

*Regular Paper*

## Development of Explicit Eulerian Finite Difference Solver for Large-Scale Fluid-Structure Interaction Systems

KAZUYASU SUGIYAMA,<sup>†1</sup> YASUHIRO KAWASHIMA,<sup>†2</sup>  
HIROSHI KOYAMA,<sup>†3</sup> SHIGEHO NODA,<sup>†3</sup> SATOSHI II,<sup>†1</sup>  
SHU TAKAGI,<sup>†1,†3</sup> YOICHIRO MATSUMOTO<sup>†1</sup> and RYUTARO HIMENO<sup>†3</sup>

A scalable numerical algorithm has been reconsidered for massively parallel computations of fluid-structure interaction systems as biological applications. A new Eulerian method using a fixed mesh has been developed to solve the basic equation set for the incompressible Newtonian fluid and hyperelastic material in a finite difference manner. A new artificial compressibility method, corresponding to one of full explicit time-stepping algorithms, with adaptive parameters is proposed. The advocated solver easily attains excellent scalability since it makes the workload on each core equivalent and reduces the amount of node-to-node communication required for the iterative computation. It is applied to wall-bounded flows with biconcave particles, which replicate the shape of red blood cells. The computational performance on a Xeon cluster is presented in terms of a weak scaling and also of a strong scaling with  $O(10^9)$  grid points up to 8,192 cores.

### 1. Introduction

Fluid-Structure Interaction (FSI) phenomena appear in a number of biological systems. In general, the system (e.g. a blood flow involving flexibly deformable Red Blood Cells (RBCs)) has complexity associated with its multi-scale/physics nature, the complicated geometry, and a large number of dispersed bodies. In the biocomputing research field, technical progresses in the coupling of the fluid and structure dynamics have been vigorously made and large-scale computations have been performed<sup>1)–4)</sup>.

When dealing with moving interface problems, one has preferably employed a Lagrangian description using a finite element mesh since it is suited for treating an elastic constitutive law. The numerical methods include Arbitrary Lagrangian Eulerian<sup>5)</sup>, Deforming-Spatial-Domain/Stabilized Space-Time<sup>6)</sup> and Immersed Boundary<sup>7)</sup> approaches. Once the Lagrangian mesh is provided, the state-of-the-art approaches are satisfactory for achieving accurate predictions, and have been applied to a wide variety of biological problems. However, in a massively parallel computation, it is formidable to reduce a computational-load imbalance in the mesh nodes for the system involving complicated geometry of solid and/or

a large number of objects. To release the FSI simulation from the mesh distortion and/or the mesh reconstruction procedure and also from the load-balancing task, full Eulerian (fixed-mesh) approaches<sup>8)–19)</sup> have been explored. The authors formulated the basic equations suited to the finite difference method, and then demonstrated that the proposed method provided significant advances in our understanding of the geometrical flexibility<sup>12)–17)</sup>. Since a uniform cubic grid is used to discretize the basic equation set, it easily makes the computational load on each core for a rectangularly-divided subdomain equivalent.

The full Eulerian FSI method is characterized by the feasibility in implementing the hyperelastic constitutive law into the standard incompressible fluid flow algorithm, in which the pressure Poisson equation is implicitly solved in an iterative way to satisfy the mass conservation. One may expect that efficient computational techniques cultivated in the field of the computational fluid dynamics can be utilized, which would be an advantage in the realization of massively parallel computation. However, the iterative computation requires the repetitive nearest-neighbor communications, and thus it is more difficult to scale to a larger number of cores. To achieve a viable compromise between the numerical consistency and the scalability, a new Artificial Compressibility Method (ACM) has been developed. Unlike the original ACM<sup>20)</sup> for the steady flow problem or the

---

<sup>†1</sup> The University of Tokyo

<sup>†2</sup> Fujitsu Nagano Systems Engineering Ltd.

<sup>†3</sup> Riken

implicit ACM<sup>21</sup>), we employ a full explicit time-stepping scheme<sup>22</sup>). In contrast to the conventional ACM, in which the model parameters are fixed, we introduce an optimization procedure, which dynamically determines the model parameters to be adapted to the flow state in a way such as an error minimization.

In this paper, we focus on wall-bounded flows with discolored biconcave particles, which replicate the shape of RBC. We perform parallel simulations using up to 8,192 cores on a Xeon cluster (MPC-RICC, RIKEN), and weak and strong scalability tests.

## 2. Simulation methods

### 2.1 Eulerian description for fluid-structure interaction problem

The fluid and solid are assumed to be incompressible and to possess the same density and viscosity, as in many analyses for biological systems. The governing equations are comprised of the mass and momentum conservations:

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0, \quad (1) \\ \rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) &= -\nabla p + \nabla \cdot \boldsymbol{\tau} + \left( -\frac{\Delta P}{L_x} \right) \mathbf{e}_x, \quad (2) \end{aligned}$$

where  $\rho$  denotes the density,  $\mathbf{e}_x$  the unit vector in  $x$  direction,  $\mathbf{v}$  the velocity vector,  $p$  the pressure deviation from the driving pressure, and  $\boldsymbol{\tau}$  the deviatoric Cauchy stress. To pump the fluid and solid, the uniform pressure gradient  $-\Delta P/L_x$  (here,  $L_x$  is the inlet-outlet length in  $x$  direction of the computational domain, and  $\Delta P$  is the inlet-outlet pressure drop) is applied to the system. The deviatoric Cauchy stress is written in a mixture form of the Newtonian fluid and the neo-Hookean material, namely,

$$\boldsymbol{\tau}(\mathbf{x}) = 2\mu \mathbf{D}(\mathbf{x}) + G\phi_s^{1/2}(\mathbf{x}) \left( \tilde{\mathbf{B}}(\mathbf{x}) - \frac{1}{3}\text{tr}(\tilde{\mathbf{B}}(\mathbf{x}))\mathbf{I} \right), \quad (3)$$

where  $\mu$  denotes the viscosity,  $G$  the modulus of transverse elasticity,  $\mathbf{I}$  the unit tensor, and  $\mathbf{D}(= \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T))$  the strain rate tensor. The operator  $\text{tr}(\dots)$  stands for the tensor trace. The quantities  $\phi_s$  and  $\tilde{\mathbf{B}}$  denote the local volume fraction of solid and the modified left Cauchy-Green deformation tensor<sup>14</sup>), respectively, which obey the transport equations

$$\partial_t \phi_s + \mathbf{v} \cdot \nabla \phi_s = 0, \quad (4)$$

$$\partial_t \tilde{\mathbf{B}} + \mathbf{v} \cdot \nabla \tilde{\mathbf{B}} = \mathbf{L} \cdot \tilde{\mathbf{B}} + \tilde{\mathbf{B}} \cdot \mathbf{L}^T, \quad (5)$$

where  $\mathbf{L}(= \nabla \mathbf{v})$  denotes the velocity gradient tensor.

The full Eulerian FSI description is well-suited to the voxel-based geometry. A set of the

voxel data is converted into the initial solid volume fraction  $\phi_{s0}$  (here, the subscript 0 stands for the initial quantity). Once the  $\phi_{s0}$  field is provided, one can carry out the FSI simulation without mesh generation/reconstruction procedure. The validity of the full Eulerian simulation method based on the equation set (1)–(5) has been established in various manners<sup>13),14),17</sup>), including comparisons with well validated results using a finite element method for complicated flow/deformation.

### 2.2 Artificial compressibility method with adaptive parameters

The pressure is defined to satisfy a relation  $\langle p \rangle_\Omega = 0$ , here  $\langle \dots \rangle_\Omega$  stands for the volume average over the whole computational domain. The time-stepping algorithm to update the variables at the  $(N+1)$ -th time level from the  $N$ -th time level follows the unprojection step:

$$\mathbf{v}^* = \mathbf{v}^N + (\Delta t)\rho^{-1} \{ -\nabla p^N + \mathbf{F}(\mathbf{v}^N, \tilde{\mathbf{B}}^N, \phi_s^N) \}, \quad (6)$$

and the projection step:

$$p^{N+1} = p^N + \delta p, \quad (7)$$

$$\mathbf{v}^{N+1} = \mathbf{v}^* - (\Delta t)\rho^{-1} \nabla(\delta p), \quad (8)$$

where the superscript  $*$  represents the unprojected quantity, and  $\mathbf{F}$  is the summation of all the terms except for the pressure gradient term in the momentum equation (2). If one follows the SMAC algorithm<sup>23</sup>), which aims at satisfying the divergence-free condition ( $\nabla \cdot \mathbf{v}^{N+1} = 0$ ), the incremental pressure correction  $\delta p$  is given by the solution to the Poisson equation

$$\nabla^2 \delta p = (\Delta t)^{-1} D^*, \quad (9)$$

where  $D(= \rho \nabla \cdot \mathbf{v})$  is the divergence of the mass flux. To numerically solve the partial differential equation (9) together with the boundary conditions, the iterative method is usually employed, and then the core-to-core communication is required in the parallel computation. In the present study, instead, we follow the pressure evolution of the revived ACM<sup>22</sup>) written in an algebraic form

$$\delta p = -\beta^2 (\Delta t) (\gamma p^N + D^*), \quad (10)$$

where  $\beta$  denotes the pseudo acoustic speed, and  $\gamma$  the relaxation coefficient, of which the positive value is effective in the suppression of the checkerboard instability. Although the solenoidal condition (1) is not perfectly satisfied in the ACM, we try to approximate (1) as exactly as possible<sup>24</sup>). The model parameters  $\beta$  and  $\gamma$  are spatially uniform and determined dynamically at each time step. The mass flux divergence at the  $(N+1)$ -th time level is written

as

$$D^{N+1} = D^* + \beta^2(\Delta t)^2 \nabla^2 D^* + \gamma \beta^2(\Delta t)^2 \nabla^2 p^N, \quad (11)$$

Together with a constrained condition of the non-negative relaxation coefficient  $\gamma \geq 0$ , the mean-square of the mass flux divergence at the  $(N+1)$ -th time level,  $\langle (D^{N+1})^2 \rangle_\Omega$ , is minimized provided that

$$\beta = \frac{1}{(\Delta t)} \sqrt{\frac{A_1}{A_2}}, \quad \gamma = 0 \text{ if } \frac{-(A_1 B_2 + A_2 B_1)}{A_2 A_3 - B_2^2} < 0,$$

$$\beta = \frac{1}{(\Delta t)} \sqrt{\frac{A_1 A_3 + B_1 B_2}{A_2 A_3 - B_2^2}}, \quad \gamma = \frac{-(A_1 B_2 + A_2 B_1)}{A_1 A_3 + B_1 B_2} \text{ otherwise.} \quad (12)$$

where

$$\begin{aligned} A_1 &= -\langle D^* \nabla^2 D^* \rangle_\Omega, \quad A_2 = \langle (\nabla^2 D^*)^2 \rangle_\Omega, \\ A_3 &= \langle (\nabla^2 p^N)^2 \rangle_\Omega, \quad B_1 = \langle D^* \nabla^2 p^N \rangle_\Omega, \\ B_2 &= \langle (\nabla^2 D^*) (\nabla^2 p^N) \rangle_\Omega. \end{aligned} \quad (13)$$

In the parallel computation, the advocated ACM requires the *allreduce* operation only to find  $A_1$ ,  $A_2$ ,  $A_3$ ,  $B_1$  and  $B_2$  in (13). Thus, it is likely to considerably reduce the amount of the global communication as compared with the standard Poisson equation-based approach, which requires the estimation of the residual error at each iteration step.

The ACM-based solution inevitably has the *pseudo* compressibility since the solenoidal condition (1) is more or less violated. Here, its level is discussed. Let us introduce  $U_{\max}$  as the maximum velocity in the system and  $(\Delta x)$  as the side length of the cubic grid. It should be noticed that from the analytical study<sup>24)</sup>, the pseudo Mach number  $Ma (= U_{\max}/\beta)$ , which indicates the level of the pseudo compressibility, has the upper limit  $2\sqrt{3}\eta$ , here  $\eta = U_{\max}(\Delta t)/(\Delta x)$  denotes the Courant-Friedrichs-Lewy (CFL) number. Therefore, if the CFL number is sufficiently smaller than unity, the computed velocity field is guaranteed to be nearly incompressible.

### 2.3 Initial and boundary conditions

We will address the three-dimensional problem of the multiple neo-Hookean particle motion in a Poiseuille flow. We shall restrict our attention to the system bounded by the bottom ( $y = 0$ ) and top ( $y = L_y$ ) plates and periodic in  $x$  (streamwise) and  $z$  (spanwise) directions with the periodicity of  $L_x$  and  $L_z$ , respectively. The system is supposed in stationary equilibrium (i.e.  $\mathbf{v}_0 = p_0 = 0$  and  $\tilde{\mathbf{B}}_0 = \phi_{s0}^{1/2} \mathbf{I}$ ) before the driving pressure is imposed. The initial and neutral shape of the particle is discoid bicon-

cave. In temporally updating  $\mathbf{v}$ ,  $p$ ,  $\phi_s$ , and  $\tilde{\mathbf{B}}$ , we impose the no-slip condition on the plates.

### 2.4 General descriptions

We follow a conventional staggered grid arrangement, where the velocity components are located on the cell faces, the pressure, the solid volume fraction and the diagonal components of  $\tilde{\mathbf{B}}$  are on the cell centroid, and the non-diagonal components of  $\tilde{\mathbf{B}}$  are on the cell sides. The spatial derivatives are approximated by the second-order central differences, except for those of the advection terms, to which the fifth-order Weighted Essentially Non-Oscillating (WENO5) scheme<sup>25)</sup> is applied. To integrate the equations in time, we employ the first-order explicit Euler scheme.

### 2.5 Parallelization

The solver code is written in Fortran 77. It is implemented into an object-oriented framework *V-Sphere*<sup>26)</sup>, which includes class libraries to facilitate the software development especially for massively parallel computations. The framework *V-Sphere* provides common functions such as data management, parsing of various parameters described by XML language, and file I/O.

We utilize Massively Parallel Cluster (MPC; Fujitsu PRIMERGY RX200S5 (1048 nodes)) of RIKEN Integrated Cluster of Clusters (RICC) with a theoretical peak performance of 98.2 TFlops. Each computational node has Intel Xeon 5570 (quad SMP processor chip 2.93GHz, memory bandwidth: 0.54Byte/Flop)  $\times 2$  with a memory of 12GB and a harddisk of 500GB (RAID-0, SAS). In total, there are 8,384 cores. The system is equipped with InfiniBand: X4 DDR (Qlogic SilverStorm 9024  $\times$  60 and 9120  $\times$  2) for inter-node networking via two-way communication with a performance of 16Gbps per way. The network topology is a fat-tree type (bisection bandwidth: 240GB/s). There are 60 leaf switches connecting with 20 computational nodes for each, and two spine switches connecting with all the leaf switches.

We employ a domain decomposition method for the whole computational domain given as a set of regularly divided cubic cells using a MPI library (Fujitsu MPI). Since the spatial derivative of any quantity is discretized in the finite difference manner, a nearest-neighbor communication at the boundary of the decomposed domain is essential for referring to a datum at the adjacent node. To suppress the overhead due to the communication rise-time, we take

the nearest-neighbor communication for all the quantities all at once. The total amount of the nearest-neighbor communication is dependent upon the number of grid points on the interfacial area of the decomposed domain and upon the number of points of the finite difference stencils. To evaluate the volume averaged quantities in (13) or the summation of the residual error during an iterative computation, the global communication is required. The global communication is efficiently controlled by the RICC system, in which the network topology is taken into account.

Let us estimate the amount of the nearest-neighbor and global communications at each time step based on the proposed ACM. For comparison, we also consider the standard semi-implicit algorithm, in which the pressure Poisson equation (9) is solved by means of a four-color SOR method, and the deviatoric Cauchy stress term involved in (2) is temporally approximated by the second-order Crank-Nicolson method.

### 2.5.1 Nearest-neighbor communication

Let  $N_{dx} \times N_{dy} \times N_{dz}$  be the number of grid points in the decomposed domain on each core. The number of grid points on the node boundary is given by  $2(N_{dx}N_{dy} + N_{dx}N_{dz} + N_{dy}N_{dz})$ . Let  $N_{st}$  be the number of points of the finite difference stencil in each direction. For the second-order central difference method and the WENO5 scheme,  $N_{st} = 3$  (for the velocity components ( $\times 3$ ) and the pressure ( $\times 1$ )) and  $N_{st} = 7$  (for the solid volume fraction ( $\times 1$ ) and the left Cauchy-Green deformation tensor components ( $\times 6$ )), respectively. In describing the spatial derivative, the number of nearest-neighbor communications is estimated as  $(N_{st} - 1)(N_{dx}N_{dy} + N_{dx}N_{dz} + N_{dy}N_{dz})$  for each quantity. Hence, in the explicit ACM method, the total number is  $50(N_{dx}N_{dy} + N_{dx}N_{dz} + N_{dy}N_{dz})$  for all the quantities.

In the semi-implicit algorithm, repetitive nearest-neighbor communications are additionally needed in the iteration process. Let  $N_C$  and  $N_S$  be the numbers of iterations for the Crank-Nicolson and SOR methods, respectively. In describing the divergence of the Cauchy stress tensor involved in (2), the number of stencil points is  $N_{st} = 3$  for the Cauchy stress tensor components ( $\times 6$ ) and the velocity components ( $\times 3$ ). Hence, the number of nearest-neighbor communications in the iteration for the deviatoric stress term is estimated as  $18N_C(N_{dx}N_{dy} +$

$N_{dx}N_{dz} + N_{dy}N_{dz})$ . In the four-color SOR method, the decomposed domain is divided into  $2 \times 2 \times 2$  sub-domains. Hence, the number of grid points on the boundary in each sub-domain is  $(N_{dx}N_{dy} + N_{dx}N_{dz} + N_{dy}N_{dz})/2$ . In solving the Poisson equation (9), since the number of stencil points is  $N_{st} = 3$  for  $\delta p$  and the number of the sub-domains is 8, the number of nearest-neighbor communications is estimated as  $4N_S(N_{dx}N_{dy} + N_{dx}N_{dz} + N_{dy}N_{dz})$ .

It should be noticed that when one applies the multi-color SOR method, the parallel efficiency can be considerably enhanced by means of an overlapping of communication with computation (i.e. a hiding communication) conducted through an asynchronous communication technique. Nevertheless, in consideration that a relation  $50 \ll 18N_C + 4N_S$  usually holds and its right-hand-side generally becomes greater with increasing the system size, the amount of the nearest-neighbor communication in the present ACM is much less than that in the semi-implicit algorithm.

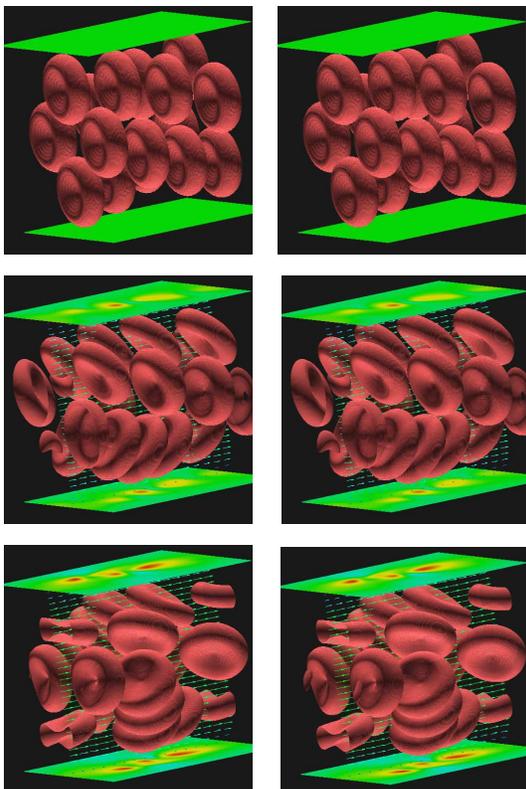
### 2.5.2 Global communication

In the ACM, the number of *allreduce* operations is 6, corresponding to the number of the volume averaged quantities in (13) and the pressure to be zero when being volume averaged. On the other hand, in the semi-implicit algorithm, the *allreduce* operation is not involved. However, the evaluation of the overall residual error is required unless the convergence speed is known. In practical simulations, although the error estimation is not always needed at each iteration step, the total number of *allreduce* operations for the Crank-Nicolson and SOR methods is likely to be more than 6, corresponding to the number of operations in the ACM.

## 3. Results and discussion

### 3.1 Overview

In the present study, for convenience in characterizing the elasticity of solid, all quantities are non-dimensionalized using the density, the viscosity, and the driving pressure gradient. Thus, the equation set (1)–(5) in a dimensional form is solved together with  $\rho = \mu = -\Delta P/L_x = 1$ . We exemplify simulation results involving 16 biconcave discoid particles with the modulus of  $G = 50$ . The computational extent is  $L_x \times L_y \times L_z = 7.2 \times 7.2 \times 7.2$ , and the number of grid points is  $N_x \times N_y \times N_z = 128 \times 128 \times 128$ . The numerical simulations are

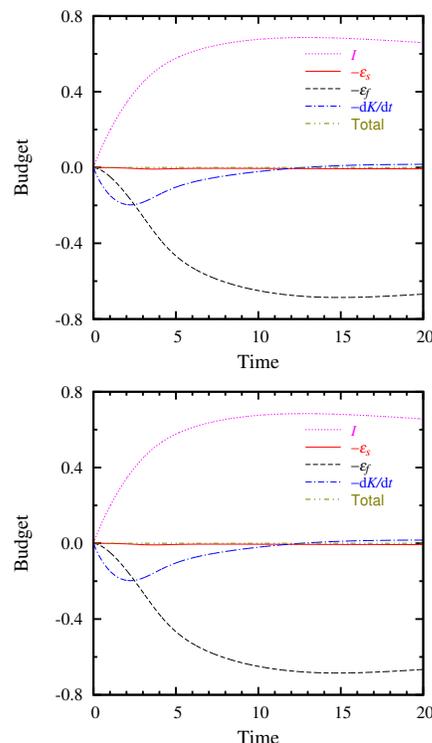


**Fig. 1** The snapshots of discoid biconcave particles of the neo-Hookean material in a three-dimensional Poiseuille flow. The top, middle, and bottom panels show the particle interfaces at  $t = 0$ ,  $t = 5$ , and  $t = 20$ , respectively. The colors on the walls indicate the shear stress distributions. The left and right panels show the results based on the SMAC method using FFT-TDMA and those on the present ACM, respectively.

carried out using a single node. For comparison, two kinds of simulations are performed: one is based on the SMAC algorithm, in which the Fast Fourier Transform (FFT) in streamwise ( $x$ ) and spanwise ( $z$ ) directions and the Tri-Diagonal Matrix Algorithm (TDMA) in wall-normal ( $y$ ) direction are applied to exactly solving the pressure Poisson equation (9) in a finite difference form<sup>27</sup>, while the other is based on the present ACM. In both the methods, the first-order explicit Euler scheme is applied to integrating the equations in time with a time increment of  $\Delta t = 5 \times 10^{-4}$ , which is determined to fully capture the elastic wave. The Reynolds number using a flow rate in a fully developed state and the channel height  $L_y$  is about 10. The CFL number based on the maximum advection speed is about 0.018, while that based

on the elastic wave speed  $\sqrt{G/\rho}$  is 0.063.

The particle position and orientation are shown in Figure 1. The particles deform and translate in the downstream. As the time goes on, they rotate and tend to be more mixed, revealing somehow complicated behavior. There are no significant discrepancies in the particle position and shape between the results based on the SMAC algorithm and present ACM.



**Fig. 2** The budget of the kinetic-energy transport in the Poiseuille flow containing 16 neo-Hookean particles. (a) top panel: SMAC method using FFT-TDMA; (b) bottom panel: present ACM.

To check whether an energy conservation is correctly captured, we examine a budget of the overall kinetic-energy transport written as

$$-\frac{dK}{dt} + I - \varepsilon_s - \varepsilon_f = 0, \quad (14)$$

where  $K(= \langle \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \rangle_\Omega)$ ,  $I(= \langle v_x (-\Delta P/L) \rangle_\Omega)$ ,  $\varepsilon_s(= \langle \phi_s G \mathbf{B}' : \mathbf{D}' \rangle_\Omega)$ , and  $\varepsilon_f(= \langle 2\mu \mathbf{D}' : \mathbf{D}' \rangle_\Omega)$  denote the kinetic-energy, the energy input rate, the strain energy rate, and the kinetic-energy dissipation rate, respectively. Figure 2 shows the time history of each contribution in the left hand side of (14). The temporal profiles of the individual terms in the present ACM are in good agreement with those in the SMAC

method. The double-chained curve in Figure 2 shows the summation of the left-hand-side terms of (14). Its absolute value is much smaller than the variation of the contributions of the individual terms. Therefore, the system is well conserved during the simulation in view of the energy balance, and the energy exchange between the fluid and solid phases via the solid deformation is reasonably guaranteed.

### 3.2 Weak scaling

Here, to analyze the impact of the number of cores  $M$  on the parallel performance, we perform weak scaling studies. For each core, the computational extent and the number of grid points are fixed at  $L_x \times L_y \times L_z = 3.6 \times 3.6 \times 3.6$  and  $N_x \times N_y \times N_z = 64 \times 64 \times 64$ , respectively. Each decomposed domain contains two biconcave discoid particles. The material properties are the same as those in §3.1. The number of cores utilized is an integer power of 2 ranging from  $M = 1$  to  $M = 8,192$ . For comparison, not only the simulation based on the present ACM but also that on the SMAC algorithm using the four-color SOR method as stated in §2.5 are performed.

In the four-color SOR method, the acceleration coefficient during the iteration is set to 1.6. Usually, with increasing the number of cores, the decay of the overall residual error becomes slower, and thus the number of iterations becomes larger. However, to perform the weak scalability test, one should fix the computational cost per core. Therefore, firstly, we perform a computation using a single core, and monitor the ratio of the root-mean-square of the residual error of the pressure Poisson equation at each iteration step to that at the initial step. Then, we judge the convergence from the ratio being smaller than a criterion  $\epsilon$ , and determine the number of iterations  $N_S$ , which is fixed for all the runs, irrespective of the number of cores. To examine the impact of the communication in the iterative Poisson equation solver on the performance, we vary convergence criteria at  $\epsilon = 10^{-3}$  and  $\epsilon = 10^{-4}$ . Moreover, we apply different communication techniques, namely, synchronous and asynchronous ones. The decay of the residual error in the Crank-Nicolson method is rather faster, and the number of iterations is set to  $N_C = 20$  at each time step, at which the residual error is much less than  $10^{-6}$ .

In  $50(N_{dx}N_{dy} + N_{dx}N_{dz} + N_{dy}N_{dz})$  nearest-neighbor communications in updating the

quantities explained in §2.5.1 and the *allreduce* operation in the present ACM (corresponding to the remaining communication), the synchronous communication is employed.

For the SOR method, the time increment is set to  $(\Delta t) = 6.25 \times 10^{-5}$ , while for the present ACM, it is set to the one fifth, namely,  $(\Delta t) = 1.25 \times 10^{-5}$ , since it should be small enough to avoid the violation of the mass conservation<sup>24</sup>). The simulation conditions related to the performance are listed in Table 1. The total numbers of iterations with the criteria  $\epsilon = 10^{-3}$  and  $\epsilon = 10^{-4}$  are fixed as shown in Table 1.

**Table 1** Simulation conditions in the weak scalability tests for the SMAC/four-color SOR method and for the present ACM with a  $64 \times 64 \times 64$  mesh per core.

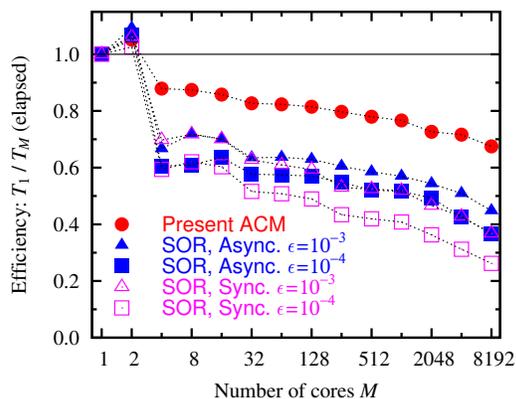
	SOR		ACM
	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	
# of time steps	10	10	50
$N_S$ in total	1,000	2,200	-
$N_C$ in total	200	200	-

**Table 2** Elapsed wall time  $T_M$  (sec) in the weak scalability tests with a  $64 \times 64 \times 64$  mesh per core.

Cores ( $M$ )	SOR ( $\epsilon = 10^{-3}$ )		SOR ( $\epsilon = 10^{-4}$ )		ACM
	Sync.	Async.	Sync.	Async.	
1	24.94	24.30	26.81	27.26	21.57
2	23.49	22.25	26.13	25.50	20.52
4	35.76	36.44	45.16	44.96	24.55
8	34.77	33.81	43.20	44.78	24.68
16	35.62	34.66	44.44	42.84	25.15
32	39.75	38.28	51.95	47.25	26.09
64	40.92	38.17	52.79	47.45	26.20
128	42.01	38.61	54.77	47.80	26.47
256	47.08	40.18	61.80	49.79	27.07
512	47.56	41.52	63.91	52.28	27.67
1,024	48.09	42.65	65.68	52.86	28.15
2,048	53.54	44.73	73.88	55.26	29.70
4,096	58.33	47.76	85.90	64.08	30.11
8,192	67.84	54.23	102.20	74.04	31.96

For various number of cores  $M$ , the elapsed time  $T_M$  and the relative performance to the base one using one core  $M = 1$  (i.e.  $T_{M=1}/T_M$ ) are reported in Table 2 and in Figure 3, respectively. It is seen from Table 2 that in the SOR method, the asynchronous communication technique is substantially effective for reducing the run time as compared with the results using the synchronous communication. Further notably, for all  $M$ , the elapsed times in the ACM are considerably shorter than those in the SOR method tuned up with the asynchronous

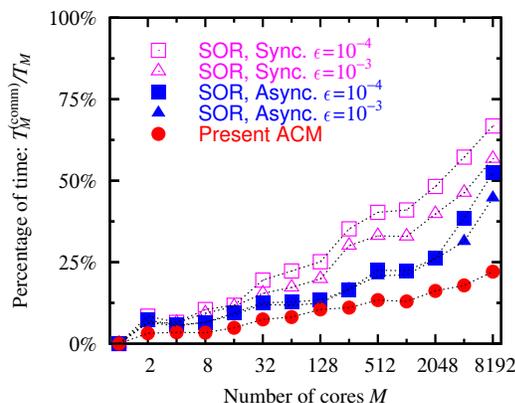
communication, indicating the advantage in the reduction of the communication overweighs the disadvantage in excessive number of time steps. As shown in Figure 3, the plots reveal somehow strange behaviors for  $M \leq 8$ , because the system comprised of multi-core processors conducts only the core-to-core communication inside the node without node-to-node communication. In the SOR method, the larger number of iterations owing to the smaller convergence criterion  $\epsilon$  leads to the longer run time (Table 2) and the less weak scaling performance (Figure 3), indicating that one obtains worse parallel performance when seeking to improve the simulation accuracy. As shown in Figure 3, the scalability in the present ACM is remarkably better than those in the SOR method.



**Fig. 3** The efficiency, corresponding to the inverse of elapsed time  $T_M$  normalized by that at one core  $T_1$ , versus the number of cores  $M$ , showing the weak scaling performance with a  $64 \times 64 \times 64$  mesh per core. The filled circles indicate the results based on the present ACM. The triangles and squares indicate the results based on the SMAC algorithm using the four-color SOR method with the criteria  $\epsilon = 10^{-3}$  ( $\sum N_S = 1,000$ ) and  $\epsilon = 10^{-4}$  ( $\sum N_S = 2,200$ ), respectively, and the filled and open symbols of them show the results using the synchronous and asynchronous communications in solving the Poisson equation, respectively.

To demonstrate the communication overhead, the ratio of the communication time  $T_M^{(\text{comm})}$  to the elapsed time  $T_M$  in percent is reported in Figure 4. The communication time fraction  $T_M^{(\text{comm})}/T_M$  in the ACM is much lower than those in the SOR method. The ACM requires only 22% or less of the elapsed time for

communication up to 8,192 cores. It should be noticed that with increasing the number of cores  $M$ , the amount of the nearest-neighbor communication per core is fixed in the weak scalability test, but the increase in the amount of the global communication detailed in §2.5.2 is inevitable for the present ACM, which may have no small impact on the positive correlation between  $T_M^{(\text{comm})}/T_M$  and  $M$  shown in Figure 4.



**Fig. 4** Weak scaling performance in terms of communication time  $T_M^{(\text{comm})}$  as a percentage of the elapsed time  $T_M$ . The meanings of the symbols are the same as those in Figure 3.

### 3.3 Strong scaling

Here, we perform strong scaling studies at a fixed problem size using thousands of cores. The computational extent and the number of grid points are set to  $L_x \times L_y \times L_z = 115.2 \times 57.6 \times 57.6$  and  $N_x \times N_y \times N_z = 2,048 \times 1,024 \times 1,024$  (i.e. 2,147,483,648 grid points in total), respectively. The whole domain contains 16,384 particles. The material properties are the same as those in §3.1. Similar to the weak scalability tests in §3.2, we make comparisons between the parallel performances of the present ACM and the SMAC algorithm using the four-color SOR method. The time increments for the SOR method and for the ACM are respectively set to the same as those in §3.2. The SOR process is iterated until the residual error relative to that at the initial iteration step becomes less than the criterion  $\epsilon$ . The total numbers of iterations determined thereby and the number of time steps are listed in Table 3.

The run on 1,024 cores is used as the base to evaluate the strong scaling performance. The

**Table 3** Simulation conditions in the strong scalability tests for the SMAC/four-color SOR method and for the present ACM with a  $2,048 \times 1,024 \times 1,024$  mesh in total.

	SOR		ACM
	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	
# of time steps	10	10	50
$N_S$ in total	1,058	6,128	-
$N_C$ in total	70	70	-

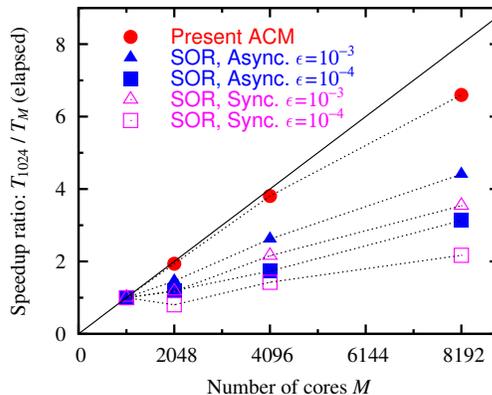
elapsed time  $T_M$  and the speedup ratio defined as  $T_{M=1024}/T_M$  are reported in Table 4 and in Figure 5, respectively. Since a great effort has been made to tune the Poisson equation solver, the parallel performance based on the SMAC algorithm is substantially enhanced by introducing the asynchronous communication technique even at thousands of cores as shown in Figure 5. Further remarkably, the profile of the present ACM exhibits more excellent scalability, and the parallel efficiency of 82% is maintained up to  $M = 8,192$ . It is seen from Table 4 that at  $M = 1,024$ , the run time in the ACM is longer than that in the SOR method with the criterion  $\epsilon = 10^{-3}$  using the asynchronous communication, while at the larger number of cores  $M \geq 2,048$ , the run time in the ACM becomes shorter, on which the better scalability is reflected.

**Table 4** Elapsed wall time  $T_M$  (sec) in the strong scalability tests with a  $2,048 \times 1,024 \times 1,024$  mesh in total

Cores ( $M$ )	SOR ( $\epsilon = 10^{-3}$ )		SOR ( $\epsilon = 10^{-4}$ )		ACM
	Sync.	Async.	Sync.	Async.	
1,024	187.60	180.87	437.05	398.11	210.86
2,048	159.83	124.05	543.48	334.10	108.87
4,096	86.94	69.09	306.38	229.73	55.37
8,192	53.02	41.05	201.40	127.16	31.96

#### 4. Conclusion

We proposed the scalable full explicit Eulerian finite difference method for solving Fluid-Structure Interaction (FSI) problems, which reduces the amount of core-to-core communications required for the iterative computation, and easily makes the computational load on each core equivalent. We performed the large-scale FSI simulations of the channel flow with biconcave discoid particles, which replicate the shape of Red Blood Cells (RBCs). The present full explicit Eulerian solver was proven to be excellently scalable in terms of the weak scaling and also of the strong scaling with  $O(10^9)$  grid points up to 8,192 cores, as compared with



**Fig. 5** The speedup ratio, corresponding to the inverse of elapsed time  $T_M$  normalized by that at 1,024 cores  $T_{1024}$ , versus the number of cores  $M$ , showing the strong scaling performance with a  $2,048 \times 1,024 \times 1,024$  mesh in total. The meanings of the symbols are the same as those in Figure 3. The solid line indicates the linear scaling.

the conventional solver using the multi-color SOR method well tuned up for massively parallel computations.

It should be emphasized that since the multi-color SOR method with the asynchronous communication is quite scalable in view of parallelization, we believe that the comparisons in terms of the weak and strong scalings therewith are sufficiently meaningful. However, the multi-color SOR method is more time-consuming than faster convergence approaches such as a multigrid method. To further discuss the parallel efficiency in terms of the peak performance ratio as well as the weak/strong scaling, a multigrid solver is currently being tuned up for massively parallel computations, and will be applied to performance tests for comparison.

As a future perspective, the present numerical approach will be applied to Petascale computations of blood flows involving RBCs and platelets in order to improve the understanding of the initial process of the thrombus formation. For this purpose, better serial and parallel efficiencies are desirable from the numerical point of view. The OpenMP-MPI hybrid parallelization has been implemented into the code, and currently the performance tuning is ongoing.

#### 5. Acknowledgments

The present simulations were conducted using the RIKEN Integrated Cluster of Clusters (RICC). A part of the calculation was done

for the performance evaluation process on the Next-Generation Supercomputer. The authors are grateful to Kenji Ono (RIKEN) for fruitful discussion. The authors thank the referees for their useful comments and suggestions to improve the paper. This research was supported by Research and Development of the Next-Generation Integrated Simulation of Living Matter, a part of the Development and Use of the Next-Generation Supercomputer Project of the Ministry of Education, Culture, Sports, Science and Technology (MEXT). K.S. was supported in part from MEXT (Grant-in-Aid for Young Scientist (B), No. 21760120).

### References

- 1) T. Washio, J. Okada and T. Hisada, "A parallel multilevel technique for solving the biodomain equation on a human heart with Purkinje fibers and a torso model", *SIAM J. Sci. Comput.*, 30: 2855–2881, 2008.
- 2) O. Sahni, M. Zhou, M.S. Shephard and K.E. Jansen, "Scalable implicit finite element solver for massively parallel processing with demonstration to 160K cores", *Proc. 2009 ACM/IEEE Conf. on Supercomputing*, Portland, Oregon, doi: 10.1145/1654059.1654129.
- 3) A. Rahimian, I. Lashuk, S.K. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin and G. Biros, "Petascale direct numerical simulation of blood flow on 200K cores and heterogeneous architectures", *Proc. 2010 ACM/IEEE Conf. on Supercomputing*, New Orleans, doi: 10.1109/SC.2010.42.
- 4) A. Peters, S. Melchionna, E. Kaxiras, J. Lätt, H. Sircar, M. Bernaschi, M. Bisson and S. Succi, "Multiscale simulation of cardiovascular flows on the IBM Blue Gene/P: full heart-circulation system at near red-blood cell resolution", *Proc. 2010 ACM/IEEE Conf. on Supercomputing*, New Orleans, doi: 10.1109/SC.2010.33.
- 5) C.W. Hirt, A.A. Amsden and J.L. Cook, "An arbitrary Lagrangian-Eulerian computing method for all flow speeds", *J. Comput. Phys.*, 14: 227–253, 1974.
- 6) T.E. Tezduyar, M. Behr and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces - The deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests", *Comput. Methods Appl. Mech. Engrg.*, 94, 339–351, 1992.
- 7) C.S. Peskin, "Flow patterns around heart valves: a numerical method", *J. Comput. Phys.*, 10: 252–271, 1972.
- 8) P.A.A. Van Hoogstraten, P.M.A. Slaats and F.P.T. Baaijens, "A Eulerian approach to the finite element modelling of neo-Hookean rubber material", *Appl. Sci. Res.*, 48: 193–210, 1991.
- 9) C. Liu and N.J. Walkington, "An Eulerian description of fluids containing visco-elastic particles", *Arch. Ration. Mech. Anal.*, 159: 229–252, 2001.
- 10) T. Dunne, "An Eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation", *Int. J. Numer. Meth. Fluids*, 51: 1017–1039, 2006.
- 11) G.-H. Cottet, E. Maitre and T. Milcent, "Eulerian formulation and level set models for incompressible fluid-structure interaction", *Math. Modelling and Numer. Anal.* 42: 471–492, 2008.
- 12) K. Sugiyama, S. Ii, S. Takeuchi, S. Takagi, and Y. Matsumoto, "Full Eulerian simulations of biconcave neo-Hookean particles in a Poiseuille flow", *Comput. Mech.*, 46: 147–157, 2010.
- 13) N. Nagano, K. Sugiyama, S. Takeuchi, S. Ii, S. Takagi and Y. Matsumoto, "Full Eulerian finite-difference simulation of fluid flow in hyperelastic wavy channel", *J. Fluid Sci. Tech.*, 5: 475–490, 2010.
- 14) K. Sugiyama, S. Ii, S. Takeuchi, S. Takagi, and Y. Matsumoto, "A full Eulerian finite difference approach for solving fluid-structure coupling problems", *J. Comput. Phys.*, 230: 596–627, 2011.
- 15) S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi and Y. Matsumoto, "An implicit full Eulerian method for the fluid-structure interaction problem", *Int. J. Numer. Meth. Fluids*, 65: 150–165, 2011.
- 16) K. Sugiyama, N. Nagano, S. Takeuchi, S. Ii, S. Takagi and Y. Matsumoto, "Particle-in-cell method for fluid-structure interaction simulations of neo-Hookean tube flows", *Theor. Appl. Mech. Jpn.*, 59: 245–256, 2011.
- 17) S. Takagi, K. Sugiyama, S. Ii and Y. Matsumoto, "A review of full Eulerian methods for fluid structure interaction problems", *J. Appl. Mech.*, 2011, published online, doi:10.1115/1.4005184.
- 18) S. Ii, X. Gong, K. Sugiyama, J. Wu, H. Huang and S. Takagi, "A full Eulerian fluid-membrane coupling method with a smoothed volume-of-fluid approach", *Comm. in Comput. Phys.*, 2011, accepted.
- 19) S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi, Y. Matsumoto and F. Xiao, "An interface capturing method with a continuous function: the THINC method with multi-dimensional reconstruction", *J. Comput. Phys.* 2011, accepted,

- doi:10.1016/j.jcp.2011.11.038.
- 20) A.J. Chorin, “A numerical method for solving incompressible viscous flow problems”, *J. Comput. Phys.*, 2: 12–26, 1967.
  - 21) S.E. Rogers and D. Kwak, “Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations”, *AIAA J.*, 28: 253–262, 1990.
  - 22) T. Ohwada and P. Asinari, “Artificial compressibility method: asymptotic numerical method for incompressible Navier-Stokes equations”, *J. Comput. Phys.*, 229: 1698–1723, 2010.
  - 23) A.A. Amsden and F.H. Harlow, “A simplified MAC technique for incompressible fluid flow calculations”, *J. Comput. Phys.*, 6: 322–325, 1970.
  - 24) K. Sugiyama, S. Ii, S. Takagi, and Y. Matsumoto, “Artificial compressibility method with adaptive parameters”, in preparation.
  - 25) G.-S. Jiang and C.-W. Shu, “Efficient implementation of weighted ENO scheme”, *J. Comput. Phys.*, 126: 202–228, 1996.
  - 26) K. Ono, T. Tamaki and H. Yoshikawa, “Development of a framework for parallel simulators with various physics and its performance”, *Lecture Notes in Comput. Sci. Engrg.*, 67: 9–18, 2009.
  - 27) M. Sbragaglia and K. Sugiyama, “Boundary induced nonlinearities at small Reynolds numbers”, *Physica D*, 228: 140–147, 2007.
-