Electronic Research Archive

ERA, 30 (9): 3377–3396.
DOI: 10.3934/era.2022172
Received: 30 December 2021
Revised: 29 April 2022
Accepted: 16 June 2022
Published: 18 July 2022

http://www.aimspress.com/journal/ERA

Research article

A parallel domain decomposition algorithm for fluid-structure interaction simulations of the left ventricle with patient-specific shape

Yujia Chang[1,2], Yi Jiang[1,2,*] and Rongliang Chen[1,2,*]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong Province, China
[2] Shenzhen Key Laboratory for Exascale Engineering and Scientific Computing, Shenzhen, Guangdong Province, China

* **Correspondence:** Email: yi.jiang@siat.ac.cn, rl.chen@siat.ac.cn; Tel: +86-755-86382312; Fax: +86-755-86392312.

Abstract: In this paper, we propose a scalable parallel algorithm for simulating the cardiac fluid-structure interactions (FSI) of a patient-specific human left ventricle. It provides an efficient forward solver to deal with the induced sub-problems in solving an inverse problem that can be used to quantify the interested parameters. The FSI between the blood flow and the myocardium is described in an arbitrary Lagrangian-Eulerian (ALU) framework, in which the velocity and stress are assumed being continuous across the fluid-structure interface. The governing equations are discretized by using a finite element method and a fully implicit backward Eulerian formula, and the resulting algebraic system is solved by using a parallel Newton-Krylov-Schwarz algorithm. We numerically show that the algorithm is robust with respect to multiple model parameters and scales well up to 2300 processor cores. The ability of the proposed method to produce qualitatively true prediction is also demonstrated via comparing the simulation results with the clinic data.

Keywords: fluid-structure interaction; human heart; left ventricle; finite element method; parallel computing

1. Introduction

The Biomechanics properties of the human heart have been attracting interests in both clinical and research communities, as they can reveal the fundamental physiological status of the heart, and are useful to predict and diagnose its diseases, such as myocardial ischemia and infarction, atrial and

ventricular arrhythmias, systolic and diastolic dysfunctions, and so on [1,2]. Numerical simulation provides a practical way to study the cardiac mechanics [3,4], by which the kinetical and kinematical behaviors of the heart muscles and the enclosed blood are depicted via a system of partial differential equations. Within the framework, the interested mechanical features are summarized into corresponding model parameters, and can be quantified by solving an inverse problem. Following this path, a series of induced forward sub-problems are repeatedly solved, and the values of the interested parameters are approximated to match the prior knowledges in an optimal sense [5–16].

In designing a numerical solver for such a system, a critical issue lays on the handling of the fluid-structure interaction [17–19] between the heart muscle and the enclosed blood. One popular approach is the Arbitrary Lagrange-Euler (ALE) method [20,21], in which the fluid sub-system is described in the Euler coordinate system, the structure sub-system is described in the Lagrange coordinate systems, and the relevant field variables are coupled through the fluid-structure interface continuation conditions. Since the fluid-structure interface is explicitly treated, the ALE method can achieve a good accuracy to approximate the physical quantities that are defined on the surface, such as the wall shear stress. Another type of methods, including the immersed boundary(IB) and its variants such as the immersed boundary/finite element (IB/FE) method [22–27], works on a fixed mesh but requires a very fine mesh in the neighborhood of the interface to confine the interpolation error between the fluid mesh and structure mesh, which inevitably leads to a huge number of degrees of freedom. Watanabe et al. [28,29]. developed a program based on the finite element method in order to simulate the fluid-structure interaction of the human left ventricle during systole and diastole. Doyle et al. [30] used an ideal geometry to perform a fluid-structure interaction simulation of canine left ventricular blood flow, and analyzed in detail the parallel scalability and stability of their method. Nordsletten et al. [31] reconstructed the geometry of left ventricle based on Magnetic Resonance Imaging (MRI) and simulated the fluid-structure interaction of the heart to investigate the blood flow pattern and the pressure distribution inside the left ventricle. Gao et al. [32,33] used the IB/FE method to simulate the dynamic process of the left ventricle from end-diastole to end-systole in a heart beat, whose results show a very good match to the clinical observation.

In this paper, we propose an efficient parallel domain decomposition algorithm to solve the cardiac FSI system, and demonstrate the simulation results of large-scaled problems on a supercomputer. More specifically, we verify the computation efficiency, numerical stability and the parallel scalability of the proposed method, and demonstrate the fidelity of the simulated results by comparing them with clinical data. The rest of this paper is organized as follows: Section 2 briefly describes the reconstruction of the heart geometry from a medical image and the finite element mesh generation. The governing equations, boundary conditions, numerical solution algorithm, as well as the parameter settings are also introduced; in Section 3, we present the numerical experiments on the mesh size convergence, time step size convergence, and the parallel scalability of the solving algorithm, as well as give detailed analyses for a one-heart-beat simulation; the proposed method is summarized in Section 4, in which several possible improvements are also suggested for the future work.

2. Model and method

2.1. Geometry reconstruction and mesh generation

A CT image of a 38-year-old male is adopted to construct the 3D geometry model for our numerical

simulations, which was taken at the end of a diastole phase with spatial resolution 0.5 mm. The axial, coronal and sagittal views of CT images are shown in Figure 1(a–c), respectively, in which the red region represents the blood contained in the left ventricle, while the myocardium is colored in cyan. Figure 1(d) shows the initial shape of the left ventricle obtained from the CT image reconstruction process, which is smoothed and repaired to get the final geometry as shown in Figure 2(a). In our tests, five meshes, denoted by $\mathcal{L}_i$ $(i = 1,2,3,4,5)$, are generated for investigating the parallel efficiency and mesh convergence. The total number of mesh nodes and elements, the elements in the ventricle and cavity domains, and the element sizes are listed in Table 1. In particular, Figure 2(b) and (c) show the mesh structures of myocardium part and blood part of $\mathcal{L}_1$ in separate views.

Table 1. Number of nodes and elements of the meshes used in the simulations.

	Element size (mm)	Number of nodes	Number of elements	Ventricle elements	Cavity elements
$\mathcal{L}_1$	2.0	89679	507954	261596	246358
$\mathcal{L}_2$	1.5	209355	1202878	612265	590613
$\mathcal{L}_3$	1.2	405248	2348547	1197750	1150797
$\mathcal{L}_4$	1.0	696275	4058421	2065739	1992682
$\mathcal{L}_5$	0.8	1385804	7923279	4017108	3906171

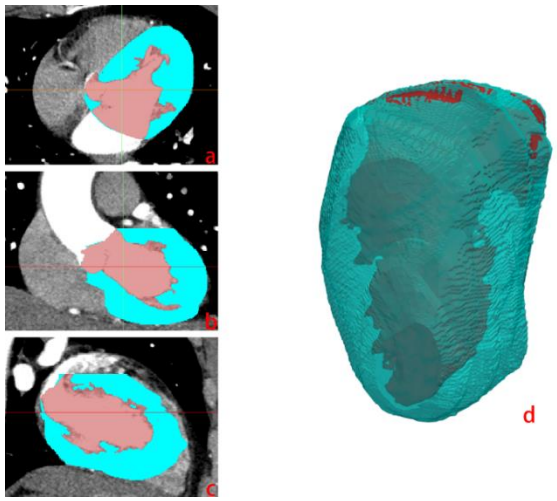

Figure 1. Original geometry of left ventricle reconstructed from CT images. (a–c) An axial, coronal, and sagittal plane of the CT image; (d) The reconstructed initial geometry of the left ventricle by using Mimics.

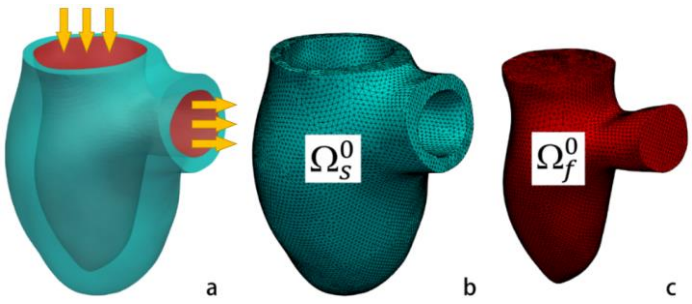

Figure 2. The smoothed geometry and a representative mesh of the left ventricle. (a) The smoothed geometry of left ventricle, of which the inflow and outflow tracts are marked by the arrows; (b) and (c) Mesh of the solid (myocardium), and fluid region (blood), respectively. The superscript 0 indicates they are at the initial configuration.

2.2. Governing Equations

Define the computational region at time t as $\Omega^t = \Omega_f^t \cup \Omega_s^t \subset R^3$, where Ω_f^t is fluid domain, Ω_s^t is structure domain, and $\Gamma_w^t = \partial\Omega_f^t \cap \partial\Omega_s^t$ is the interface between fluid and structure. In addition, $\Gamma_{f,I}^t$ and $\Gamma_{f,O}^t$ are the inlet and outlet boundaries of the fluid domain at time t, and $\Gamma_{s,I}^t$ and $\Gamma_{s,O}^t$ are the inlet and outlet boundaries of the structure domain at time t. In particular, the initial configuration of each of the above is denoted by letting $t = 0$, as shown in Figure 3.

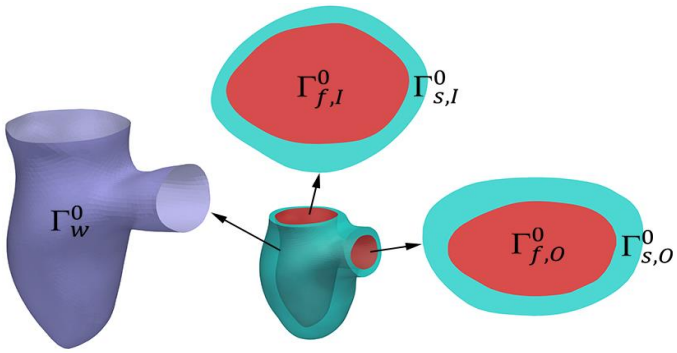

Figure 3. Initial configuration of the left ventricle domain, boundaries and interface.

The myocardium of the left ventricle is modeled as a linear elastic material. Namely, the movement of each material point in the myocardium region is described by the following equation:

$$\begin{cases} \rho_s \dfrac{\partial^2 \mathbf{x}_s}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma}_s = 0 & \text{in} \quad \Omega_s^0 \ , \\ \mathbf{x}_s = 0 & \text{on} \quad \Gamma_{s,I}^0 \cup \ \Gamma_{s,O}^0 \end{cases} \tag{1}$$

where $\mathbf{x}_s$ represents the displacement of a material point in Ω_s^0, ρ_s is density of myocardium, $\boldsymbol{\sigma}_s = \lambda_s(\nabla \cdot \mathbf{x}_s)\boldsymbol{I} + \mu_s(\nabla \mathbf{x}_s + \nabla \mathbf{x}_s{}^T)$ is the Cauchy stress tensor in which $\boldsymbol{I}$ denoting the rank two identity tensor, α is a damping parameter, λ_s and μ_s are two Lamé coefficients, which are generally calculated from provided material Young's modulus E and Poisson's ratio v_s : $\lambda_s = v_s E/\big((1+v_s)(1-2v_s)\big)$, $\mu_s = E/\big(2(1+v_s\ )\big)$.

On the other hand, we use an unsteady incompressible Navier-Stokes equation to depict the blood flow. Note that, because of the deformation of the myocardium, the fluid domain occupied by the blood changes in time. To deal with this issue, an ALE method [34] is used to describe the fluid domain Ω_f^t at time t:

$$A_t(\mathrm{Y}) = \mathrm{Y} + \mathbf{x}_f(\mathrm{Y}), \quad \forall \mathrm{Y} \in \Omega_f^0,$$

Where Y represents a reference point in the initial configuration of the fluid domain, and $\mathbf{x}_f$ is its displacement. In practice, the reference points match the mesh vertices, and $\mathbf{x}_f$ are assumed to satisfy the following Laplace equation:

$$\begin{cases} \Delta\ \mathbf{x}_f = 0 & \text{in} \quad \Omega_f^0\ , \\ \mathbf{x}_f = 0 & \text{on} \quad \Gamma_{f,I}^0 \cup\ \Gamma_{f,O}^0\ , \\ \mathbf{x}_f = \mathbf{x}_s & \text{on} \quad \Gamma_w^0\ . \end{cases} \tag{2}$$

Under the above settings, the ALE form of the fluid equation reads:

$$\begin{cases} \rho_f \left.\dfrac{\partial \mathbf{u}_f}{\partial t}\right|_Y + \rho_f\big[(\mathbf{u}_f - \boldsymbol{\omega}_g)\cdot\nabla\big]\mathbf{u}_f - \nabla\cdot\boldsymbol{\sigma}_f = 0 & \text{in} \quad \Omega_f^t\ , \\ \nabla\cdot\mathbf{u}_f = 0 & \text{in} \quad \Omega_f^t\ . \end{cases} \tag{3}$$

where ρ_f is fluid density, $\mathbf{u}_f$ is fluid velocity, and $\boldsymbol{\sigma}_f = -p_f\boldsymbol{I} + \mu_f\big(\nabla\mathbf{u}_f + \nabla\mathbf{u}_f{}^T\big)$ is Cauchy stress tensor, p_f is fluid pressure, μ_f is fluid viscosity. $\boldsymbol{\omega}_g = \partial\mathbf{x}_f/\partial t$ represents mesh moving velocity, and the first term in Eq (2) $(\partial\mathbf{u}_f/\partial t)\big|_Y$ denotes the time derivative of velocity is taken as the ALE coordinates are unchanged.

Besides, the following conditions are satisfied at the fluid-structure interface：

$$\begin{cases} \mathbf{u}_f = \dfrac{\partial\mathbf{x}_s}{\partial t} & \text{on} \quad \Gamma_w^t\ , \\ \boldsymbol{\sigma}_s\cdot\mathbf{n}_s = -\boldsymbol{\sigma}_f\cdot\mathbf{n}_f & \text{on} \quad \Gamma_w^t\ , \end{cases} \tag{4}$$

where $\mathbf{n}_s$ and $\mathbf{n}_f$ are the unit outer normal vectors in structure and fluid domains, respectively.

2.3. Material parameters and boundary conditions settings

For all experiments performed in this paper, the duration of a cardiac cycle was set to 0.80 s, with the diastolic phase lasts for 0.48 s and the systolic phase for 0.32 s, if not mentioned otherwise. Fluid

density is $1.05\,g/cm^3$, and viscosity is $0.03\,g/(cm\cdot s)$, structure density is $1.37\,g/cm^3$, Young's modulus is $6\times10^6\,g/(cm\cdot s^2)$, and Poisson's ratio is 0.49. Each numerical simulation starts from a diastolic phase, during this stage, the boundary conditions are given in Eq (5). That is, the inlet of left ventricle, $\Gamma_{f,I}^0$ keeps open, and the blood flow passing this boundary according to a prescribed rate, as the time-variant curve shown in Figure 4. Note that, this plot indicates two peaks of the inflow, the first is the early diastolic filling wave (E wave) and the second is the late diastolic filling wave (A wave). In the meantime, the outlet of left ventricle is closed, thus the blood flow rate on $\Gamma_{f,O}^0$ keeps at zero. Boundary conditions of the systolic phase are prescribed by Eq (6): during this stage, the inlet of left ventricle is closed (no blood inflow), while its outlet is open and the contained blood flows out freely in a pressure-free state.

$$\begin{cases}\mathbf{u}_f = v_f^d & \text{on} \quad \Gamma_{f,I}^t \\ \mathbf{u}_f = 0 & \text{on} \quad \Gamma_{f,O}^t\end{cases} \tag{5}$$

$$\begin{cases}\mathbf{u}_f = 0 & \text{on} \quad \Gamma_{f,I}^t \\ \boldsymbol{\sigma}_f\cdot\mathbf{n}_f = 0 & \text{on} \quad \Gamma_{f,O}^t\end{cases} \tag{6}$$

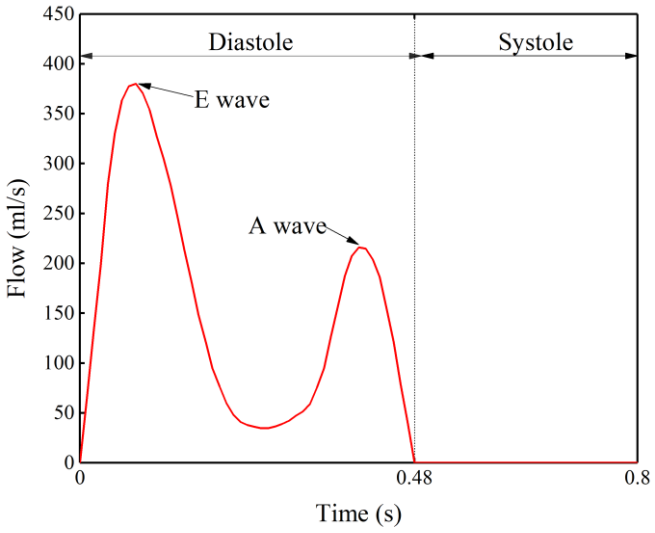

Figure 4. Inlet flow rate.

2.4. Numerical algorithms

We used a finite element method to spatially discretize the model equations, and solve the resulting system in a monolithic way. More specifically, the solid equation is discretized by the conventional P1 finite, and the fluid domain equations are discretized by a P1-P1 finite element that is enhanced with stabilization terms [35]. Further, a one-step backward-difference method [36] is used to get a fully discretized system, of which the solution vector is denoted by x. To solve the system, an inexact Newton method is invoked at each time step: Denote the nonlinear algebraic system as $\mathcal{F}(x)=0$, and

given the initial value $x^{(0)}$, the following iterations are performed:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}. \tag{7}$$

Here, $\alpha^{(k)}$ is the step size obtained by a line search step, and the search direction $s^{(k)}$ is obtained by using Krylov subspace method to approximately solve the Jacobian system as follows:

$$J_k M_k^{-1} M_k s^{(k)} = -\mathcal{F}\left(x^{(k)}\right), \tag{8}$$

$$\left\| J_k (M_k)^{-1} M_k s^{(k)} + \mathcal{F}(x^{(k)}) \right\| \leq \eta_k \left\| \mathcal{F}\left(x^{(k)}\right) \right\|, \tag{9}$$

where $J_k = \nabla \mathcal{F}\left(x^{(k)}\right)$ is the Jacobian matrix of the nonlinear function $\mathcal{F}(x)$ at point $x^{(k)}$. Due to the intrinsic properties of Navier-Stokes equations, J_k is non-symmetric, so the GMRES method is used to approximately solve the linear system (8). η_k is the termination condition of the linear iteration, and 10^{-6} is chosen in this paper.

In above, one key argument is on the construction of M_k, which represents a precondition operator for accelerating the convergence of Krylov iterations. In this paper, we use the Restricted Additive Schwarz (RAS) precondition method [37] to construct M_k. To do so, the whole mesh Ω is divided into N non-overlapping sub-domains:

$$\Omega = \bigcup_{l=1}^{N} \Omega_l \ , \qquad l = 1,2 \ldots N \quad and \quad \Omega_i \cap \Omega_j = \emptyset, i \neq j$$

then each sub-domain Ω_l is extended by δ layers to obtain overlapped sub-domain Ω_l^δ. The preconditioner M_k is constructed by:

$$M_k = \sum_{l=1}^{N} (R_l^0)^T B_l^{-1} R_l^\delta \ ,$$

where R_l^δ and R_0^δ denote the restriction operators of Ω to Ω_l^δ and Ω_l:

$$R_l^\delta v = v|_{\Omega_l^0} \qquad \text{and} \qquad R_l^\delta v = v|_{\Omega_l^\delta},$$

for v represents a vector defined on the global mesh domain. Beside, their transposes represent the corresponding prolongation operators, B_l^{-1} denotes the subproblem solver defined on Ω_l^δ, which is set as an in-complete LU decomposition operator of the subproblem Jacobian matrix in this paper. More details on the implementation of the preconditioner can be found in [32].

For clarity, we summarize Newton-Krylov-Schwarz(NKS) algorithm for solving the system of nonlinear algebraic equations of the FSI problem in each time step as follows.

step1 Set $x^{(0)}$ according to the initial conditions or the results of previous time step
step2 k ← 0
do
  step3 Compute $\mathcal{F}(x^{(k)})$ and its Jacobian matrix J_k
  step4 Check the stopping condition. If it is satisfied, then stop the loop and get the solution, otherwise go to the next step
  step5 Using Krylov subspace method to approximately solve the preconditioned Jacobian system (8) to get $s^{(k)}$
  step6 Using a line search method to obtain $\alpha^{(k)}$
  step7 Let $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}$
  step8 k ← k + 1 and go to **step3**
end

3. Numerical experiments

3.1. Mesh convergence tests

It is well-known that, when solving the above fluid-structure interaction problem, the smaller size of mesh element, the higher accuracy of the solution will achieve, which, however, consumes more computational resources and time. In this section, we discuss the mesh convergence of the proposed algorithm and analyze the effect of the size of the mesh element on the solution accuracy. For this purpose, four meshes, $\mathcal{L}_1$,$\mathcal{L}_2$, $\mathcal{L}_3$,$\mathcal{L}_4$ are used to repeat a same simulation for one cardiac cycle. The time step is set to 0.0005 s, and 20 nodes of Tianhe-2A supercomputer (a total of $20 \times 24 = 480$ processor cores) are used for each case.

Time and memory cost by using each of the meshes are listed in Table 2, where "time" represents the total compute time for all time steps and "memory" is the maximum memory allocated by all processor cores during the simulation. Effect of the element size on the compute time and memory cost is given in Figure 5. As it is expected, as mesh element size decreases, both time and memory required for simulation get increased.

Table 2. Time and memory used for solving with different meshes.

	Element Size (mm)	Number of Nodes	Number of Elements	Time (s)	Memory (GB)
$\mathcal{L}_1$	2.0	89679	507954	6972	96.31
$\mathcal{L}_2$	1.5	209355	1202878	19710	110.78
$\mathcal{L}_3$	1.2	405248	2348547	31083	133.39
$\mathcal{L}_4$	1.0	696275	4058421	52360	162.90

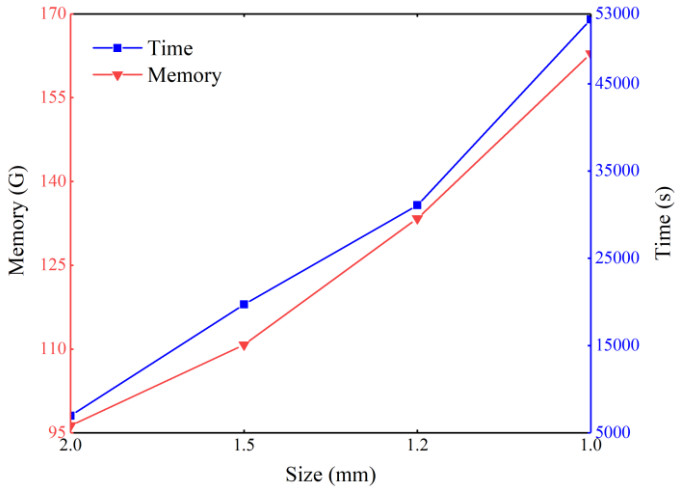

Figure 5. Variation of time and memory used for simulation with mesh element size. Note that the mesh cell size are listed in decreasing order from left to right on the horizontal axis.

To verify our results, the calculated left ventricular cavity volume and the blood flow rate at its outlet by using different meshes are compared: Let $V_i(t)$ be the left ventricular cavity volume at time t obtained by using mesh $\mathcal{L}_i$ $(i = 1,2,3,4)$, and $Q_i(t)$ the flow rate through the outlet. We regard the results of the finest mesh $\mathcal{L}_4$ as a reference, and calculate the deviations of the cavity volume and outlet flow rate of other coarser meshes. Deviations at several representative moments are listed in Table 3 and 4, and the overall pattern are plotted in Figures 6 and 7 for the whole cardiac cycle. All of these clearly show that the deviation becomes smaller as the element size decreases, which indicates that the solution accuracy gets improved.

Table 3. Differences of left ventricular cavity volumes obtained from simulations by using different meshes.

Volume difference (mL)	t = 0.16	t = 0.32	t = 0.48	t = 0.6	t = 0.7	t = 0.8
$\|\boldsymbol{V_1(t) - V_4(t)}\|$	1.622	1.981	2.741	2.322	1.948	1.62
$\|\boldsymbol{V_2(t) - V_4(t)}\|$	0.865	1.058	1.47	1.174	0.965	0.794
$\|\boldsymbol{V_3(t) - V_4(t)}\|$	0.382	0.467	0.651	0.504	0.407	0.332

Table 4. Difference of outlet flow rates obtained from simulations by using different meshes.

Flow rate difference (mL/s)	t = 0.6	t = 0.7	t = 0.8
$\|\boldsymbol{Q_1(t) - Q_4(t)}\|$	2.938	3.0923	4.7726
$\|\boldsymbol{Q_2(t) - Q_4(t)}\|$	1.664	1.6414	3.3274
$\|\boldsymbol{Q_3(t) - Q_4(t)}\|$	0.763	0.7587	2.6136

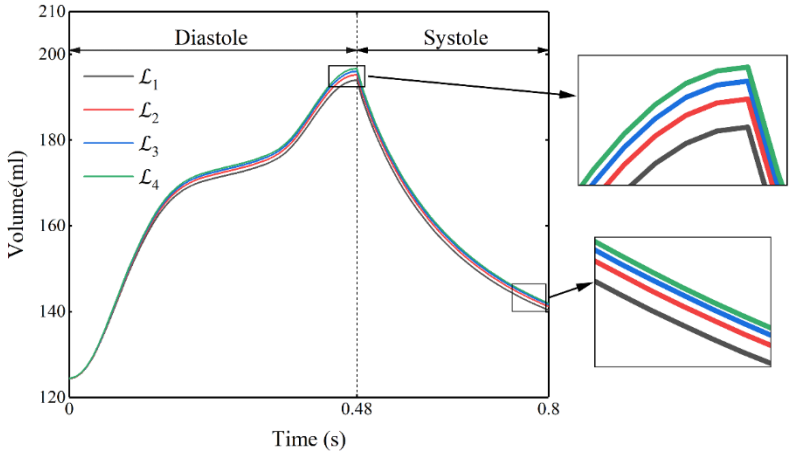

Figure 6. Variation of left ventricular cavity volume with time obtained from simulations by using different meshes.

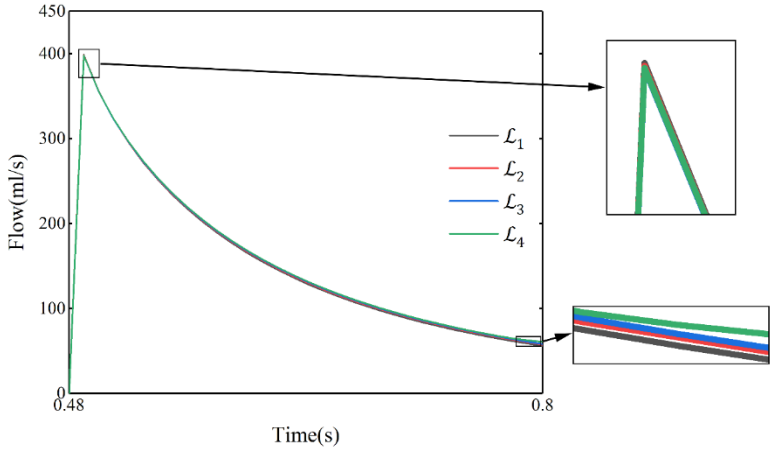

Figure 7. Variation of outlet flow rate with time obtained from simulations by using different meshes.

3.2. Time-step convergence

In order to verify the convergence of time step size, we use $\mathcal{L}_1$ mesh and set the size of the time step to be $\Delta t_1 = 0.005, \Delta t_2 = 0.002, \Delta t_3 = 0.001, \Delta t_4 = 0.0005$ (unit: second) to perform simulations, respectively. $V^i(t)$ denotes the volume occupied by the blood contained in left ventricle at time t by using time step Δt_i, and $Q^i(t)$ denotes the rate of the blood flow that passes through the left ventricular outlet. Taking the results computed by using the smallest Δt_4 as the reference, the deviations of left ventricle blood volume and outlet flow rate obtained by using other time steps at several representative moments are presented in Tables 5 and 6, and the overall patterns during the cardiac cycle are presented in Figures 8 and 9. Based on these results, it can be seen that the difference of different computing results becomes smaller as the time step decreases, as desired.

Table 5. Differences of left ventricular cavity volumes obtained from simulations by using different time steps.

Volume difference (mL)	t = 0.16	t = 0.32	t = 0.48	t = 0.6	t = 0.7	t = 0.8
$\left\|\boldsymbol{V^1(t) - V^4(t)}\right\|$	0.292	0.015	0.118	0.227	0.217	0.179
$\left\|\boldsymbol{V^2(t) - V^4(t)}\right\|$	0.097	0.005	0.039	0.075	0.072	0.06
$\left\|\boldsymbol{V^3(t) - V^4(t)}\right\|$	0.032	0.002	0.013	0.024	0.023	0.02

Table 6. Difference of outlet flow rates obtained from simulations by using different time steps.

Flow rate difference (mL/s)	t = 0.6	t = 0.7	t = 0.8
$\left\|\boldsymbol{Q^1(t) - Q^4(t)}\right\|$	2.074	1.378	1.129
$\left\|\boldsymbol{Q^2(t) - Q^4(t)}\right\|$	0.674	0.446	0.412
$\left\|\boldsymbol{Q^3(t) - Q^4(t)}\right\|$	0.214	0.131	0.146

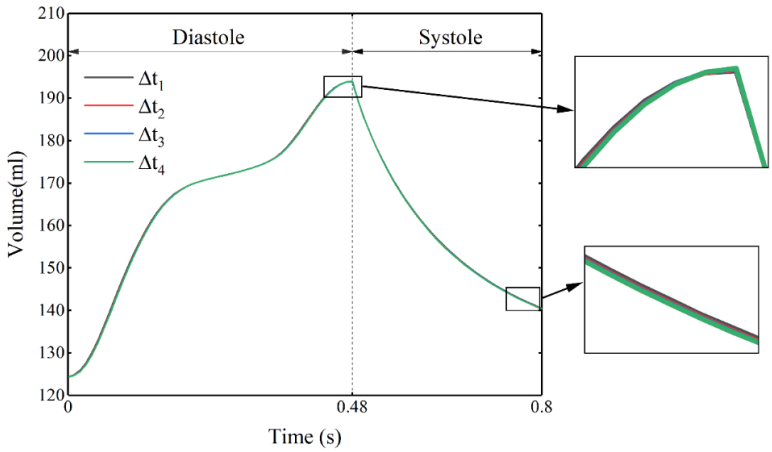

Figure 8. Variation of left ventricular cavity volume with time obtained from simulations by using different time steps.

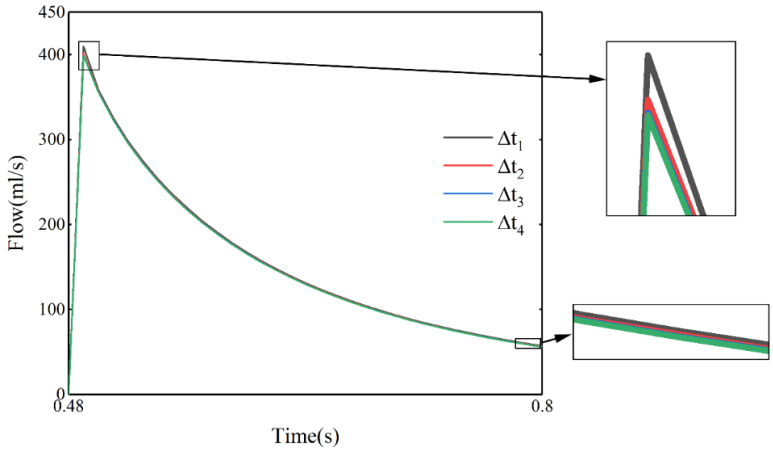

Figure 9. Variation of outlet flow rate with time obtained from simulations by using different time steps.

3.3. Parallel performance

To investigate the parallel scalability of the proposed algorithm, two computational meshes, $\mathcal{L}_4$ (4.06×10^6 mesh elements) and $\mathcal{L}_5$ (7.92×10^6 mesh elements), are chosen to repeat simulations on the Tianhe-2A supercomputer, by using 72, 144, 288, 576, 1152, and 2304 processor cores, respectively. For each case, the average compute time of the first 10 time steps is counted to calculate the speedup. More specifically, let T_{ini} be the average computation time per time step when using 72 processors, and T_p the average computation time per time step when using p processors, the speedup and parallel efficiency are defined as follows:

$$\text{Speedup} = \frac{T_{ini}}{T_p}, \qquad \text{Efficiency} = \frac{\text{Speedup}}{p \div 72}, \qquad p = 72, 144, 288, 576, 1152, 2304.$$

Table 7 , Figures 10 and 11 present the parallel scalability performance of the numerical algorithm for solving the left ventricle fluid-structure interaction problem, where "np" denotes the number of processor cores, "Newton" the average number of Newton iteration steps per time step, "GMRES" the average number of GMRES iteration steps in each Newton step, "Time" the average compute time per time step, "Speedup" the speedup of the algorithm, and "Efficiency" the parallel efficiency of the algorithm. It can be seen that when the number of processor cores gets increased while the problem size keeps unchanged, the number of Newton iteration steps remains almost the same and the number of GMRES iterations increases just few steps, and the total compute time decreases significantly. Moreover, as the number of mesh elements is increased from 4.06×10^6 to 7.92×10^6, both speedup and parallel efficiency are improved, indicating that the algorithms can achieve higher parallel efficiency when the scale of the problem matches appropriately the number of processor cores. Nevertheless, it can be found that, when for case of 7.92×10^6 elements, the algorithm still has 40% parallel efficiency when the number of processor cores is expanded to 2304 cores. These results verify that the proposed algorithm has good parallel scalability for solving the left ventricle fluid-structure interaction problem.

Table 7. Parallel scalability results of the algorithm.

	np	Newton	GMRES	Time(s)	Speedup	Ideal	Efficiency
	72	2	49.5	52.4	1.0	1	100%
	144	2	49.3	35.1	1.5	2	75%
$\mathcal{L}_4$	288	2	50.6	17.8	2.9	4	74%
4.06×10^6	576	2	54.2	9.3	5.6	8	71%
	1152	2	57.1	6.8	7.7	16	48%
	2304	2	59.4	5.1	10.2	32	32%
	72	2	61.6	107.5	1.0	1	100%
	144	2	64.6	58.5	1.8	2	92%
$\mathcal{L}_5$	288	2	66.4	32.3	3.3	4	83%
7.92×10^6	576	2	69.7	18.5	5.8	8	73%
	1152	2	70.7	11.6	9.3	16	58%
	2304	2	74.7	8.3	12.9	32	40%

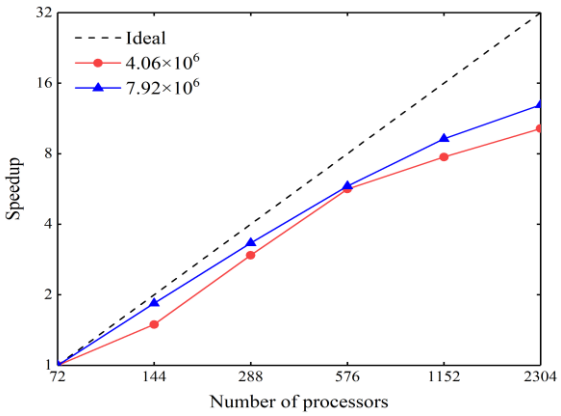

Figure 10. Speedup of the algorithm for using up to 2304 processor cores. The dash line represents the ideal speedup rate.

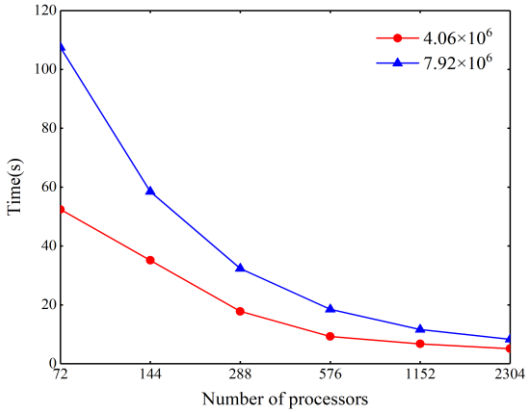

Figure 11. The average compute time per time step of the algorithm for using up to 2304 processor cores.

3.4. Analysis on simulation results

In the following, we use 480 processors on Tianhe-2A supercomputer to perform a cardiac cycle simulation based on $\mathcal{L}_4$, and the time step size is set to 0.0005 s. Due to the simplification of the model, we mainly discuss the velocity field distribution in the blood region contained left ventricle, as well as the deformation and stress status of the myocardium.

Figure 12 shows the rate of blood flow at the inlet and outlet of left ventricle as a time-dependent function during a cardiac cycle, and Figure 13 shows instantaneous blood flow velocity streamline in left ventricle at several time moments. It can be seen that the blood flow velocity increased rapidly at the beginning stage, and the first local maximal flow rate appears as the early diastolic filling wave (E wave) arrives. Besides, an obvious vortex formed at 0.08s can be observed. After that, the inlet flow rate drops, but the previously observed vortex area gets expanded, with some new vortexes generated. At 0.25 s, the inlet flow velocity drops to a low level, the velocity field also gradually gets stabilized, and the vortex area decrease to almost vanished. The late diastolic filling wave (A wave) arrives around

0.4 s, which causes the inlet flow rate increased again. It can be seen that the velocity field in the left ventricle fluctuated, and a vortex appears again. The diastole phase terminates at 0.48 s, and the systole begins right away, during which, the left ventricle inlet keeps closed and the outlet is open. The outlet blood flow velocity quickly reaches its peak, and then falls gradually. During the systolic phase, a vortex can also be observed at 0.70 s.

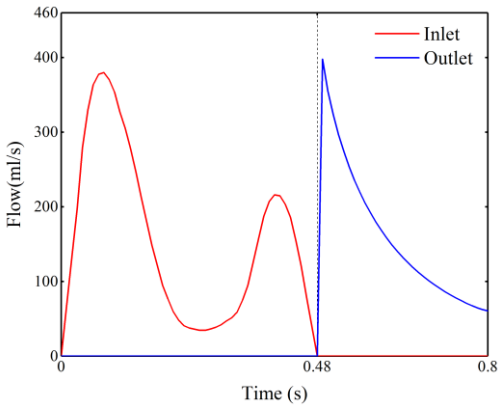

Figure 12. Variation of flow rate at the inlet and outlet of left ventricle with time during one cardiac cycle.

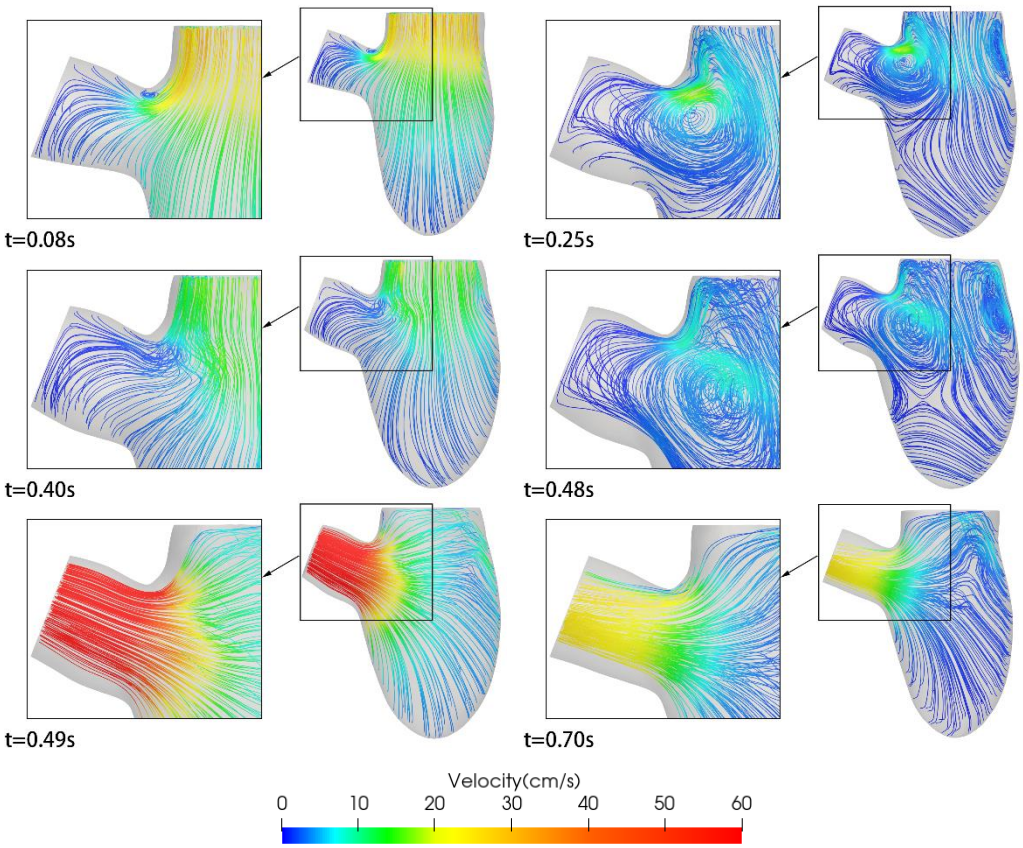

Figure 13. Instantaneous velocity streamline in the left ventricle.

In addition, Figure 14 presents the history of the cavity volume of the left ventricle during one cardiac cycle, and Figure 15 demonstrates the instantaneous velocity field of the blood as well as the displacement field of the myocardium. According to these results, it can be seen that the velocity field inside the left ventricle fluctuates with the inlet flow rate throughout diastole, and the volume of the left ventricular cavity and the displacement of the myocardium undergo two drastic changes after the arrival of the E and A waves, as large amount of blood rapidly flushes into the left ventricle. Besides, during the systolic phase, the outlet blood flow reached its peak at the beginning of contraction and then decrease gradually, and the velocity near the outlet region was significantly higher than elsewhere. As expected, after blood flowed out of left ventricle, the volume of the left ventricular cavity and the displacement of the myocardium around the left ventricle get decreased.

Furthermore, we cut the left ventricle on three representative cross section along the axial axis, coronal axis and sagittal axis, respectively, and compare the simulation results with the CT image data at the end of the systole, as shown in Figure 16. It can be seen that the results obtained by simulation are within the normal physiological range and visually match the CT images. But at the same time, due to the simplicity of the underlying model and experimental settings (e.g., the left ventricle is linearly elastic and the active contraction is ignored), some artifacts and flaws can be found in the current simulation results. For example, as shown in Figure 14, the volume of the left ventricle cavity at the end of the drainage stage is larger than the initial one, and we found that the blood flowing into the left ventricle during diastole is larger than that flowing out from during the drainage stage (74 mL vs 56 mL), indicating an un-normal situation. In addition, backflow has already appeared at the outlet at the end of contraction, and the backflow speed is relatively high. At this moment, the blood flow rate is about 20cm/s, but the speed of backflow has reached about 80 cm/s. If the simulation continues, the backflow speed may reach up to 1000 cm/s, which does not lay in a physiological range. In the future, we will revise our model, algorithm and experimental settings, in aiming to get more realistic simulation results.

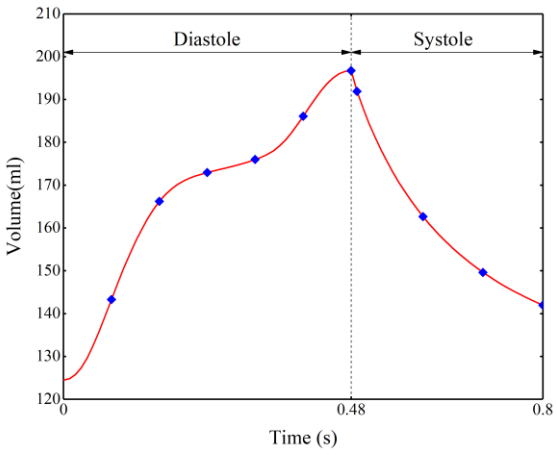

Figure14. Change of the left ventricular cavity volume with time. The blue points correspond to the left ventricular cavity volumes at the time showed in Figure 15.

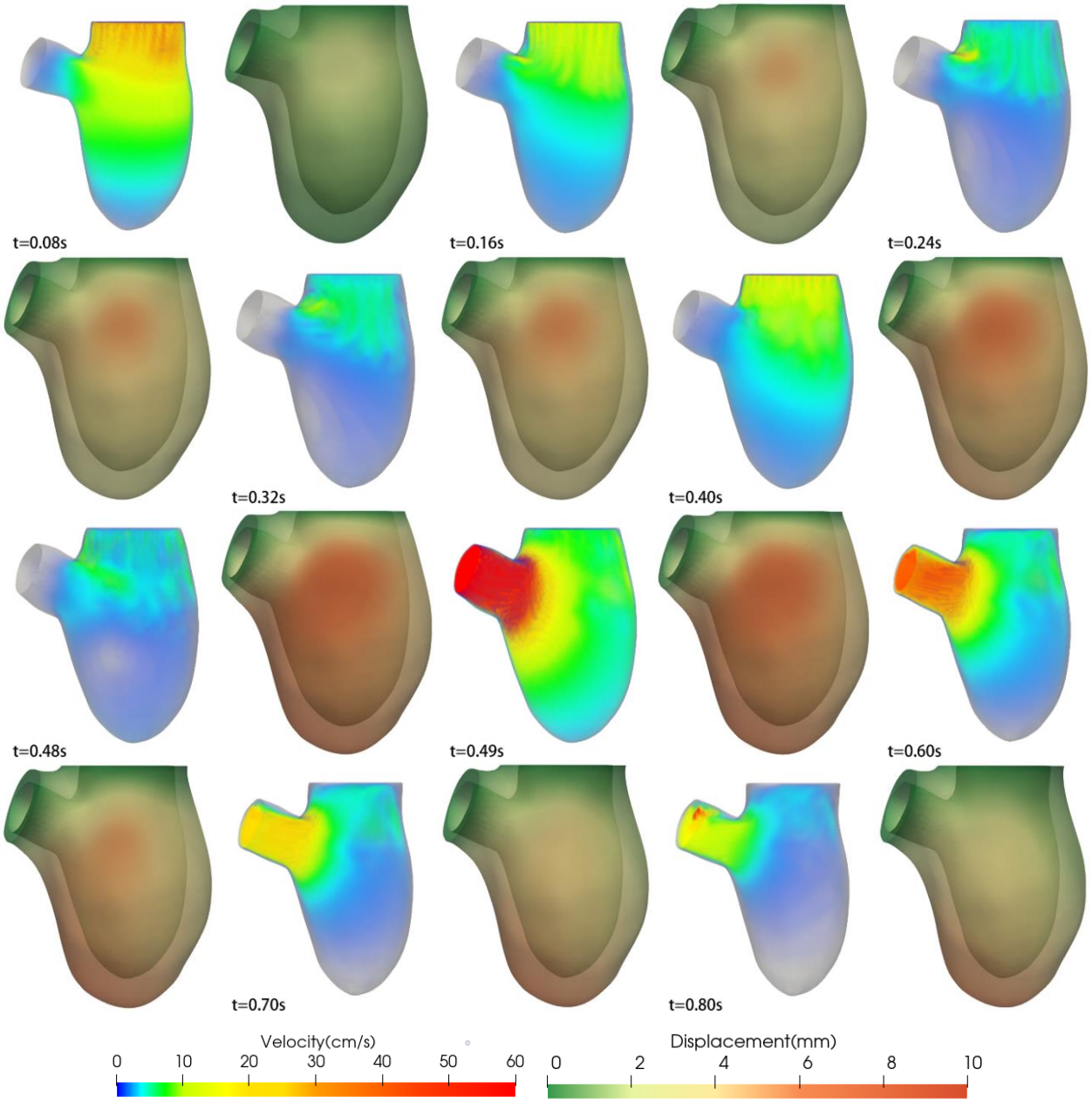

Figure 15. Instantaneous velocity distribution of blood in the left ventricle (left) and displacement change of myocardium (right) during a cardiac cycle.

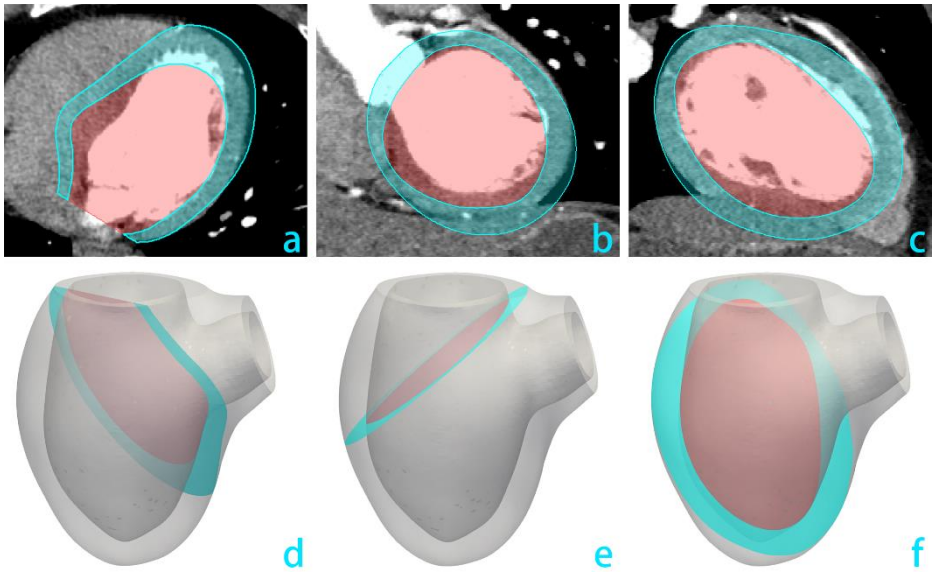

Figure 16. Comparison of the simulation results with the CT images at the end of diastole. (a)–(c) are axial, coronal and sagittal sections of the CT images, respectively. The white and dark gray areas are the blood and myocardial tissue, and the red and cyan areas are the distribution of blood and myocardium on the cross-section obtained from the simulation. (d)–(f) mark the spatial locations of the three sections in (a)–(c), respectively.

4. Conclusions

Due to the complexity of human heart, fluid-structure interaction simulation on the hemodynamic is a very challenging task. In this paper, we proposed a high-performance parallel solver to perform efficient FSI simulations of the human heart left ventricle on Tianhe-2A supercomputer. Through experiments, we verify that the algorithm is highly efficient, and has good numerical stability and parallel scalability up to more than 2000 processor cores. By using the solver, we further provide a preliminary demonstration of its possible application scenarios in clinical research.

Acknowledgements

This work was partially supported by the National Key R&D Program of China 2018YFE0198400, NSFC grants under 12001520, 12071461 and 62161160312, and Shenzhen Fund RCYX20200714114735074.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. M. Nash, P. Hunter, Computational mechanics of the heart, *J. Elasticity*, **61** (2000), 113–141. https://doi.org/10.1023/A:1011084330767

2. D. Kass, C.-H. Chen, C. Curry, M. Talbot, R. Berger, B. Fetics, et al., Improved left ventricular mechanics from acute VDD pacing in patients with dilated cardiomyopathy and ventricular conduction delay, *Circulation*, **99** (1999), 1567–1573. https://doi.org/10.1161/01.CIR.99.12.1567
3. S. N. Doost, D. Ghista, B. Su, L. Zhong, Y. S. Morsi, Heart blood flow simulation: a perspective review, *Biomed. Eng. Online*, **15** (2016), 1–28. https://doi.org/10.1186/s12938-016-0224-8
4. R. Mittal, J. H. Seo, V. Vedula, Y. J. Choi, H. Liu, H. H. Huang, et al., Computational modeling of cardiac hemodynamics: Current status and future outlook, *J. Comput. Phys.*, **305** (2016), 1065–1082. https://doi.org/10.1016/j.jcp.2015.11.022
5. J. Li, J. M. Melenk, B. Wohlmuth, J. Zou, Optimal a priori estimates for higher order finite elements for elliptic interface problems, *Appl. Numer. Math.*, **60** (2010), 19–37. https://doi.org/10.1016/j.apnum.2009.08.005
6. J. Li, J. Xie, J. Zou, An adaptive finite element reconstruction of distributed fluxes, *Inverse Probl.*, **27** (2011), 075009. https://doi.org/10.1088/0266-5611/27/7/075009
7. X. Cao, H. Diao, J. Li, Some recent progress on inverse scattering problems within general polyhedral geometry, *Electron. Res. Arch.*, **29** (2021), 1753–1782. https://doi.org/10.3934/era.2020090
8. J. Li, H. Liu, H. Sun, On an inverse elastic wave imaging scheme for nearly incompressible materials, *IMA J. Appl. Math.*, **84** (2018), 229–257. https://doi.org/10.1093/imamat/hxy056
9. H. Diao, H. Liu, L. Wang, On generalized Holmgren's principle to the Lame operator with applications to inverse elastic problems, *Calc. Var. Partial Differ. Equ.*, **59** (2020), 179. https://doi.org/10.1007/s00526-020-01830-5
10. H. Diao, H. Liu, B. Sun, On a local geometric property of the generalized elastic transmission eigenfunctions and application, *Inverse Probl.*, **37** (2021), 105015. https://doi.org/10.1088/1361-6420/ac23c2
11. X. Wang, Y. Guo, S. Bousba, Direct imaging for the moment tensor point sources of elastic waves, *J. Comput. Phys.*, **448** (2022), 110731. https://doi.org/10.1016/j.jcp.2021.110731
12. H. Li, H. Liu, J. Zou, Minnaert resonances for bubbles in soft elastic materials, *SIAM J. Appl. Math.*, **82** (2022), 119–141. https://doi.org/10.1137/21M1400572
13. H. Liu, W.-Y. Tsui, A. Wahab, X. Wang, Three-dimensional elastic scattering coefficients and enhancement of the elastic near cloaking, *J. Elasticity*, **143** (2021), 111–146. https://doi.org/10.1007/s10659-020-09807-3
14. Y. Deng, H. Li, H. Liu, On spectral properties of Neuman-Poincare operator and plasmonic resonances in 3D elastostatics, *J. Spectr. Theory*, **9** (2019), 767–789. https://doi.org/10.4171/JST/262
15. H. Li, J. Li, H. Liu, On novel elastic structures inducing polariton resonances with finite frequencies and cloaking due to anomalous localized resonances, *J. Math. Pures Appl.*, **120** (2018), 195–219. https://doi.org/10.1016/j.matpur.2018.06.014
16. H. Li, H. Liu, On anomalous localized resonance for the elastostatic system, *SIAM J. Math. Anal.,* **48** (2016), 3322–3344. https://doi.org/10.1137/16M1059023
17. E. H. Dowell, K. C. Hall, Modeling of fluid-structure interaction, *Annu. Rev. Fluid Mech.*, **33** (2001), 445–490. https://doi.org/10.1146/annurev.fluid.33.1.445
18. G. Hou, J. Wang, A. Layton, Numerical methods for fluid-structure interaction---a review, *Commun. Comput. Phys.*, **12** (2012), 337–377. https://doi.org/10.4208/cicp.291210.290411s

19. F. Jiang, Stabilizing effect of elasticity on the motion of viscoelastic/elastic fluids, *Electron. Res. Arch.*, **29**, 2021, 4051–4074. https://doi.org/10.3934/era.2021071
20. C. Hirt, A. Anthony, J. L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, *J. Comput. Phys.*, **14** (1974), 227–253. https://doi.org/10.1016/0021-9991(74)90051-5
21. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, T. JR Hughes, Fluid-structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation, *Comput. Mech.*, **54** (2014), 1055–1071. https://doi.org/10.1007/s00466-014-1059-4
22. C. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.*, **10** (1972), 252–271. https://doi.org/10.1016/0021-9991(72)90065-4
23. F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid-structure interaction, *Prog. Aerosp. Sci.*, **65** (2014), 1–21. https://doi.org/10.1016/j.paerosci.2013.09.003
24. W. Kim, H. Choi, Immersed boundary methods for fluid-structure interaction: A review, *Int. J. Heat Fluid Flow*, **75** (2019), 301–309. https://doi.org/10.1016/j.ijheatfluidflow.2019.01.010
25. D. Jones, X. Zhang, A conforming-nonconforming mixed immersed finite element method for unsteady stokes equations with moving interfaces, *Electron. Res. Arch.*, **29**, (2021), 3171–3191. https://doi.org/10.3934/era.2021032
26. D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, *Comput. Struct.*, **81** (2003), 491–501. https://doi.org/10.1016/S0045-7949(02)00404-2
27. B. E. Griffith, X. Luo, Hybrid finite difference/finite element immersed boundary method, *Int. J. Numer. Method Biomed. Eng.*, **33** (2017), e2888. https://doi.org/10.1002/cnm.2888
28. H. Watanabe, T. Hisada, S. Sugiura, J. Okada, H. Fukunari, Computer Simulation of Blood Flow, Left Ventricular Wall Motion and Their Interrelationship by Fluid-Structure Interaction Finite Element Method, *JSME Int. J. C-Mech. SY*, **45** (2002), 1003–1021. https://doi.org/10.1299/jsmec.45.1003
29. H. Watanabe, S. Sugiura, H. Kafuku, T. Hisada, Multiphysics Simulation of Left Ventricular Filling Dynamics Using Fluid-Structure Interaction Finite Element Method, *Biophys. J.*, **87** (2004), 2074–2085. https://doi.org/10.1529/biophysj.103.035840
30. M. Doyle, S. Tavoularis, Y. Bourgault, Application of Parallel Processing to the Simulation of Heart Mechanics, *International Symposium on High Performance Computing Systems and Applications*, Springer, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-642-12659-8_3
31. D. Nordsletten, M. McCormick, P. J. Kilner, D. Kay, N. P. Smith, Fluid–solid coupling for the investigation of diastolic and systolic human left ventricular function, *Int. J. Numer. Methods Biomed. Eng.*, **27** (2011), 1017–1039. https://doi.org/10.1002/cnm.1405
32. H. Gao, D. Carrick, C. Berry, B. Griffith, X. Luo, Dynamic finite-strain modelling of the human left ventricle in health and disease using an immersed boundary-finite element method, *IMA J. Appl. Math.*, **79** (2014), 978–1010. https://doi.org/10.1093/imamat/hxu029
33. H. Gao, H. Wang, C. Berry, X. Luo, B. Griffith, Quasi-static image-based immersed boundary-finite element model of left ventricle under diastolic loading, *Int. J. Numer. Methods Biomed. Eng.*, **30** (2014), 1199–1222. https://doi.org/10.1002/cnm.2652
34. Y. Wu, X.-C. Cai, A fully implicit domain decomposition based ALE framework for three-dimensional fluid–structure interaction with application in blood flow computation, *J. Comput. Phys.*, **258** (2014), 524–537. https://doi.org/10.1016/j.jcp.2013.10.046

35. L. Franca, S. Frey, Stabilized finite element methods: II. The incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.*, **99** (1992), 209–233. https://doi.org/10.1016/0045-7825(92)90041-H
36. W. Ames, W. Rheinboldt, A. Jeffrey, Numerical Methods Partial Differential Equation, Second Edition, Academic Press, 1977.
37. X.-C. Cai, M. Sarkis, A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems, *SIAM J. Sci. Comput.*, **21** (1999), 792–797. https://doi.org/10.1137/S106482759732678X

AIMS Press