \text{ (strong stop); } \quad \|\mathbf{r}_h^{(l_{k^*})}\|_2 \leq \frac{1}{\sqrt{\Lambda_h^{(l)}}} \eta_h^{(l_{k^*})} \text{ (weak stop)}. \quad (26)$$

Here $\lambda_h^{(l)}$ and $\Lambda_h^{(l)}$ are the smallest and the largest eigenvalues respectively of the generalized eigenvalue problem for E_h and $(F_h^{(l)})^T F_h^{(l)}$.

A natural norm for measuring errors arising from the mixed FEM formulation (22) is $\|(\vec{u}, p)\|_{\mathcal{E}} := \|\nabla \vec{u}\|_{L^2(\Omega)} + \|p\|_{L^2(\Omega)}$, $\forall (\vec{u}, p) \in \mathbf{H}_{E_0}^1(\Omega) \times L^2(\Omega)$. The a posteriori error estimator $\eta_h^{(l_k)}$ that is employed in (26) is equivalent to the total error (approximation error at the k th iteration) in the sense that

$$c \eta_h^{(l_k)} \leq \|\nabla(\delta \vec{u}^{(l)} - \delta \vec{u}_h^{(l_k)})\|_{L^2(\Omega)} + \|\delta p^{(l)} - \delta p_h^{(l_k)}\|_{L^2(\Omega)} \leq C \eta_h^{(l_k)}, \quad (27)$$

with $\frac{c}{C} \sim O(1)$. At the l th iterative step $\vec{u}_h^{(l)}, p_h^{(l)}$ is known. It follows from (23) that the correction $\delta \vec{u}_h^{(l)} = \vec{u}_h^{(l+1)} - \vec{u}_h^{(l)}$ and $\delta p_h^{(l)} = p_h^{(l+1)} - p_h^{(l)}$. This implies that (24) or (25) solves for the basis coefficients of $(\vec{u}_h^{(l+1)}, p_h^{(l+1)})$. Thus, essentially one can use the same a posteriori approximation error estimator to estimate approximation errors a posteriori for $(\delta \vec{u}_h^{(l)}, \delta p_h^{(l)})$ as those for $(\vec{u}_h^{(l+1)}, p_h^{(l+1)})$.⁶

5.3 A balanced black-box stopping test for nonlinear solver

Using $\|(\vec{u}_h, p_h)\|_{\mathcal{E}} := \|\nabla \vec{u}_h\|_{L^2(\Omega)} + \|p_h\|_{L^2(\Omega)}$, if (\vec{u}, p) denotes the true solution, then following section 3 leads to

$$\begin{aligned} & \|\nabla(\vec{u}_h^{(l+1)} - \vec{u})\|_{L^2(\Omega)} + \|(p_h^{(l+1)} - p)\|_{L^2(\Omega)} \\ & \leq \left(\|\nabla(\vec{u}_h^{(l)} - \vec{u})\|_{L^2(\Omega)} + \|p_h^{(l)} - p\|_{L^2(\Omega)} \right) + \left(\|\nabla \delta \vec{u}_h^{(l)}\|_{L^2(\Omega)} + \|\delta p_h^{(l)}\|_{L^2(\Omega)} \right). \end{aligned} \quad (28)$$

The vector norm $\|\cdot\|_{E_h}$ associated with $\|(\vec{u}_h, p_h)\|_{\mathcal{E}}$ is defined as

$$\|\mathbf{e}_h\|_{E_h} := \sqrt{\mathbf{e}_h^T E_h \mathbf{e}_h} = \sqrt{\mathbf{e}_1^T \mathbf{A}_h \mathbf{e}_1 + \mathbf{e}_2^T Q_h \mathbf{e}_2}, \quad \forall \mathbf{e}_h = [\mathbf{e}_1^T, \mathbf{e}_2^T]^T \in \mathbb{R}^{n_u + n_p},$$

⁶ This is not a rigorous mathematical statement. A proof for this statement is an ongoing research.

where $E_h := \begin{pmatrix} \mathbf{A}_h & \mathbf{O} \\ \mathbf{O} & Q_h \end{pmatrix}$ is a symmetric positive-definite matrix. The pressure mass matrix $Q_h = [q_{kj}]$, $q_{kj} := \int_{\Omega} \psi_k \psi_j \forall k, j = 1, \dots, n_p$. By construction, for a given approximation, E_h is independent of nonlinear iterative step l . Hence, it needs to be constructed only once for a nonlinear iterative procedure. Observe that unlike the convection-diffusion case, E_h is not simply the symmetric positive-definite part of the Navier–Stokes coefficient matrix in (24) or (25).

For any two nonnegative real numbers a and b [3, p. 213]

$$\sqrt{a+b} \leq \sqrt{a} + \sqrt{b} \leq \sqrt{2}\sqrt{a+b}. \quad (29)$$

Using (29), for any $(\vec{v}_h, q_h) \in \mathbf{X}_{E_0}^h \times M^h$,

$$\sqrt{\mathbf{v}_h^T \mathbf{A}_h \mathbf{v}_h + \mathbf{q}_h^T Q_h \mathbf{q}_h} \leq \|(\vec{v}_h, q_h)\|_{\mathcal{E}} \leq \sqrt{2} \sqrt{\mathbf{v}_h^T \mathbf{A}_h \mathbf{v}_h + \mathbf{q}_h^T Q_h \mathbf{q}_h}, \quad (30)$$

where $\mathbf{v}_h, \mathbf{q}_h$ are the coordinates of \vec{v}_h, q_h with respect to velocity and pressure basis respectively. From (30) it follows that

$$\|\nabla \delta \vec{u}_h^{(l)}\|_{L^2(\Omega)} + \|\delta p_h^{(l)}\|_{L^2(\Omega)} \simeq \sqrt{(\Delta \mathbf{u}_h^{(l)})^T \mathbf{A}_h \Delta \mathbf{u}_h^{(l)} + (\Delta \mathbf{p}_h^{(l)})^T Q_h \Delta \mathbf{p}_h^{(l)}}. \quad (31)$$

In presence of ‘tight’ a posteriori error estimator $\eta_h^{(l)}$, which is equivalent to the approximation error at the l th nonlinear iteration in the sense that

$$c \eta_h^{(l)} \leq \|\nabla(\vec{u}_h^{(l)} - \vec{u})\|_{L^2(\Omega)} + \|p_h^{(l)} - p\|_{L^2(\Omega)} \leq C \eta_h^{(l)}, \quad \text{with } \frac{C}{c} \sim O(1), \quad (32)$$

using (31) and (32) in (28) leads to

$$\eta_h^{(l+1)} \simeq \eta_h^{(l)} + \sqrt{(\Delta \mathbf{u}_h^{(l)})^T \mathbf{A}_h \Delta \mathbf{u}_h^{(l)} + (\Delta \mathbf{p}_h^{(l)})^T Q_h \Delta \mathbf{p}_h^{(l)}}. \quad (33)$$

Using the strong or weak stopping criterion (26) or any other stopping criterion for linear iteration, $(\Delta \mathbf{u}_h^{(l)}, \Delta \mathbf{p}_h^{(l)})$ is replaced by $(\Delta \mathbf{u}_h^{(l_{k^*})}, \Delta \mathbf{p}_h^{(l_{k^*})})^T$ in (33) which becomes

$$\eta_h^{(l+1)} \simeq \eta_h^{(l)} + \sqrt{(\Delta \mathbf{u}_h^{(l_{k^*})})^T \mathbf{A}_h \Delta \mathbf{u}_h^{(l_{k^*})} + (\Delta \mathbf{p}_h^{(l_{k^*})})^T Q_h \Delta \mathbf{p}_h^{(l_{k^*})}}. \quad (34)$$

Thus, in spirit of section 3, the nonlinear iteration is stopped at l^* which is the smallest value of $(l+1)$ such that

$$\sqrt{(\Delta \mathbf{u}_h^{(l_{k^*})})^T \mathbf{A}_h \Delta \mathbf{u}_h^{(l_{k^*})} + (\Delta \mathbf{p}_h^{(l_{k^*})})^T Q_h \Delta \mathbf{p}_h^{(l_{k^*})}} \leq \eta_h^{(l^*+1)}. \quad (35)$$

Note that alternative nonlinear iteration stopping strategies do exist, see [22] for more details. However, these are neither balanced nor black-box in the sense presented above.

The resulting algorithm NAVIER_NEWTON_GMRES is presented in Figure 4.

⁷ This k^* will in general be different for different l .

```

Algorithm: NAVIER_NEWTON_GMRES
given functions GMRES_Navier_balanced, matvecA, matvecQ, Navier_error_est
.....
solve the corresponding Stokes problem to obtain starting guess:  $(\vec{u}_h^{(0)}, p_h^{(0)})$ 
.....
for  $l = 0, 1, 2, \dots$  until convergence do

    Inner iteration (GMRES solver)
    % GMRES_Navier_balanced: solves (24) or (25) using preconditioned GMRES
      with (or without) stopping test (26)

    % Coefficient matrix  $F_h^{(l)}$ , right-hand-side  $[\mathbf{f}_h^{(l)T}, \mathbf{g}_h^{(l)T}]^T$ , preconditioner  $M_h^{(l)}$ 

    compute the vector of basis coefficients for  $\delta \vec{u}_h^{(l)}$  and  $\delta p_h^{(l)}$ :
     $[\Delta \mathbf{u}_h^{(l)T}, \Delta \mathbf{p}_h^{(l)T}]^T = \text{GMRES\_Navier\_balanced}(F_h^{(l)}, [\mathbf{f}_h^{(l)T}, \mathbf{g}_h^{(l)T}]^T, M_h^{(l)})$ 

    Outer iteration (Nonlinear solver)
    update solution:  $\vec{u}_h^{(l+1)} = \vec{u}_h^{(l)} + \delta \vec{u}_h^{(l)}$ ,  $p_h^{(l+1)} = p_h^{(l)} + \delta p_h^{(l)}$ 

    % Navier_error_est computes the a posteriori error estimate
    compute a posteriori error estimate:  $\eta_h^{(l+1)} = \text{Navier\_error\_est}(\vec{u}_h^{(l+1)}, p_h^{(l+1)})$ 

    % matvecA( $\cdot$ ), matvecQ( $\cdot$ ) compute the action of  $\mathbf{A}_h$ ,  $\mathbf{Q}_h$  on a vector respectively.
    stopping test:
    if  $\sqrt{(\Delta \mathbf{u}_h^{(l)})^T \text{matvecA}(\Delta \mathbf{u}_h^{(l)}) + (\Delta \mathbf{p}_h^{(l)})^T \text{matvecQ}(\Delta \mathbf{p}_h^{(l)})} \leq \eta_h^{(l+1)}$ 
      convergence, break  $l$  loop
    endif

enddo

```

Fig. 4 The NAVIER_NEWTON_GMRES algorithm expressed in pseudocode.

5.4 Experimental results

Results of some computational experiments in IFISS are presented in this section as a proof-of-concept. The test problem for this purpose is the flow over a backward-facing step problem; see [9], [16]. In order to illustrate the robustness of the linear and the nonlinear balanced stopping test (26) and (35) respectively, results are presented for viscosity $\nu = 1/50, 1/100$ (hence varying Reynolds number). The grid level $h = 6$ is fixed and $\mathbf{Q}_1\text{-P}_0$ rectangular finite elements are employed on $2^h \times (2^h \times 3)$ grid.

Since no stabilization for the convection term is built in IFISS for the Navier–Stokes equations, the a posteriori error estimator is expected to overestimate the true error. Thus, employing the weak stopping test in (26) for linear iterations might lead to premature stopping. So, the strong stopping test in (26) is used. Results are presented for the Newton iterations only. Note that the initial guess for the Newton iteration in each case is the built in IFISS solution of the corresponding Stokes problem.

At each grid level and for various values of viscosity, a reference PDE solution is computed using Newton iteration to a tight nonlinear relative residual tolerance of $1\text{e-}12$ and a reference a posteriori error estimate η_h is computed from it. Also,

let $e_{\eta_h}^{(l)} := |\eta_h^{(l)} - \eta_h|$ where $\eta_h^{(l)}$ is the a posteriori error estimate computed at nonlinear iteration l from the PDE solution obtained therein.

Similarly, on each grid level, a reference a posteriori error estimate $\eta^{(h_l)}$ is also computed using a reference MATLAB Gaussian elimination solution for each linear system arising at each step of the nonlinear iteration. Let the difference between the reference and the computed a posteriori error estimate⁸ at linear balanced stopping iteration k^* be $e_{\eta_h}^{(l_{k^*})} := |\eta_h^{(l_{k^*})} - \eta^{(h_l)}|$.

For a comparison with balanced stopping GMRES solver, each linear system is also solved using preconditioned GMRES to $\|\mathbf{r}_h^{(l_k)}\|_2/\|\mathbf{r}_h^{(l_0)}\|_2$ tolerance of $1\mathbf{e}-6$, $1\mathbf{e}-9$ and the corresponding iteration counts l_{k_1} , l_{k_2} (respectively) are tabulated. The modified pressure convection-diffusion preconditioner [3, chapter 9] is employed with GMRES in all cases for solving the linear(ized) system arising at each nonlinear iterative step. The same initial random vector is used in all these solvers for solving any particular linear system.

The Navier–Stokes PDE (21) is defined on a L-shaped (flow over a backward-facing step) domain $\Omega = (-1, 5) \times (-1, 1) \setminus (-1, 0] \times (-1, 0]$ with $\vec{f} = \vec{0}$. On the outflow boundary ($x_1 = 5, -1 < x_2 < 1, \vec{x} = (x_1, x_2)$), Neumann boundary conditions are defined everywhere while a Poiseuille flow profile is imposed on the inflow boundary ($x_1 = -1, 0 \leq x_2 \leq 1$) and zero velocity condition is imposed on the walls [3, p. 335]. This problem can be generated in IFISS by choosing example 2 when running the driver `navier_testproblem`.

Table 6 Navier–Stokes test problem with Newton iteration on a $2^h \times (2^h \times 3)$ ($h = 6$) grid with $\nu = 1/50$ (left), $\nu = 1/100$ (right), both with the modified pressure convection-diffusion GMRES preconditioning.

l	l_{k_1}	l_{k_2}	l_{k^*}	$e_{\eta_h}^{(l_{k^*})}$	$e_{\eta_h}^{(l)}$
1	29	39	23	3.6e-05	3.1e-02
2	38	48	33	7.1e-08	3.3e-04
3	42	53	36	1.3e-09	5.2e-07

l	l_{k_1}	l_{k_2}	l_{k^*}	$e_{\eta_h}^{(l_{k^*})}$	$e_{\eta_h}^{(l)}$
1	35	46	46	2.0e-05	2.9e-01
2	53	66	52	2.0e-08	7.1e-03
3	55	68	52	1.8e-08	2.2e-04
4	59	73	55	8.2e-10	8.4e-07

In Table 6, a comparison of l_{k_1} , l_{k_2} numbers with the corresponding l_{k^*} values shows that employing the (strong) linear balanced stopping test (26) leads to savings in iteration counts especially when compared to $1\mathbf{e}-9$ tolerance iteration counts. Note that l_{k^*} is not significantly reduced (or even increases) compared to l_{k_1} in the right table. The reason is that the actual linear solver $\|\mathbf{r}_h^{(l_k)}\|_2/\|\mathbf{r}_h^{(l_0)}\|_2$ tolerance beyond which the approximation error cannot be reduced significantly is approximately $1\mathbf{e}-6$ (which is usually not be known to the user in advance) here. Since the strong linear solver stopping test is used here, the linear solver takes slightly more iterations (only for $l = 1$ here) or is not significantly reduced as compared to l_{k_1} , which is the iteration counts corresponding to actual linear solver tolerance $1\mathbf{e}-6$ in the right table.

At the l th Newton iteration and at the linear balanced black-box stopping iteration l_{k^*} , $e_{\eta_h}^{(l_{k^*})}$ columns show that the preconditioned GMRES solution of the

⁸ This a posteriori approximation error estimate is for the linearized part $(\delta \vec{u}_h^{(l_k)}, \delta p_h^{(l_k)})$.

linearized part has converged with some accuracy to the reference linearized solution. In other words, the sequence $\{\eta_h^{(l_k)}\}$ has converged with some accuracy to reference $\eta^{(h_l)}$. The last row $e_{\eta_h}^{(l)}$ entries (corresponding to the balanced nonlinear stopping iteration l^*) exhibit convergence of the sequence $\{\eta_h^{(l)}\}$ with good accuracy to the reference η_h .

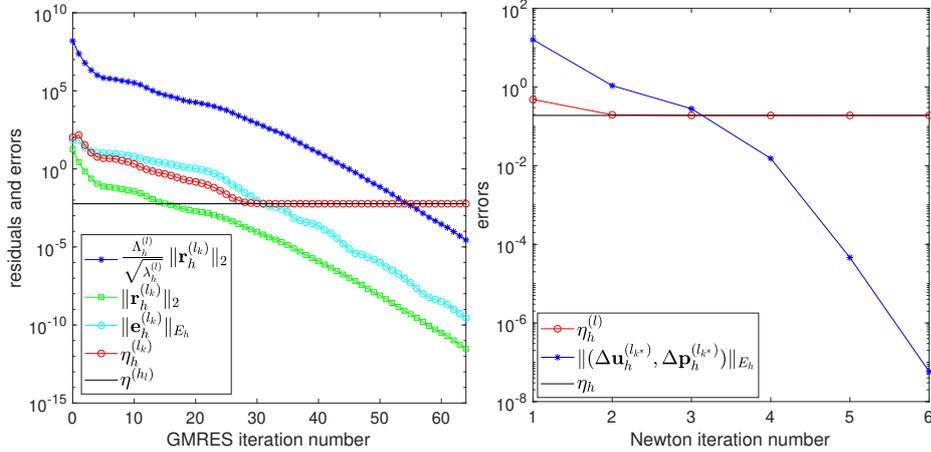


Fig. 5 Errors vs iteration number for Navier–Stokes test problem on a 64×192 grid with $\nu = 1/100$ for Newton iteration (right) and GMRES iteration (left) at $l = 4$ th Newton iteration.

Evolution of errors with iteration number are plotted in Figure 5 at 4th Newton iteration on 64×192 grid for $\nu = 1/100$. On the plot for linear iteration observe that at the balanced black-box linear stopping iteration l_{k^*} , the curve for $\eta_h^{(l_k)}$ converges with some accuracy to the line for the reference $\eta^{(h_l)}$. This convergence is further illustrated by continuing for 9 more iterations after balanced linear stopping. Also, on the plot for Newton iteration (right plot) notice that at the balanced black-box stopping nonlinear iteration number four, the curve for $\eta_h^{(l)}$ converges with some accuracy to the line for the reference a posteriori approximation error estimate η_h . This convergence is further illustrated by continuing for 2 more iterations after balanced black-box nonlinear stopping.

Since a cheap method of computing the eigenvalues involved in balanced stopping of nonsymmetric linear solvers is still a work in progress, the cost of computing the eigenvalues for the linear solver balanced stopping test may offset the computational savings if large number of nonlinear iterations are required. Hence, at present, it is prudent to use the balanced black-box nonlinear stopping test only.

6 Conclusions

Balanced black-box stopping criteria have been devised in linear and nonlinear solvers of systems arising from FEM approximation of PDEs. Typically employing these stopping criteria would lead to computational savings by reducing unnecessary iterations. In cases where an effective preconditioner is used, such stopping

criteria would indicate that the solver has not stopped prematurely. By balancing the algebraic error and the approximation error, the iteration is terminated as soon as the algebraic error is insignificant compared to the approximation error.

The balanced black-box stopping strategies presented in this paper are quite generic. They can be suitably modified to cater for varied linear and nonlinear iterative procedures for solving nonsymmetric linear(ized) systems arising from numerical approximation of a PDE. This is possible provided cheap and tight a posteriori (or a priori) approximation error estimators are available along with cheap tractable bounds on the relevant errors (here the algebraic error) that are generally difficult to compute.

References

1. Arioli, M., Lohin, D., Wathen, A.J.: Stopping criteria for iterations in finite element methods. *Numer. Math.* **99**(3), 381–410 (2005). <https://doi.org/10.1007/s00211-004-0568-z>
2. Brenner, S.C., Scott, L.R.: *The Mathematical Theory of Finite Element Methods*. Springer, USA (2008). Third Edition
3. Elman, H., Silvester, D., Wathen, A.: *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, UK (2014). Second Edition
4. Elman, H.C., Ramage, A., Silvester, D.J.: IFISS: A computational laboratory for investigating incompressible flow problems. *SIAM Review* **56**(2), 261–273 (2014). <https://doi.org/10.1137/120891393>
5. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: *Computational Differential Equations*. Cambridge University Press, USA (1996). First Edition
6. Freund, R.W.: A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. *SIAM J. Sci. Comput.* **14**(2), 470–482 (1993). <https://doi.org/10.1137/0914029>
7. Freund, R.W., Gutknecht, M.H., Nachtigal, N.M.: An implementation of the look-ahead Lanczos algorithm for non-hermitian matrices. *SIAM J. Sci. Comput.* **14**(1), 137–158 (1993). <https://doi.org/10.1137/0914009>
8. Golub, G.H., Van Loan, C.F.: *Matrix Computations*. The John Hopkins University Press, USA (2013). Fourth Edition
9. Gresho, P.M., Gartling, D.K., Torczynski, J.R., Cliffe, K.A., Winters, K.H., Garratt, T.J., Spence, A., Goodrich, J.W.: Is the steady viscous incompressible two-dimensional flow over a backward-facing step at $re = 800$ stable? *Int. J. Numer. Methods Fluids* **17**(6), 501–541 (1993). <https://doi.org/10.1002/flid.1650170605>
10. Hughes, T.J.R., Brooks, A.: A multi-dimensional upwind scheme with no cross-wind diffusion. In: *Finite Element Methods for Convection Dominated Flows*, ASME Winter Annual Meeting, T. Hughes (Ed.), New York, USA **34**, 19–35 (1979). <https://www.researchgate.net/publication/297681092>
11. Jiránek, P., Strakos, Z., Vohralík, M.: A posteriori error estimates including algebraic error and stopping criteria for iterative solvers. *SIAM J. Sci. Comput.* **32**(3), 1567–1590 (2010). <https://doi.org/10.1137/08073706X>
12. Oden, J.T., Demkowicz, L.F.: *Applied Functional Analysis*. CRC Press, USA (1996). First Edition
13. Paige, C.C., Saunders, M.A.: Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.* **12**(4), 617–629 (1975). <https://doi.org/10.1137/0712047>
14. Pietro, D.A.D., Flauraud, E., Vohralík, M., Yousef, S.: A posteriori error estimates, stopping criteria, and adaptivity for multiphase compositional refinement for thermal multiphase compositional flows in porous media. *Journal of Comp. Phys.* **276**, 163–187 (2014). <https://doi.org/10.1016/j.jcp.2014.06.061>
15. Pietro, D.A.D., Vohralík, M., Yousef, S.: An a posteriori-based, fully adaptive algorithm with adaptive stopping criteria and mesh refinement for thermal multiphase compositional flows in porous media. *Comput. Math. Appl.* **68**(12 B), 2331–2347 (2014). <https://doi.org/10.1016/j.camwa.2014.08.008>

16. Powell, C.E., Silvester, D.J.: Preconditioning steady-state Navier–Stokes equations with random data. *SIAM J. Sci. Comput.* **34**(5), A2482–A2506 (2012). <https://doi.org/10.1137/120870578>
17. Pranjal: Optimal iterative solvers for linear systems with stochastic PDE origins: Balanced black-box stopping tests, PhD thesis. University of Manchester, UK (2017). <http://eprints.maths.manchester.ac.uk/2596/>
18. Saad, Y., Schultz, M.: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.* **7**(3), 856–869 (1986). <https://doi.org/10.1137/0907058>
19. Shishkin, G.I.: Methods of constructing grid approximations for singularly perturbed boundary-value problems. *Condensing grid methods. Russian J. Numer. Anal. Math. Modelling* **7**(6), 537–562 (1992). <https://doi.org/10.1515/rnam.1992.7.6.537>
20. Silvester, D., Pranjal: An optimal solver for linear systems arising from stochastic FEM approximation of diffusion equations with random coefficients. *SIAM/ASA J. Uncertainty Quantification* **4**(1), 298–311 (2016). <https://doi.org/10.1137/15M1017740>
21. Sleijpen, G.L.G., Fokkema, D.R.: BICGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum. *Elec. Trans. Numer. Anal.* **1**, 11–32 (1993). <https://etna.mcs.kent.edu/vol.1.1993/pp11-32.dir/pp11-32.pdf>
22. Syamsudhuha, Silvester, D.J.: Efficient solution of the steady-state Navier–Stokes equations using a multigrid preconditioned Newton–Krylov method. *Int. J. Numer. Methods Fluids* **43**(12), 1407–1427 (2003). <https://doi.org/10.1002/fld.627>
23. Verfürth, R.: *A Posteriori Error Estimation Techniques for Finite Element Methods*. Oxford University Press, UK (2013). First Edition
24. Wathen, A.: Preconditioning and convergence in the right norm. *Int. J. Comput. Math.* **84**(8), 1199–1209 (2007). <https://doi.org/10.1080/00207160701355961>
25. Wu, C.T.: On the implementation of an accurate and efficient solver for convection-diffusion equations, PhD thesis. University of Maryland, USA (2003). <https://drum.lib.umd.edu/handle/1903/32>