Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A highly parallel fully implicit domain decomposition method for the simulation of the hemodynamics of a patient-specific artery at the full-body scale

Shanlin Qin [a,1], Rongliang Chen [a,b,1], Bokai Wu [a], Xiao-Chuan Cai [c,*]

[a] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China*
[b] *Shenzhen Key Laboratory for Exascale Engineering and Scientific Computing, Shenzhen, Guangdong 518055, China*
[c] *Department of Mathematics, University of Macau, Macau, China*

ARTICLE INFO

Article history:
Received 3 July 2022
Received in revised form 6 October 2022
Accepted 23 October 2022
Available online 26 October 2022

Keywords:
Patient-specific-full-body artery
Blood flow distribution
Incompressible Navier-Stokes equations
Finite element method
Domain decomposition method
Parallel computing

ABSTRACT

The numerical simulation of blood flows in the human body with a certain level of clinical accuracy is important for the understanding of the human physiology. The success of the modeling relies on a robust numerical method with the corresponding software that can handle the complex geometry, the complex fluid flows and run efficiently on a supercomputer. In this work, we introduce a highly parallel domain decomposition method to solve the three-dimensional incompressible Navier-Stokes equations on a patient-specific artery at the full-body scale from neck to feet with 222 outlets and a minimum diameter around 1.0 mm. A locally refined, unstructured mesh is used to resolve the complex fluid flow. Moreover, a two-level method is introduced to determine the model parameters in the Windkessel outlet boundary condition to guarantee clinically correct flow distributions to 14 major regions. A fully implicit Newton-Krylov-Schwarz method is used to solve the nonlinear algebraic system at each time step and numerical experiments show that the proposed method is robust with respect to the complex geometry, the graph-based partition of the complex mesh, the ill-conditioned sparse systems with locally dense blocks, and different model parameters and is scalable with up to 15,360 processor cores. With the proposed method, one simulation of the blood flow in a full-body arterial network can be obtained in about 8 hours per cardiac cycle, which enables its potential use in a wide range of clinical scenarios.

1. Introduction

Image-based numerical simulations of the human hemodynamics have now been widely recognized as a potentially useful tool for the assessment and management of various cardiovascular diseases, including, for example, the hypertension, the atherosclerosis and the aortic aneurysm and dissection [1–3]. In terms of the complexity of the geometrical models, the simulation of blood flows has progressed from a single bifurcation to multiple junctions and then to the arterial tree over the past few decades [4]. More recently, the advances of the medical imaging technology and the computational capability have

* Corresponding author.
E-mail address: xccai@um.edu.mo (X.-C. Cai).

[1] Co-first author.

https://doi.org/10.1016/j.jcp.2022.111730

made it possible for the simulation of blood flows in a full-body arterial network, which has a wide range of application scenarios such as the estimation of several important hemodynamic metrics including the central artery pulse pressure and augmentation index [5], the understanding of the multi-organ interactions [6] and the delivery of oxygen, drugs or other substances [7].

Due to the low computational costs, the reduced-dimensional (zero or one-dimensional) models are preferred when the concern is about the blood pressure and the flow waveform, and therefore they are frequently used in the analysis of the wave intensity and the pulse wave velocity [8–10]. We are interested in the full three-dimensional (3D) model since it can capture the details of the local fluid patterns, such as the helix and vortex flows [11,12]. Simulations are carried out on the vascular system consisting of arteries to all the major regions with low-dimensional models imposed on the outlets as the boundary condition to consider the influence of the peripheral flows [13–15]. This multi-scale model is a direct reflection of the multi-scale nature of the arterial tree, which is composed of arteries, arterioles and capillaries. The main challenges of the 3D blood flow simulation are its high computational cost and difficult to resolve flow fields, which requires the use of supercomputers with a properly designed parallel algorithm that is scalable to a large number of processor cores and robust with respect to different model parameters and spatiotemporal mesh sizes.

We first provide a brief literature review of large-scale 3D blood flow simulations, with focus on their numerical methods and computational capabilities. Grinberg et al. introduced a spectral/hp element method to simulate the blood flow in a cranial arterial network that has 65 branch vessels. By using 3265 processor cores of a Cray XT3 supercomputer, the simulation took 80 hours per cardiac cycle based on a mesh with 459,250 spectral elements [4]. Zhou et al. adopted a finite element package PHASTA to solve the Navier-Stokes equtions for blood flow simulations in a whole-body arterial tree that has 78 outlets. Using 2,048 processor cores of a Cray XT5 supercomputer, it takes 1.5 hours for one cardiac cycle simulation based on a mesh with 42.8 million elements [16]. Xiao et al. simulated the blood flow in a whole-body artery with 82 outlets. For a mesh with 14,438,720 tetrahedral elements, the calculation took 48 hours per cardiac cycle by using 384 processor cores of the Ranger supercomputer at the Texas Advanced Computing Center [17]. Based on a regular Cartesian grid, Randles et al. adopted a lattice Boltzmann method to simulate the hemodynamics in a systemic arterial tree by using 1.57 million cores of the Sequoia Blue Gene/Q supercomputer [18]. Liu et al. applied a three-level nested block preconditioner to solve the discretized governing equation and simulate the hemodynamics in a pulmonary artery consisting of 722 branches. By using 720 processor cores of the Stampede2 supercomputer, the simulation took 4 hours for one cardiac cycle based on a mesh with 26.1 million tetrahedral elements [19]. In short, existing researches have showed the feasibility of blood flow simulations in a large arterial network, however, parallel simulations that can maintain high scalability while providing clinically meaningful hemodynamic results are still scarce.

In our previous work, we investigated a parallel Newton-Krylov-Schwarz (NKS) method for the study of hemodynamics in the abdominal aorta [20]. In the present work, we push the limit to include a full-body artery from neck to feet that includes 222 outlet branches. In order to carry out a simulation with clinically meaningful flow distributions to specific regions, the mesh generation is a very important step. With a sufficiently fine surface mesh, it is not too difficult to obtain a mesh that captures all the geometrical features of the artery, but to obtain the correct flow field, more refinements are necessary and mesh independence studies should be performed, especially at sites with acute geometrical changes where complex flows are usually developed, such as the aortic arch and arterial branching. Another important issue to consider is the choice of the outlet boundary condition. In this work, the three-element Windkessel model is applied to all outlets, which is physiologically more accurate than the constant pressure and the resistive boundary conditions [13], but nontrivial to calibrate the model parameters to achieve correct blood flow distributions. Iterative methods are often used to convert this problem into an optimization problem for adjusting values for the terminal model parameters. Such methods have been used, for example, by Blanco et al. to obtain target blood flow distributions for 16 and 144 regions in the 1D arm and full-body blood flow simulations respectively [21,22]; and by Xiao et al. to guarantee correct blood flow distributions for 9 regions during a 3D full-body blood flow simulation [17]. The method needs to solve the governing equation repeatedly and therefore significantly increases the total computing time. Based on a relationship between the resistance and the flow rate introduced in [23], we propose a two-level method to obtain clinically measured blood flow distributions to 14 regions, including the head, heart, liver, stomach, spleen, intestine, kidneys, adrenal glands, arms and legs. This method requires no additional computation since it predetermines the values of the terminal parameters by two steps, namely splitting the total resistance and compliance to each region based on the clinical measurements, and from each region to individual outlets based on the cross-sectional area of the outlets. The aim of the present work is to apply a robust and scalable numerical method for the blood flow simulation at the full-body scale and investigate the numerical accuracy in terms of mesh sizes and the clinical accuracy in providing correct blood flow distributions to various regions.

The rest of this paper is organized as follows. In section 2, we first describe the procedures adopted for the image segmentation and mesh generation, and then introduce the governing equations and its finite element discretization with an emphasis on the outlet boundary condition and NKS solver. In section 3, we validate our method by solving a benchmark problem and then apply it to investigate the hemodynamics in a full-body artery, where regional blood flow distributions are investigated. We also show the robustness and scalability of the proposed algorithm. Finally, we draw some conclusions in section 4.

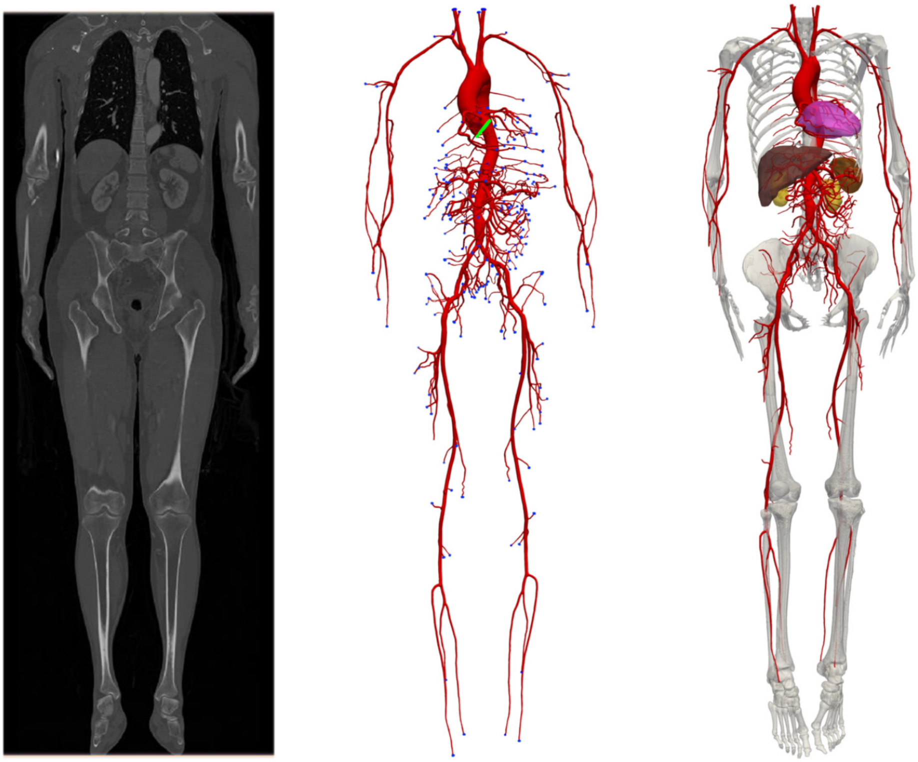

Fig. 1. Geometrical model of a full-body artery from neck to feet with 1 inlet (marked in green) and 222 outlets (marked in blue) reconstructed from the CT image. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2. Methods

2.1. Geometry reconstruction

As shown in the left figure of Fig. 1, the computed tomography (CT) angiography image of a person with a slight abdominal aneurysm is used to reconstruct the geometrical model of a full-body artery from neck to feet by using a commercial software package. The image consists of $512 \times 512 \times 1995$ voxels with a resolution of $0.98 \times 0.98 \times 0.75$ mm. The preliminary segmentation and reconstruction consist of the following basic steps:

1. Adjust the image contrast and highlight the vascular anatomy with threshold values 259 and 3071 so that the entire artery is included;
2. Segment the artery by using a dynamic region-growing method [24];
3. Check the vascular network connectivity and make manual corrections guided by the human anatomical atlases in [25];
4. Repeat steps 2 and 3 until a preliminary arterial network is obtained;
5. Reconstruct the rough 3D geometrical model from the segmentation obtained from step 4 and export the arterial geometry.

After the above steps of reconstruction, some manual adjusting and connecting are often necessary for the small arteries of the geometrical model. Then, a smoothing step is required to remove obvious defects and sharp angles without seriously altering the original features of the artery. Furthermore, blood vessels are truncated and identified at the proximal and distal ends for the prescription of inlet and outlet boundary conditions, respectively. After these steps, we obtain a geometrical model that contains all the primary arteries in the body, as shown in the middle figure of Fig. 1. It includes 1 inlet with a diameter of 38.03 mm and 222 outlets with the minimum and maximum diameters of 0.93 and 7.15 mm, respectively. We remark that this geometrical model includes a more complex and realistic full-body artery comparing with the 78-outlet case in [16] and the 82-outlet case in [17], both of which have a minimum outlet diameter 2 mm. In the right figure of Fig. 1, we show the arterial network against a background of the human skeleton along with some of the primary regions, to which the blood supplies will be considered in the simulation.

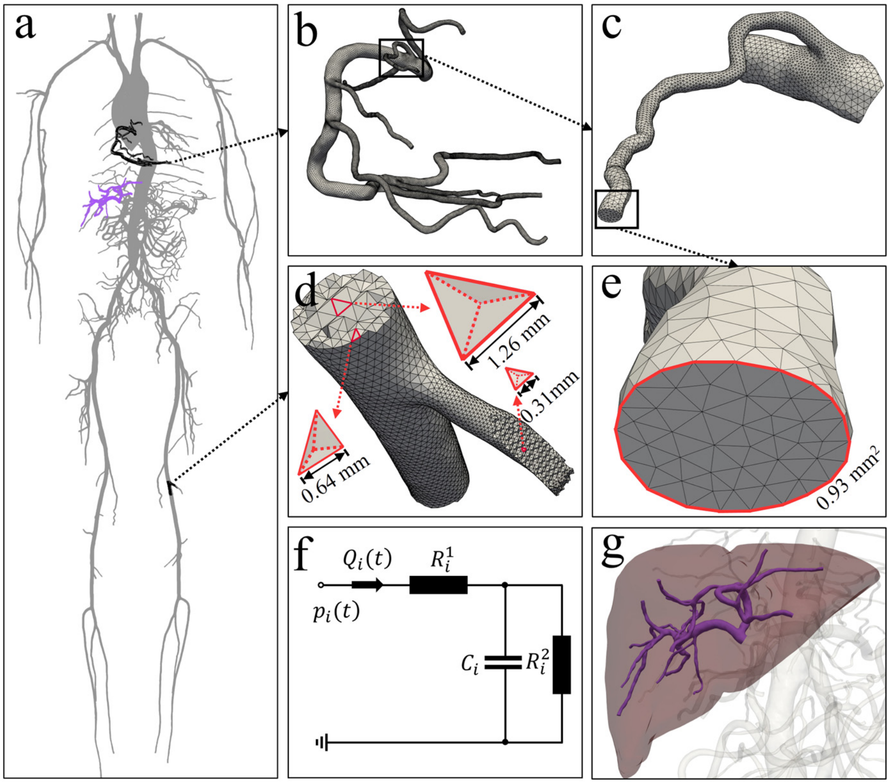

Fig. 2. (a) a global vision of the entire mesh; (b-e) the close-up views to display the change of mesh sizes in the right coronary artery and a small portion of the artery in the left leg; (f) the three-element Windkessel model for outlet boundary conditions and (g) the hepatic artery that supplies blood to the liver.

2.2. Mesh generation

Mesh generation plays a critical role in obtaining numerically accurate results, especially for such a complex geometry that involves branch vessels of diameters ranging from 0.93 to 38.03 mm, as shown in Fig. 2. To balance the computational cost and the solution accuracy, non-uniform meshes with adjusted local mesh densities are required and generated by using some commercial software packages to cover the computational domain of the arterial network. Our meshing strategy is two-fold:

1. *Geometry preservation by controlling the surface mesh size*. Specifically, the surface mesh is non-uniformly generated with its size depending on the equivalent diameter of the artery. For example, in Fig. 2 of a mesh with 21,168,339 elements, the mean edge length of the surface mesh is 0.63 mm for large arteries with diameters >4 mm, which ensures at least 20 elements placed around the perimeter of the artery with a diameter about 4 mm. The mean edge length decreases to 0.31 mm for small arteries with diameters between 1-4 mm, which ensures 10 elements for the perimeter of the artery with a diameter about 1 mm. Fig. 2 b and c show the use of different surface mesh sizes to describe the geometrical details of the right coronary artery and its branches. With this method, the generated surface mesh can preserve the multi-scale nature of the arterial geometry and also limit the total number of elements to control the overall cost of the computation. Meanwhile, since the inlet and outlet surfaces are identified before the mesh generation, the sizes of their meshes can be adjusted individually. For example, for a small branch of the right coronary artery shown in Fig. 2e, we create 71 triangles for the outlet surface with an area of 0.93 mm^2 by using a mean edge length of 0.17 mm.
2. *Flow resolving by adjusting the interior volume mesh size*. Generally, the size of the interior volume mesh is controlled to be larger than the size of the surface mesh to limit the total number of elements. This can be shown in a small portion of the artery in Fig. 2d, where edge lengths of the representative volume and surface elements are 1.62 and 0.64 mm respectively. However, for regions where flow patterns are interested, mesh independence studies should be carried out to resolve the complex fluid flow. In this case, local mesh refinements are considered since it is virtually impossible to refine the mesh globally due to the prohibitive computational cost. For example, the locally isotropic mesh refinement with a mean edge length of 0.31 mm is applied on the mesh of a branch vessel in Fig. 2d. In the numerical experiments, we investigate results obtained by using several different meshes since different quantities of interests at different locations require different mesh sizes to resolve.

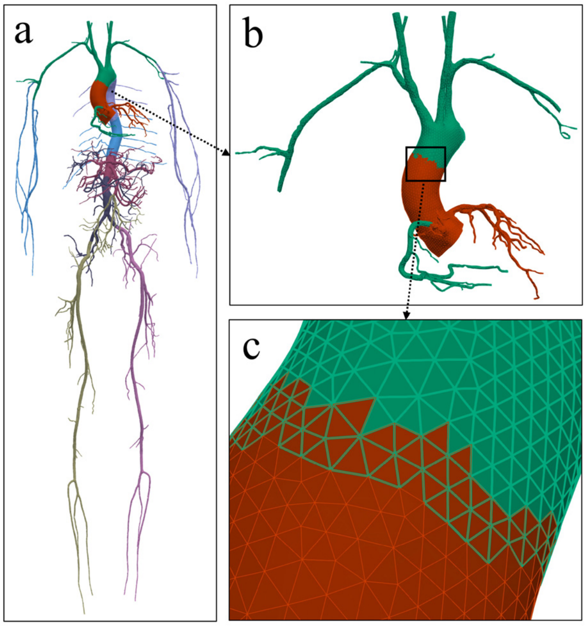

Fig. 3. (a) a sample coarse mesh of the artery partitioned into 8 sub-meshes represented by different colors; (b) the enlarged view of two sub-meshes, where the green one contains two pieces of the artery that are not directly connected and (c) the overlaps between the adjacent sub-meshes to be used by the preconditioning algorithm.

We mention that the image segmentation and the mesh generation steps are quite time consuming. For the purpose of parallel processing, ParMETIS [26] is adopted to partition the global mesh into a number of sub-meshes with roughly the same number of elements to ensure the load balance and the partition also tries to minimize the size of the interfaces between the sub-meshes to minimize the inter-processor communication cost. The sub-meshes are then distributed to the same number of processor cores for the computation. Fig. 3 shows a sample of the mesh partitioning, where a small mesh is partitioned into 8 overlapping sub-meshes indicated by 8 different colors. It should be pointed out that sub-meshes may contain some pieces of the artery that are not directly connected which is not desirable but difficult to avoid.

2.3. Mathematical model of the blood flow

In large arteries, where the velocity and the shear rate are high, the blood flow can be considered as a Newtonian flow and described by the following incompressible Navier-Stokes equations

$$\begin{cases} \rho\left(\dfrac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right) - \nabla\cdot\boldsymbol{\sigma} = 0, \quad \text{in} \quad \Omega\times(0,T], \\ \nabla\cdot\boldsymbol{u} = 0, \quad \text{in} \quad \Omega\times(0,T], \end{cases} \tag{1}$$

where $\Omega \in R^3$ is the arterial domain shown in Fig. 1, T is the time period of the simulation which is a cardiac cycle, $\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u})$ is the stress tensor with $\boldsymbol{\epsilon}(\boldsymbol{u}) = (\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)/2$ and I is a 3×3 identity matrix, ρ is the fluid density, μ is the fluid dynamic viscosity, and $p = p(\boldsymbol{x},t)$ and $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x},t) = (u_1,u_2,u_3)^T$ are the pressure and velocity vector to be computed approximately.

To solve (1), proper initial and boundary conditions should be specified. Since the physiologically correct initial conditions are impossible to obtain, we produce the initial conditions by running the simulation with zero velocity and pressure for a few time steps, and the resulting initial conditions are quite reasonable. $\Gamma = \partial\Omega$ is the boundary of the computational domain, and we label Γ_I, Γ_W and Γ_O^i as the boundaries at the inlet, the wall and the i^{th} outlet, respectively. The boundary conditions have a great influence on the computational results. In this work, we impose the following boundary conditions on the surfaces of the computational domain:

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{v}_I(t), \quad \text{on} \quad \Gamma_I\times(0,T], \tag{2}$$

$$\boldsymbol{u}(\boldsymbol{x},t) = 0, \quad \text{on} \quad \Gamma_W\times(0,T], \tag{3}$$

$$p_i(t) = \Big(p_i(0) - R_i^1 Q_i(0)\Big)e^{-t/\tau_i} + R_i^1 Q_i(t) + \int_0^t \frac{e^{-(t-s)/\tau_i}}{C_i} Q_i(s)ds$$

$$= \Big(p_i(0) - R_i^1 \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x},0)\cdot \boldsymbol{n}_O^i d\Gamma\Big)e^{-t/\tau_i} + R_i^1 \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x},t)\cdot \boldsymbol{n}_O^i d\Gamma$$

$$+ \int_0^t \Big(\frac{e^{-(t-s)/\tau_i}}{C_i} \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x},s)\cdot \boldsymbol{n}_O^i d\Gamma\Big)ds, \quad \text{where} \quad \tau_i = R_i^2 C_i,$$

$$\text{on} \quad \Gamma_O^i \times (0,T], \; (i = 1,2,\cdots,M), \tag{4}$$

where M is the total number of outlets. Specifically, (2) is a Dirichlet boundary condition on the inlet Γ_I with $\boldsymbol{v}_I(t)$ being a plug flow velocity [27] and can be calculated by $\boldsymbol{v}_I(t) = Q(t)\cdot \boldsymbol{n}_I/S$, where $Q(t)$ is a clinically measured flow rate at the inlet, S is the inlet area and $\boldsymbol{n}_I$ is the inward unit normal vector to the inlet surface. (3) is a no-slip boundary condition on the wall Γ_W. To account for the effect of the truncated downstream vasculature on the region of interests, we use (4) as the boundary condition for each outlet Γ_O^i, which relates the pressure $p_i(t)$ with the flow rate $Q_i(t)$ at the i^{th} outlet surface, where $Q_i(t) = \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x},t)\cdot \boldsymbol{n}_O^i d\Gamma$ with $\boldsymbol{n}_O^i$ being the outward unit normal vector at the i^{th} outlet surface. (4) is derived from the three-element Windkessel model shown in Fig. 2f, where a proximal resistance R_i^1 is in series with a component consisting of a capacitance C_i and a distal resistance R_i^2 [13,28]. In this Windkessel model, we assume the distal pressure is 0.

2.4. Parameters in outlet boundary conditions

At the beginning of the simulation, a total resistance R_T and a total compliance C_T are determined by the patient-specific systolic and diastolic pressures at the inlet [29]. After obtaining R_T and C_T, we compute the values of the resistance R_i and the compliance C_i of the i^{th} $(i = 1,2,\cdots,M)$ outlet from R_T and C_T. For an artery with multiple outlets, the way to split R_T greatly impacts the blood flow distribution to different parts of the body. Therefore, we introduce the concept of a region, which is defined as a specific organ (including the heart, the liver, the stomach, the spleen, the intestine, the kidneys and the suprarenal glands) or an anatomical territory (including the head, the arms and the legs). A region receives the blood from one or more arterial vessels. For example, the liver is regarded as a region supplied by the hepatic artery, as shown in Fig. 2g. To achieve correct blood flow distributions to specific regions, we define the blood flow fraction F_{region}^k as follows,

$$F_{region}^k = \frac{\overline{Q}_{region}^k}{\overline{Q}_T}, \tag{5}$$

where $\overline{Q}_T$ and $\overline{Q}_{region}^k$ are the mean flow rates of the inlet and the k^{th} region over one cardiac cycle respectively. [23] showed that the flow rate through a vessel is approximately inverse to the terminal resistance. In this work, we generalize this idea by introducing the regional resistance R_{region}^k and compliance C_{region}^k and adopt a two-level method for the split of R_T and C_T as follows:

In the first level, we split R_T and C_T for the k^{th} regional resistance R_{region}^k and compliance C_{region}^k according to the given blood flow fraction F_{region}^k,

$$R_{region}^k = \frac{R_T}{F_{region}^k}, \tag{6}$$

$$C_{region}^k = C_T F_{region}^k. \tag{7}$$

In the second level, we distribute the computed R_{region}^k and C_{region}^k into each outlet for R_i and C_i within this region according to the effective radius of the branch outlet surface,

$$R_i = R_{region}^k (r_i^3)^{-1} \sum_{l=1}^{m_k} r_l^3, \tag{8}$$

$$C_i = C_{region}^k r_i^3 \big(\sum_{l=1}^{m_k} r_l^3\big)^{-1}, \tag{9}$$

where m_k is the total number of outlets in the k^{th} region, and r_i and r_l are the effective radii of the i^{th} and l^{th} branch outlet surfaces within this region. After obtaining R_i, R_i^1 and R_i^2 (marked in Fig. 2) of the i^{th} outlet are calculated according to $R_i = R_i^1 + R_i^2$ and $R_i^1 = 66R_i^2/1200$ [17].

2.5. A fully implicit finite element discretization and a parallel domain decomposition solver

We define the trial and weighting function spaces as follows,

$$\mathcal{V} = \{\boldsymbol{u} \in [H^1(\Omega)]^3 : \boldsymbol{u}|_{\Gamma_I} = \boldsymbol{v}_I(t) \text{ and } \boldsymbol{u}|_{\Gamma_W} = 0\}, \tag{10}$$

$$\mathcal{V}_0 = \{\boldsymbol{u} \in [H^1(\Omega)]^3 : \boldsymbol{u}|_{\partial\Omega} = 0\}, \tag{11}$$

$$\mathcal{P} = \{p \in L^2(\Omega)\}. \tag{12}$$

Then the weak form of (1) reads: find $\boldsymbol{u} \in \mathcal{V}$ and $p \in \mathcal{P}$, such that for $\forall\, \boldsymbol{\phi} \in \mathcal{V}_0$ and $\psi \in \mathcal{P}$,

$$\mathcal{A}\Big(\{\boldsymbol{u}, p\}, \{\boldsymbol{\phi}, \psi\}\Big) = 0, \tag{13}$$

with

$$\begin{aligned}\mathcal{A}\Big(\{\boldsymbol{u}, p\}, \{\boldsymbol{\phi}, \psi\}\Big) = {} & \rho\Big(\int_\Omega \frac{\partial \boldsymbol{u}}{\partial t} \cdot \boldsymbol{\phi} d\Omega + \int_\Omega (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{\phi} d\Omega\Big) - \int_\Omega p \nabla \cdot \boldsymbol{\phi} d\Omega \\ & + 2\mu \int_\Omega \boldsymbol{\epsilon}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\boldsymbol{\phi}) d\Omega + \int_\Omega (\nabla \cdot \boldsymbol{u})\psi d\Omega \\ & - \int_{\Gamma_O^i} \Big(-p\boldsymbol{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u})\Big) \cdot \boldsymbol{n}_O^i \cdot \boldsymbol{\phi} d\Gamma.\end{aligned} \tag{14}$$

Due to the use of the Windkessel boundary condition on the outlets, the last term in (14) can be written as

$$\begin{aligned}& \int_{\Gamma_O^i} \Big(-p\boldsymbol{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u})\Big) \cdot \boldsymbol{n}_O^i \cdot \boldsymbol{\phi} d\Gamma \\ & = \sum_{i=1}^{M} \int_{\Gamma_O^i} \Bigg\{ -\Bigg[\Big(p_i(0) - R_i^1 \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x}, 0) \cdot \boldsymbol{n}_O^i d\Gamma\Big) e^{-t/\tau_i} + R_i^1 \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x}, t) \cdot \boldsymbol{n}_O^i d\Gamma \\ & + \int_0^t \Big(\frac{e^{-(t-s)/\tau_i}}{C_i} \int_{\Gamma_O^i} \boldsymbol{u}(\boldsymbol{x}, s) \cdot \boldsymbol{n}_O^i d\Gamma\Big) ds \Bigg]\boldsymbol{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u}) \Bigg\} \cdot \boldsymbol{n}_O^i \cdot \boldsymbol{\phi} d\Gamma.\end{aligned} \tag{15}$$

To discretize (13) in the space, we cover the computational domain Ω with an unstructure tetrahedral mesh $\Omega^h = \{\mathcal{K}\}$ and define the P_1-P_1 finite element spaces $\mathcal{V}^h$, $\mathcal{V}_0^h$ and $\mathcal{P}^h$ corresponding to their infinite dimensional subspaces. Then the semi-discretization of (13) reads: find $\boldsymbol{u}^h \in \mathcal{V}^h$ and $p^h \in \mathcal{P}^h$, such that for $\forall\, \boldsymbol{\phi}^h \in \mathcal{V}_0^h$ and $\psi^h \in \mathcal{P}^h$,

$$\mathcal{B}\Big(\{\boldsymbol{u}^h, p^h\}, \{\boldsymbol{\phi}^h, \psi^h\}\Big) = 0, \tag{16}$$

with

$$\begin{aligned}\mathcal{B}\Big(\{\boldsymbol{u}^h, p^h\}, \{\boldsymbol{\phi}^h, \psi^h\}\Big) = {} & \mathcal{A}\Big(\{\boldsymbol{u}^h, p^h\}, \{\boldsymbol{\phi}^h, \psi^h\}\Big) \underline{- \sum_{\mathcal{K} \in \Omega^h} \Big(\nabla \cdot \boldsymbol{u}^h, \tau_c \nabla \cdot \boldsymbol{\phi}^h\Big)_{\mathcal{K}}} \\ & \underline{- \sum_{\mathcal{K} \in \Omega^h} \Big(\frac{\partial \boldsymbol{u}^h}{\partial t} + \Big(\boldsymbol{u}^h \cdot \nabla\Big)\boldsymbol{u}^h + \nabla p^h, \tau_m(\boldsymbol{u}^h \cdot \nabla\boldsymbol{\phi}^h + \nabla\psi^h)\Big)_{\mathcal{K}}},\end{aligned} \tag{17}$$

where the underlined terms are the stabilization terms with parameters τ_c and τ_m given in reference [30].

The spatial discretization leads to a semi-discretized system that can be rewritten as

$$\frac{d\mathcal{X}(t)}{dt} = \mathcal{S}\Big(\mathcal{X}(t)\Big), \tag{18}$$

where $\mathcal{X}(t) = (\boldsymbol{u}^h, p^h)$ represents the time-dependent solution vector of the pressure and velocity at the nodal points; and $\mathcal{S}$ is a nonlinear function of $\mathcal{X}(t)$. For the temporal discretization of (18), we adopt the following fully implicit backward Euler method

$$\frac{\mathcal{X}^n - \mathcal{X}^{n-1}}{\Delta t} = \mathcal{S}(\mathcal{X}^n), \tag{19}$$

where $\mathcal{X}^n = \mathcal{X}(t_n)$ represents the pressure and velocity at the n^{th} time step and Δt is the time-step size. This fully implicit scheme is unconditionally stable and allows the use of larger time-step sizes than an explicit one, but is more computationally expensive due to the requirement of solving the following large and nonlinear system at each time step

$$\mathcal{F}(\mathcal{X}^n) = 0. \tag{20}$$

The nonlinear function is mostly sparse, except the nodes on the outlets because of the integral nature of the Windkessel boundary condition. To solve (20), we adopt a parallel NKS method, which consists of an inexact Newton method for solving the nonlinear problem and a preconditioned Krylov subspace method for solving the linear Jacobian problem within each Newton iteration [31,32]. Specifically, NKS performs the following steps to find the solution at each time step:

Step 1: Set the convergence tolerances ξ and η for the linear and nonlinear solvers respectively;
Step 2: Use the solution of the previous time step as the initial guess $\mathcal{X}_0^n = \mathcal{X}^{n-1}$;
Step 3: For each Newton step $k = 1, 2, \cdots$,

- Compute the Jacobian matrix analytically $J_k^n = \nabla\mathcal{F}(\mathcal{X}_k^n)$ and construct the restricted additive Schwarz (RAS) preconditioner $(M_k^n)^{-1}$;
- Find the inexact Newton direction $\delta\mathcal{X}_k^n$ by approximately solving the following preconditioned Jacobian system

$$\| J_k^n (M_k^n)^{-1} M_k^n \delta\mathcal{X}_k^n + \mathcal{F}(\mathcal{X}_k^n)\| \le \xi \|\mathcal{F}(\mathcal{X}_k^n)\|; \tag{21}$$

- Find the step length τ_k^n via a cubic linesearch method and update the solution by $\mathcal{X}_{k+1}^n = \mathcal{X}_k^n + \tau_k^n \delta\mathcal{X}_k^n$;
- Store $\mathcal{X}_{k+1}^n$ as the solution of the current time step if $\|\mathcal{F}(\mathcal{X}_k^n)\|$ or $\frac{\|\mathcal{F}(\mathcal{X}_{k+1}^n)\|}{\|\mathcal{F}(\mathcal{X}_0^n)\|} < \eta$ and go to **Step 2** with $n = n+1$.

Here, we provide some details of the computation of the Jacobian matrix in Appendix A. Moreover, we omit the scripts k and n for notational simplicity and rewrite the RAS preconditioner $(M_k^n)^{-1}$ as M^{-1} with its definition given as follows,

$$M^{-1} = \sum_{l=1}^{np} (R_l^0)^T B_l^{-1} R_l^{\delta}, \tag{22}$$

where np is the number of subdomains (also equals to the number of processor cores used for the computation), B_l is the subdomian Jacobian matrix and R_l^0 and R_l^{δ} are the restriction operators defined on the non-overlapping and overlapping subdomains.

3. Results and discussions

In this section, we first validate our numerical method by comparing its results with that obtained from a popular software package. Then, we simulate the hemodynamics of a full-body artery with a focus on the regional flow distribution. Besides, the robustness and the parallel scalability of the proposed algorithm are investigated. The algorithm is implemented on the Portable Extensible Toolkit for Scientific computing (PETSc) library [33]. All simulations are carried out on a supercomputer, whose compute node consists of two 12-core CPUs and 64 GB shared memory. If it is not specifically mentioned, we claim that the nonlinear algorithm converges when either the relative or absolute residual is smaller than $\eta = 1 \times 10^{-6}$. For the Jacobian solver we use a relative residual of $\xi = 1 \times 10^{-3}$. The blood is characterized with a density of $\rho = 1.06$ g/cm^3 and a dynamic viscosity of $\mu = 3.5 \times 10^{-2}$ g/(cm $\cdot$ s).

3.1. Blood flow in a tube with one bifurcation

In this subsection, we verify the correctness of the proposed numerical method by comparing it with the results obtained with the software package SimVascular [34] for the simulation of a transient blood flow in a tube with one bifurcation. As shown in Fig. 4a, the configuration consists of a parent tube of diameter 4 cm with its left end serving as the inlet, and its right end connecting to a hemisphere of radius 2.0 cm. Two daughter branches intersect the hemisphere with “Outlet

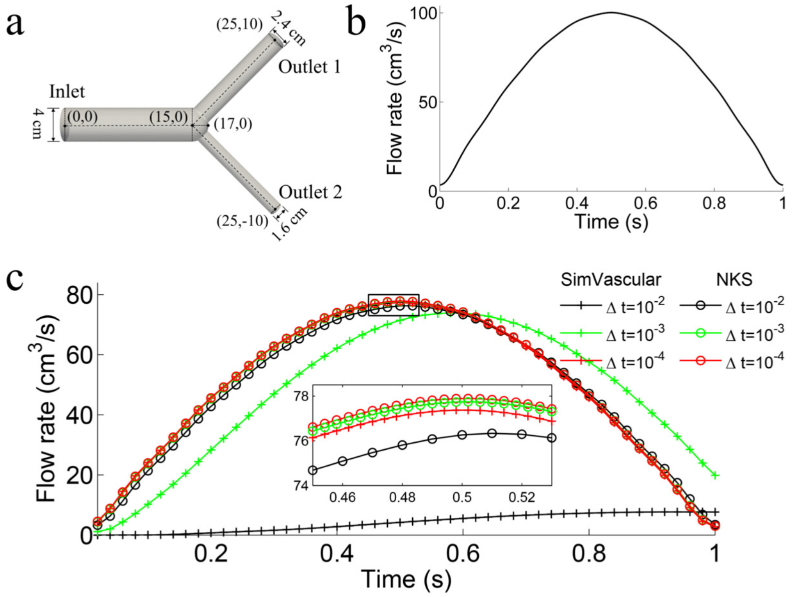

Fig. 4. (a) the geometry of a benchmark problem; (b) the pulsatile flow rate imposed on the inlet boundary and (c) the comparison of the flow rates at outlet 1 computed with SimVascular and NKS by using different time-step sizes $\Delta t = 1\times\ 10^{-2},\ 10^{-3}$ and 10^{-4} s.

1" of diameter 2.4 cm and "Outlet 2" of diameter 1.6 cm. The coordinates in Fig. 4a provide the length of the tubes. As shown in Fig. 4b, a pulsatile volumetric flow rate is prescribed to the inlet with an axisymmetric parabolic profile. In the computation, the flow rate is obtained by a 10-term Fourier series approximation of $100\sin(\pi t/T)$, where $t \in [0, T]$ with a total time $T = 1$ s. For SimVascular, the simulation is performed by using a mesh with 505, 381 linear tetrahedral elements on a computer with 16 GB RAM. The residual convergence criteria for the linear and nonlinear solvers are both 1×10^{-3}. For NKS, the simulation is carried out on a mesh with 506, 642 elements on a parallel computer with 120 processor cores. The Windkessel boundary condition is imposed on each outlet with the values of the proximal resistance, the distal resistance and the capacitance being 8.99×10^1 dyn · s/cm^5, 1.64×10^3 dyn · s/cm^5 and 1.16×10^{-5} cm^5/dyn at outlet 1 and 3.06×10^2 dyn · s/cm^5, 5.56×10^3 dyn · s/cm^5 and 3.41×10^{-6} cm^5/dyn at outlet 2.

For SimVascular, the choice of the time-step size Δt is restricted by the CFL number [13]. Let h be the mesh size and v the velocity, then according to $\Delta t = h/\|v\|$, a time step-size of 2×10^{-3} s is needed such that the CFL number is around 1 for the present problem, since the minimum h is about 0.06 cm and the maximum speed is about 30 cm/s. Therefore, as shown in Fig. 4c, a time-step size of 1×10^{-2} s fails to give a convergent flow rate for outlet 1. A decrease of the time-step size to 1×10^{-3} s increases the accuracy but is still not accurate enough and a further reduction to 1×10^{-4} s yields a convergent solution. Because of a fully implicit temporal scheme, NKS is not restricted by the CFL condition and a time-step size of 1×10^{-2} s is enough to give a convergent solution. It can be seen that the convergent solutions from both methods are consistent of each other. Note that the ratio of the mean flow rate $\overline{Q}_1/\overline{Q}_2 = 49.23/14.49 = 3.40$ nearly equals to the ratio of the resistance $R_2/R_1 = 5.86 \times 10^3/1.73 \times 10^3 = 3.39$, where $\overline{Q}_1$ and $\overline{Q}_2$ are the mean flow rates through outlets 1 and 2 for a cardiac cycle, respectively, and R_1 and R_2 are the values of the outlet resistances assigned to outlets 1 and 2, respectively. Overall, the proposed numerical method is validated by its agreement with SimVascular.

3.2. Blood flow in a full-body artery

In this subsection, we discuss the simulation of blood flow in the patient-specific-full-body artery given in Fig. 1. For the inlet boundary condition, we prescribe a representative flow rate profile given in Fig. 5, which is extracted from [35] and scaled to an average cardiac output of a resting person 5 L/min with the cardiac cycle $T = 1.0$ s. Hemodynamic analysis is carried out at a characteristic time $t = 0.12$ s during the peak systole, as marked by a circle in Fig. 5. The values of the total resistance and compliance are chosen as $R_T = 1.57 \times 10^3$ dyn · s/cm^5 and $C_T = 8.49 \times 10^{-4}$ cm^5/dyn so that the computed systolic and diastolic pressures are within physiologically normal ranges. The initial pressure of the outlet $p_i(0)$ in the Windkessel model is set to be 76 mmHg.

3.2.1. Local mesh refinement

Before analyzing the simulated results, it is necessary to conduct a mesh independence study to evaluate the accuracy of the solution for such a large and complex geometry. As shown in Fig. 7, local mesh refinements are performed at four different regions: A (the ascending aorta), B (the hepatic artery), C (the right ulnar artery) and D (the left posterior tibial artery). Note that position B is located at the right hepatic artery before a bifurcation, as shown in a close-up view in Fig. 8n.

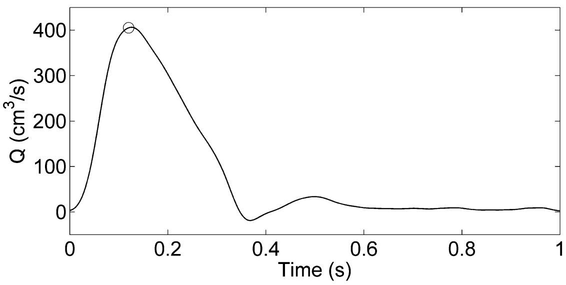

Fig. 5. The flow rate profile for a cardiac cycle imposed at the inlet with the peak systole $t = 1.2$ s marked by a circle.

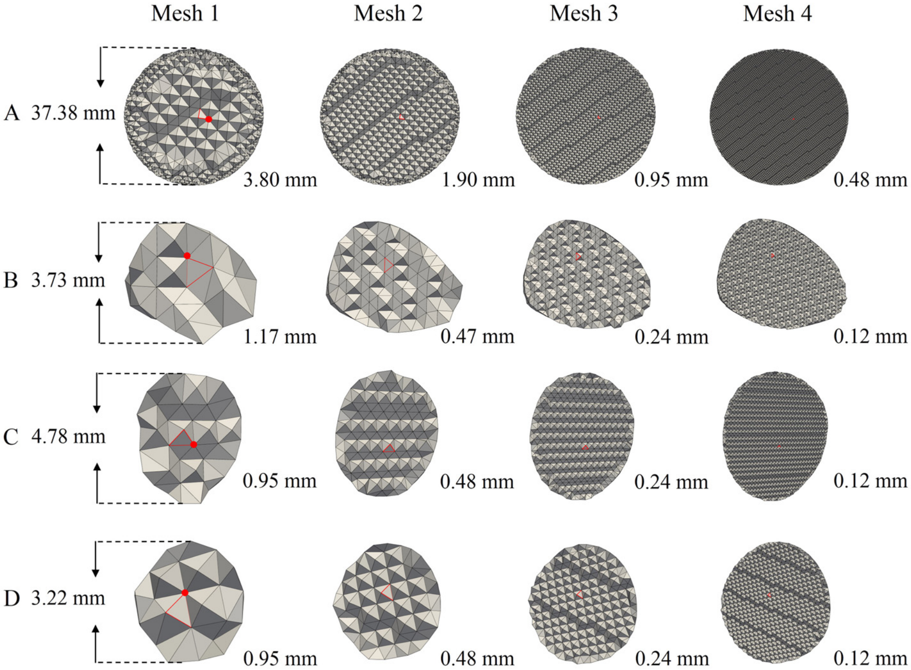

Fig. 6. Slices of four meshes at four different locations where local mesh refinements are performed: A (the ascending aorta), B (the hepatic artery), C (the right ulnar artery) and D (the left posterior tibial artery). The monitoring nodal points are labeled by red dots in Mesh 1. The maximum edge sizes of the highlighted elements are listed to show the refinement of the mesh.

The refinement results in four meshes with Mesh 1 with 10,072,134, Mesh 2 with 10,250,197, Mesh 3 with 11,924,493 and Mesh 4 with 25,640,395 tetrahedral elements. Fig. 6 shows some slices of the meshes at the four locally refined regions. Point-wise solutions of the pressure and velocity are monitored at the nodal points marked by red dots in Mesh 1. We see that the refinement leads to a decrease about a half of the maximum edge size of the highlighted element near these nodal points. For example, the maximum edge sizes are 3.80 mm in Mesh 1, 1.90 mm in Mesh 2, 0.95 mm in Mesh 3 and 0.48 mm in Mesh 4 in region A. Comparing with the other three interested regions, the refinement in region A begins with a larger element size 3.80 mm due to its large vessel size 37.38 mm. It is noted that, for Mesh 1, the geometrical shape of the vessel is changed at regions B, C and D due to the use of coarse meshes, but is well maintained in region A where the surface mesh is controlled to be smaller than the volume mesh. All simulations are carried out by using 720 processor cores with a time-step size 1 ms.

Fig. 7 shows the influence of the mesh refinement on the temporal changes of the pressure and the velocity at the four monitoring nodal points within the locally refined regions. For the pressure, we see that even the coarsest Mesh 1 can yield converged results for all the monitoring points comparing with the results of the finest Mesh 4. To see it more clearly, pressures during a systolic period are shown in the enlarged images. We quantify the difference by calculating the maximum error and obtain values of 0.03 mmHg, 0.56 mmHg, 0.06 mmHg and 0.22 mmHg for the monitoring points in regions A, B, C and D respectively, which are quite small comparing to the value of the systolic pressure. However, for the velocity, the influence of the mesh size is quite significant. We find that Mesh 1 can provide general trends but not accurate enough values for the velocity. For example, it produces a maximum relative error 64.70% for the velocity at the monitoring point in region B and fails to characterize the curvature of the velocity profile during the diastolic period at the

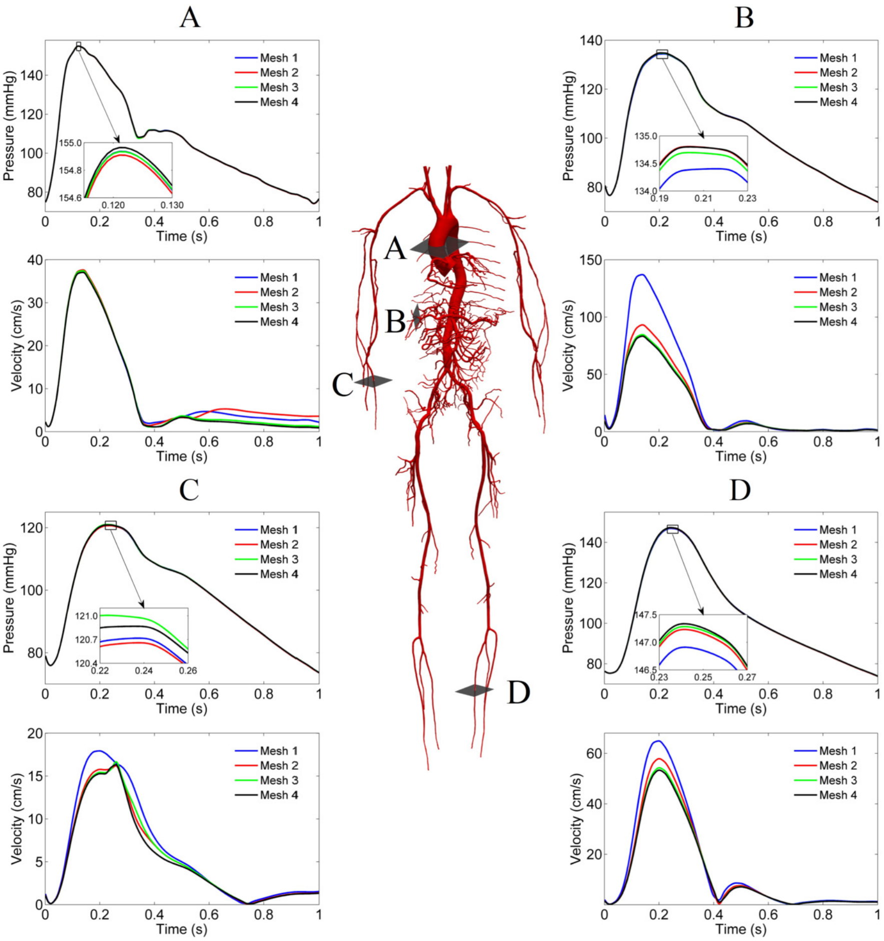

Fig. 7. The influence of the mesh refinement on the temporal changes of the pressure and the velocity at the four monitoring nodal points within the locally refined regions: A (the ascending aorta), B (the hepatic artery), C (the right ulnar artery) and D (the left posterior tibial artery).

monitoring point in region C. By reducing the mesh size, Mesh 2 improves the accuracy of the velocity, but not as accurate as Mesh 3. A further refinement of the mesh to Mesh 4 can not improve much on the accuracy of the velocity. Therefore, Mesh 3 is accurate enough to provide physiological realistic waveforms for the pressure and the velocity, as shown by the experimental data in [36]. Overall, the pressure is not, but the velocity is, sensitive to the mesh size. To resolve the locally interested flow field, we need of a mesh size about 1 mm for large arteries with a diameter of 30 − 40 mm and 0.2 mm for small arteries with a diameter of 3 − 5 mm.

3.2.2. Distribution of regional blood flows

To understand the two-level method for the model parameters selection, we estimate the blood flow distributions to 14 regions based on the results of Mesh 4 to reduce the influence of the mesh size. As shown in Fig. 8, branch vessels that deliver the blood to the same region are grouped together, including 7 to the head, 21 to the heart, 16 to the liver, 7 to the stomach, 9 to the spleen, 49 to the intestine, 4 to each of the left and right kidneys, 4 to each of the left and right suprarenal glands, 7 to the left and 5 to the right arms, and 26 to the left and 31 to the right legs. Other 28 branch vessels that do not belong to any interested region are classified into one group. To quantify the flow distribution to specific regions, we calculate the blood flow fraction F^k_{region} with its definition given in (5). The following relative error is used to evaluate the discrepancy between the computed and measured results

$$e = \frac{|F^k_{region}(\text{measured}) - F^k_{region}(\text{computed})|}{F^k_{region}(\text{measured})}, \tag{23}$$

where $F^k_{region}(\text{computed})$ is the computed blood flow fraction and $F^k_{region}(\text{measured})$ is the corresponding measured data reported in [17,22,37]. As listed in Table 1, regional resistances R^k_{region} and compliances C^k_{region} are calculated based on the

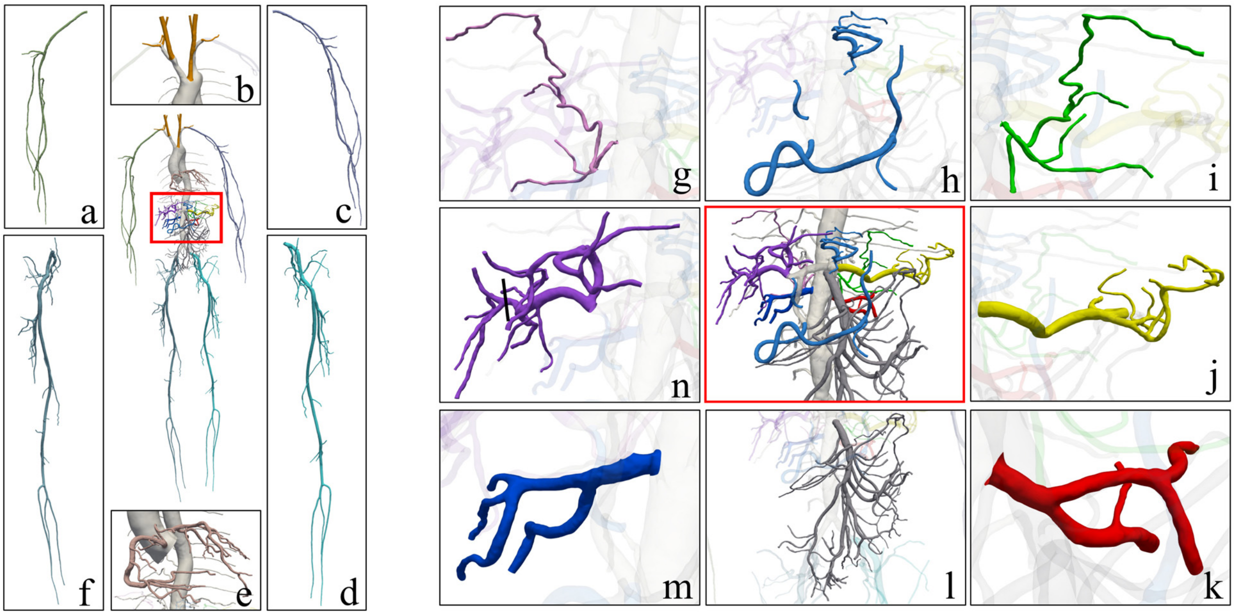

Fig. 8. The close-up views of the arteries that supply the blood to different regions of the body, including (a) the right arm, (b) the head, (c) the left arm, (d) the left leg, (e) the heart, (f) the right leg, (g) the right suprarenal gland, (h) the stomach, (i) the left suprarenal gland, (j) the spleen, (k) the left kidney, (l) the intestine, (m) the right kidney and (n) the liver. The enlarged view in the red box shows the arteries that branch from the abdominal aorta.

Table 1
Comparison between the computed regional blood flow fractions and the reported data from the literature. "No. vessel" is the number of branch vessels supplying the target regions. R^k_{region} and C^k_{region} are the regional resistance and compliance respectively. F^k_{region} is the blood flow fraction and e is the relative error.

Regions	No. vessels	Measured		R^k_{region}	C^k_{region}	Computed	
		F^k_{region}	References	$(\text{dyn}\cdot\text{s/cm}^5)$	(cm^5/dyn)	F^k_{region}	e
Head	7	18.30%	[17]	8.55E+03	1.55E-04	19.24%	4.75%
Heart	21	4.00%	[22,37]	3.91E+04	3.40E-05	4.29%	7.25%
Liver	16	6.50%	[22,37]	2.41E+04	5.52E-05	6.35%	2.31%
Stomach	7	1.00%	[22,37]	1.57E+05	8.49E-06	1.02%	2.00%
Spleen	9	3.00%	[22,37]	5.22E+04	2.55E-05	3.05%	1.67%
Intestine	49	13.25%	[22,37]	1.18E+04	1.13E-04	13.34%	0.68%
Left kidney	4	9.50%	[22,37]	1.65E+04	8.07E-05	9.39%	1.16%
Right kidney	4	9.50%	[22,37]	1.65E+04	8.07E-05	8.96%	5.68%
Left adrenal gland	4	0.15%	[22,37]	1.04E+06	1.27E-06	0.15%	0.00%
Right adrenal gland	4	0.15%	[22,37]	1.04E+06	1.27E-06	0.16%	6.67%
Left arm	7	5.60%	[17]	2.79E+04	4.76E-05	5.39%	3.75%
Right arm	5	5.60%	[17]	2.79E+04	4.76E-05	5.23%	6.61%
Left leg	26	11.20%	[17]	1.40E+04	9.51E-05	11.27%	0.63%
Right leg	31	11.20%	[17]	1.40E+04	9.51E-05	11.41%	1.88%
Others	28	–	–	1.49E+05	8.92E-06	1.11%	–
Total	222	–	–	1.57E+03	8.49E-04	100.36%	–

measured blood flow fraction F^k_{region}(measured). By introducing the regional resistances and compliances, the computed blood flow fraction matches quite well with the measured values, with maximum and minimum relative errors 7.25% in the heart and 0.00% in the left adrenal gland. The accuracy of the solution can also be reflected by the total output 100.36%, which is quite close to 100.00% for the satisfaction of the mass conservation. We remark that the error of the blood flow fraction can be reduced by using finer meshes. Thus, we show that our two-level method for determining the terminal parameters is useful in obtaining target regional blood flow distributions.

3.2.3. Distribution of hemodynamic parameters

Fig. 9 presents the spatial distributions of the pressure and velocity at the peak systole $t = 0.12$ s in the whole body and the four cross sections shown in Fig. 7. In the whole body, the calculated systolic pressure reaches 155.70 mmHg at the aortic root and drops gradually from proximal to distal, which is similar to the results in [16]. The velocity and streamline distributions show generally lower values in the ascending aorta and higher values in the descending aorta. This trend is

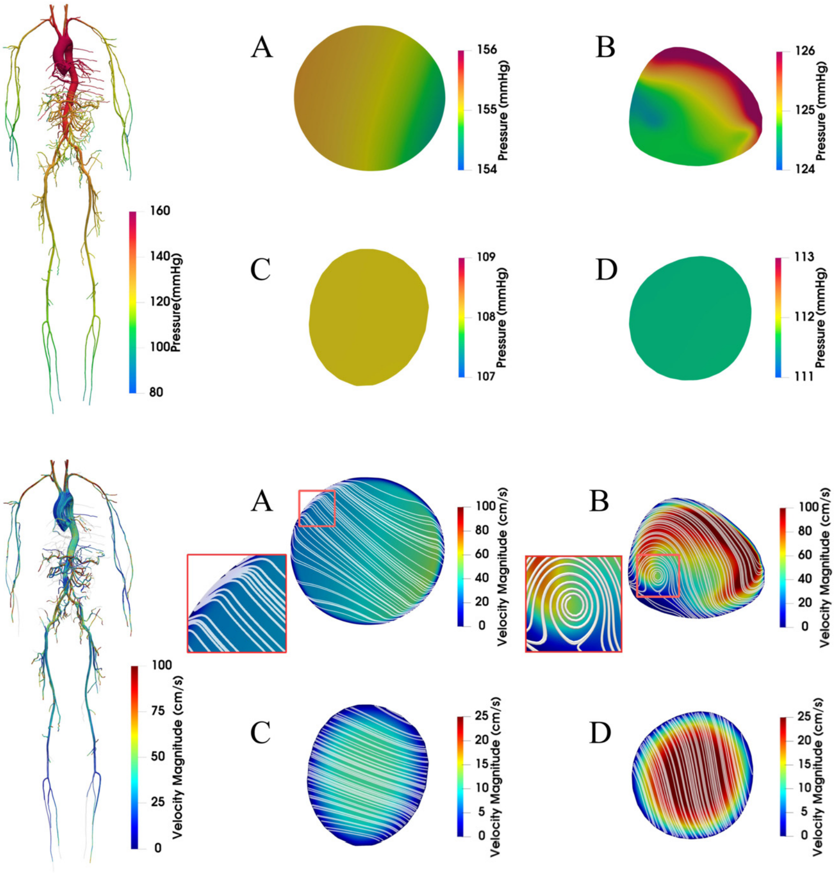

Fig. 9. The spatial distributions of the pressure and velocity at the peak systole $t = 0.12$ s in the whole body and the four cross sections in Fig. 7.

similar to the results in [17] but different in the region of the abdominal aorta because of a slight aneurysm enlargement in our case. In the cross section, pressures are displayed in a color-encoded form with a scale of 2 mmHg. We see clear pressure gradients in the cross sections A and B but not in C and D, since the ascending aorta bends in A and the right hepatic artery bifurcates in B, while arteries in C and D are relatively straight without apparent curvatures. The profile of the velocity is closely related to the pressure gradient. Therefore, skewed flows are observed in A and B, where pressure gradients exist in both the radial and axial directions. In contrast, the flow is relatively symmetric in C and D, where it predominantly drives by the axial pressure gradient. To study the local flow pattern, we generate the 2D streamlines by projecting the velocity vector onto the 2D plane. As shown in the enlarged view of the cross section A, curved streamlines are observed in the vicinity of the aortic wall, which are helical flows developed due to the curvature of the ascending aorta, as has been shown experimentally by the 4D flow MRI in [12,38]. In the enlarged view of the cross section B, vortical flows are developed due to the flow separation induced by the bifurcation of the artery geometry, as also reported by the numerical study in [39].

Based on the computed pressure and velocity, one can derive many clinically important hemodynamic parameters, such as the wall shear stress (WSS). The definition of WSS is described as the difference between the overall shear stress and its normal projection [30],

$$\mathrm{WSS} = \boldsymbol{\sigma}\boldsymbol{n}_p - (\boldsymbol{\sigma}\boldsymbol{n}_p \cdot \boldsymbol{n}_p)\boldsymbol{n}_p, \tag{24}$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{n}_p$ is the normal vector at a nodal point on the wall. Fig. 10 presents the spatial distributions of WSS at the peak systole $t = 0.12$ s for the whole body and the local disease-prone regions, including the right coronary artery and the abdominal aorta. Comparing to the results in [16,17], our results generally show a similar trend of the WSS distribution in the arterial network, with relative low values in the thoracic aorta and high values in part of the abdominal aortic branches, for example, the intestine. WSS has been associated with many different vascular diseases. In the coronary artery, low WSS patches appear in regions of arterial bifurcations and bends, as marked in the boxes of Fig. 10, where plaques are usually developed [40]. In the abdominal aorta, low WSS patches emerge in the lumen wall of the aneurysm and has been extensively associated with the thrombus deposition and the aneurysmal rupture [41]. Note

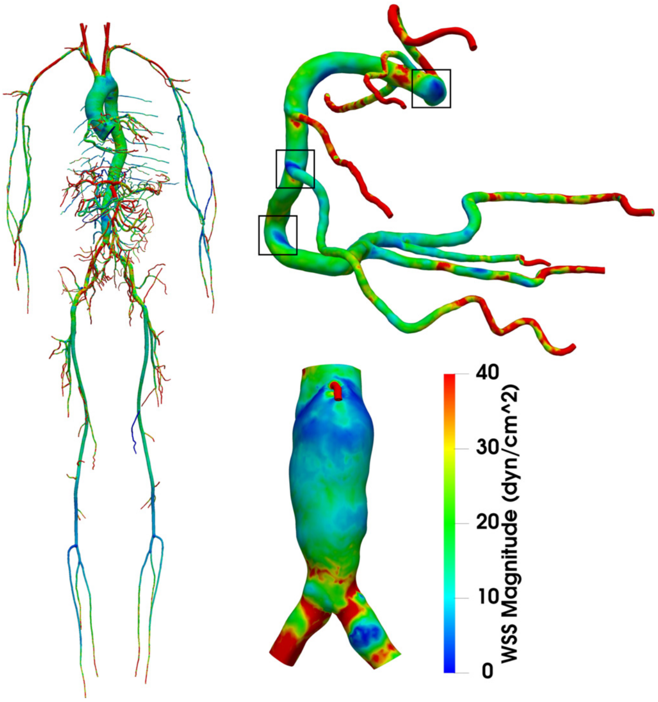

Fig. 10. The spatial distributions of the wall shear stress at the peak systole $t = 0.12$ s for the whole body and the close-up views of the right coronary artery and the abdominal aorta.

that the distribution of the hemodynamic parameters changes due to the pulsatile nature of the blood flow and becomes more complex in the period of diastole, as shown in our previous work [42].

3.3. Robustness and scalability

In this subsection, we investigate the robustness and parallel scalability of the proposed algorithm for the simulation of blood flows in the full-body artery shown in Fig. 1. In the following tables, “Newton” is the average number of nonlinear iterations per time step, “GMRES” is the average number of linear iterations in each Newton step, “Time” is the average compute time spent on each time step. All results are collected based on the first 20 time steps. To measure the parallel performance of our algorithm, we use the speedup and the parallel efficiency defined as follows,

$$\text{Speedup} = \frac{t_{N_0}}{t_N}, \tag{25}$$

$$\text{Efficiency} = \frac{t_{N_0} \times N_0}{t_N \times N}, \tag{26}$$

where t_{N_0} and t_N are the compute times using N_0 and N $(N_0 \le N)$ processor cores respectively. N_0 is the minimum number of processor cores that can solve the problem.

Table 2 shows the robustness of the proposed algorithm with respect to several important parameters, including the time-step size Δt, the viscosity μ, the total resistance R_T and the total capacitance C_T. The influence of the parameters are tested by changing one parameter each time while keeping others fixed with default values $\Delta t = 1 \times 10^{-3}$ s, $\mu = 4 \times 10^{-2}$ g/cm $\cdot$ s, $R_T = 1.57 \times 10^3$ dyn $\cdot$ s/cm^5 and $C_T = 8.49 \times 10^{-4}$ cm^5/dyn. All tests are carried out on a mesh with 1.45×10^7 elements using 960 processor cores. Firstly, we study the impact of the time-step size on the performance of our algorithm. Four time-step sizes 2×10^{-4}, 5×10^{-4}, 1×10^{-3} and 2×10^{-3} s are considered. We see that due to the use of a fully implicit scheme in time, our algorithm converges and is stable for all the tested time-step sizes, which are much larger than 5×10^{-5} s used in [17]. However, as the time-step size increases from 2×10^{-4} to 2×10^{-3} s, the numbers of Newton and GMRES iterations increase, which results in a longer compute time at each time step. This is mainly due to the fact that the initial guess of the Newton’s method becomes less accurate when the time-step size becomes larger. From the table, we see that the choice of a time-step size 1×10^{-3} is reasonable in terms of the total computing time.

Then, we investigate the impact of the viscosity μ on the algorithm. The viscosity is variant depending on the hemodynamic conditions and closely related to the Reynolds number by the relationship $Re \propto 1/\mu$ [43]. Therefore, with the decrease of the viscosity, the Reynolds number increases and the problem becomes more difficult to solve. This explains the slight increase of the numbers of Newton and GMRES iterations and the compute time as well, as the viscosity

Table 2
The robustness of the proposed algorithm with respect to the time-step size Δt, the viscosity μ, the total resistance R_T and the total capacitance C_T.

		Newton	GMRES	Time (s)
Δt (s)	2×10^{-4}	2.0	104.9	23.9
	5×10^{-4}	2.1	128.8	27.4
	1×10^{-3}	2.6	203.4	46.4
	2×10^{-3}	4.1	271.9	90.5
μ (g/cm·s)	1×10^{-2}	2.7	214.1	49.3
	4×10^{-2}	2.6	211.9	48.3
	7×10^{-2}	2.6	206.4	46.5
	1×10^{-1}	2.4	190.6	40.4
R_T $(\text{dyn} \cdot \text{s/cm}^5)$	7.85×10^{2}	2.6	195.7	43.6
	1.57×10^{3}	2.6	203.4	46.4
	3.14×10^{3}	2.7	252.7	55.1
	6.28×10^{3}	2.9	306.8	69.8
C_T (cm^5/dyn)	4.25×10^{-4}	2.6	199.4	44.6
	8.49×10^{-4}	2.6	203.4	46.4
	1.70×10^{-3}	2.6	215.6	47.4
	3.40×10^{-3}	2.6	218.4	49.0

Table 3
Parallel performance of the proposed algorithm tested on two meshes with 1.45×10^7 and 2.45×10^7 elements.

Mesh	*np*	Newton	GMRES	Time (s)	Speedup	Efficiency
1.45×10^7	480	2.6	201.4	86.5	1.0	100%
	960	2.6	203.4	46.4	1.9	93%
	1920	2.7	248.9	31.2	2.8	69%
	3840	2.6	400.7	27.6	3.1	39%
	7680	2.6	657.0	25.4	3.4	21%
2.45×10^7	720	2.9	306.5	151.0	1.0	100%
	1440	2.9	288.2	78.2	1.9	96%
	2880	2.9	374.1	54.0	2.8	70%
	5760	3.0	571.9	46.7	3.2	40%
	11520	3.0	687.3	37.2	4.1	25%
	15360	2.7	756.9	34.4	4.4	21%

decreases from 1×10^{-1} to 1×10^{-2} $\text{g/cm} \cdot \text{s}$. Overall, the convergence of our algorithm implies its robustness to the viscosity.

Finally, we explore the influence of the model parameters in the Windkessel boundary condition, namely the total resistance R_T and the total compliance C_T, whose values can be changing due to the alterations in the morphology and composition of the artery caused by the disease and age [44]. Results show that our algorithm is robust to R_T and C_T since the numbers of Newton iterations and GMRES iterations are almost stable for all the tests, especially for C_T.

Table 3 shows the parallel scalability tested on two meshes: a coarse mesh with 1.45×10^7 elements and a fine mesh with 2.45×10^7 elements. For both meshes, as the number of processor core increases, the number of Newton iterations stays almost as a constant. However, the number of GMRES iterations grows slowly. This is consistent with the classical convergence theory of the RAS perconditioner, which holds that the condition number increases with the increase of the number of subdomains [45]. Moreover, our results show a low parallel efficiency of 21% for the coarse mesh with 7,680 cores and the fine mesh with 15,360 cores. The main reason for the low efficiency is the heavy burden of the communication between processor cores, which can be improved by increasing the overall size of the problem. For the sake of clarity, we plot the compute time vs. the number of processor cores in Fig. 11, with the percentage showing the corresponding parallel efficiencies. As we can see, the compute time drops quickly at first and then slowly when the number of processor cores reaches 3,840 for the coarse mesh and 5,760 for the fine mesh, where both of the parallel efficiencies are about 40%. Overall, our algorithm can be scalable to 15,360 processor cores and still maintains about 20% parallel efficiency. With this highly scalable method, the average compute time per time step can be reduced to about 30 seconds, which means the blood flow simulation of a full-body artery from neck to feet can be as short as 8 hours per cardiac cycle.

4. Concluding remarks

In this work, we investigate a parallel domain decomposition method to solve the unsteady incompressible Navier-Stokes equations for the simulation of blood flows in a full-body arterial network with 222 outlet branches. For this complex arterial network, it is a mandatory requirement to consider the regional blood flow distribution. Therefore, we introduce a

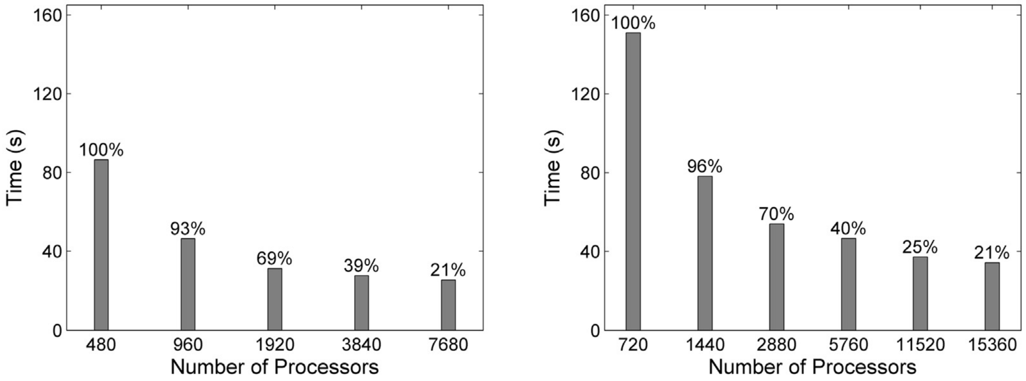

Fig. 11. The average compute time vs. the number of processor cores for the two meshes of 1.45×10^7 (left) and 2.45×10^7 (right) elements. The percentages are the corresponding parallel efficiencies.

two-level method to determine the model parameters in the Windkessel outlet boundary condition, which guarantees the correct amount of blood supply to 14 regions. We validate the proposed numerical method by comparing its results with that obtained from the software package SimVascular on a benchmark problem and show our superiority in choosing large time-step sizes due to the use of a fully implicit time scheme. Moreover, local mesh refinements are performed and found to be an effective way to study the influence of the mesh size on the accuracy of the solution and results indicate that the pressure is not, but the velocity is, sensitive to the mesh size. Numerical experiments show that the proposed method is scalable with up to 15,360 processor cores and robust with respect to different model parameters, which makes it possible for a comprehensive study of a vascular disease under the system of a whole-body arterial network.

CRediT authorship contribution statement

Shanlin Qin: Methodology, Validation, Writing. Rongliang Chen: Software, Methodology, Writing. Bokai Wu: Visualization. Xiao-Chuan Cai: Conceptualization, Supervision.

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.

Data availability

The data that has been used is confidential.

Acknowledgements

This work was supported by the Shenzhen grants (Grant Nos. JCYJ20190806165805433 and RCYX20200714114735074) and the NSFC grants (Grant Nos. 62161160312 and 12071461).

Appendix A. Construction of the Jacobian

There are many ways to construct the Jacobian matrix. In this work, we construct the Jacobian analytically instead of approximately to enhance the efficiency and accuracy of the nonlinear solver. The Jacobian matrix of (20) can be obtained by taking the first order partial derivatives with respect to each of the unknowns. The details of the linear and nonlinear terms in the Jacobian matrix of (20) without the consideration of the Windkessel boundary condition can refer to [31,46]. Here, we only introduce what the Windkessel boundary condition, namely (15), brings to the Jacobian matrix.

First, we introduce some notations. We write the velocity vector of the n^{th} time step in the form $\boldsymbol{u}^n = (u_1^n, u_2^n, u_3^n)^T$ and the outward unit normal vector at the i^{th} outlet surface $\boldsymbol{n}_o^i = (n_1^i, n_2^i, n_3^i)^T$. We also approximate u_1^n, u_2^n and u_3^n in the finite-dimensional space as follows,

$$u_1^n = \sum_{k'=1}^{N_T} u_{1,k'}^n \varphi_{k'},\ u_2^n = \sum_{k'=1}^{N_T} u_{2,k'}^n \varphi_{k'} \text{ and } u_3^n = \sum_{k'=1}^{N_T} u_{3,k'}^n \varphi_{k'}, \tag{27}$$

where k' is the k'^{th} nodal points, $\varphi_{k'}$ denotes the corresponding nodal basis and N_T represents the total number of mesh nodes. Then, in the x direction, we obtain the following function from (15) that contributes to the Jacobian matrix of (20) due to the use of the Windkessel boundary condition,

$$\widetilde{\mathcal{F}}_{i'}^{(x)} = \sum_{i=1}^{M}\int_{\Gamma_O^i}\Bigg[\Big(2\mu\sum_{k'=1}^{N_T}u_{1,k'}^{n}\frac{\partial\varphi_{k'}}{\partial x}-\mathcal{C}-\mathcal{D}\Big)n_1^i+\mu\Big(\sum_{k'=1}^{N_T}u_{1,k'}^{n}\frac{\partial\varphi_{k'}}{\partial y}+\sum_{k'=1}^{N_T}u_{2,k'}^{n}\frac{\partial\varphi_{k'}}{\partial x}\Big)n_2^i$$
$$+\mu\Big(\sum_{k'=1}^{N_T}u_{1,k'}^{n}\frac{\partial\varphi_{k'}}{\partial z}+\sum_{k'=1}^{N_T}u_{3,k'}^{n}\frac{\partial\varphi_{k'}}{\partial x}\Big)n_3^i\Bigg]\varphi_{i'}d\Gamma, \tag{28}$$

where $i' = 1, 2, \cdots, N_T$ and we have

$$\mathcal{C} = R_i^1\int_{\Gamma_O^i}\sum_{q=1}^{3}\sum_{k'=1}^{N_T}u_{q,k'}^{n}\varphi_{k'}n_q^i d\Gamma, \tag{29}$$

$$\mathcal{D} = \Delta t\sum_{m=1}^{n}\frac{e^{-(t_n-t_m)/\tau_i}}{C_i}\int_{\Gamma_O^i}\sum_{q=1}^{3}\sum_{k'=1}^{N_T}u_{q,k'}^{m}\varphi_{k'}n_q^i d\Gamma. \tag{30}$$

Taking the derivatives of (28) with respect to $u_{1,j'}^n$, $u_{2,j'}^n$ and $u_{3,j'}^n$, we obtain

$$\frac{\partial\widetilde{\mathcal{F}}_{i'}^{(x)}}{\partial u_{1,j'}^n} = \sum_{i=1}^{M}\int_{\Gamma_O^i}\Bigg[\Big(2\mu\frac{\partial\varphi_{j'}}{\partial x}\varphi_{i'}-R_i^1\Phi_1\varphi_{i'}-\frac{\Delta t}{C_i}\Phi_1\varphi_{i'}\Big)n_1^i+\mu\frac{\partial\varphi_{j'}}{\partial y}\varphi_{i'}n_2^i+\mu\frac{\partial\varphi_{j'}}{\partial z}\varphi_{i'}n_3^i\Bigg]d\Gamma, \tag{31}$$

$$\frac{\partial\widetilde{\mathcal{F}}_{i'}^{(x)}}{\partial u_{2,j'}^n} = \sum_{i=1}^{M}\int_{\Gamma_O^i}\Bigg[\Big(-R_i^1\Phi_2\varphi_{i'}-\frac{\Delta t}{C_i}\Phi_2\varphi_{i'}\Big)n_1^i+\mu\frac{\partial\varphi_{j'}}{\partial x}\varphi_{i'}n_2^i\Bigg]d\Gamma, \tag{32}$$

$$\frac{\partial\widetilde{\mathcal{F}}_{i'}^{(x)}}{\partial u_{3,j'}^n} = \sum_{i=1}^{M}\int_{\Gamma_O^i}\Bigg[\Big(-R_i^1\Phi_3\varphi_{i'}-\frac{\Delta t}{C_i}\Phi_3\varphi_{i'}\Big)n_1^i+\mu\frac{\partial\varphi_{j'}}{\partial x}\varphi_{i'}n_3^i\Bigg]d\Gamma, \tag{33}$$

where $i', j' = 1, 2, \cdots, N_T$ and $\Phi_q = \int_{\Gamma_O^i}\varphi_{j'}n_q^i d\Gamma$ ($q = 1, 2$ and 3). We see that Φ_q involves the integration over the boundary, which makes the Jacobian system difficult to solve since it produces a dense block in the Jacobian matrix. The equivalents of (28) in the y and z directions can be used to compute the derivatives for the Jacobian similarly.

References

[1] C.A. Taylor, J. Humphrey, Open problems in computational vascular biomechanics: hemodynamics and arterial wall mechanics, Comput. Methods Appl. Mech. Eng. 198 (45–46) (2009) 3514–3523.

[2] A. Quarteroni, A. Manzoni, C. Vergara, The cardiovascular system: mathematical modelling, numerical algorithms and clinical applications, Acta Numer. 26 (2017) 365–590.

[3] C.W. Ong, I. Wee, N. Syn, S. Ng, H.L. Leo, A.M. Richards, A.M. Choong, Computational fluid dynamics modeling of hemodynamic parameters in the human diseased aorta: a systematic review, Ann. Vasc. Surg. 63 (2020) 336–381.

[4] L. Grinberg, T. Anor, J. Madsen, A. Yakhot, G. Karniadakis, Large-scale simulation of the human arterial tree, Clin. Exp. Pharmacol. Physiol. 36 (2) (2009) 194–205.

[5] F.E. Fossan, J. Mariscal-Harana, J. Alastruey, L.R. Hellevik, Optimization of topological complexity for one-dimensional arterial blood flow models, J. R. Soc. Interface 15 (149) (2018) 20180546.

[6] S.M. Shavik, Z. Jiang, S. Baek, L.C. Lee, High spatial resolution multi-organ finite element modeling of ventricular-arterial coupling, Front. Physiol. 9 (2018) 119.

[7] T. Bomberna, G.A. Koudehi, C. Claerebout, C. Verslype, G. Maleux, C. Debbaut, Transarterial drug delivery for liver cancer: numerical simulations and experimental validation of particle distribution in patient-specific livers, Expert. Opin. Drug. Deliv. 18 (3) (2021) 409–422.

[8] T. Du, D. Hu, D. Cai, Outflow boundary conditions for blood flow in arterial trees, PLoS ONE 10 (5) (2015) e0128597.

[9] J.P. Mynard, J.J. Smolich, One-dimensional haemodynamic modeling and wave dynamics in the entire adult circulation, Ann. Biomed. Eng. 43 (6) (2015) 1443–1460.

[10] P.J. Blanco, L.O. Müller, S.M. Watanabe, R.A. Feijóo, On the anatomical definition of arterial networks in blood flow simulations: comparison of detailed and simplified models, Biomech. Model. Mechanobiol. (2020) 1–16.

[11] S. Schnell, D.A. Smith, A.J. Barker, P. Entezari, A.R. Honarmand, M.L. Carr, S.C. Malaisrie, P.M. McCarthy, J. Collins, J.C. Carr, et al., Altered aortic shape in bicuspid aortic valve relatives influences blood flow patterns, Eur. J. Echocardiogr. 17 (11) (2016) 1239–1247.

[12] O.A. Jarral, M.K. Tan, M.Y. Salmasi, S. Pirola, J.R. Pepper, D.P. O'Regan, X.Y. Xu, T. Athanasiou, Phase-contrast magnetic resonance imaging and computational fluid dynamics assessment of thoracic aorta blood flow: a literature review, Eur. J. Cardio-Thorac. Surg. 57 (3) (2020) 438–446.

[13] 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 (29–32) (2006) 3776–3796.

[14] H. Lei, D.A. Fedosov, G.E. Karniadakis, Time-dependent and outflow boundary conditions for dissipative particle dynamics, J. Comput. Phys. 230 (10) (2011) 3765–3779.

[15] S. Dong, G.E. Karniadakis, C. Chryssostomidis, A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains, J. Comput. Phys. 261 (2014) 83–105.

[16] M. Zhou, O. Sahni, H.J. Kim, C.A. Figueroa, C.A. Taylor, M.S. Shephard, K.E. Jansen, Cardiovascular flow simulation at extreme scale, Comput. Mech. 46 (1) (2010) 71–82.

[17] N. Xiao, J.D. Humphrey, C.A. Figueroa, Multi-scale computational model of three-dimensional hemodynamics within a deformable full-body arterial network, J. Comput. Phys. 244 (2013) 22–40.

[18] A. Randles, E.W. Draeger, T. Oppelstrup, L. Krauss, J.A. Gunnels, Massively parallel models of the human circulatory system, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2015, pp. 1–11.

[19] J. Liu, W. Yang, M. Dong, A.L. Marsden, The nested block preconditioning technique for the incompressible Navier–Stokes equations with emphasis on hemodynamic simulations, Comput. Methods Appl. Mech. Eng. 367 (2020) 113122.

[20] S. Qin, R. Chen, B. Wu, W.-S. Shiu, X.-C. Cai, Numerical simulation of blood flows in patient-specific abdominal aorta with primary organs, Biomech. Model. Mechanobiol. 20 (2021) 909–924.

[21] P. Blanco, S. Watanabe, R. Feijóo, Identification of vascular territory resistances in one-dimensional hemodynamics simulations, J. Biomech. 45 (12) (2012) 2066–2073.

[22] P.J. Blanco, S.M. Watanabe, E.A. Dari, M.A.R. Passos, R.A. Feijóo, Blood flow distribution in an anatomically detailed arterial network model: criteria and algorithms, Biomech. Model. Mechanobiol. 13 (6) (2014) 1303–1330.

[23] L. Grinberg, G.E. Karniadakis, Outflow boundary conditions for arterial networks with multiple outlets, Ann. Biomed. Eng. 36 (9) (2008) 1496–1514.

[24] D.L. Pham, C. Xu, J.L. Prince, Current methods in medical image segmentation, Annu. Rev. Biomed. Eng. 2 (1) (2000) 315–337.

[25] R.W. Dudek, T.M. Louis, High-Yield Gross Anatomy, Lippincott Williams & Wilkins, 2013.

[26] G. Karypis, K. Schloegel, ParMETIS: Parallel graph partitioning and sparse matrix ordering library, Version 4.0, Tech. Rep., University of Minnesota, 2013.

[27] S. Madhavan, E.M.C. Kemmerling, The effect of inlet and outlet boundary conditions in image-based CFD modeling of aortic flow, Biomed. Eng. Online 17 (1) (2018) 1–20.

[28] I.E. Vignon-Clementel, C. Figueroa, K. Jansen, C. Taylor, Outflow boundary conditions for 3D simulations of non-periodic blood flow and pressure fields in deformable arteries, Comput. Methods Biomech. Biomed. Eng. 13 (5) (2010) 625–640.

[29] N. Xiao, J. Alastruey, C. Alberto Figueroa, A systematic comparison between 1-D and 3-D hemodynamics in compliant arterial models, Int. J. Numer. Methods Biomed. Eng. 30 (2) (2014) 204–231.

[30] R. Chen, B. Wu, Z. Cheng, W.-S. Shiu, J. Liu, L. Liu, Y. Wang, X. Wang, X.-C. Cai, A parallel non-nested two-level domain decomposition method for simulating blood flows in cerebral artery of stroke patient, Int. J. Numer. Methods Biomed. Eng. 36 (11) (2020) e3392.

[31] A.T. Barker, X.-C. Cai, Scalable parallel methods for monolithic coupling in fluid–structure interaction with application to blood flow modeling, J. Comput. Phys. 229 (3) (2010) 642–659.

[32] F. Kong, X.-C. Cai, A scalable nonlinear fluid–structure interaction solver based on a Schwarz preconditioner with isogeometric unstructured coarse spaces in 3D, J. Comput. Phys. 340 (2017) 498–518.

[33] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, et al., PETSc Users Manual, Tech. Rep., Argonne National Laboratory, 2022.

[34] A. Updegrove, N.M. Wilson, J. Merkow, H. Lan, A.L. Marsden, S.C. Shadden, SimVascular: an open source pipeline for cardiovascular simulation, Ann. Biomed. Eng. 45 (3) (2017) 525–541.

[35] J. Alastruey, N. Xiao, H. Fok, T. Schaeffter, C.A. Figueroa, On the impact of modelling assumptions in multi-scale, subject-specific models of aortic haemodynamics, J. R. Soc. Interface 13 (119) (2016) 20160073.

[36] I.T. Gabe, J.H. Gault, J. Ross Jr, D.T. Mason, C.J. Mills, J.P. Schillingford, E. Braunwald, Measurement of instantaneous blood flow velocity and pressure in conscious man with a catheter-tip velocity probe, Circulation 40 (5) (1969) 603–614.

[37] J. Valentin, Basic anatomical and physiological data for use in radiological protection: reference values, Ann. ICRP 32 (3–4) (2002) 1–277.

[38] A. Frydrychowicz, A. Berger, A.M. Del Rio, M.F. Russe, J. Bock, A. Harloff, M. Markl, Interdependencies of aortic arch secondary flow patterns, geometry, and age analysed by 4-dimensional phase contrast magnetic resonance imaging at 3 tesla, Eur. J. Radiol. 22 (5) (2012) 1122–1130.

[39] K.G. Lyras, J. Lee, Comparison of numerical implementations for modelling flow through arterial stenoses, Int. J. Mech. Sci. 211 (2021) 106780.

[40] F. Gijsen, Y. Katagiri, P. Barlis, C. Bourantas, C. Collet, U. Coskun, J. Daemen, J. Dijkstra, E. Edelman, P. Evans, et al., Expert recommendations on the assessment of wall shear stress in human coronary arteries: existing methodologies, technical considerations, and clinical applications, Eur. Heart J. 40 (41) (2019) 3421–3433.

[41] A.J. Boyd, D.C. Kuhn, R.J. Lozowy, G.P. Kulbisky, Low wall shear stress predominates at sites of abdominal aortic aneurysm rupture, J. Vasc. Surg. 63 (6) (2016) 1613–1619.

[42] S. Qin, B. Wu, J. Liu, W.-S. Shiu, Z. Yan, R. Chen, X.-C. Cai, Efficient parallel simulation of hemodynamics in patient-specific abdominal aorta with aneurysm, Comput. Biol. Med. 136 (2021) 104652.

[43] E. Nader, S. Skinner, M. Romana, R. Fort, N. Lemonne, N. Guillot, A. Gauthier, S. Antoine-Jonville, C. Renoux, M.-D. Hardy-Dessources, et al., Blood rheology: key parameters, impact on blood flow, role in sickle cell disease and effects of exercise, Front. Physiol. 10 (2019) 1329.

[44] S. Greenwald, Ageing of the conduit arteries, J. Pathol. 211 (2) (2007) 157–172.

[45] C. Yang, J. Cao, X.-C. Cai, A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere, SIAM J. Sci. Comput. 32 (1) (2010) 418–438.

[46] R. Chen, X.-C. Cai, Parallel one-shot Lagrange–Newton–Krylov–Schwarz algorithms for shape optimization of steady incompressible flows, SIAM J. Sci. Comput. 34 (5) (2012) 584–605.