A nonlinear elimination preconditioned inexact Newton algorithm for steady state incompressible flow problems on 3D unstructured meshes

Li Luo, Rongliang Chen, Xiao-Chuan Cai, David E. Keyes

The Newton algorithm and its variants are frequently used to obtain the numerical solution of large nonlinear systems arising from the discretization of partial differential equations, e.g., the incompressible Navier-Stokes equations in computational fluid dynamics. Near quadratic convergence can be observed when the nonlinearities in the system are well-balanced. However, if some of the equations have stronger nonlinearities than the others in the system, a Newton-like algorithm may suffer from slow convergence in the form of a long stagnation in the residual history, or not converge at all.

Nonlinear preconditioning aims to tackle this problem by creating an inner iteration with improved balance, which can be thought of as making the residual contours more spherical (i.e., hypersphericity in high dimension). Nonlinear preconditioners require solving nonlinear subproblems in inner iterations to remove implicitly local high nonlinearities that cause Newton's method to take small updates, so that the fast convergence of global Newton iteration can be restored. A nonlinear preconditioner can be applied on the left or on the right of the nonlinear function. The idea of left preconditioning [Cai and Keyes, 2002] is to replace the nonlinear function by a preconditioned one with more balanced nonlinearities, and then solve the new system using

Li Luo

Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, e-mail: `li.luo@kaust.edu.sa`

Rongliang Chen

Shenzhen Institutes of Advanced Technology, Shenzhen, China, e-mail: `rl.chen@siat.ac.cn`

Xiao-Chuan Cai

Department of Computer Science, University of Colorado Boulder, Boulder, USA, e-mail: `cai@cs.colorado.edu`

David E. Keyes

Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, e-mail: `david.keyes@kaust.edu.sa`

a Newton-like algorithm. In contrast, right preconditioning such as nonlinear elimination (NE) [Cresta et al., 2007, Cai and Li, 2011] does not change the nonlinear function but modifies the unknown variables of the original system. The application of NE can be viewed as a subspace correction step to provide a new starting point for the global Newton iteration, then the solution is updated in the whole space.

In this paper, we develop a nonlinear elimination preconditioned inexact Newton algorithm for steady state flow problems in 3D. It is well known that such problems are usually difficult to solve if a good initial guess is not available. The Newton-like algorithms may diverge though applied with some globalization techniques such as line search. To overcome the difficulty, we introduce an iterative restricted elimination approach based on the magnitude of the local residual, which successfully reduces the number of global Newton iterations. Numerical experiments show the value of the proposed algorithm in comparison to the classical inexact Newton method applied globally, and the impact of tuning parameters.

1 The nonlinear elimination preconditioned inexact Newton algorithm

Consider $F : R^n \to R^n$. We aim to find $x^* \in R^n$, such that

$$F(x^*) = 0, \tag{1}$$

starting from an initial guess $x^0 \in R^n$, where $F = (F_1, \ldots, F_n)^T$, $F_i = F_i(x_1, \ldots, x_n)$, and $x = (x_1, \ldots, x_n)^T$. We first recall the inexact Newton algorithm with backtracking (INB). Assume x^k is the current approximate solution. A new x^{k+1} can be computed via

$$x^{k+1} = x^k + \lambda^k s^k, \tag{2}$$

where λ^k is the step length, and the inexact Newton direction s^k satisfies

$$\|F'\left(x^k\right) s^k + F\left(x^k\right)\| \leq \eta^k \|F(x^k)\|. \tag{3}$$

Here $\eta^k \in [0, 1)$ is a forcing term that determines how accurately the Jacobian system needs to be solved. To enhance the robustness of INB, we adapt the choice of the forcing term based on norms that are by-products of the iteration, as suggested by Eisenstat and Walker [Eisenstat and Walker, 1996].

In many practical situations, especially for nonlinear equations that have unbalanced nonlinearities, λ^k is much smaller than 1 since it is often determined by the components with the strongest nonlinearities. The objective of nonlinear elimination (NE) is to balance the overall nonlinearities of the system through subspace correction. To illustrate the algorithm, we denote

by $y = L(\mathcal{F}, x)$ the operation of the subspace correction step, where $\mathcal{F}$ is a modified nonlinear function and x is an intermediate approximate solution. The basic algorithm of NE preconditioned inexact Newton method with backtracking (INB-NE) can be described as follows:

Algorithm 1 The nonlinear elimination preconditioned inexact Newton method with backtracking (INB-NE)

Step 1   Start from the initial guess x^0 and set $k = 0$, $x^{-1} = x^0$.

Step 2   Check convergence:
- If the global condition $\|F(x^k)\| \leq \gamma_r \|F(x^0)\|$ is satisfied, stop.
- If $\|F(x^k)\|/\|F(x^{k-1})\| > \rho_0$ and $k < N_{ne}$, go to **Step 3**; otherwise, go to **Step 4**.

Step 3   The NE step: perform subspace correction iteratively.
- Set $x^{(0)} = x^k$.
- For $l = 0, \ldots, N_l - 1$:
    - (i)   Construct the nonlinear function $\mathcal{F}(x)$.
    - (ii) Evaluate $x^{(l+1)} = L(\mathcal{F}, x^{(l)})$.
    - (iii) If $\|F(x^{(l+1)})\|/\|F(x^{(0)})\| < \rho_1$, break.
- Set $x^k = x^{(l+1)}$, go to **Step 4**.

Step 4   The global INB step:
- Inexactly solve $F'(x^k)s^k = -F(x^k)$.
- Compute λ^k using the cubic backtracking technique.
- Update $x^{k+1} = x^k + \lambda^k s^k$.
- Set $k = k + 1$, go to **Step 2**.

In the algorithm, γ_r is the relative tolerance for the nonlinear solver, ρ_0 and ρ_1 are preselected factors to measure the relative reduction of the global residual, and N_{ne} is used to control the number of applications of NE.

Next, we discuss the construction of $\mathcal{F}(x)$ and $L(\mathcal{F}, x)$ in detail. In this paper, we consider a point-based elimination approach, i.e., when one variable on some particular mesh point is selected to eliminate, all other variables corresponding to that mesh point are also eliminated. Specifically, let I be an index set of M mesh points, where each index corresponds to m unknown components x_{i_c} and m nonlinear residual components F_{i_c}, $c = 0, \ldots, m - 1$. At each subspace correction step, we decompose I into a "bad" subset $I_b^{(l)}$ with $M_b^{(l)}$ mesh points and a "good" subset $I_g^{(l)} = I \backslash I_b^{(l)}$ with $M - M_b^{(l)}$ mesh points, where $I_b^{(l)}$ and $I_g^{(l)}$ correspond to the variables that have strong and weak nonlinearities, respectively. In this paper we consider the bad subset of mesh points $I_b^{(l)}$ as

$$I_b^{(l)} = \left\{ i \mid \text{If } \max_c\{|F_{i_c}(x^{(l)})|\} > \beta \|F(x^{(l)})\|_\infty, c = 0, \ldots, m - 1 \right\}, \quad (4)$$

where $\beta > 0$ is a preselected factor. With this subset, we define two subspaces

$$V_b^{(l)} = \left\{ v \mid v = (v_0, \ldots, v_{n-1})^T \in R^n, \ v_{i_c} = 0 \text{ if } i \notin I_b^{(l)} \right\}, \quad (5)$$

$$\text{and } V_g^{(l)} = \left\{ v \mid v = (v_0, \ldots, v_{n-1})^T \in R^n, \ v_{i_c} = 0 \text{ if } i \in I_b^{(l)} \right\}. \quad (6)$$

The corresponding restriction operators are denoted as $R_b^{(l)}$ and $R_g^{(l)}$, which map the vectors from R^n to $V_b^{(l)}$ and $V_g^{(l)}$, respectively. Then, the modified nonlinear function $\mathcal{F}$ is defined as

$$\mathcal{F}(x) = R_g^{(l)}(x - x^{(l)}) + R_b^{(l)}(F(x)). \tag{7}$$

The nonlinear system $\mathcal{F}(x) = 0$ is solved by using the classical INB algorithm with the initial guess $x^{(l)}$. $x^{(*)}$ is accepted as the approximate solution if the stopping condition $\|\mathcal{F}(x^{(*)})\| \leq \gamma_r^{ne}\|\mathcal{F}(x^{(l)})\|$ is satisfied, where γ_r^{ne} is the relative tolerance for the nonlinear solver. In practice, we replace the equations corresponding to the good components by $x_{i_c} - x_{i_c}^{(l)} = 0$ and keep the others unchanged. Therefore, the solve of $\mathcal{F}(x) = 0$ can be performed in the whole space.

To construct the operator L, we introduce a restricted bad subset

$$I_{b,\varepsilon}^{(l)} = \left\{ i \,|\, \text{If } \max_c\{|F_{i_c}(x^{(l)})|\} > (\beta + \varepsilon)\|F(x^{(l)})\|_\infty, c = 0, \ldots, m - 1 \right\}, \tag{8}$$

where the restricted size $\varepsilon > 0$ is a given parameter. With this subset, we define the corresponding subspaces $V_{b,\varepsilon}^{(l)}$ and $V_{g,\varepsilon}^{(l)}$ in a similar way to (5) and (6). The corresponding restriction operators are denoted as $R_{b,\varepsilon}^{(l)}$ and $R_{g,\varepsilon}^{(l)}$, respectively. Then, with the approximate solution $x^{(*)}$, we define the corrected solution $x^{(l+1)}$ for the subspace correction step as

$$x^{(l+1)} = L(\mathcal{F}, x^{(l)}) = R_{g,\varepsilon}^{(l)}(x^{(l)}) + R_{b,\varepsilon}^{(l)}(x^{(*)}). \tag{9}$$

Remark 1. In this paper, an additive Schwarz preconditioned GMRES method is employed as the linear solver to obtain the solution of the Jacobian systems in both the global INB process and the NE step.

2 Numerical experiments

Let Ω be a bounded domain in R^3. The system of interest can be described by the steady state incompressible Navier-Stokes equations, as follows:

$$\begin{cases} \rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, & \text{in} \quad \Omega, \\ \nabla \cdot \mathbf{u} = 0, & \text{in} \quad \Omega. \end{cases}$$

Here $\mathbf{u} = (u, v, w)^T$ is the velocity, ρ is the density, and $\boldsymbol{\sigma} = -p\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$ is the Cauchy stress tensor, where $\mathbf{I}$ is an identity matrix, p is the pressure, and μ is the viscosity coefficient. A P_1-P_1 stabilized finite element method is used to discretize the incompressible Navier-Stokes equations, which results in a nonlinear system $F(x) = 0$ to be solved. Here x is a vector of the velocity and pressure unknowns defined at the mesh points.

The algorithms studied in this paper are implemented in PETSc [Balay et al., 2019]. All computations are carried out on the Tianhe-2 supercomputer. The Jacobian matrices arising from both the global INB process and the NE step are computed analytically. The relative tolerances γ_r and γ_r^{ne} are set to be 10^{-10} and 10^{-3}, respectively. The restart value of GMRES is fixed at 200. In the linear Schwarz preconditioner, the size of overlap is fixed to 2. A point-block incomplete LU (ILU) factorization with 3 fill-in levels is used to apply the approximate inverse action of the subdomain matrix. For all the numerical tests, we fix the parameters $\rho_0 = 0.8$, $\rho_1 = 0.2$, and $N_{ne} = 3$. A zero vector is used as the initial guess, i.e., $x^0 = \mathbf{0}$. It is observed in our tests that only one application of NE is needed for the global Newton to converge quickly, which usually happens at the 3rd global Newton step.

We first consider a lid-driven cubical cavity flow with different Reynolds numbers. The length of the cavity is $D = 1\text{m}$. A fluid with density $\rho = 1\text{kg/m}^3$ is driven by the wall at $y = D$ which moves tangentially in the x direction with a constant velocity $U = 1\text{m/s}$. The other walls impose a no slip boundary condition. The Reynolds number is defined as $Re = \rho U D/\mu$. We vary the viscosity μ to test different Reynolds numbers 1000, 1600, 1800, and 2000. In this case, we use $\beta = 10^{-2}$, $\varepsilon = 0$, and $N_l = 1$. A tetrahedral mesh with 1,761,316 elements and 313,858 nodes is used for the test. The simulation is conducted using 240 processor cores.

For $Re = 2000$, the projections of streamlines on equidistant planes are shown in Fig. 1 (left). In Fig. 1 (right), we show the histories of the nonlinear residuals by using the classical INB and the proposed INB-NE. It is observed that, for the classical INB, the residual curve converges quickly for case $Re = 1000$, but stagnates longer for cases $Re = 1600$ and $Re = 1800$, and diverges for case $Re = 2000$. For INB-NE, the residual curves for these four cases converge within 16 global nonlinear steps. Comparatively, the proposed algorithm is more robust with respect to higher Reynolds numbers.

To study how NE removes the strongest nonlinearities, we show in Fig. 2 the residual contour of component u before and after the application of NE, and the corresponding "bad" subset colored in red, at the 3rd global Newton step for case $Re = 2000$. Table 1 shows the numbers of iterations and compute times obtained using different Re. In the table, "NI$_{global}$" denotes the number of global Newton iterations, "LI$_{global}$" denotes the averaged number of GMRES iterations per global Newton, "NI$_{ne}$" refers to the averaged number of Newton iterations per subspace correction in NE, "LI$_{ne}$" is the averaged number of GMRES iterations per Newton in NE, "T$_{ne}$(s)" is the compute time in second for the NE preconditioner, and "T$_{total}$(s)" is the total compute time in second. As Re increases, though extra compute time is spent on the nonlinear preconditioning, the total compute time of INB-NE is less than that of the classical INB due to significant decrease of the number of global Newton iterations.

We next consider another well-understood benchmark problem, flow around a cylinder, as defined in [Schäfer and Turek, 1996]. The detailed geome-
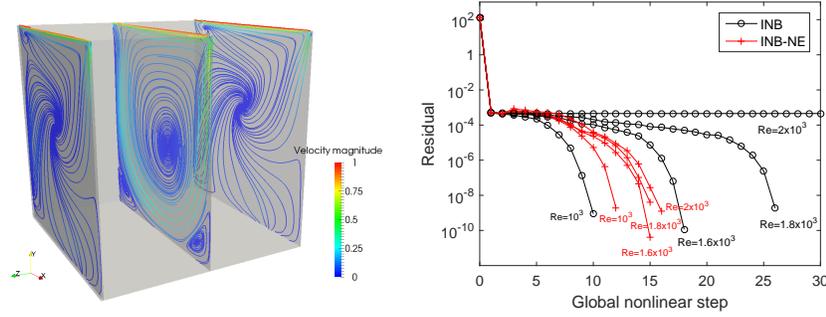
Fig. 1 Lid-driven cavity flow: (left) streamlines for case $Re = 2000$, (right) nonlinear residual history. $\beta = 10^{-2}$, $\varepsilon = 0$, and $N_l = 1$.
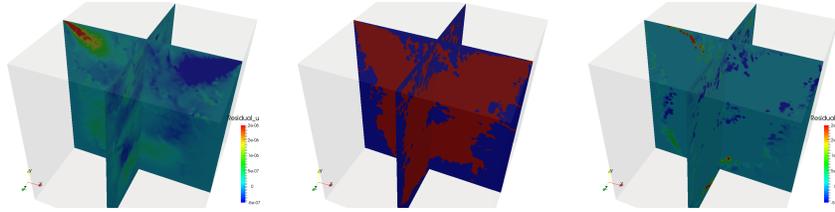

Fig. 2 Lid-driven cavity flow at the 3rd global Newton step for case $Re = 2000$: (left) residual contour of component u before NE, (middle) "bad" subset in red, (right) residual contour of component u after NE.

Table 1 The numbers of iterations and compute times obtained using different Re in the case of lid-driven cavity flow. "–" indicates that the case fails to converge.

	INB			INB-NE					
Re	NI_{global}	LI_{global}	$T_{total}(s)$	NI_{global}	LI_{global}	NI_{ne}	LI_{ne}	$T_{ne}(s)$	$T_{total}(s)$
1000	10	24.40	31.44	12	29.25	8	6.62	21.38	60.90
1600	18	32.29	64.72	15	28.00	8	6.87	21.86	70.82
1800	26	39.54	101.52	15	34.60	8	7.38	21.85	73.84
2000	–	–	–	16	37.93	8	7.63	21.73	78.23

try can be found in the reference. The height and width of the channel is $H = 0.41$m, and the diameter of the cylinder is $D = 0.1$m. The inflow condition is $U(0, y, z) = 16U_m yz(H - y)(H - z)/H^4$, yielding $Re = \rho \bar{U} D/\mu$, where $\bar{U} = 4U_m/9$. The outlet is imposed with a natural outflow boundary condition. We fix the density to $\rho = 1$kg/m^3, the velocity to $U_m = 0.45$m/s, and vary the viscosity μ to test different Reynolds numbers $Re = 20, 120, 170,$ and 200. In this case, we use $\beta = \varepsilon = 5 \times 10^{-4}$ and $N_l = 2$. A tetrahedral mesh with 4,909,056 elements and 851,024 nodes is used for the test. The simulation is conducted using 480 processor cores.

The velocity contour on the plane $z = 0.5H$ for case $Re = 20$ is shown in Fig. 3 (left). The histories of the nonlinear residuals obtained using the classical INB and the the present INB-NE are shown in Fig. 3 (right). As

Re increases, the number of Newton iterations of the classical INB increases rapidly. When Re is up to 200, INB fails to converge. In contrast, INB-NE converges well for all the four cases and requires fewer nonlinear iterations than the classical INB. Table 2 shows the numbers of iterations and compute times obtained using different Re. Similar to the case of driven cavity flow, when Re becomes larger, the proposed INB-NE is more efficient than the classical INB in terms of the total compute time.
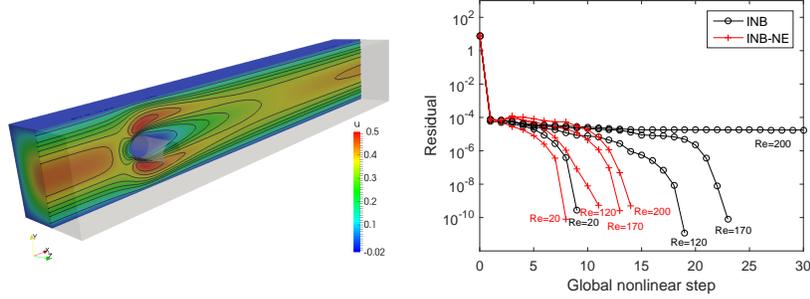

Fig. 3 Flow around a cylinder: (left) velocity contour for case $Re = 20$, (right) nonlinear residual history. $\beta = \varepsilon = 5 \times 10^{-4}$, $N_l = 2$.

Table 2 The numbers of iterations and compute times obtained using different Re in the case of flow around a cylinder. "–" indicates that the case fails to converge.

	INB			INB-NE					
Re	NI_{global}	LI_{global}	$T_{total}(s)$	NI_{global}	LI_{global}	NI_{ne}	LI_{ne}	$T_{ne}(s)$	$T_{total}(s)$
20	9	31.67	41.51	8	34.38	5.5	12.73	43.72	81.76
120	19	52.05	106.85	11	52.27	6.5	11.85	49.65	108.29
170	23	64.13	142.63	13	45.62	6.5	12.62	50.34	116.87
200	–	–	–	14	43.57	7.5	12.27	57.69	127.43

To study the impact of the parameters on the performance of the NE preconditioner, we test the case of flow around a cylinder at $Re = 200$ with different values of N_l, β, and ε. Results are listed in Table 3. In general, when increasing the number of subspace correction steps N_l or decreasing the preselected factor β, the number of global Newton iterations decreases, but this does not necessarily result in a better performance in terms of the total compute time. On the other hand, a suitable choice of the restricted size ε improves the convergence of the global Newton iteration. It is seen form the table that the configuration of $N_l = 2$ and $\beta = \varepsilon = 5 \times 10^{-4}$ leads to the smallest compute time for the concerned problem.

3 Conclusions

We demonstrated the robustifying effect of a nonlinearly preconditioned inexact Newton algorithm for steady state incompressible flow problems in 3D. The key idea is to perform iterative subspace correction steps to remove

Table 3 The numbers of iterations and compute times obtained using different values of parameters for the case of flow around a cylinder at $Re = 200$.

$\beta = \varepsilon = 5 \times 10^{-4}$						
N_l	NI_{global}	LI_{global}	NI_{ne}	LI_{ne}	$\mathrm{T}_{ne}(\mathrm{s})$	$\mathrm{T}_{total}(\mathrm{s})$
1	20	54.7	8.5	12.6	66.8	181.8
2	14	43.6	7.5	12.3	57.7	127.4
3	14	49.1	6.7	9.5	73.6	146.5
$\varepsilon = \beta$, $N_l = 2$						
β	NI_{global}	LI_{global}	NI_{ne}	LI_{ne}	$\mathrm{T}_{ne}(\mathrm{s})$	$\mathrm{T}_{total}(\mathrm{s})$
2×10^{-3}	17	41.6	7.5	6.5	53.3	137.1
10^{-3}	16	46.2	7	8.2	51.3	133.8
5×10^{-4}	14	43.6	7.5	12.3	57.7	127.4
$\beta = 5 \times 10^{-4}$, $N_l = 2$						
ε	NI_{global}	LI_{global}	NI_{ne}	LI_{ne}	$\mathrm{T}_{ne}(\mathrm{s})$	$\mathrm{T}_{total}(\mathrm{s})$
0	18	49.7	6	14.9	46.7	142.4
5×10^{-5}	16	43.3	6	15.0	47.3	127.9
5×10^{-4}	14	43.6	7.5	12.3	57.7	127.4
5×10^{-3}	16	42.1	8.5	11.2	67.0	149.3

the local high nonlinearities that cause difficulty for classical Newton-like algorithms. We tested the algorithm using two well-understood examples including a lid-driven cavity flow and the flow around a cylinder. Results of numerical experiments show that the proposed algorithm is more robust and converges faster than the classical algorithm in problems with high Reynolds numbers where globalized Newton methods may stagnate.

References

S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Users Manual. Technical Report ANL-95/11 Revision 3.10, Argonne National Laboratory, 2019.

X.-C. Cai and D. E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.*, 24:183–200, 2002.

X.-C. Cai and X. Li. Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity. *SIAM J. Sci. Comput.*, 33:746–762, 2011.

P. Cresta, O. Allix, C. Rey, and S. Guinard. Nonlinear localization strategies for domain decomposition methods: Application to post-buckling analyses. *Comput. Methods Appl. Mech. Eng.*, 196:1436–1446, 2007.

S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, 17:16–32, 1996.

M. Schäfer and S. Turek. Benchmark computations of laminar flow around a cylinder. *Notes Numer. Fluid Mech.*, 52:547–566, 1996.