

# A nonlinear elimination preconditioned inexact Newton method for blood flow problems in human artery with stenosis



Li Luo<sup>a</sup>, Wen-Shin Shiu<sup>b</sup>, Rongliang Chen<sup>b,c</sup>, Xiao-Chuan Cai<sup>d,\*</sup>

<sup>a</sup> Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia

<sup>b</sup> Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China

<sup>c</sup> Shenzhen Key Laboratory for Exascale Engineering and Scientific Computing, Shenzhen, China

<sup>d</sup> Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA

## ARTICLE INFO

### Article history:

Received 9 February 2019

Received in revised form 11 August 2019

Accepted 31 August 2019

Available online 9 September 2019

### Keywords:

Blood flow problem

Navier-Stokes equations with resistive boundary conditions

Finite element

Inexact Newton method

Nonlinear preconditioning

Parallel computing

## ABSTRACT

Simulation of blood flows in the human artery is a promising tool for understanding the hemodynamics. The blood flow is often smooth in a healthy artery, but may become locally chaotic in a diseased artery with stenosis, and as a result, a traditional solver may take many iterations to converge or does not converge at all. To overcome the problem, we develop a nonlinearly preconditioned Newton method in which the variables associated with the stenosis are iteratively eliminated and then a global Newton method is applied to the smooth part of the system. More specifically, we model the blood flow in a patient-specific artery based on the unsteady incompressible Navier-Stokes equations with resistive boundary conditions discretized by a fully implicit finite element method. The resulting nonlinear system at each time step is solved by using an inexact Newton method with a domain decomposition based Jacobian solver. To improve the convergence and robustness of the Newton method for arteries with stenosis, we develop an adaptive restricted region-based nonlinear elimination preconditioner which performs subspace correction to remove the local high nonlinearities. Numerical experiments for several cerebral arteries are presented to demonstrate the superiority of the proposed algorithm over the classical method with respect to some physical and numerical parameters. We also report the parallel scalability of the proposed algorithm on a supercomputer with thousands of processor cores.

© 2019 Elsevier Inc. All rights reserved.

## 1. Introduction

With the rapid development of computing technology, numerical simulation of blood flows has become a useful tool for the understanding of the hemodynamics in many biomechanical applications, such as the assessment of blood characteristics [8,32] and the prediction of surgery outcomes [26,31]. The computational based approaches are expected to become more popular due to their attractive merits: non-invasive, flexible and fast. However, there are several challenges. The complex geometry of the arterial tree, with stenosis in diseased patients, can generate highly nonuniform flows with recirculation and

\* Corresponding author.

E-mail addresses: li.luo@kaust.edu.sa (L. Luo), xuwx@siat.ac.cn (W.-S. Shiu), rl.chen@siat.ac.cn (R. Chen), cai@cs.colorado.edu (X.-C. Cai).

possibly turbulence, and sometimes extra difficulties are induced by the physiological inflow/outflow boundary conditions needed in order to obtain a truncated domain of interest.

Several classes of numerical methods have been developed recently for the accurate calculation of blood flows in patient-specific arteries. Taylor et al. [18,33,34] developed a framework of stabilized finite element method for solving the 3D incompressible Navier-Stokes equations coupled with 0D lumped parameter models for inflow and outflow boundary conditions. The computational technology was used to study cardiac hemodynamics for coronary arteries with stenosis in [32]. Most of the published works for studying blood flows in human arteries with stenosis [20,21,25] were based on commercial software which is easy to utilize but offers limited parallelism, for instance, ANSYS-CFX [1] is scalable with only a few hundred processor cores. If a full 3D model is used for the blood flow, the discretization of incompressible Navier-Stokes equations leads to a large system (millions of degrees of freedom) of nonlinear equations, and the system is required to solve many times in the unsteady cases. To make the computational technology clinically feasible, the use of supercomputers and scalable parallel algorithms are indispensable.

In this paper, we introduce a fully implicit finite element method for the 3D unsteady incompressible Navier-Stokes equations, and aim to develop efficient, robust and scalable solution algorithms for the resulting discretized system. An inexact Newton method is used to solve the nonlinear algebraic system, within which a Krylov subspace method is used to solve the analytically computed Jacobian systems. The parallel scalability of the inexact Newton-Krylov method depends heavily on the performance of the linear preconditioner [9,19,36,37]. In this work, we consider the overlapping restricted additive Schwarz (RAS) preconditioner which has been shown to be scalable for a class of integral type resistive boundary conditions [36] adopted here.

The inexact Newton method is efficient when the nonlinearities in the system are well-balanced and a good initial guess is available. However, for arteries with stenosis, the nonlinearities of the system are usually not well-balanced, and additionally a good initial guess is difficult to obtain. In such cases, the classical inexact Newton method often suffers from slow convergence or fails to converge, and the convergence is rather sensitive to the system parameters, e.g. the Reynolds number, the mesh size, and the time step size, etc. Moreover, during these slow nonlinear iterations, the preconditioned Jacobian solver is called repeatedly by the outer Newton steps, thus wasting a lot of compute time. To overcome this difficulty, we introduce a class of nonlinear elimination (NE) preconditioners to improve the convergence of the classical Newton method. The NE preconditioner is applied in the intermediate Newton iteration to implicitly remove the local high nonlinearities. The NE preconditioner has been applied successfully to some challenging nonlinear problems, such as transonic full potential flow [16,17], lid-driven cavity flow at high Reynolds numbers [7,15,41], and two-phase flow in porous media [24,39,40]. Different strategies have been proposed to identify the bad components that slow down the convergence. Huang et al. [15] and Yang et al. [41] applied a point-wise approach to eliminate the components associated with certain mesh points that cause the local high nonlinearities for multi-component systems, such as steady lattice Boltzmann equations and Navier-Stokes equations in the velocity-vorticity form. More recently, Yang et al. [39,40] proposed a field-based approach to eliminate the components associated with some field variables that have stronger nonlinearity than the others in the system, i.e. the saturation field in reservoir simulation. The authors systematically studied several performance-related parameters in the NE preconditioner to achieve optimal performance for the simulation. In these studies, only model problems in regular computational domains are considered. In this paper, we apply the NE preconditioner to the more realistic blood flow problem in patient-specific artery with stenosis. We propose an adaptive region-based elimination strategy and the region corresponds to the complexity of the fluid. The new method outperforms the classical method and other available elimination strategies in terms of global nonlinear iterations and the total compute time. We perform extensive numerical experiments to show the robustness and efficiency of the proposed algorithm with respect to certain physical and numerical parameters. Parallel scalability of the algorithm on a supercomputer with thousands of processor cores is also reported.

The remainder of this paper is organized as follows. In Section 2, the system of incompressible Navier-Stokes equations with a fully implicit finite element discretization is presented. In Section 3, we discuss in detail the proposed nonlinear elimination preconditioned inexact Newton method. Numerical experiments for several arteries with stenosis are reported in Section 4. The robustness and scalability of the proposed method are comprehensively studied in this section. Some concluding remarks are given in Section 5.

## 2. Governing equations and discretization

As shown in Fig. 1, we consider a human artery as the domain  $\Omega$ , that consists of the artery wall  $\Gamma_W$ , one or two inlets  $\Gamma_{I_i}$ ,  $i = 1, 2$ , and multiple outlets  $\Gamma_{O_j}$ ,  $j = 1, \dots, m$ . Although blood flow exhibits complex rheological properties, we model it as a Newtonian fluid described by the incompressible Navier-Stokes equations [32]:

$$\begin{cases} \rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{f}, & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0, & \text{in } \Omega. \end{cases} \quad (1)$$

Here  $\mathbf{u} = (u, v, w)^T$  is the velocity,  $\rho$  is the blood density,  $\mathbf{f}$  is a given body force per unit mass, and  $\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$  is the Cauchy stress tensor, where  $\mathbf{I}$  is an identity matrix,  $p$  is the pressure, and  $\mu$  is the viscosity coefficient.

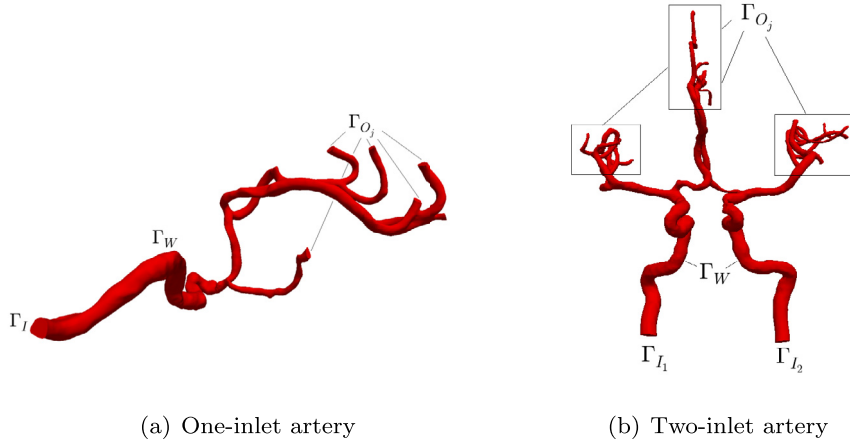


Fig. 1. Two sample arteries.

We consider a no slip boundary condition on the wall

$$\mathbf{u} = \mathbf{0}, \quad \text{on } \Gamma_W, \quad (2)$$

and a velocity profile derived from a flow rate on the inlets

$$\mathbf{u} = \mathbf{v}_{I_i}, \quad \text{on } \Gamma_{I_i}, \quad i = 1, 2. \quad (3)$$

On the outlets, we apply a resistive boundary condition which is based on the assumption of a linear dependence between the pressure and the flow rate

$$p = R_j Q_j, \quad \text{on } \Gamma_{O_j}, \quad j = 1, \dots, m. \quad (4)$$

Here  $R_j$  is the resistance, and  $Q_j$  is the flow rate through the  $j$ th outlet:

$$Q_j = \int_{\Gamma_{O_j}} \mathbf{u} \cdot \mathbf{n}_j d\Gamma, \quad j = 1, \dots, m, \quad (5)$$

where  $\mathbf{n}_j$  is the outward unit normal vector on  $\Gamma_{O_j}$ . It was first proposed by Murray [28] that the flow rate is related to the vessel size by  $Q_j \propto d_j^3$ , where  $d_j$  is the diameter of the  $j$ th outlet. Taylor et al. [32] indicates that such a morphometry law provides additional physiological information for assessing the resistance relative to the flow rate. Under resting conditions, the mean pressure  $p$  is largely constant through the artery. In accordance with the boundary condition (4), the Murray law also implies that  $R_j \propto d_j^{-3}$ , in the sense that small branches have a higher resistance compared with larger branches. Assuming the total resistance is given as  $R_T$ , the resistance for each outlet can be computed as

$$R_j = R_T \frac{\sum_{j=1}^m d_j^3}{d_j^3}, \quad j = 1, \dots, m. \quad (6)$$

Different values of  $R_T$  will be considered in the Numerical Experiments section of the paper.

For a patient-specific problem, the initial fields of velocity and pressure are usually not known in advance. Therefore we often start with zero as the initial guess, i.e.,  $\mathbf{u} = \mathbf{0}$  and  $p = 0$  over the domain at  $t = 0$ . We run the numerical simulation for one cardiac cycle, and the use the resulting  $\mathbf{u}$  and  $p$  as the actual initial guess for the next cardiac cycle.

Next, we introduce the weak form and a stabilized finite element discretization of the governing equations. The trial function spaces for the velocity and the pressure are defined as

$$\mathbf{V} = \left\{ \mathbf{u} \mid \mathbf{u} \in [H^1(\Omega)]^3, \mathbf{u} = \mathbf{0} \text{ on } \Gamma_W \text{ and } \mathbf{u} = \mathbf{v}_{I_i} \text{ on } \Gamma_{I_i}, i = 1, 2 \right\},$$

$$P = \left\{ p \mid p \in L^2(\Omega) \right\}.$$

The weighting function space for the velocity is defined as:

$$\mathbf{V}_0 = \left\{ \mathbf{u} \mid \mathbf{u} \in [H^1(\Omega)]^3, \mathbf{u} = \mathbf{0} \text{ on } \Gamma_W \cup \Gamma_{I_i}, i = 1, 2 \right\}.$$

Following standard notations, the weak form reads: find  $(\mathbf{u}, p) \in \mathbf{V} \times P$ , such that for  $\forall (\mathbf{w}, q) \in \mathbf{V}_0 \times P$ ,

$$B(\mathbf{u}, p; \mathbf{w}, q) = 0, \quad (7)$$

with

$$\begin{aligned} B(\mathbf{u}, p; \mathbf{w}, q) = & \int_{\Omega} \rho \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{w} d\Omega + \int_{\Omega} \rho (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{w} d\Omega - \int_{\Omega} p \nabla \cdot \mathbf{w} d\Omega + \int_{\Omega} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) : \nabla \mathbf{w} d\Omega \\ & + \int_{\Omega} (\nabla \cdot \mathbf{u}) q d\Omega - \int_{\Omega} \rho \mathbf{f} \cdot \mathbf{w} d\Omega + \sum_{j=1}^m \int_{\Gamma_{O_j}} (R_j Q_j \mathbf{n}_j \cdot \mathbf{w} - \mu \mathbf{n}_j \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \mathbf{w}) d\Gamma. \end{aligned} \quad (8)$$

We consider a  $P_1$ - $P_1$  stabilized finite element method [13,35] for the spatial discretization of the weak form (7). Additional stabilization terms are needed in order to meet the Ladyzenskaja-Babuska-Brezzi (LBB) inf-sup condition. Let  $\Omega_h = \{K\}$  be a conforming tetrahedral mesh of  $\Omega$  with  $h_K$  the diameter of the element  $K \in \Omega_h$ , denote by  $(\cdot, \cdot)_K$  the  $L^2$ -inner product over element  $K$ . We define the finite element subspaces  $\mathbf{V}^h$ ,  $\mathbf{V}_0^h$ , and  $P^h$  as the counterparts of their infinite dimensional subspaces. Then, the semi-discrete system of (7) is described as follows: find  $(\mathbf{u}_h, p_h) \in \mathbf{V}^h \times P^h$ , such that for  $\forall (\mathbf{w}_h, q_h) \in \mathbf{V}_0^h \times P^h$ ,

$$B_S(\mathbf{u}_h, p_h; \mathbf{w}_h, q_h) = 0, \quad (9)$$

with

$$B_S(\mathbf{u}_h, p_h; \mathbf{w}_h, q_h) = B(\mathbf{u}_h, p_h; \mathbf{w}_h, q_h) + \sum_{K \in \Omega_h} (\mathcal{S}_h, \tau_m ((\mathbf{u}_h \cdot \nabla) \mathbf{w}_h + \nabla q_h))_K + \sum_{K \in \Omega_h} (\nabla \cdot \mathbf{u}_h, \tau_c \nabla \cdot \mathbf{w}_h)_K, \quad (10)$$

where  $\mathcal{S}_h$ ,  $\tau_m$ , and  $\tau_c$  are defined as

$$\begin{aligned} \mathcal{S}_h &= \rho \frac{\partial \mathbf{u}_h}{\partial t} + \rho (\mathbf{u}_h \cdot \nabla) \mathbf{u}_h + \nabla p_h - \rho \mathbf{f}_h, \\ \tau_m &= \left( \sqrt{4/\Delta t^2 + \mathbf{u}_h \cdot G \cdot \mathbf{u}_h + 36\mu^2/\rho^2 G : G} \right)^{-1}, \\ \tau_c &= (8\tau_m \text{tr}(G))^{-1}. \end{aligned}$$

Here  $(G)_{ij} = \sum_{l=1}^3 \frac{\partial \xi_l}{\partial x_i} \frac{\partial \xi_l}{\partial x_j}$  ( $i, j = 1, 2, 3$ ) is the covariant metric tensor and  $\frac{\partial \xi}{\partial \mathbf{x}}$  refers to the Jacobian of the mapping between the reference and the physical element.  $\Delta t$  is the step size to be introduced below for the temporal discretization. We refer to [30,35] for more details of the spatial discretization scheme.

With the spatial discretization, (9) can be rewritten as a time-dependent nonlinear system

$$\frac{\partial \mathbf{x}(t)}{\partial t} = \mathcal{N}(\mathbf{x}(t)), \quad (11)$$

where  $\mathcal{N}(\mathbf{x}(t))$  is a nonlinear function, and  $\mathbf{x}(t)$  is a time-dependent vector of unknowns defined at the nodal points of  $\Omega_h$  at time  $t$ . To describe the temporal discretization scheme, we first define  $\mathbf{x}^n$  as the approximation of  $\mathbf{x}(t)$  at the  $n$ th time step. Then, a second-order backward differentiation formula (BDF2) is used to further discretize (11) in a fully implicit manner

$$\mathbf{x}^n - \frac{4}{3}\mathbf{x}^{n-1} + \frac{1}{3}\mathbf{x}^{n-2} = \frac{2\Delta t}{3}\mathcal{N}(\mathbf{x}^n), \quad \text{for } n \geq 2. \quad (12)$$

To provide an accurate approximation of  $\mathbf{x}^1$  in the above scheme, we separate the first time step into two fractional steps: (i)  $(0, \frac{\Delta t}{2})$  and (ii)  $(\frac{\Delta t}{2}, \Delta t)$ , and proceed as follows:

$$(i) \text{ Compute } \mathbf{x}^{\frac{1}{2}}: \quad \mathbf{x}^{\frac{1}{2}} - \mathbf{x}^0 = \frac{\Delta t}{2}\mathcal{N}(\mathbf{x}^{\frac{1}{2}}), \quad (13)$$

$$(ii) \text{ Update } \mathbf{x}^1: \quad \mathbf{x}^1 - \frac{4}{3}\mathbf{x}^{\frac{1}{2}} + \frac{1}{3}\mathbf{x}^0 = \frac{\Delta t}{3}\mathcal{N}(\mathbf{x}^1). \quad (14)$$

The above spatial and temporal discretization result in a nonlinear system

$$\mathcal{F}(\mathbf{x}^n) = \mathbf{0} \quad (15)$$

that has to be solved at each time step. We remark that the ordering of unknowns in  $\mathbf{x}^n$  and  $\mathcal{F}(\mathbf{x}^n)$  has a significant impact on the convergence properties of the algebraic solver and the parallel scalability of the overall method [14]. In contrast to

field by field ordering as usually required by other methods, here the components of velocity and pressure are ordered node by node, for example,

$$\mathbf{x}^n = (u_0^n, v_0^n, w_0^n, p_0^n, u_1^n, v_1^n, w_1^n, p_1^n, \dots, u_{M-1}^n, v_{M-1}^n, w_{M-1}^n, p_{M-1}^n)^T,$$

where  $M$  is the total number of mesh points. We refer to this ordering as the point-block ordering, in which the unknowns associated with each mesh point are always together in the same small block, which improves the robustness of ILU preconditioner, as well as the cache performance.

### 3. A nonlinearly preconditioned inexact Newton method

In this section, we describe the proposed nonlinearly preconditioned inexact Newton method for solving (15). We first recall the popular inexact Newton method with backtracking (INB) [10–12] as the outer iterative process, which consists of the following steps:

1. Take the initial guess or the solution of the previous time step as the initial guess  $\mathbf{x}_0^n = \mathbf{x}^{n-1}$ .
2. Construct the Jacobian matrix, analytically,  $J_k^n = \mathcal{F}'(\mathbf{x}_k^n)$ .
3. Find the inexact Newton direction  $\mathbf{s}_k^n$  by approximately solving the linear Jacobian system

$$J_k^n \mathbf{s}_k^n = -\mathcal{F}(\mathbf{x}_k^n), \quad (16)$$

4. Find a step length  $\lambda_k$  using a line search procedure [10] and compute the new approximation,

$$\mathbf{x}_{k+1}^n = \mathbf{x}_k^n + \lambda_k \mathbf{s}_k^n. \quad (17)$$

5. Set  $k = k + 1$ , go to Step 2.

In the INB method, the linear solver in Step 3 is stopped if

$$\|J_k^n \mathbf{s}_k^n + \mathcal{F}(\mathbf{x}_k^n)\| \leq \eta_r \|\mathcal{F}(\mathbf{x}_k^n)\|, \quad (18)$$

where  $\eta_r$  is a prescribed relative tolerance. The nonlinear iteration is stopped if

$$\|\mathcal{F}(\mathbf{x}_{k+1}^n)\| \leq \gamma_r \|\mathcal{F}(\mathbf{x}_0^n)\|, \quad (19)$$

where  $\gamma_r$  is a prescribed relative tolerance for the nonlinear solver. We remark that  $\lambda_k$  is a critically important parameter in INB. The slow convergence of INB happens when the value of  $\lambda_k$  is too small. In practice, the value of  $\lambda_k$  is often determined by a small number of components of  $\mathcal{F}$  that are more nonlinear than the others. A nonlinear preconditioner will be introduced in the next subsection to approximately eliminate these ‘bad’ components.

#### 3.1. The nonlinear elimination preconditioner

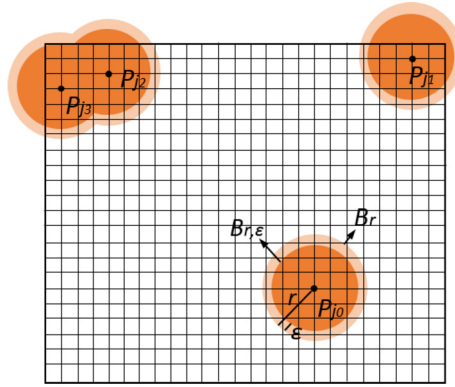
To balance the nonlinearities in  $\mathcal{F}$  so that larger step length can be used, we introduce a nonlinear preconditioner. We consider  $\rho_k^n = \frac{\|\mathcal{F}(\mathbf{x}_k^n)\|}{\|\mathcal{F}(\mathbf{x}_{k-1}^n)\|}$  as a measure of the relative reduction of the residual at the  $k$ th Newton iteration. If  $\rho_k^n$  is too large (for example, if  $\rho_k^n > \rho_0$ , where  $0 < \rho_0 < 1$  is a preselected parameter), then we consider the current Newton direction is not effective, and a preconditioning step is then introduced to produce a different  $\mathbf{x}_k^n$ . In the preconditioning step, we first identify the components of  $\mathcal{F}$  that slow down the convergence, and then eliminate them using a few subspace Newton iterations. For the rest of the paper, we refer to this step as a nonlinear elimination (NE) preconditioner and it is applied as a subspace correction step within a Newton iteration. This can be viewed as a right nonlinear preconditioning method [3,7], as opposed to the left nonlinear preconditioning method [6,22–24]. A high level description of the INB-NE method for solving (15) at each time step is presented in **Algorithm 1**.

---

#### **Algorithm 1** A nonlinear elimination preconditioned inexact Newton method with backtracking (INB-NE).

---

1. Take the initial guess or the solution of the previous time step as the initial guess  $\mathbf{x}_0^n = \mathbf{x}^{n-1}$ . Set  $k = 0$ ,  $\mathbf{x}_{-1}^n = \mathbf{x}_0^n$ .
  2. If the global nonlinear stopping condition (19) is satisfied, stop and set  $\mathbf{x}^n = \mathbf{x}_k^n$ ; otherwise, go to Step 3.
  3. Compute  $\rho_k^n$  and if  $\rho_k^n > \rho_0$ , go to Step 4; otherwise, go to Step 5.
  4. The nonlinear elimination step:
    - Identify the slow components of the nonlinear residual.
    - Approximately eliminate the slow components using a few subspace Newton iterations.
    - Update  $\mathbf{x}_k^n$ . Go to Step 5.
  5. The global update step:
    - Analytically construct the global Jacobian matrix  $J_k^n$ .
    - Inexactly solve  $J_k^n \mathbf{s}_k^n = -\mathcal{F}(\mathbf{x}_k^n)$ .
    - Update  $\mathbf{x}_{k+1}^n = \mathbf{x}_k^n + \lambda_k \mathbf{s}_k^n$ .
    - Set  $k = k + 1$ . Go to Step 2.
-



**Fig. 2.** Decomposition of a 2D mesh to a ‘good’ region (uncolored) and a ‘bad’ region (colored). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

The key idea of NE is to remove local high nonlinearities before performing a global nonlinear update, so it is essential to effectively identify the components to be eliminated in the nonlinear system. Several strategies have been studied for different types of problems including the component-wise approach [17,41], the point-wise approach [15,41], and the field-based approach [38–40]. For problems under consideration in this paper, we observe that a small number of components contribute a large percentage of the total nonlinear residual norm, and these components often cluster around the one corresponding to a local maximum of the absolute value of the residual function. Further use of the structure of the nonlinear residual function in the region containing these bad components can effectively improve the convergence of the global nonlinear iteration. Hence, we propose and test an adaptive region-based nonlinear elimination approach in this paper. In contrast to the physics-based approach [16,17] that the components to be eliminated are known in advance, here we use the information of the intermediate solution to identify the location of the bad region. We compare and discuss different elimination strategies in Section 3.2.

In the remaining part of this subsection, we focus on Step 4 of the algorithm which is the nonlinear preconditioning step. We first introduce some notations. Let  $I = \{0, 1, \dots, M - 1\}$  be an index set of the whole mesh with  $M$  points

$$\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{M-1}\},$$

where each index corresponds to four unknown components  $x_{ic} \in \{u_i^n, v_i^n, w_i^n, p_i^n\}$  and four nonlinear residual components  $\mathcal{F}_{ic}$ ,  $c = 0, \dots, 3$ . At the  $k$ th Newton iteration, we decompose  $I$  into a ‘bad’ subset  $I_b^k$  with  $M_b^k$  mesh points and a ‘good’ subset  $I_g^k$  with  $(M - M_b^k)$  mesh points, such that  $I_g^k = I \setminus I_b^k$ . In our approach, we introduce a collection of ‘seed points’

$$\{\mathbf{p}_{j_0}, \mathbf{p}_{j_1}, \dots, \mathbf{p}_{j_{L-1}}\} \subset \mathbf{P}.$$

Each of these seed points owns the component with local maximum absolute residual value  $\beta_l \|\mathcal{F}\|_\infty$ ,  $l = 0, \dots, L - 1$ . Here  $\{\beta_l\}$  are a series of prescribed parameters. Then, we consider the bad region as an union set of  $r$ -balls from these seed points, i.e.  $\bigcup_{l=0}^{L-1} B_r(\mathbf{p}_{j_l})$ . Here  $r$  is a given positive constant. An illustration of the bad region on a 2D mesh is shown in Fig. 2. The corresponding index set  $I_b^k$  is defined as

$$I_b^k = I_{b,0}^k \cup \dots \cup I_{b,L-1}^k, \tag{20}$$

where

$$I_{b,l}^k = \{i \mid \|\mathbf{p}_i - \mathbf{p}_{j_l}\| \leq r, i = 0, \dots, M - 1\}, l = 0, \dots, L - 1. \tag{21}$$

With the index set  $I_b^k$ , we define two subspaces

$$V_b^k = \left\{ \mathbf{v} \mid \mathbf{v} = (v_0, \dots, v_{N-1})^T \in \mathbb{R}^N, v_{ic} = 0 \text{ if } i \notin I_b^k, c = 0, \dots, 3 \right\}, \tag{22}$$

$$V_g^k = \left\{ \mathbf{v} \mid \mathbf{v} = (v_0, \dots, v_{N-1})^T \in \mathbb{R}^N, v_{ic} = 0 \text{ if } i \in I_b^k, c = 0, \dots, 3 \right\}, \tag{23}$$

respectively, where  $N = 4M$  is the total number of unknowns. The corresponding restriction operators are denoted as  $\mathcal{R}_b^k$  and  $\mathcal{R}_g^k$ , that are sub-identity matrices mapping the vectors from  $\mathbb{R}^N$  to  $V_b^k$  and  $V_g^k$ , respectively. Then, we define the following nonlinear system

$$\mathcal{G}(\mathbf{x}) \equiv \mathcal{R}_b^k(\mathcal{F}(\mathbf{x})) + \mathcal{R}_g^k(\mathbf{x} - \mathbf{x}_k^n) = \mathbf{0}. \tag{24}$$

$\mathcal{G}$  is simply a restriction of  $\mathcal{F}$  to the bad region. The ‘inverse’ of  $\mathcal{G}$  is considered as a nonlinear preconditioner of  $\mathcal{F}$ . Here by ‘inverse’, we mean a solve of (24) by using the classical INB method with the initial guess  $\mathbf{x}_k^n$ .  $\mathbf{x}_k^*$  is accepted as the approximate solution if the stopping condition  $\|\mathcal{G}(\mathbf{x}_k^*)\| \leq \gamma_r^{\text{NE}} \|\mathcal{G}(\mathbf{x}_k^n)\|$  is satisfied, where  $\gamma_r^{\text{NE}}$  is the relative tolerance for solving (24). Note that the nonlinear preconditioner may change from iteration to iteration since the selection of the bad region may change.

Since  $\mathbf{x}_k^*$  is the same as  $\mathbf{x}_k^n$  in the ‘good’ region and is replaced by a new solution in the ‘bad’ region,  $\mathcal{F}(\mathbf{x}_k^*)$  may have a sharp jump near the interface between the bad region and the good region since the calculation of  $\mathcal{F}(\mathbf{x}_k^*)$  involves the derivatives with respect to a function which is not guaranteed to be smooth across the interface. Below we introduce a restricted version of the NE preconditioner which keeps the corrected solution only in the interior of the bad region and throws away the values near the interface. For each seed point  $\mathbf{p}_{j_l}$ , we consider a restricted  $r$ -ball  $B_{r,\varepsilon}(\mathbf{p}_{j_l}) \subseteq B_r(\mathbf{p}_{j_l})$  with restricted size  $\varepsilon \in [0, r)$ . The corresponding index set  $I_b^{k,\varepsilon}$  for the union set is defined as

$$I_b^{k,\varepsilon} = I_{b,0}^{k,\varepsilon} \cup \dots \cup I_{b,L-1}^{k,\varepsilon}, \tag{25}$$

where

$$I_{b,l}^{k,\varepsilon} = \{i \mid \|\mathbf{p}_i - \mathbf{p}_{j_l}\| \leq r - \varepsilon, i = 0, \dots, M - 1\}, l = 0, \dots, L - 1. \tag{26}$$

With the index set  $I_b^{k,\varepsilon}$ , we define two subspaces

$$\begin{aligned} V_b^{k,\varepsilon} &= \left\{ v \mid v = (v_0, \dots, v_{N-1})^T \in \mathbb{R}^N, v_{i_c} = 0 \text{ if } i \notin I_b^{k,\varepsilon}, c = 0, \dots, 3 \right\}, \\ V_g^{k,\varepsilon} &= \left\{ v \mid v = (v_0, \dots, v_{N-1})^T \in \mathbb{R}^N, v_{i_c} = 0 \text{ if } i \in I_b^{k,\varepsilon}, c = 0, \dots, 3 \right\}, \end{aligned}$$

respectively. The corresponding restriction operators are denoted as  $\mathcal{R}_b^{k,\varepsilon}$  and  $\mathcal{R}_g^{k,\varepsilon}$ , which map the vectors from  $\mathbb{R}^N$  to  $V_b^{k,\varepsilon}$  and  $V_g^{k,\varepsilon}$ , respectively. Then, with the approximate solution  $\mathbf{x}_k^*$ , we update the solution  $\mathbf{x}_k^n$  by

$$\mathbf{x}_k^n := \mathcal{R}_g^{k,\varepsilon}(\mathbf{x}_k^n) + \mathcal{R}_b^{k,\varepsilon}(\mathbf{x}_k^*), \tag{27}$$

so that the good part of  $\mathbf{x}_k^n$  is retained while only the interior of the bad part is replaced by the corrected solution.

**Remark 3.1.** Generally, different  $r, \varepsilon$  can be used for different  $r$ -balls. For the particular test cases in this paper, a single  $r$  value is enough to capture the bad region for the artery with one stenosis, hence we only consider the case of one seed point that owns the component with global maximum value of  $\|\mathcal{F}\|_\infty$ .

### 3.2. Discussion on different elimination strategies

For multi-component systems, different strategies have been proposed to choose the bad components that slow down the convergence of the classical INB method, such as the component-wise approach [17,41], the point-wise approach [15, 41], and the field-based approach [38–40]. However, designing a good strategy is often problem-dependent, and there is no available theoretical guidelines [41]. For the purpose of comparison, we describe different approaches within the framework of **Algorithm 1**, as follows:

- *Component-wise approach.* At the  $k$ th Newton iteration, the bad subset of components  $N_b^k$  is chosen as

$$N_b^k = \{n \mid |\mathcal{F}_n| > \beta \|\mathcal{F}\|_\infty, n = 0, \dots, N - 1\}, \tag{28}$$

where  $\beta > 0$  is a preselected constant. Then, the subspaces  $V_b^k$  and  $V_g^k$  corresponding to (22) and (23) are defined as

$$\begin{aligned} V_b^k &= \left\{ v \mid v = (v_0, \dots, v_{N-1})^T \in \mathbb{R}^N, v_n = 0 \text{ if } n \notin N_b^k \right\}, \\ V_g^k &= \left\{ v \mid v = (v_0, \dots, v_{N-1})^T \in \mathbb{R}^N, v_n = 0 \text{ if } n \in N_b^k \right\}. \end{aligned}$$

The restriction operators can be defined correspondingly and applied directly to (24). The updated solution is obtained by solving (24) and defined as  $\mathbf{x}_k^n := \mathbf{x}_k^*$ . The component-wise approach was initially designed for scalar systems [17]. For multi-component systems, the approach is blind to the physical nature of different variables and select the bad components according to the algebraic criteria (28). The component-wise approach may eliminate only one of the variables associated with a mesh point if its residual component dominates the magnitude of the whole residual.

- *Point-wise approach.* In this approach, when one variable at a mesh point is selected to be eliminated, all other variables associated with this mesh point are also eliminated. The bad subset of mesh points  $I_b^k$  is defined as

$$I_b^k = \{i \mid \text{If } \max_c \{|\mathcal{F}_{i_c}|\} > \beta \|\mathcal{F}\|_\infty, c = 0, \dots, 3\}. \quad (29)$$

The subspaces  $V_b^k$  and  $V_g^k$  are defined the same as (22) and (23). Following the idea of restricted elimination, we define the restricted bad subset  $I_b^{k,\varepsilon}$  by excluding some mesh points near the interface, such that

$$I_b^{k,\varepsilon} = \{i \mid \text{If } \max_c \{|\mathcal{F}_{i_c}|\} > (\beta + \varepsilon_1) \|\mathcal{F}\|_\infty, c = 0, \dots, 3\}, \quad (30)$$

where  $\varepsilon_1$  is a preselected constant. The point-wise approach has been successfully used for multi-component systems such as steady lattice Boltzmann equations [15] and Navier-Stokes equations in the velocity-vorticity form [41], both on 2D structured meshes. The point-wise approach can be viewed as an extension of the domain decomposition based elimination method NKS-RAS [7].

- *Field-based approach.* The field-based approach was studied in [38–40] for the situation when the nonlinearity of one field variable is stronger than the others in the multi-component system (for example, the pressure field in the Navier-Stokes equations). In the approach, all components corresponding to one particular field are selected to eliminate, without the need of any feedback from the intermediate solution or residual. Once the field to eliminate is determined, the ‘bad’ components are fixed through the whole computation. Hence, this approach is considered to be of a static fashion.
- *Field-split approach.* This approach alternatively eliminates each field of the system, i.e., the velocity field and the pressure field of our problem. The subspace correction step (Step 4 in **Algorithm 1**) is performed in a multiplicative manner, and an algebraic criterion is introduced to select the bad components adaptively. Specifically, in the first phase, the velocity field is corrected by choosing the bad components with index  $i_c$  if  $\max_c \{|\mathcal{F}_{i_c}|\} > \beta \|\mathcal{F}_v\|_\infty$ ,  $c = 0, 1, 2$ , where  $\mathcal{F}_v$  is the subvector of residual associated with the velocity components. Then, the pressure field is corrected in the second phase by choosing the bad components using the condition  $|\mathcal{F}_{i_3}| > \beta \|\mathcal{F}_p\|_\infty$ , where  $\mathcal{F}_p$  is the subvector of residual associated with the pressure components. Practically, the bad components in each phase belong to the subset of those in the point-wise approach, hence the implementation can be done easily by sweeping the subspace correction step of the point-wise approach and choosing one particular field to eliminate at each sweep. The restricted subsets of components can be defined similarly as (30).
- *Region-based approach.* The region-based approach can be viewed as an extension of the point-wise approach. In contrast to the algebraic criteria used in other approaches, the criteria in the region-based approach is of a hybrid physical-algebraic type, accounting for the continuous domain where the blood flow may become locally chaotic. In order to resolve the flow structures in the local region, all variables should be included and eliminated simultaneously. We show later in the numerical tests that this approach leads to better convergence than other approaches for our problem.

### 3.3. The linear solver

A linear solver is required for obtaining the solution of the Jacobian systems arising from both the outer Newton iteration and the nonlinear elimination process. Good candidates include the class of Krylov subspace methods with effective linear preconditioners. In this study, a restricted additive Schwarz (RAS) preconditioned Generalized Minimal RESidual (GMRES) method [4,29] is employed to solve the Jacobian systems.

We consider a right-preconditioned problem

$$AM_{\text{RAS}}^{-1}y = b, \quad \text{with } x = M_{\text{RAS}}^{-1}y, \quad (31)$$

where  $A$  is the Jacobian matrix,  $M_{\text{RAS}}$  is the RAS preconditioner,  $x$  is the solution, and  $b$  is the right-hand side. Denote by  $np$  the number of processor cores of the parallel computer, we partition the computational domain  $\Omega_h$  into  $np$  nonoverlapping subdomains  $\Omega_l$  (i.e.  $\Omega_i \cap \Omega_j = \emptyset$ ,  $\forall i \neq j$ ) for  $l = 1, \dots, np$ , such that  $\Omega_h = \Omega_1 \cup \dots \cup \Omega_{np}$ . The subvector associated with  $\Omega_l$  is denoted as  $y_l$ . We then extend  $\Omega_l$  to overlap with its neighbors by  $\delta$  layers of mesh elements and denote the overlapping subdomain as  $\Omega_l^\delta$ . On each overlapping subdomain, we define the corresponding subvector  $y_l^\delta$  and the restriction operator  $R_l^\delta$  that maps the global vector of unknowns in  $\Omega_h$  to  $y_l^\delta$ , i.e.

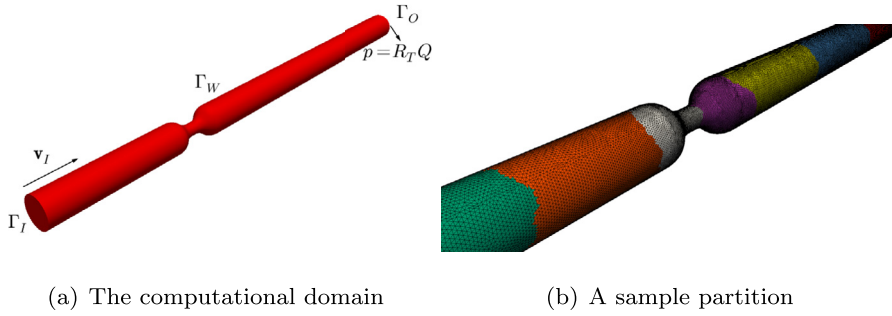
$$y_l^\delta = R_l^\delta y = (I \quad 0) \begin{pmatrix} y_l^\delta \\ y \setminus y_l^\delta \end{pmatrix}.$$

We denote  $R_l^0$  as the restriction operator that returns  $y_l$  defined on the nonoverlapping subdomain. Then, the RAS preconditioner [5] is defined as

$$M_{\text{RAS}}^{-1} = \sum_{l=1}^{np} (R_l^0)^T (A_l)^{-1} R_l^\delta, \quad (32)$$

$$A_l = R_l^\delta A (R_l^\delta)^T. \quad (33)$$





**Fig. 3.** A straight cylindrical vessel with an idealized stenosis. In (b), different colors refer to different partitions.

In (32),  $(A_I)^{-1}$  is understood as the inverse of the subdomain Jacobian matrix, its product with a vector is computed by solving a subdomain linear system inexactly. In our work, this is done by using a point-block incomplete LU (ILU) factorization of  $A_I$  based on the reverse Cuthill-McKee ordering.

#### 4. Numerical experiments

In this section, we provide some examples to study the robustness and efficiency of the proposed algorithm. We first verify the correctness of the proposed discretization scheme by calculating the pressure distribution along a straight cylindrical vessel with an idealized stenosis. Next, we conduct simulations of blood flows in some patient-specific cerebral arteries. We focus on: (1) the robustness of the algorithm with respect to some physical and numerical parameters, (2) a comparison of the numerical performance between the new INB-NE method and the classical INB method, and (3) the parallel scalability of the new algorithm.

The algorithms are implemented using PETSc [2]. All computations are carried out on a supercomputer which has two 12-core Intel Ivy Bridge Xeon CPUs and 64 GB local memory in each of its compute nodes. The relative tolerance for the global nonlinear solver is set to be  $10^{-6}$ . The relative tolerance for the linear solver is set to be  $10^{-3}$ . The Jacobian matrices arising from both the outer Newton iteration and the NE step are computed analytically. The restart value of GMRES is fixed at 400. For the NE preconditioner, we set the preselected parameter  $\rho_0$  to be 0.8 for judging if the residual is reduced slowly. In the rest of this paper, ‘ $NI_{global}$ ’ denotes the averaged number of outer Newton iterations per time step, ‘ $LI_{global}$ ’ denotes the averaged number of GMRES iterations per outer Newton iteration, ‘ $N_{ne}$ ’ is the averaged number of applications of the NE preconditioner per time step, ‘ $NI_{ne}$ ’ refers to the averaged number of Newton iterations per NE application, ‘ $LI_{ne}$ ’ is the averaged number of GMRES iterations per Newton in NE, ‘ $Time_{ne}$  (s)’ is the compute time in second per NE application, and ‘ $Time_{total}$  (s)’ is the total compute time in second per time step. In the Navier-Stokes equations, the body force is ignored for all experiments. The blood is characterized with viscosity  $\mu = 0.035$  g/(cm s), and density  $\rho = 1.06$  g/cm<sup>3</sup>.

##### 4.1. A straight cylindrical vessel with idealized stenosis

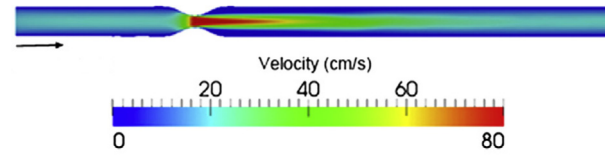
To verify the proposed fully implicit finite element discretization scheme, we first consider a straight cylindrical vessel with an idealized stenosis. Here the classical INB method is used to solve the discretized nonlinear system, while the NE-based preconditioner will be examined in the next section.

In this test, a circular tube with reference length  $L = 6.16$  cm, diameter  $D = 0.35$  cm and a stenosis of 60% diameter reduction is used. The computational domain and a sample partition into 8 subdomains are shown in Fig. 3. This simple case was also studied in Taylor et al. [32] for modeling the left anterior descending coronary artery. The resistance is chosen as  $R_T = 120,000$  dynes·s/cm<sup>2</sup>, and the inflow velocity is  $\mathbf{v}_I = (u_I, 0, 0)$  with  $u_I = 10.525$  cm/s.

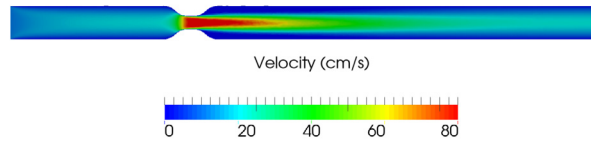
The solution is considered to be steady if  $\|\mathbf{x}^n - \mathbf{x}^{n-1}\|_\infty \leq 10^{-7}$ . The time step size is given as  $\Delta t = 0.01$  s. A sequence of unstructured meshes ranging from 207,467 elements to 1,118,548 elements are used for the tests. The overlap size of the RAS preconditioner is fixed to  $\delta = 1$ , ILU(2) is used as the subdomain solver. The velocity and pressure distribution along the tube are shown in Fig. 4. It is observed that a jet caused by sudden acceleration of velocity appears behind the stenotic territory and dissipates gradually along the central line of the vessel. We compare our results with those obtained in Taylor et al. [32] by calculating the outflow rate  $Q_{\Gamma_O}$  (cm<sup>3</sup>/s), the pressure difference  $p_a - p_d$  (mmHg), as well as the pressure ratio  $p_d/p_a$  between the proximal point  $P_a(0, 0, 0)$  and the distal point  $P_d(6.16, 0, 0)$ . Details of the comparison are listed in Table 1. As the mesh is refined to over a million elements, the estimated quantities match quite well with those results in the reference [32], which confirm the correctness of the proposed discretization scheme.

##### 4.2. Two patient-specific cerebral arteries

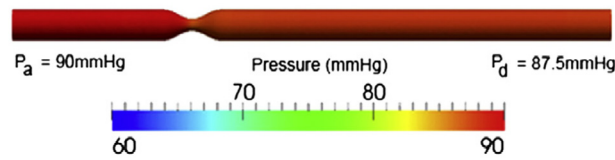
To further study the numerical performance of the proposed algorithm, we consider two patient-specific cerebral arteries, respectively, the one-inlet case and the two-inlet case, as shown in Fig. 1. The 3D geometry of the arteries is generated from



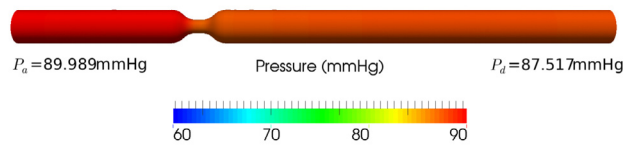
(a) Velocity (in Figure 3 of [32])



(b) Velocity (present)



(c) Pressure (in Figure 3 of [32])



(d) Pressure (present)

**Fig. 4.** The distribution of velocity and pressure for the straight cylindrical vessel with an idealized stenosis, in comparison of the published results in [32].**Table 1**

Comparison of the outflow rate  $Q_{\Gamma_o}$  ( $\text{cm}^3/\text{s}$ ), the pressure difference  $p_a - p_d$  (mmHg), and the pressure ratio  $p_d/p_a$  obtained using different meshes for the case of a straight vessel.

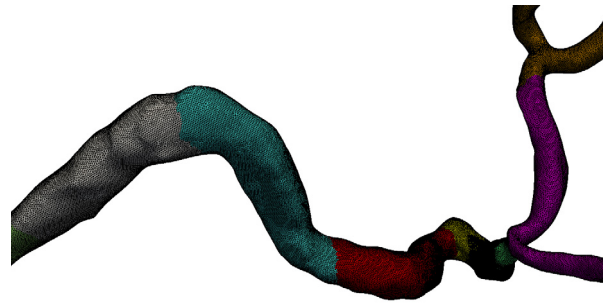
Case	# of nodes	# of elements	$Q_{\Gamma_o}$	$p_a - p_d$	$\frac{p_d}{p_a}$
Mesh_1	39,506	207,467	0.927	3.231	0.9627
Mesh_2	126,697	604,231	0.933	2.307	0.9732
Mesh_3	221,281	1,118,548	0.972	2.472	0.9725
Taylor et al. [32]			0.972	2.500	0.9722

**Table 2**

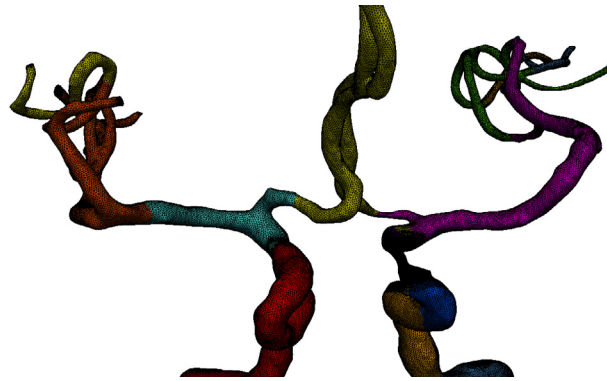
The case configuration and mesh information for two patient-specific cerebral arteries.

Case	Mean $Re$	$R_T$ (dynes $\text{s}/\text{cm}^2$ )	# of inlets	# of outlets	# of elements	# of nodes
One-inlet	443.2	21,923.0	1	6	437,538	2,190,164
Two-inlet	262.6	6,642.2	2	15	1,069,767	5,225,949

MRI scans and reconstructed by the software Mimics [27]. We then use ANSYS [1] to create the volume mesh that is finer nearby the stenosis. The case configuration and mesh information are summarized in Table 2. Fig. 5 shows the meshes used for these two cases with sample partitions into 16 subdomains.

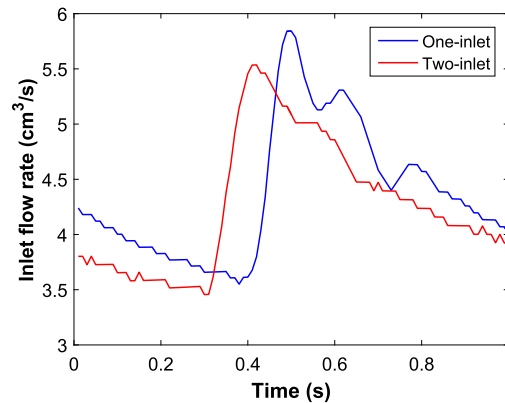


(a) The one-inlet case



(b) The two-inlet case

**Fig. 5.** The meshes used for two patient-specific cerebral arteries with sample partitions into 16 subdomains. Different colors refer to different partitions.



**Fig. 6.** The flow rate profiles imposed on the artery inlets. For the two-inlet case, the flow rate is adjusted by multiplying a factor of 1.53 for the left inlet  $\Gamma_{I_1}$  and 0.51 for the right inlet  $\Gamma_{I_2}$ , respectively.

A flow rate profile with a period of 1 s is imposed on the artery inlet  $\Gamma_I$  for the one-inlet case and on  $\Gamma_{I_i}$  ( $i = 1, 2$ ) for the two-inlet case, as plotted in Fig. 6. Here, for the two-inlet case, the flow rate is adjusted by multiplying two different factors, 1.53 for  $\Gamma_{I_1}$  and 0.51 for  $\Gamma_{I_2}$ , respectively. The mean Reynolds number  $Re$  is defined based on the mean inlet velocity and the diameter of the artery. The total resistance  $R_T$  for these two cases are given according to different clinical experiments in [25]. The time step size is set as  $\Delta t = 0.01$  s. The region-based approach is used for the NE preconditioner, the radius of  $r$ -ball and the restricted size are given as  $r = 0.4$  cm and  $\varepsilon = 0.1$  cm, respectively. The relative tolerance for the nonlinear solver in the NE step is  $\gamma_r^{NE} = 10^{-3}$ . For the linear solver, we fix the overlap size as  $\delta = 1$  and use ILU(3) as the subdomain solver. The effect of these numerical parameters will be studied later. The simulations are carried out using 480 processor cores.

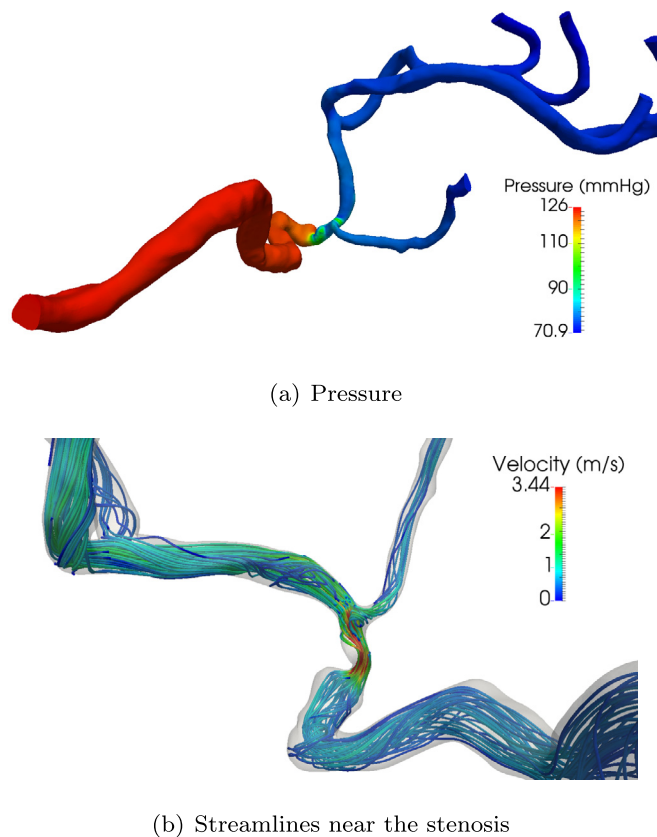


Fig. 7. Numerical solution at  $t = 0.7s$  for the one-inlet case.

Numerical solutions for these two cases at  $t = 0.7s$  are shown in Figs. 7 and 8, including the distribution of pressure and the views of streamlines colored by velocity magnitude. Distal pressure loss is apparently observed due to the severe intracranial stenosis.

Fig. 9 shows the bad region in the region-based INB-NE method for the one-inlet case at two states: (a) the fourth nonlinear step in the first time step and (b) the second nonlinear step in the second time step. When the flow moves forward to a developed state, the bad region is located right behind the stenosis, where some complex flow structures are generated. Fig. 10(a) shows the residual contour for the  $u$ -component in the bad region at the second nonlinear step in the second time step. After the subspace correction is performed, such local high nonlinearities are removed as shown in Fig. 10(b).

#### 4.2.1. Comparison of INB and different INB-NE approaches

The complex geometry of the artery and the use of physical parameters impose great challenges to the nonlinear solution algorithms. It is worthwhile to compare the performance of INB and different INB-NE approaches for such difficult problems. In Figs. 11 and 12, we show the histories of the nonlinear residuals by using INB and INB-NE for the two test cases. In the figures, we denote by Field-based(velocity/pressure) the field-based approach that eliminates only the velocity field or the pressure field. Optimal parameters are used for different approaches:  $r = 0.4$  cm,  $\varepsilon = 0.1$  cm, and  $\beta = \varepsilon_1 = 0.01$ . For the first time step (fractional step (i)), the component-wise INB-NE, the point-wise INB-NE, and the field-based(velocity) INB-NE stagnate with a longer period than the classical INB, while the field-based(pressure) INB-NE performs almost identically with INB. On the other hand, though the field-split and region-based INB-NE methods endure a small overshoot at the beginning, they reach the tolerance with fewer number of iterations. Once the first time step solution is obtained, all INB-NE approaches perform better than the classical INB in the second time step. A detailed comparison for the average number of Newton iterations and compute time for different approaches are shown in Table 3. It is seen that although the cost of the region-based approach is comparable to the other approaches, it leads to the smallest number of global Newton iterations and the least amount of total compute time. As discussed in Section 3.2, the region-based approach returns a better local solution in the domain enclosed by the bad region, which is believed to be crucial for the blood flow in an artery with stenosis. Overall, the region-based approach results in the best performance among different elimination strategies. Therefore, we focus only on the region-based INB-NE for the rest of the paper.

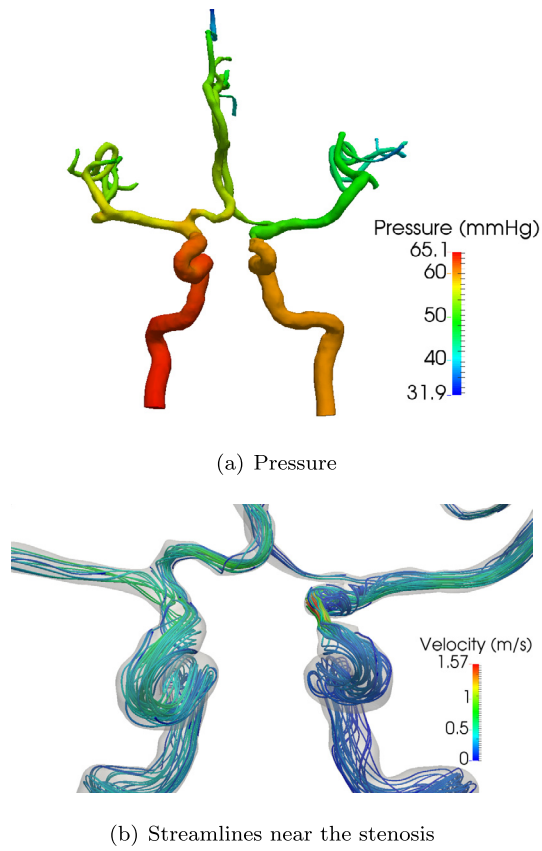


Fig. 8. Numerical solution at  $t = 0.7s$  for the two-inlet case.

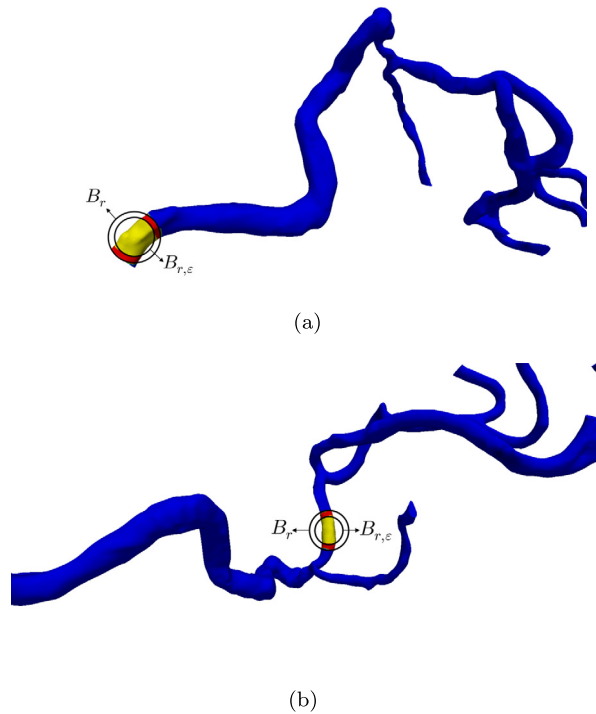
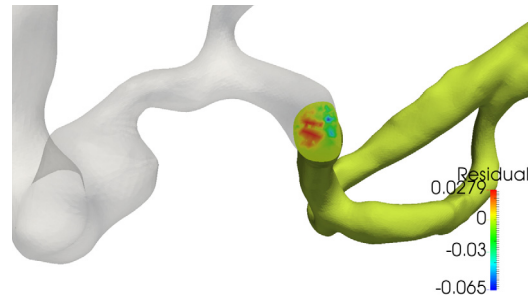
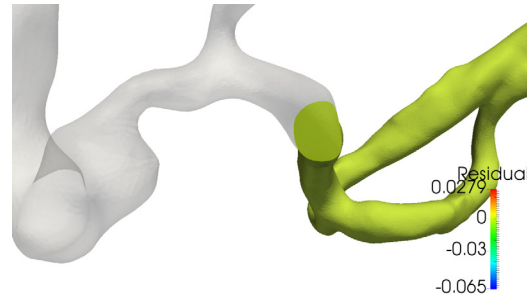


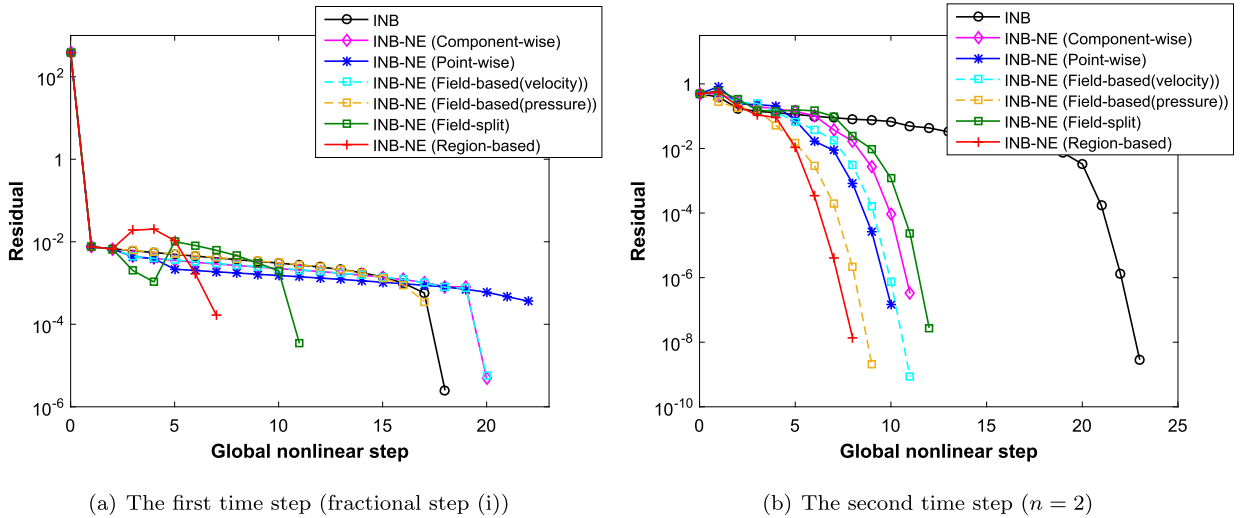
Fig. 9. The bad region  $B_r$  and its restricted part  $B_{r,\epsilon}$  for the one-inlet case at the fourth nonlinear step in the first time step (a) and the second nonlinear step in the second time step (b).  $r = 0.4$  cm,  $\epsilon = 0.1$  cm.



(a) Before the subspace correction



(b) After the subspace correction

**Fig. 10.** Residual contours for the  $u$ -component at the second nonlinear step in the second time step for the one-inlet case.

(a) The first time step (fractional step (i))

(b) The second time step ( $n = 2$ )**Fig. 11.** Nonlinear residual history for the one-inlet case in the first and second time steps.

#### 4.2.2. Impact of the parameters in nonlinear elimination

To understand the impact of the parameters on the performance of the NE preconditioner, we test the one-inlet case with different values of  $r$ ,  $\varepsilon$  and  $\gamma_r^{\text{NE}}$  while fixing other parameters as mentioned above. In Table 4, we show the effect of  $r$  and  $\varepsilon$  on the number of nonlinear and linear iterations, as well as the compute time. When  $r$  becomes larger, the number of bad components increases, in general the cost for solving subspace nonlinear problems may also increase. Paying such a price helps reduce the number of global Newton iterations. On the other hand, adjusting the restricted size  $\varepsilon$  changes the number of global GMRES iterations needed for solving the linear Jacobian system. From Table 4, we find that the pair of values  $(r, \varepsilon) = (0.4, 0.1)$  is suitable to minimize the total compute time and guarantees the convergence of the overall algorithm. In Table 5 we show the impact of  $\gamma_r^{\text{NE}}$  on the performance of the NE preconditioner. It is seen when a smaller  $\gamma_r^{\text{NE}}$  is used, the numbers of global Newton and GMRES iterations decrease, while the time spent on the inner Newton iterations increases. The choice of  $\gamma_r^{\text{NE}} = 10^{-3}$  gives the best results in terms of the total compute time. Therefore, we keep

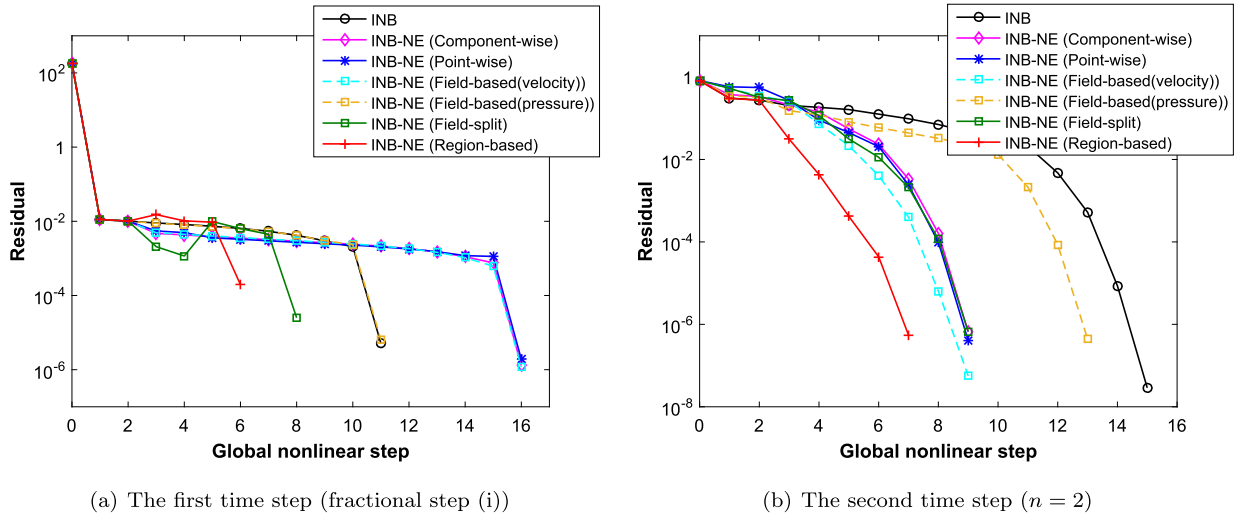


Fig. 12. Nonlinear residual history for the two-inlet case in the first and second time steps.

Table 3

Comparison of INB and different INB-NE approaches for the two test cases. Optimal parameters are used for different approaches:  $r = 0.4$  cm,  $\varepsilon = 0.1$  cm, and  $\beta = \varepsilon_1 = 0.01$ . 'Pct.' denotes the maximum percentage of the eliminated components.

Method	$N_{l_{global}}$	$Time_{total}$ (s)	$N_{ne}$	$N_{l_{ne}}$	$Time_{ne}$ (s)	Pct.
The one-inlet case						
INB	10.50	458.04	-	-	-	-
INB-NE (Component-wise)	8.67	376.80	1.50	3.22	49.27	0.5%
INB-NE (Point-wise)	8.50	501.69	1.83	3.82	22.97	2.3%
INB-NE (Field-based(velocity))	8.50	368.46	1.67	2.90	49.90	75%
INB-NE (Field-based(pressure))	8.00	319.66	1.50	2.00	33.70	25%
INB-NE (Field-split)	7.17	363.13	1.33	4.31	28.46	6.1%/0.1%†
INB-NE (Region-based)	5.67	225.94	1.67	3.20	20.55	7.7%
The two-inlet case						
INB	7.17	206.72	-	-	-	-
INB-NE (Component-wise)	7.00	184.72	1.33	2.25	9.53	0.6%
INB-NE (Point-wise)	6.83	256.05	1.33	3.12	13.02	2.4%
INB-NE (Field-based(velocity))	7.00	191.08	1.17	2.14	9.62	75%
INB-NE (Field-based(pressure))	6.83	160.37	1.50	2.11	9.41	25%
INB-NE (Field-split)	5.50	200.00	1.08	4.15	19.20	5.7%/1.1%†
INB-NE (Region-based)	5.17	149.17	1.33	2.88	14.36	6.3%

a/b† here means that Pct. for the first phase is a and for the second phase is b.

using these optimal choices for all the following tests. It is also seen from the table that the compute times for the NE preconditioner are about 15% of the total compute times.

#### 4.2.3. Robustness with respect to physical parameters

We next study the robustness of both the classical INB method and the proposed INB-NE method with respect to two significant physiological parameters: the total resistance  $R_T$  and the blood viscosity  $\mu$ . Here,  $R_T$  evaluates the resistivity and compliance in the distal artery, and  $\mu$  characterizes the blood viscosity of a patient in certain physical situation. Numerical results for the one-inlet case with different values of  $R_T$  and  $\mu$  are summarized in Table 6 and Table 7, respectively.

As shown in Table 6, when  $R_T$  varies from  $O(10^2)$  to  $O(10^5)$ , the number of global GMRES iterations in INB increases considerably, leading to a substantial growth of the total compute time. For INB-NE, both the numbers of global Newton iterations and GMRES iterations are fewer than those in INB. Overall, the total compute times for INB-NE are roughly 2/3 of those required for INB.

When we decrease the blood viscosity  $\mu$ , the mean Reynolds number in the system increases. It is seen from Table 7 that for INB, both the numbers of global Newton iterations and GMRES iterations increase when  $\mu$  decreases, as a result the total compute time increases as well. In contrast, the total compute times of INB-NE are always less than those of INB, saving over 40% of the total cost. Note that the classical INB fails to converge when  $\mu$  reduces to 0.025 g/(cm.s). Comparatively, the proposed method is more robust with respect to a smaller viscosity.

**Table 4**

Results for the one-inlet case obtained using different values for the preselected constants  $r$  and  $\varepsilon$ . The time step size is 0.01 s, the resistance is  $R_T = 2.1923 \times 10^4$  dynes s/cm<sup>2</sup>, the density and viscosity of blood are  $\rho = 1.06$  g/cm<sup>3</sup> and  $\mu = 0.035$  g/(cm·s), respectively. The tests are carried out on 480 processor cores. The mark '-' indicates that the case fails to converge.

$\varepsilon$ (cm)	0.05	0.1	0.15	0.05	0.1	0.15
	$r = 0.2$ cm			$r = 0.3$ cm		
$NI_{global}$	6.83	7.00	9.66	6.33	6.00	6.50
$LI_{global}$	547.27	514.64	490.83	548.08	431.31	450.54
$N_{ne}$	1.50	1.50	1.50	1.67	1.50	1.83
$NI_{ne}$	3.00	2.78	4.00	4.30	3.11	3.82
$LI_{ne}$	12.30	6.60	19.72	25.12	10.29	10.89
$Time_{ne}$ (s)	18.50	16.64	27.11	29.95	21.10	22.85
$Time_{total}$ (s)	306.05	299.34	411.41	307.46	258.21	276.91
	$r = 0.4$ cm			$r = 0.5$ cm		
$NI_{global}$	6.17	5.67	6.00	5.67	5.33	-
$LI_{global}$	546.65	403.21	415.17	552.62	446.03	-
$N_{ne}$	1.50	1.67	1.50	1.50	1.50	-
$NI_{ne}$	4.67	3.20	3.00	3.33	4.00	-
$LI_{ne}$	31.83	17.19	11.74	22.97	24.17	-
$Time_{ne}$ (s)	33.75	20.55	18.73	22.51	26.55	-
$Time_{total}$ (s)	301.12	225.94	234.37	269.50	229.63	-

**Table 5**

Results for the one-inlet case obtained using different values of  $\gamma_r^{NE}$ . The time step size is 0.01 s, the density and viscosity of blood are  $\rho = 1.06$  g/cm<sup>3</sup> and  $\mu = 0.035$  g/(cm·s), respectively. The resistance is  $R_T = 2.1923 \times 10^4$  dynes s/cm<sup>2</sup>. The tests are carried out on 480 processor cores.

$\gamma_r^{NE}$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$
$NI_{global}$	5.83	5.83	5.67	5.67
$LI_{global}$	422.08	407.34	403.21	377.62
$N_{ne}$	1.50	1.50	1.67	1.67
$NI_{ne}$	2.33	2.77	3.20	3.70
$LI_{ne}$	17.52	18.08	17.19	16.60
$Time_{ne}$ (s)	16.51	18.83	20.55	23.27
$Time_{total}$ (s)	230.57	230.69	225.94	226.98

**Table 6**

Results for the one-inlet case obtained using different values for the resistance  $R_T$ . The time step size is 0.01 s, the density and viscosity of blood are  $\rho = 1.06$  g/cm<sup>3</sup> and  $\mu = 0.035$  g/(cm·s), respectively. The tests are carried out on 480 processor cores.

$R_T$ (dynes·s/cm <sup>2</sup> )	$2.1923 \times 10^2$	$2.1923 \times 10^3$	$2.1923 \times 10^4$	$2.1923 \times 10^5$
The classical INB method				
$NI_{global}$	9.83	10.33	10.50	8.17
$LI_{global}$	246.49	295.73	592.75	1431.74
$Time_{total}$ (s)	277.77	306.55	458.04	676.65
The proposed INB-NE method				
$NI_{global}$	5.83	5.83	5.67	5.67
$LI_{global}$	216.43	235.46	403.21	1080.85
$N_{ne}$	1.67	1.67	1.67	1.33
$NI_{ne}$	3.40	4.30	3.20	3.00
$LI_{ne}$	18.68	27.35	17.19	19.08
$Time_{ne}$ (s)	22.82	31.22	20.55	20.00
$Time_{total}$ (s)	186.87	205.82	225.94	400.82

#### 4.2.4. Robustness with respect to the time step size

We further investigate how the time step size influences the performance of the algorithms. Although the fully implicit scheme (12) allows to use a large time step size, it is still a challenge to solve the nonlinear system if  $\Delta t$  is given inappropriately. Results for the one-inlet case obtained using different values of  $\Delta t$  are shown in Table 8. For INB, the numbers of global Newton iterations and GMRES iterations increase rapidly as  $\Delta t$  increases. INB fails to converge when  $\Delta t$  increases to 0.04 s. In contrast, INB-NE features a good convergence when  $\Delta t$  is up to 0.06 s. It is observed that the performance of NE preconditioning is not sensitive to the change of the time step size. In all, the total compute time of INB-NE increases slowly when a larger  $\Delta t$  is used, this merit is favorable for long time calculation of medical applications.



**Table 7**

Results for the one-inlet case obtained using different values for the viscosity  $\mu$ . The time step size is 0.01 s, the blood density is  $\rho = 1.06 \text{ g/cm}^3$ , and the resistance is chosen as  $R_T = 2.1923 \times 10^4 \text{ dynes s/cm}^2$ . The tests are carried out on 480 processor cores. The mark '-' indicates that the case fails to converge.

$\mu \text{ (g/(cm}\cdot\text{s))}$	0.025	0.03	0.035	0.045	0.055
The classical INB method					
$N_{l_{global}}$	-	10.83	10.50	7.67	6.67
$L_{l_{global}}$	-	615.92	592.75	593.72	561.88
$\text{Time}_{total} \text{ (s)}$	-	504.64	458.04	334.26	280.38
The proposed INB-NE method					
$\mu \text{ (g/(cm}\cdot\text{s))}$	0.025	0.03	0.035	0.045	0.055
$N_{l_{global}}$	7.17	6.17	5.67	5.33	5.17
$L_{l_{global}}$	440.72	468.49	403.21	345.09	383.45
$N_{ne}$	2.00	1.67	1.67	1.50	1.17
$N_{l_{ne}}$	4.25	3.90	3.20	3.00	2.57
$L_{l_{ne}}$	22.82	21.39	17.19	16.48	11.50
$\text{Time}_{ne} \text{ (s)}$	28.29	26.12	20.55	20.27	16.98
$\text{Time}_{total} \text{ (s)}$	316.61	274.80	225.94	198.96	193.20

**Table 8**

Results for the one-inlet case obtained using different values for the time step size  $\Delta t$ . The density and viscosity of blood are  $\rho = 1.06 \text{ g/cm}^3$  and  $\mu = 0.035 \text{ g/(cm s)}$ , respectively. The resistance is  $R_T = 2.1923 \times 10^4 \text{ dynes s/cm}^2$ . The tests are carried out on 480 processor cores. The mark '-' indicates that the case fails to converge.

$\Delta t \text{ (s)}$	0.005	0.01	0.02	0.04	0.06
The classical INB method					
$N_{l_{global}}$	6.33	10.50	11.66	-	-
$L_{l_{global}}$	522.16	592.75	605.17	-	-
$\text{Time}_{total} \text{ (s)}$	253.95	458.04	537.44	-	-
The proposed INB-NE method					
$N_{l_{global}}$	5.50	5.67	7.00	8.17	8.83
$L_{l_{global}}$	410.39	403.21	357.26	320.35	314.60
$N_{ne}$	1.33	1.67	1.33	1.50	1.50
$N_{l_{ne}}$	2.63	3.20	2.87	2.56	3.00
$L_{l_{ne}}$	16.48	17.19	10.74	11.74	15.44
$\text{Time}_{ne} \text{ (s)}$	17.76	20.55	19.40	17.35	20.06
$\text{Time}_{total} \text{ (s)}$	214.08	225.94	254.34	277.00	299.98

#### 4.2.5. Study of the parallel scalability

In this subsection, we report the parallel performance of INB-NE and focus on two important factors: the fill-in level  $k$  of the ILU subsolver and the overlap size  $\delta$  of the linear RAS preconditioner. We first fix  $\delta = 1$  and study the effect of  $k$  on the strong scalability by increasing the number of processor cores  $np$  from 240 to 1,920. In this test, a fixed mesh with 5,225,949 elements and 1,069,767 nodes is used for the two-inlet case, and the time step size is  $\Delta t = 0.0025\text{s}$ . The simulation is stopped after five time steps. Numerical results are summarized in Table 9. From the table we observe that, as the number of processor cores increases, the number of Newton iterations stays almost constant while the number of GMRES iterations grows mildly. In general, the linear solver converges better when a larger fill-in level is used, but this does not necessarily result in a better performance in terms of the total compute time. It is observed that ILU(2) leads to the smallest amount of total compute time when  $np$  is up to 1,920. Next, we fix the fill-in level to  $k = 2$  and study the effect of  $\delta$  on the parallel performance by varying  $np$  from 512 to 3,072. We use a refined mesh with 9,807,822 elements and 1,854,661 nodes for the one-inlet case in this experiment. As shown in Table 10, a larger overlapping size usually results in fewer number of GMRES iterations, but the cost per iteration for communication and computation increases. It is seen that the small overlap  $\delta = 1$  provides a good trade off between the linear iterations and the total compute time. In Fig. 13, we report the speedup for INB-NE with the above optimized parameters. Overall, a reasonably good efficiency of around 65% is achieved for both cases.

## 5. Concluding remarks

We developed a nonlinearly preconditioned inexact Newton method for the simulation of blood flow in human artery with stenosis. The blood flow is modeled by the unsteady incompressible Navier-Stokes equations discretized by a fully implicit finite element method. When a classical inexact Newton method with backtracking is used to solve the resulting nonlinear system, it often suffers from slow convergence or stagnations. In this work, we proposed a nonlinear elimination preconditioner to handle this issue. The key idea is to perform subspace correction to remove the local high nonlineari-

**Table 9**

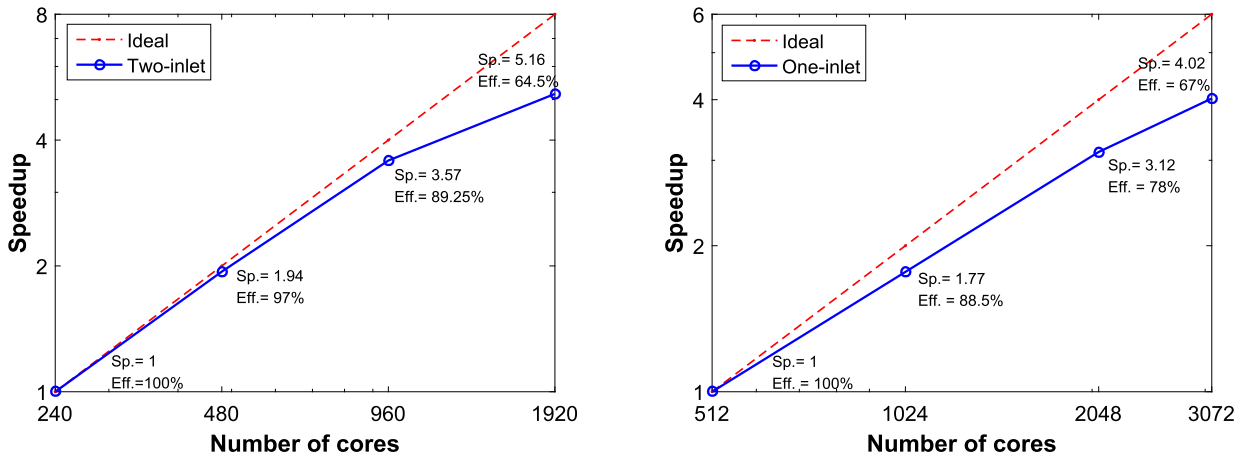
Scalability test for the two-inlet case obtained using the INB-NE method with different fill-in levels of the ILU subsolver. A fixed mesh with 5,225,949 elements and 1,069,767 nodes is used. The overlap size of RAS preconditioner is  $\delta = 1$ . The time step size is 0.0025 s.

$np$	Subsolve	$Nl_{global}$	$Ll_{global}$	$Nl_{ne}$	$Ll_{ne}$	$Time_{ne}$ (s)	$Time_{total}$ (s)
240	ILU(0)	3.83	1068.52	1.33	6.75	11.04	243.79
	ILU(1)	3.67	668.59	1.20	3.66	10.42	171.54
	ILU(2)	3.83	525.70	1.20	3.33	10.82	171.83
	ILU(3)	3.50	488.47	1.20	3.00	11.54	179.63
480	ILU(0)	3.83	1115.04	1.33	6.50	6.19	133.81
	ILU(1)	3.50	683.47	1.20	4.00	5.83	87.84
	ILU(2)	3.67	532.91	1.20	3.67	6.05	88.66
	ILU(3)	3.50	491.91	1.20	3.50	6.47	96.40
960	ILU(0)	3.83	1179.70	1.33	6.63	3.72	77.45
	ILU(1)	3.50	743.81	1.20	4.00	3.69	52.88
	ILU(2)	3.50	541.76	1.20	3.67	3.76	48.08
	ILU(3)	3.67	505.50	1.20	3.50	4.02	56.74
1,920	ILU(0)	3.83	1436.56	1.20	5.00	2.27	53.17
	ILU(1)	3.67	900.36	1.20	4.16	2.30	38.33
	ILU(2)	3.67	653.05	1.20	3.67	2.34	33.31
	ILU(3)	3.67	630.18	1.20	3.83	2.43	38.58

**Table 10**

Scalability test for the one-inlet case obtained using the INB-NE method with different overlap size  $\delta$  in the RAS preconditioner. A fixed mesh with 9,807,822 elements and 1,854,661 nodes is used. ILU(2) is used as the subdomain solver. The time step size is 0.0025 s.

$np$	$\delta$	$Nl_{global}$	$Ll_{global}$	$Nl_{ne}$	$Ll_{ne}$	$Time_{ne}$ (s)	$Time_{total}$ (s)
512	0	3.83	763.30	1.40	12.57	13.17	225.36
	1	3.83	677.39	1.40	11.29	13.19	212.89
	2	4.00	643.38	1.40	10.57	13.28	218.39
1,024	0	4.17	811.52	1.40	13.43	7.69	135.79
	1	4.17	665.52	1.40	11.86	7.77	120.54
	2	4.17	633.04	1.40	11.14	7.83	120.34
2,048	0	4.00	805.46	1.50	18.00	5.79	88.80
	1	4.00	692.75	1.40	13.86	5.21	76.47
	2	4.00	645.00	1.40	12.71	5.31	73.14
3,072	0	4.17	774.56	1.40	16.86	4.08	58.37
	1	4.00	687.17	1.40	14.71	4.17	52.92
	2	4.00	687.04	1.40	13.71	4.19	56.65



**Fig. 13.** Scalability results for the two-inlet case (left) and the one-inlet case (right). ‘Sp.’ denotes the speedup and ‘Eff.’ denotes the efficiency. The mesh used for the two-inlet case has 5,225,949 elements and 1,069,767 nodes, and the mesh used for the one-inlet case has 9,807,822 elements and 1,854,661 nodes. The overlap size of RAS preconditioner is  $\delta = 1$ . ILU(2) is used as the subdomain solver. The time step size is 0.0025 s.

ties that cause difficulty for the convergence of the inexact Newton method. We tested the algorithm using two realistic cases of human cerebral artery with stenosis. Results of numerical experiments show that the proposed method is more robust and faster than the classical method with respect to some physical and numerical parameters, and is scalable on a supercomputer with thousands of processor cores.

### Declaration of competing interest

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

### Acknowledgement

The first author was supported in part by NSF of China under grant 11701547. The second and third authors were supported in part by the National Key R&D Program of China under grant 2016YFB0200601 and by the Shenzhen Basic Research Program under grants JCYJ20160331193229720. The fourth author was supported in part by NSF of China under DMS-1720366.

### References

- [1] ANSYS, <https://www.ansys.com/>, 2019.
- [2] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc Users Manual, Argonne National Laboratory, 2019, <http://www.mcs.anl.gov/petsc>.
- [3] P. Brune, M. Knepley, B. Smith, X. Tu, Composing scalable nonlinear algebraic solvers, *SIAM Rev.* 57 (2015) 535–565.
- [4] X.-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, D.P. Young, Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation, *SIAM J. Sci. Comput.* 19 (1998) 246–265.
- [5] X.-C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, *SIAM J. Sci. Comput.* 21 (1999) 792–797.
- [6] X.-C. Cai, D.E. Keyes, Nonlinearly preconditioned inexact Newton algorithms, *SIAM J. Sci. Comput.* 24 (2002) 183–200.
- [7] X.-C. Cai, X. Li, Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity, *SIAM J. Sci. Comput.* 33 (2011) 746–762.
- [8] J.R. Cebral, M.A. Castro, J.E. Burgess, R.S. Pergolizzi, M.J. Sheridan, C.M. Putman, Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models, *Am. J. Neuroradiol.* 26 (2005) 2550–2559.
- [9] P. Crosetto, S. Deparis, G. Fouresteray, A. Quarteroni, Parallel algorithm for fluid-structure interaction problems in haemodynamics, *SIAM J. Sci. Comput.* 33 (2011) 1598–1622.
- [10] J.E. Dennis, R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.
- [11] S.C. Eisenstat, H.F. Walker, Globally convergent inexact Newton method, *SIAM J. Optim.* 4 (1994) 393–422.
- [12] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, *SIAM J. Sci. Comput.* 17 (1996) 16–32.
- [13] L.P. Franca, S.L. Frey, Stabilized finite element methods, II: the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.* 99 (1992) 209–233.
- [14] W.D. Gropp, D.K. Kaushik, D.E. Keyes, B.F. Smith, High-performance parallel implicit CFD, *Parallel Comput.* 27 (2001) 337–362.
- [15] J. Huang, C. Yang, X.-C. Cai, A nonlinearly preconditioned inexact Newton algorithm for steady lattice Boltzmann equations, *SIAM J. Sci. Comput.* 38 (2016) A1701–A1724.
- [16] F.-N. Hwang, H.-L. Lin, X.-C. Cai, Two-level nonlinear elimination based preconditioners for inexact Newton methods with application in shocked duct flow calculation, *Electron. Trans. Numer. Anal.* 37 (2010) 239–251.
- [17] F.-N. Hwang, Y.-C. Su, X.-C. Cai, A parallel adaptive nonlinear elimination preconditioned inexact Newton method for transonic full potential equation, *Comput. Fluids* 110 (2015) 96–107.
- [18] H.J. Kim, I.E. Vignon-Clementel, C.A. Figueroa, J.F. LaDisa, K.E. Jansen, J.A. Feinstein, C.A. Taylor, On coupling a lumped parameter heart model and a three-dimensional finite element aorta model, *Ann. Biomed. Eng.* 37 (2009) 2153–2169.
- [19] F. Kong, X.-C. Cai, A highly scalable multilevel Schwarz method with boundary geometry preserving coarse spaces for 3D elasticity problems on domains with complex geometry, *SIAM J. Sci. Comput.* 38 (2016) C73–C95.
- [20] X.Y. Leng, K.S. Wong, D.S. Liebeskind, Evaluating intracranial atherosclerosis rather than intracranial stenosis, *Stroke* 45 (2014) 645–651.
- [21] D.S. Liebeskind, E. Feldmann, Fractional flow in cerebrovascular disorders, *Interv. Neurol.* 1 (2012) 87–99.
- [22] L. Liu, D.E. Keyes, Field-split preconditioned inexact Newton algorithms, *SIAM J. Sci. Comput.* 37 (2015), A1388–A1409.
- [23] L. Liu, D.E. Keyes, Convergence analysis for the multiplicative Schwarz preconditioned inexact Newton algorithm, *SIAM J. Numer. Anal.* 54 (2016) 3145–3166.
- [24] L. Liu, D.E. Keyes, R. Krause, A note on adaptive nonlinear preconditioning techniques, *SIAM J. Sci. Comput.* 40 (2018) A1171–A1186.
- [25] J. Liu, Z. Yan, Y. Pu, W.-S. Shiu, J. Wu, R. Chen, X. Leng, H. Qin, X. Liu, B. Jia, L. Song, Y. Wang, Z. Miao, Y. Wang, L. Liu, X.-C. Cai, Functional assessment of cerebral artery stenosis: a pilot study based on computational fluid dynamics, *J. Cereb. Blood Flow Metab.* 37 (2017) 2567–2576.
- [26] F. Migliavacca, R. Balossino, G. Pennati, G. Dubini, T.Y. Hsia, M.R. de Leval, E.L. Bove, Multiscale modelling in biofluidynamics: application to reconstructive paediatric cardiac surgery, *J. Biomech.* 39 (2006) 1010–1020.
- [27] Mimics, <https://www.materialise.com/en/medical/software/mimics>, 2019.
- [28] C.D. Murray, The physiological principle of minimum work, I: the vascular system and the cost of blood volume, *Proc. Natl. Acad. Sci. USA* 12 (1926) 207–214.
- [29] Y. Saad, *Iterative Methods for Sparse Linear Systems*, second ed., SIAM, 2003.
- [30] C.A. Taylor, T.J.R. Hughes, C.K. Zarins, Finite element modeling of blood flow in arteries, *Comput. Methods Appl. Mech. Eng.* 158 (1998) 155–196.
- [31] C.A. Taylor, M.T. Draney, J.P. Ku, D. Parker, B.N. Steele, K. Wang, C.K. Zarins, Predictive medicine: computational techniques in therapeutic decision-making, *Comput. Aided Surg.* 4 (1999) 231–247.
- [32] C.A. Taylor, T.A. Fonte, J.K. Min, Computational fluid dynamics applied to cardiac computed tomography for noninvasive quantification of fractional flow reserve, *J. Am. Coll. Cardiol.* 61 (2013) 2233–2241.
- [33] I.E. Vignon-Clementel, C.A. Figueroa, K.E. Jansen, C.A. Taylor, Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 3776–3796.

- [34] I.E. Vignon-Clementel, C.A. Figueroa, K.E. Jansen, C.A. Taylor, Outflow boundary conditions for 3D simulations of non-periodic blood flow and pressure fields in deformable arteries, *Comput. Methods Biomech. Biomed. Eng.* 13 (2010) 625–640.
- [35] C.H. Whiting, K.E. Jansen, A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis, *Int. J. Numer. Methods Fluids* 35 (2001) 93–116.
- [36] Y. Wu, X.-C. Cai, A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition, *Comput. Fluids* 45 (2011) 92–102.
- [37] Y. Wu, X.-C. Cai, A fully implicit domain decomposition based ALE framework for three-dimensional fluid-structure interaction with application in blood flow computation, *J. Comput. Phys.* 258 (2014) 524–537.
- [38] H. Yang, F.-N. Hwang, X.-C. Cai, Nonlinear preconditioning techniques for full-space Lagrange-Newton solution of PDE-constrained optimization problems, *SIAM J. Sci. Comput.* 38 (2016) A2756–A2778.
- [39] H. Yang, C. Yang, S. Sun, Active-set reduced-space methods with nonlinear elimination for two-phase flow problems in porous media, *SIAM J. Sci. Comput.* 38 (2016) B593–B618.
- [40] H. Yang, S. Sun, C. Yang, Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media, *J. Comput. Phys.* 332 (2017) 1–20.
- [41] H. Yang, F.-N. Hwang, An adaptive nonlinear elimination preconditioned inexact Newton algorithm for highly local nonlinear multicomponent PDE systems, *Appl. Numer. Math.* 133 (2018) 100–115.