Efficient finite element hyperelasticity solver for blood vessel simulations

Xinhong Wang1, Zhengzheng Yan2, Yi Jiang2 and Rongliang Chen2,3

Abstract
The blood vessels play a key role in the human circulatory system. As a tremendous amount of efforts have been devoted to develop mathematical models for investigating the elastic behaviors of human blood vessels, high performance numerical algorithms aiming at solving these models have attracted attention. In this work, we present an efficient finite element solver for an elastodynamic model which is commonly used for simulating soft tissues under external pressure loadings. In particular, the elastic material is assumed to satisfy the Saint–Venant–Kirchhoff law, the governing equation is spatially discretized by a finite element method, and a fully implicit backward difference method is used for the temporal discretization. The resulting nonlinear system is then solved by a Newton–Krylov–Schwarz method. It is the first time to apply the Newton–Krylov–Schwarz method to the Saint–Venant–Kirchhoff model for a patient-specific blood vessel. Numerical tests verify the efficiency of the proposed method and demonstrate its capability for bioengineering applications.

Keywords
Blood vessel, hyperelasticity, elastodynamics, finite element method, Newton–Krylov–Schwarz method

Introduction
In human circulatory system, the blood vessel network plays a key role that carries the blood from the heart to tissues over the entire body. Understanding the properties of blood vessels is essential to understand the health and disease of the vascular system, which in turn is necessary to patient evaluation and conduct therapeutic solutions in many clinical events, including surgery, angioplasty, tissue remodeling, and engineering. In particular, it has been recognized that the elastic motion of the vessel wall has significant effects on the hemodynamic quantities of interest. As such, the mechanical property of the blood vessel keeps being interested in many research areas. By this far, there have accumulated numerous studies on this topic, such as the stiffness, compliance, vessel diameter, and the internal and/or external pressures it may be subjected to.

During the past several decades, considerable efforts have been devoted to apply computational techniques to investigate the mechanical characteristics of blood vessels based on the continuum elasticity theory.1–4 On the other hand, as the underlying mathematical models have kept constantly evolving to more sophisticated levels and the problem size become unprecedentedly large, simulations on the vascular elasticity models have become very expansive tasks. Due to these reasons, developing high performance parallel numerical algorithms for blood vessel simulations has attracted more and more attention, see...
Quarteroni et al.,\textsuperscript{5} Takizawa et al.,\textsuperscript{6} Crosetto et al.,\textsuperscript{7} Balzani et al.,\textsuperscript{8} and the references therein. In this work, we present an efficient parallel numerical solver aiming at solving nonlinear partial differential equations raised in blood vessel deformational dynamics. More specifically, the mathematical model is built on the continuum hyperelasticity theory, and a nonlinear solver is proposed based on the finite element discretization, an implicit time stepping, and the Newton–Krylov–Schwarz method.\textsuperscript{9} In this design, at each time step, we find the solution by successively solving the linearized discretized system. To achieve that, a Krylov subspace method is invoked at each Newton step. For improving the convergence and ensuring the scalability of whole method, a restricted additive Schwarz preconditioner\textsuperscript{10} is used. Finally, an efficient scalability results show the potential of the proposed algorithm, which is capable of carrying out blood vessel simulations. In the numerical experiments, we illustrate a patient-specific blood vessel under external pressure loadings. These results not only demonstrate the efficiency and capability of our solver, but also provide a basic module for more sophisticated cardiac biomechanical applications.

The rest of the paper is organized as follows: the mathematical model and the discretization scheme are described in “Modeling” section. In “Methodology” section, we introduce the Newton–Krylov–Schwarz method for solving the induced nonlinear system. Finally, numerical experiments are demonstrated in “Numerical experiments” section for simulating a human blood vessel with patient-specific shape. The scalability results show the potential of the proposed method to efficiently solve large size problems with many processors.

**Modeling**

We consider a nonlinear elastodynamic system for simulating a blood vessel with possible large elastic deformations due to external pressure loadings, such as the impact of blood flow on the interior surface or a squeeze pressure on the exterior surface. To clarify, we denote $\Omega \subset \mathbb{R}^3$ as a bounded domain that holds a segment of a blood vessel in its relaxed shape (the reference configuration). The boundary, $\partial \Omega$, is closed, piecewise differentiable surface that consists of three disjoint parts: $\partial \Omega = \Gamma_d \cup \Gamma_i \cup \Gamma_e$, where the two ends of the vessel together are denoted by $\Gamma_d$, and $\Gamma_i$ and $\Gamma_e$ represent the interior and exterior surface of the vessel, respectively. Suppose, during a period of time $[0, T]$, the vessel is fixed on $\Gamma_d$, meanwhile being applied a body force $g$ over $\Omega$, and against some surface loadings $f$ on $\Gamma_i$ and/or $\Gamma_e$. Then, according to the finite elasticity theory, at any moment $t \in (0, T)$, the displacement $d(t, X)$ and velocity $v(t, X)$ are governed by the following system of partial differential equations

$$
\begin{align*}
\frac{\partial v}{\partial t} + \nabla \cdot P &= g & \text{in } \Omega \\
\frac{\partial d}{\partial t} &= v & \text{in } \Omega 
\end{align*}
$$

that subjects to the boundary conditions

$$
\begin{align*}
P_N &= f & \text{on } \Gamma_i \cup \Gamma_e, \\
d &= 0 & \text{on } \Gamma_d
\end{align*}
$$

where $X$ denotes a material point in $\Omega$, $\rho$ is the mass density of the elastic material in consideration, $\eta$ is a velocity damping parameter, and $N$ represents the outward unit normal vector on the surfaces $\Gamma_i \cup \Gamma_e$.

In equation (1), the first Piola–Kirchhoff stress tensor $P$ is defined by

$$
P = FS
$$

where $F = I + \nabla d$ is the deformation gradient tensor, $I$ is the second rank identity tensor, $\nabla d$ is the deformation gradient tensor, and $S$ is the second Piola–Kirchhoff stress tensor. When the vessel is assumed to be exhibiting as a compressible hyperelastic Saint–Venant Kirchhoff material, it has the following specific form

$$
S = \lambda \text{trace}(E)I + 2\mu E
$$

where $\lambda$ and $\mu$ are the Lame coefficients, and

$$
E = \frac{1}{2} (F^\top F - I)
$$

is the Green–Lagrangian strain tensor.

The Saint–Venant–Kirchhoff (SVK) model is usually regarded as one type of hyperelastic models,\textsuperscript{11} for which category, there exists a strain energy density functional $W$, and the second Piola–Kirchhoff stress tensor can be defined by

$$
S = \frac{\partial W}{\partial E}
$$

In particular, for the SVK model, its energy density functional is in the form of

$$
W = \frac{\lambda}{2} \text{trace}(E)^2 + \mu \text{trace}(E^2)
$$
Despite its simplicity, the SVK model provides a basic module for many other elastic models which are more sophisticated and realistic, due to the fact that it captures some most essential features of soft materials. So far, the SVK model has been widely explored in many real-world applications, especially in studying the biomechanical characteristics of human body soft tissues, such as brain surgery, liver, respiratory system, and so on.

**Methodology**

**Spatial discretization**

We discretize the partial differential system (1) spatially by using the standard finite element method. First, the variational form is found to be: find \((d, v) \in U\), such that

\[
\mathcal{F}((d, v), (\xi, \zeta)) = 0 \quad \forall \xi, \zeta \in V
\]

for

\[
\mathcal{F}((d, v), (\xi, \zeta)) = \int_{\Omega} \frac{\partial}{\partial t} \xi dX + \int_{\Omega} v \cdot \xi dX + \int_{\partial \Omega} P : \nabla \xi dS - \int_{\Omega} g \cdot \xi dX
\]

where \(U\) and \(V\) denote the corresponding trial and test function spaces, respectively. Note that both \(U\) and \(V\) can be considered time-independently by virtue of the Lagrangian coordinates; thus, all discussions can be confined in the reference configuration \(\Omega\).

Provided a \(N\) vertices tetrahedral mesh \(T_h\) built on \(\Omega\), a \(P_1\) element is applied to all component of the displacement \(d_i\) and the velocity \(v_i\). For the case that the test function \(\zeta_i\) is chosen to be \(\phi_r\), we have

\[
\mathcal{F}_i(\phi_r) = \sum_{i=1}^{N} \frac{\partial v_{i,j}}{\partial t} \int_{\Omega} \phi_r \phi_d dX
\]

On the other hand, when \(\zeta_i\) is selected to be \(\phi_r\), we have

\[
\mathcal{F}_i(\phi_r) = \sum_{i=1}^{N} \frac{\partial v_{i,j}}{\partial t} \int_{\Omega} \phi_r \phi_d dX
\]

Here, \(i \in \{1, 2, 3\}\) and \(r \in \{1, 2, \ldots, N\}\), \(d_{i,j}\) and \(v_{i,j}\) represent the nodal values of \(d_i\) and \(v_i\) at \(l\)th node, respectively, for \(l = 1, 2, \ldots, N\). Usually, the system of nonlinear semi-discretized equations (10) and (11) is called differential algebraic equations, due to the fact that it has not yet been discretized for the differential terms with respect to time \(t\).

**Time discretization**

To get a full discretization, we apply a backward differentiable formula to the time derivative terms in equations (10) and (11). Here, without loss of generality, we show the first-order case, i.e. the backward Euler method: by denoting \(d_{i,j}^{(n)} \equiv d_i(t_n)\) and \(v_{i,j}^{(n)} \equiv v_i(t_n)\), we approximate

\[
\frac{\partial d_{i,j}}{\partial t} \approx \frac{d_{i,j}^{(n)} - d_{i,j}^{(n-1)}}{\Delta t}
\]

and

\[
\frac{\partial v_{i,j}}{\partial t} \approx \frac{v_{i,j}^{(n)} - v_{i,j}^{(n-1)}}{\Delta t}
\]

Thus, the nonlinear finite element equations (10) and (11) become

\[
\mathcal{F}_i(\phi_r) = \sum_{i=1}^{N} \left( \int_{\Omega} \phi_r \phi_d dX + \Delta t \int_{\partial \Omega} P_{k,l} \frac{\partial \phi_r}{\partial \chi_k} dS - \Delta t \int_{\partial \Omega} g \phi_r dX \right)
\]

and

\[
\mathcal{F}_i(\phi_r) = \sum_{i=1}^{N} \left( \int_{\Omega} \phi_r \phi_d dX - \Delta t \int_{\partial \Omega} f \phi_r dS \right)
\]
where \( i = 1, 2, 3 \) and \( r = 1, 2, \ldots, N \). Note that we have eliminated \( \Delta t \) from the denominators due to the fact that we are solving the equation \( \mathcal{F} = 0 \).

Provided that all information when \( t \leq t_{n-1} \) are given, equations (12) and (13) together define a system of nonlinear algebraic equations that can be written by

\[
F\left( \left\{ d^{(n,k)}_{ij}, v^{(n,k)}_{ij} \right\} \right) = 0
\]

(14)

Here, we emphasize that the unknowns are the coefficients vectors: \( \{d^{(n)}_{ij}\} \) and \( \{v^{(n)}_{ij}\} \) for \( j = 1, 2, 3 \) and \( l = 1, 2, \ldots, N \).

A Newton–Krylov–Schwarz solver

At each time step \( t_n \), we use the Newton’s method to deal with the nonlinear algebraic equation system (12) and (13). That is, we start with an initial guess \( \{d^{(0)}_{ij}, v^{(0)}_{ij}\} \) and update them by the following scheme till convergent:

- Find \( \{u_{ij}\} \) and \( \{w_{ij}\} \), such that

\[
\mathcal{J}\left( \left\{ d^{(n,k)}_{ij}, v^{(n,k)}_{ij} \right\} \right) \left\{ u_{ij}, w_{ij} \right\} = -F\left( \left\{ d^{(n,k)}_{ij}, v^{(n,k)}_{ij} \right\} \right)
\]

(15)

- Update

\[
d^{(n+1)}_{ij} = d^{(n,k)}_{ij} + au_{ij},
\]

\[
v^{(n+1)}_{ij} = v^{(n,k)}_{ij} + aw_{ij}
\]

(16)

where \( \mathcal{J} \) denotes the Jacobian matrix of \( F \). For the sake of simplicity, in the rest of this paper, we denote the combination of the unknowns \( \{u_{ij}\} \) and \( \{w_{ij}\} \) as \( x \) and refer to the linear system (3.8) in the following matrix form

\[
\mathcal{J}x = -F
\]

(17)

The Krylov subspace methods are a family of iterative methods for solving algebraic linear systems, and they are highly desired for large number of unknowns with sparse coefficient matrices, such as the ones induced from partial differential equation systems, and play an essential role in modern supercomputings for solving versatile engineering problems.17–21 In this paper, we choose the GMRES method as the linear solver inside each Newton iteration. This method constructs a sequence of orthogonal vectors in a successively expanded Krylov spaces by minimizing the equation’s residual, such that they will eventually approach to the solution. A detailed introduction and justification of the GMRES method can be found in Saad.22 In practice, applying a Krylov subspace method, such as GMRES, to solve a PDE problem often suffers a slow convergence and instability. This is mainly because the coefficient matrix of the induced linear system is ill-conditioned, which gets even worse when the problem size becomes larger. A common way to overcome this difficulty is to adopt a so-called “preconditioning.” For example, when a right preconditioner \( B^{-1} \) is applied to \( \mathcal{J}x = -F \), it means to solve an equivalent two-stage problem in the form of

\[
\begin{align*}
\{ (\mathcal{J}B^{-1})x' \} &= -F, \\
Bx &= x'
\end{align*}
\]

(18)

where \( B^{-1} \) denotes a matrix that is expected be a good approximation of the inverse of \( \mathcal{J} \). In fact, \( B^{-1} \) is often not explicitly formed; instead, it is carefully designed; such that the matrix–vector product \( B^{-1}F \) can be obtained by performing a linear solve

\[
Bx = y
\]

(19)

for vector \( y \). The rationale behind this technique is that by paying a cheap cost to solve an additional problem, the total number of Krylov iterations will be significantly reduced, such that the efficiency of whole algorithm gets improved.

Specifically, a Schwarz preconditioner \( B^{-1} \) is constructed as a linear map

\[
B^{-1} : \mathbb{R}^N \rightarrow \mathbb{R}^N
\]

(20)

based on a domain decomposition strategy: the original whole mesh \( T \) is decomposed into \( n_p \) non-overlapped subdomains: \( T = \bigcup_{j=1}^{n_p} T_j \), and each submesh \( T_j \) is extended to intersect with its neighbors by an amount of overlapping \( \delta \). Several linear maps are then created for data scattering and gathering between each submesh and the whole domain: two restriction operators \( D_j^\delta \) and \( D^\delta_j \) map a global vector \( x \) on \( T \) to local vectors \( x_j^\delta \) and \( x^\delta_j \) that are restricted on \( T_j \), respectively

\[
D_j^\delta : \mathbb{R}^N \rightarrow \mathbb{R}^{N_j^\delta} \quad \text{and} \quad D^\delta_j : \mathbb{R}^N \rightarrow \mathbb{R}^{N_j^\delta},
\]

Correspondingly, two interpolation operators

\[
(D_j^\delta)^T : \mathbb{R}^{N_j^\delta} \rightarrow \mathbb{R}^N \quad \text{and} \quad (D^\delta_j)^T : \mathbb{R}^{N_j^\delta} \rightarrow \mathbb{R}^N,
\]
are defined as the transposes for extracting data from each related subdomain. Here, \( N \) and \( N^j \) denote the number of vertices of \( T \) and \( T^j \), respectively.

In practice, the specific forms of these matrices may vary, depending on which domain decomposition strategy is adopted. In this paper, we choose the restricted additive Schwarz method, in which case \( D^j \) can be written into a rectangular form created by removing rows from the identity matrix \( I \). In addition, \( D^j \) is identical to \( D^j \) on a row that corresponds to a vertex in \( T_j \), or a zero row otherwise.

Based on the above settings, \( B^{-1} \) can be constructed via

\[
B^{-1} = \sum_{j=1}^{n_p} (D^j)^T \mathcal{J}_j^{-1} D^j
\]

Here

\[
\mathcal{J}_j = D^j \mathcal{J} (D^j)^T \quad j = 1, 2, \ldots, n_p
\]

denotes a sub-matrix obtained by restricting \( \mathcal{J} \) to \( T^j \), and \( \mathcal{J}_j^{-1} \) is regarded as an (approximate) solve for the local problem

\[
\mathcal{J}_j \mathbf{x}_j = -F_j
\]

where \( F_j \) is obtained by restricting \( F \) onto \( T^j \) and \( \mathbf{x}_j \) denotes the local solution vector.

The whole algorithm, a sequence of Newton iterations embedded with Krylov subspace linear solvers each of which is accelerated by a Schwarz preconditioner, is usually referred to the Newton–Krylov–Schwarz (NKS) method.

**Numerical experiments**

In this section, we present some numerical results for investigating deformations of a blood vessel that undergoes external loadings. Figure 1(a) shows an unstructured finite element mesh of a segment of a patient-specific carotid artery vessel used in our tests, which is composed of 19,848 vertices and 75,454 cells and corresponds to 114,003 degree of freedoms. The main body of our program package is implemented based on the well-known scientific computing toolkit PETSc, and the mesh partitioning is performed by using ParMetis for parallel computing. In our tests, we use the time step size \( \Delta t = 0.01 \text{s} \), and the Newton iteration terminates when the relative residual rtol \( \leq 1 \times 10^{-6} \) in each time step. For the linear GMRES solver, the stopping criterion is set to be rtol \( = 1.0 \times 10^{-4} \), the subdomain solver for the RAS preconditioner is the incomplete-LU (ILU) factorization with fill-in level 1 and overlapping size \( \delta = 1 \). For all tests, the computing time is measured in seconds, and all parameters are set according to Table 1 if not mentioned otherwise.

**Boundary settings**

In our simulations, the two ends of the vessel are fixed, on which a homogeneous Dirichlet boundary condition is applied. To deform the vessel, we either apply a dilatation force on the interior surface or squeeze it on the exterior surface, respectively. This force is dynamically loaded to ensure that its current direction is always normal to the surface no matter how the shape is changed. More specifically, in equation (9), we set

\[
f(t, \mathbf{X}) = \alpha_f(t)J(X)\mathbf{F}^{-T}(X)\mathbf{N}(X) \quad X \in \Gamma_i \cup \Gamma_e
\]

where \( \mathbf{N} \) is the outward unit normal direction of the vessel’s surface, \( \mathbf{F} \) is the deformation gradient tensor, and \( J = \det(\mathbf{F}) \). \( \alpha_f \) represents a time-dependent force magnitude that is set by

\[
\alpha_f(t) = \begin{cases} 
\frac{A}{2} \sin \left( \frac{\pi}{2} - \frac{t_{\text{max}} - t}{t_{\text{max}} - t_0} \right) & \text{if } t_0 < t < t_{\text{max}}, \\
\frac{A}{2} \sin \left( \frac{\pi}{2} + \frac{t_{\text{max}} - t}{t_{\text{end}} - t_0} \right) & \text{if } t_{\text{max}} < t < t_{\text{end}}, \\
\frac{A}{2} & \text{otherwise}.
\end{cases}
\]

**Figure 1.** (a) The unstructured tetrahedral mesh on the blood vessel used in our finite element simulations (19,848 vertices, 75,454 cells); (b) the time-variant pattern of the magnitude of the surface loading \( \alpha_f \).
where $A$ denotes a constant amplitude, $t_0$, $t_{\text{max}}$, and $t_{\text{end}}$ represent the time when the applied force starts, hits the maximal magnitude, and ends, respectively. A diagram of the surface loading magnitude change with respect to the time $t$ can be found in Figure 1(b).

**Incompressibility and damping**

It is usually regarded that many soft materials, including human and animal tissues, are incompressible. For taking such an effect into consideration, we add an extra term in the strain energy functional equation (7)

$$W_{\text{vol}} = \frac{\kappa}{2} (\ln J)^2$$

(24)

where $\kappa$ denotes a penalty parameter, and the larger its value, the stronger the incompressibility of the material behaves. This extra energy brings an extra term in equation (4) in the form of

$$\kappa \ln J F^{-1} F^{-T}$$

(25)

Similar to the dynamic loading term, it also induces an extra contribution into the Jacobian of the Newton’s equation.

Another practical setting is about the damping parameter $\eta$. In physics, damping is understood as an effect to prevent oscillatory motions of a dynamic system. Numerically, based on our experiences, introducing the damping term with a properly selected $\eta$ usually improves the stability of the Newton solver.

In the following, we show the numerical results obtained from these two sets of experiment.

**Vessel dilatation**

In this test, a loading is applied over the whole interior surface (except for the parts close to two ends) to expand the vessel. We first tested five cases with different choices of Young’s modulus $K$ ($K = 1 \times 10^4$, $1.5 \times 10^4$, $2 \times 10^4$, $2.5 \times 10^4$, $3 \times 10^4$, respectively). The deformed vessel geometry for each case is collected in Figure 2(a) to (e). A comparison of these five cases is also given in Figure 2(f), in which all five deformed vessels are cut through the mid plane. In another set of tests, we chose four different values for Poisson ratio $\nu$. This experiment is carried out by using four CPU processors, and the results are presented in Figure 3.

Visualization of the numerical results clearly shows that as the Young’s modulus $K$ gets larger, the smaller deformation of the vessel can be expanded. On the other hand, when the Poisson ratio $\nu$ changes, the results keep almost the same, and the most significant difference that can be observed is the thickness of the deformed vessel boundary layer. The observations agree with the common understanding of these model parameters.

**Vessel squeeze**

We repeated our tests for applying a squeezing force on the exterior surface and collected the results in Figure 4. In these tests, the force is applied on a top and bottom areas near the center of the vessel, which is dynamically loaded to ensure the normality to the exterior surface. Similarly, we also tested different Poisson ratios $\nu$, in which case, the Young’s modulus $K$ is fixed at $1 \times 10^4$ Pa. This experiment is carried out by using four CPU processors, and the results are presented in Figure 5.

**Solver performance**

We present some performances of our finite element solver for the vessel dilatation and squeeze simulations. In the following tables, we denote Newton as the average Newton iteration number per time step and

| Meaning (unit)                  | Value                  |
|--------------------------------|------------------------|
| $t_0$ Starting time to apply the surface loading (s) | 0                      |
| $t_{\text{max}}$ Time when the surface loading hits its maximum (s) | 0.3                    |
| $t_{\text{end}}$ Time when the surface loading decreases to 0 (s) | 0.6                    |
| $\rho$ Material density (g/mm$^3$) | $1.0 \times 10^{-3}$   |
| $A$ Amplitude of the surface loading (Pa) | $3.0 \times 10^3$ (dilatation) $2.0 \times 10^2$ (squeeze) |
| $\kappa$ Incompressible penalty parameter (Pa/mm) | $1.0 \times 10^3$  |
| $K$ Young’s modulus (Pa) | $2.0 \times 10^3$ (dilatation) $1.0 \times 10^3$ (squeeze) |
| $\nu$ Poisson ratio | 0.2                    |
| $\eta$ Damping parameter (g/m$^2$/s) | $5.0 \times 10^4$ (dilatation) $5.0 \times 10^5$ (squeeze) |

| Model parameters and their standard values used in the numerical tests. | Value |
|---------------------------------------------------------------|-------|
| $t_0$ Starting time to apply the surface loading (s) | 0     |
| $t_{\text{max}}$ Time when the surface loading hits its maximum (s) | 0.3   |
| $t_{\text{end}}$ Time when the surface loading decreases to 0 (s) | 0.6   |
| $\rho$ Material density (g/mm$^3$) | $1.0 \times 10^{-3}$ |
| $A$ Amplitude of the surface loading (Pa) | $3.0 \times 10^3$ (dilatation) $2.0 \times 10^2$ (squeeze) |
| $\kappa$ Incompressible penalty parameter (Pa/mm) | $1.0 \times 10^3$ |
| $K$ Young’s modulus (Pa) | $2.0 \times 10^3$ (dilatation) $1.0 \times 10^3$ (squeeze) |
| $\nu$ Poisson ratio | 0.2 |
| $\eta$ Damping parameter (g/m$^2$/s) | $5.0 \times 10^4$ (dilatation) $5.0 \times 10^5$ (squeeze) |
GMRES as the average GMRES iteration number per Newton step. The total computing time (denoted as Time in the tables) for each case is counted based on the first 30 time steps.

We first test the model parameters: Young’s modulus $K$ and Poisson ratio $\nu$, respectively. As the results are shown in Table 2, the numerical solver is able to complete all simulations very efficiently and robustly. For each case, both the nonlinear and the linear solver terminates with a small number of iterations.

Based on the same mesh, we also studied the impact on the solver performance brought by the fill-in level of the ILU subdomain solver. In particular, we repeat the dilatation and the squeeze simulations with the ILU fill-in level to be 0, 1, 2, and 3, respectively. The problem is solved by using four processors, and the results are collected in Table 3. It can be seen that the higher
ILU fill-in level can reduce the GMRES iteration numbers, but may cost more total computational time. According to the results by these tests, to set the fill-in level 1 may be the best choice for most of the cases.

In addition, we demonstrate the strong scalability of our numerical solver. The mesh used here has the same geometry as the previous case, with 129,298 vertices and 604,383 cells, and the computing time for each

Figure 4. (a)–(e) Vessel squeezing for different choices of Young’s modulus; (f) the central slice cut show for different Young’s modulus cases (Unit: Pascal): (a) $K = 1 \times 10^4$; (b) $K = 1.5 \times 10^4$; (c) $K = 2 \times 10^4$; (d) $K = 2.5 \times 10^4$; (e) $K = 3 \times 10^4$; (f) a cross-section view.

Figure 5. (a) The slice cuts across the middle plane of the vessel in pressure for different Poisson’s ratio $\nu$: 0.1 (blue), 0.2 (green), 0.3 (yellow), and 0.4 (red); (b) a close look on the top depressed area.
For two jobs, which take time of $T_1$ and $T_2$, by using $n_1$ and $n_2$ processors, respectively, we define

\[
\text{Speed-up} = \frac{T_1}{T_2} \quad \text{and} \quad \text{Efficiency} = \frac{n_1 T_1}{n_2 T_2} \times 100\%.
\]

We test on 16, 24, 32, 48, 64, and 96 CPU processors and compute the parallel scalability based on the 16-processor case. The results can be found in Table 4. It can be seen that our numerical solver has a very good efficiency, and the nonlinear and linear solver are stable in terms of the Newton and GMRES iteration numbers. This indicates that the proposed algorithm has a great potential for larger-sized elastodynamic problem with more processors. We plan to investigate this topic further in the future.

### Declaration of Conflicting Interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding
The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Shenzhen research projects: JCYJ20160331193229720, JCYJ20170307165328836, and JSGG20170824154458183 and NSFC funding 61531166003.

### ORCID iDs
Xinhong Wang  
https://orcid.org/0000-0002-0665-3382  
Yi Jiang  
https://orcid.org/0000-0001-9155-7805  
Rongliang Chen  
https://orcid.org/0000-0002-1718-2993
References

1. Holzapfel GA, Gasser TC and Stadler M. A structural model for the viscoelastic behavior of arterial walls: continuum formulation and finite element analysis. *Eur J Mech-A/Solid* 2002; 21: 441–463.

2. Alastruè V, Martinez M, Doblarè M, et al. Anisotropic micro-sphere-based finite elasticity applied to blood vessel modelling. *J Mech Phys Solid* 2009; 57: 178–203.

3. Zhou J and Fung Y. The degree of nonlinearity and anisotropy of blood vessel elasticity. *Proc Natl Acad Sci USA* 1997; 94: 14255–14260.

4. Vito RP and Dixon SA. Blood vessel constitutive models – 1995–2002. *Annu Rev Biomed Eng* 2003; 5: 413–439.

5. Quarteroni A, Veneziani A and Zunino P. Mathematical and numerical modeling of solute dynamics in blood flow and arterial walls. *SIAM J Numer Anal* 2002; 39: 1488–1511.

6. Takizawa K, Christopher J, Tezduyar TE, et al. Space-time finite element computation of arterial fluid–structure interactions with patient-specific data. *Int J Numer Meth Biomed Eng* 2010; 26: 101–116.

7. Crosetto P, Reymond P, Deparis S, et al. Fluid-structure interaction simulation of aortic blood flow. *Comput Fluid* 2011; 43: 46–57.

8. Balzani D, Deparis S, Fausten S, et al. Numerical modeling of fluid-structure interaction in arteries with anisotropic polyconvex hyperelastic and anisotropic viscoelastic material models at finite strains. *Int J Numer Meth Biomed Eng* 2016; 32: e02756.

9. Cai XC, Gropp WD, Keyes DE, et al. (1994) Newton-Krylov-Schwarz methods in CFD. In: Rannacher R, ed. *Numerical methods for the Navier-Stokes equations*. Berlin, Germany: Springer, pp.17–30.

10. Cai XC and Sarkis M. A restricted additive schwarz preconditioner for general sparse linear systems. *SIAM J Sci Comput* 1999; 21: 792–797.

11. Raoult A. Non-polygonicity of the stored energy function of a Saint Venant-Kirchhoff material. *Apl Matemat* 1986; 31: 417–419.

12. Sin FS, Schroeder D and Barbie J. Vega: non-linear fem deformable object simulator. *Computer Graphics Forum* 2013; 32: 36–48.

13. Romero I. An analysis of the stress formula for energy-momentu methods in nonlinear elastodynamics. *Comput Mech* 2012; 50: 603–610.

14. Echegaray G, Herrera I, Aguinaga I, et al. A brain surgery simulator. *IEEE Comput Graph Appl* 2014; 34: 12–18.

15. Delingette H and Ayache N. Soft tissue modeling for surgery simulation. *Handb Numer Anal* 2004; 12: 453–550.

16. Ladjal H, Azencot J, Beuve M, et al. Biomechanical modeling of the respiratory system: Human diaphragm and thorax. In: Doyle B, Miller K, Wittek A, et al. (eds) *Computational biomechanics for medicine*. Berlin, Germany: Springer, pp.101–115.

17. Kong F and Cai XC. 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* 2016; 38: C73–C95.

18. Yang H, Prudencio EE and Cai XC. Fully implicit Lagrange–Newton–Krylov–Schwarz algorithms for boundary control of unsteady incompressible flows. *Int J Numer Meth Eng* 2012; 91: 644–665.

19. Chen R and Cai XC. Parallel one-shot Lagrange–Newton–Krylov–Schwarz algorithms for shape optimization of steady incompressible flows. *SIAM J Sci Comput* 2012; 34: B584–B605.

20. Chen R, Yan Z, Zhao Y, et al. Simulating flows passing a wind turbine with a fully implicit domain decomposition method. In: Dickopf T, Gander M, Halpern L, Krause R, et al. (eds) *Domain Decomposition Methods in Science and Engineering XXII. Lecture Notes in Computational Science and Engineering*, vol 104. Berlin, Germany: Springer, 2016, pp.453–460.

21. Scacchi S. A multilevel hybrid Newton–Krylov–Schwarz method for the bidomain model of electrocardiology. *Comput Methods Appl Mech Eng* 2011; 200: 717–725.

22. Saad Y. *Iterative methods for sparse linear systems*. vol. 82. 2003. Philadelphia, PA: SIAM.