Regular Paper

Massively Parallel Computing of Novel Fluid-Structure Interaction Solver on the K Computer

Kazuyasu Sugiyama[1,2,a)] Yasuhiro Kawashima[3] Shigeho Noda[1] Satoshi Ii[4] Hiroshi Koyama[5]
Shu Takagi[6,2] Yoichiro Matsumoto[6] Ryutaro Himeno[2]

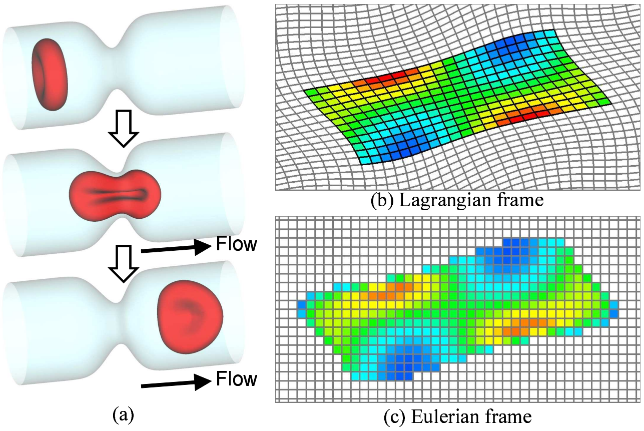
Received: October 5, 2012, Accepted: xx xx, xxxx

Abstract: An efficient and scalable numerical method for massively parallel computing of fluid-structure interaction systems has been developed for biomedical applications. To facilitate the treatment of complex geometry, a full Eulerian method is employed to couple the incompressible motions of fluid and hyperelastic materials. Instead of implicitly solving the pressure Poisson equation, a novel artificial compressibility method with adaptive parameters, which are determined to guarantee the computed field to be nearly incompressible, is employed. In both weak and strong scaling tests, the developed solver attains excellent scalability on the K computer. A sustained performance of 4.54 Pflops (42.7% of peak) has been achieved for a microchannel flow involving more than 5 million deformable bodies with 663,552 compute cores. We study arteriole blood flows to gain insight into dynamic interactions among motions of plasma and blood cells.

Keywords: Fluid-Structure Interaction, Artificial Compressibility Method, Finite Difference Method, Blood Flow

1. Introduction

Fluid-Structure Interaction (FSI) phenomena appear in a number of biological systems. Thereamong, blood flow plays important roles in several functions to sustain life. The interplay among its multi-physics nature, the complex geometry, and the suspension of blood cells gives rise to phenomenologically rich behavior. Recent advances in high-performance computing and numerical methods have promoted interest in hemodynamic simulations [1–3], and encouraged large-scale computations [4–6]. In this paper, we shall focus on deformable body motions, of which the improved understanding would gain insight into an initial thrombus formation to arrest the bleeding in hemostasis mechanism, and an exaggerated platelet aggregation to cause thrombosic vessel occlusion leading to myocardial and cerebral infarctions [7, 8]. In a microcirculation system with a sub-millimetric scale vessel, a Red Blood Cell (RBC) subjected to a shearing or squeezing fluid motion undergoes large deformation (Fig. 1(a)). Such a distinctive deformability and a dense particulate flow nature dictate rheological properties, which are relevant to transport phenomena and hemostasis processes [9]. In particlar, on the thrombus formation, the significance of the hydrodynamic effect has become commonly recognized [10–13]. The related numerical studies include simulations of three-dimensional motions of two platelets in a shear flow [14], and two-dimensional [15] and three-dimensional [16] blood flows (in vessels of 50μm height and 20μm diameter, respectively) with a number of platelets and RBCs. In order to obtain practically important knowledge of multi-scale dynamics, it is essential to perform large-scale computation at the realistic scale since the system is hydrodynamically and geometrically nonlinear in size. To tackle the challenging issue, our strategy in formulation and algorithm will be outlined below.



(a)

(b) Lagrangian frame

(c) Eulerian frame

Fig. 1 Schematic figures explaining the present simulations of the FSI problems. (a): typical simulation results revealing large deformation of a RBC. (b) and (c): on the difference between the solid deformation descriptions of the Lagrangian and Eulerian approaches. In the Lagrangian method (b), the relative displacement between adjacent material points from the reference to current configurations quantifies the deformation level. In the Eulerian method (c) [19], the left Cauchy-Green deformation tensor $\boldsymbol{B}$ defined on fixed grid points quantifies the deformation level.

[1] Advanced Center for Computing and Communication, RIKEN, 2-1 Hirosawa, Wako-shi, Saitama 351–0106, Japan
[2] RIKEN Computational Science Research Program
[3] CAE Solution Center, Fujitsu Systems East Ltd.
[4] Department of Mechanical Science and Bioengineering, Osaka University
[5] Computational Materials Science Unit, National Institute for Materials Science
[6] School of Engineering, The University of Tokyo
[a)] kazuyasu.sugiyama@riken.jp

When addressing moving interface problems, one has preferably employed a Lagrangian method using a moving finite element mesh (Fig. 1(b)). For relatively simple systems, the Lagrangian method is satisfactory and has been applied to a wide variety of biological problems. However, in parallel computing, one encounters a nontrivial issue how the respective quantities on the Eulerian and Lagrangian meshes are adequately communicated to keep the load balance of each compute node. The authors have developed a full Eulerian (fixed-mesh) method using regular Cartesian grids (Fig. 1(c)), which facilitate FSI simulations with complex geometry of solid and/or a large number of bodies, and allows us to utilize computational techniques cultivated for solving incompressible fluid flow problems [17–22]. The developed method has revealed practical advantages of geometrical flexibility [16,21] since it can directly access voxel data and avoid a breakdown in a large deformation owing to the absence of the mesh distortion problem. In view of the efficiency in massively parallel computing, it readily exploits vector processing and reduces a computational-load imbalance in a domain decomposition.

In general, state-of-the-art scalar-type supercomputers are designed to attain high sustained performance (close to the peak performance of the system) in solving dense matrix problems. In stencil computations written in a sparse matrix form, if computationally-intensive tasks are involved for each grid point, one may exploit the system performance as demonstrated in a landmark simulation of dendritic growth revealed over 1 Pflops sustained performance using a GPGPU-based supercomputer [23]. However, in solving simple sparse matrix problems such as the Laplace, Poisson and Helmholtz equations and also fluid dynamics equations in a finite-difference form, the performance is considerably limited by a memory bandwidth since the Byte-per-Flop ratio (B/F) required in the software is higher than that equipped in the hardware (irrespective of using CPUs and/or GPGPUs) even if a high parallel scalability is realized [24]. An iterative procedure for solving the Poisson equation, which is included in the standard approach for incompressible fluid flow problems, requires a high B/F value. To drastically improve the situation, we follow a revived Artificial Compressibility Method (ACM) [25], which is unlike the original ACM [26] for the steady flow problem or the implicit ACM [27] but explicitly solves a pressure evolution equation. Further, we introduce a projection step and an optimization procedure, where the model parameters are adaptively determined to mathematically guarantee the computed field to be nearly incompressible.

For given constitutive laws, our numerical scheme can faithfully capture the deformability of dispersed bodies, which is a challenging issue in the computational dynamics research but a non-negligible factor in blood flow simulations demonstrated in §5. It should be emphasized that our solver is qualitatively different from the solver treating blood cells as rigid in [5], and therefore a direct comparison with this earlier study is meaningless. Our approach requires several times larger number of grid points to resolve the deformable particle than those developed for rigid particles [28–30]. We have confirmed that the energy exchange between the continuous and dispersed phases via the particle deformation is reasonably guaranteed when 50 or more grid points are used for the RBC diameter, indicating about 600 grid points are needed for 100μm capillary diameter.

This paper is organized as follows. In §2, a system overview of the K computer is presented. In §3, the basic equation set for the full Eulerian FSI simulation and the novel ACM are explained. Then, discretization, implementation and parallelization are described. In §4, the performance in serial jobs, and the weak and strong scaling analyses in parallel jobs are reported. In §5, the application to the blood flow is demonstrated. In §6, some perspectives are provided to conclude the paper.

2. Overview of the K computer

All the numerical simulations in this paper have been performed on the K computer. An overview of the system will be presented below. For more detailed description, we refer the readers to [31]. The K computer is a distributed-memory supercomputer system, which consists of more than 80,000 compute nodes. Each node includes an eight-core CPU SPARC64$^{\text{TM}}$VIIIfx [32] operated at a clock frequency of 2 GHz, and a set of DDR3 memory modules of 16 GB. (No GPGPU is included therein.) The cores in each node share a L2 cache of 6 MB. Each core has a L1 cache of 32 KB, and four floating-point multiply-and-add execution units. The peak performance is 128 Gflops for each compute node. The system provides memory bandwidth compared with the operation speed at B/F = 0.5. There is a three-level parallelism to be well considered when one utilizes the system as efficiently as possible. The first level is SIMD processing in instruction level on each core. The second one is thread programming in a compute node, that is supported by OpenMP directives. The third one is distributed-memory parallel programming with the MPI over multiple compute nodes. Data communication among the compute nodes is conducted through an interconnection network named Tofu (a six-dimensional mesh/torus network) [33], in which a barrier hardware facility makes collective communications such as *Allreduce* and *Bcast* efficient.

3. Simulation methods

3.1 Eulerian description for fluid-structure interaction problems

The fluid and solid are assumed to be incompressible and to possess the same density and viscosity, as in many analyses for biological systems. We employ a full Eulerian finite-difference scheme to discretize the fluid-structure system. The validity of this method has been established in various manners [16, 18, 19, 21, 22], including comparisons with analytical solutions, experiments, and well-validated FSI simulations. The governing equations are the mass and momentum conservations:

$$\nabla \cdot \boldsymbol{v} = 0, \tag{1}$$

$$\rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} \right) = \nabla \cdot \boldsymbol{\sigma}, \tag{2}$$

where $\boldsymbol{v}$ denotes the velocity vector, ρ the density, t the time, and $\boldsymbol{\sigma}$ the Cauchy stress written in a mixture form of the Newtonian fluid and the neo-Hookean material, namely

$$\begin{aligned}\boldsymbol{\sigma}(\boldsymbol{x}) = -p(\boldsymbol{x})\boldsymbol{I} - \frac{\Delta P}{L_x}x\boldsymbol{I} + 2\mu\Big(\boldsymbol{D}(\boldsymbol{x}) - \tfrac{1}{3}\mathrm{tr}(\boldsymbol{D}(\boldsymbol{x}))\boldsymbol{I}\Big)\\ + G\phi_s(\boldsymbol{x})\Big(\tilde{\boldsymbol{B}}(\boldsymbol{x}) - \tfrac{1}{3}\mathrm{tr}(\tilde{\boldsymbol{B}}(\boldsymbol{x}))\boldsymbol{I}\Big),\end{aligned} \tag{3}$$

where p denotes the pressure deviation from the driving pressure, $\boldsymbol{I}$ the unit tensor, $-\Delta P$ the inlet-outlet pressure drop, L_x the inlet-outlet length in the streamwise (x) direction of the computational domain, μ the viscosity, G the modulus of transverse elasticity, and $\boldsymbol{D}(= \frac{1}{2}(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T))$ the strain rate tensor. The operator tr(...) stands for the tensor trace. The kinematics of structure is represented by the quantities ϕ_s and $\tilde{\boldsymbol{B}}$ (Fig. 1(c)), that denote the local volume fraction of solid and the left Cauchy-Green deformation tensor (modified to be $\boldsymbol{I}$ in the fluid region), respectively, and obey the transport equations

$$\partial_t\phi_s + \boldsymbol{v}\cdot\nabla\phi_s = 0, \tag{4}$$

$$\partial_t\tilde{\boldsymbol{B}} + \boldsymbol{v}\cdot\nabla\tilde{\boldsymbol{B}} = \boldsymbol{L}\cdot\tilde{\boldsymbol{B}} + \tilde{\boldsymbol{B}}\cdot\boldsymbol{L}^T, \tag{5}$$

where $\boldsymbol{L}(= \nabla\boldsymbol{v}^T)$ denotes the velocity gradient tensor. We start the FSI simulation with the initial solid volume fraction ϕ_{s0} (here, the subscript 0 stands for the initial quantity) converted from a set of the voxel data.

3.2 Artificial compressibility method with adaptive parameters

In the present study, the pressure p is given to satisfy a relation

$$\langle p\rangle = 0, \tag{6}$$

where $\langle ...\rangle$ 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:

$$\boldsymbol{v}^* = \boldsymbol{v}^N + (\Delta t)\rho^{-1}\{-\nabla p^N + \boldsymbol{F}\}, \tag{7}$$

and the projection step:

$$p^{N+1} = p^N + \delta p, \tag{8}$$

$$\boldsymbol{v}^{N+1} = \boldsymbol{v}^* - (\Delta t)\rho^{-1}\nabla(\delta p), \tag{9}$$

where the superscript $*$ represents the unprojected quantity, and $\boldsymbol{F}$ is the summation of all the terms except for the pressure gradient term in the momentum equation (2). If one follows the SMAC algorithm [34], the incremental pressure δp is given by the solution to the Poisson equation

$$\nabla^2\delta p = (\Delta t)^{-1}D^*, \tag{10}$$

where $D(= \rho\nabla\cdot\boldsymbol{v})$ is the divergence of the mass flux. A memory-access intensive procedure due to the iteration is usually required to numerically solve Eq. (10). Instead, we follow the pressure evolution of the revived ACM [25] written in an algebraic form

$$\delta p = -\beta^2(\Delta t)(\gamma p^N + D^*), \tag{11}$$

where β denotes the pseudo acoustic speed, and $\gamma(\geq 0)$ the relaxation coefficient. Although the solenoidal condition Eq. (1) is not perfectly satisfied in the ACM, we try to approximate Eq. (1) as exactly as possible in a way that the parameters β and γ are determined dynamically at each time step. With a constraint $\gamma \geq 0$, the mean-square of the mass flux divergence at the $(N+1)$-th time level, $\langle(D^{N+1})^2\rangle$ is minimized to identify the parameters

$$\begin{aligned}\beta =& \frac{1}{(\Delta t)}\sqrt{\frac{A_1}{A_2}},\ \gamma = 0 \text{ if } \frac{-(A_1B_2 + A_2B_1)}{A_2A_3 - B_2^2} < 0,\\ \beta =& \frac{1}{(\Delta t)}\sqrt{\frac{A_1A_3 + B_1B_2}{A_2A_3 - B_2^2}},\ \gamma = \frac{-(A_1B_2 + A_2B_1)}{A_1A_3 + B_1B_2}\\ & \text{otherwise,}\end{aligned} \tag{12}$$

where

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

Although the ACM-based solution inevitably has the *pseudo* compressibility, its level is regulated through the above-mentioned procedure. Among all possible distributions of D^*, there exists a lower bound $A_1/A_2 = (\Delta x)^2/12$ (here, (Δx) is the side length of the cubic grid) when the second-order central finite difference is applied to evaluating ∇^2D^*. Hence, the pseudo Mach number $Ma(= U_{\max}/\beta)$ (here, $U_{\max}$ is the maximum advection speed in the system) has the upper limit $2\sqrt{3}\eta$ (here, $\eta = U_{\max}(\Delta t)/(\Delta x)$ is the Courant-Friedrichs-Lewy (CFL) number). Since the velocity perturbation due to the pseudo compressibility is proportional to Ma^2, the computed velocity field is guaranteed to be nearly incompressible as long as the CFL number η is sufficiently smaller than unity.

One may point out that the requirement of small η is undesirable because large number of time steps, resulting in long computational time, is needed. In the present study, a flexibly deformable structure is treated, and the elastic transverse wave, which is related to the shape oscillation corresponding to one of the softness features, is considered. The wave propagation speed is usually much higher than the advection speed characterized by $U_{\max}$. If the time increment Δt is determined to resolve the wave propagation, the requirement of $\eta \ll 1$ is consequently fulfilled. Therefore, the present ACM is practically useful for the problem considered in the present study.

For the validity of the present ACM in view of the accuracy and conservation, we refer the readers to [35]. Here, to recall the appropriateness of the algorithm and to facilitate explanation of the simulation target in §4, particulate channel flow simulations are demonstrated. The particle shape is discoid biconcave with 8μm diameter, replicating the geometry of Red Blood Cell (RBC) [36]. The system including 16 particles is 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. $\boldsymbol{v}_0 = p_0 = 0$ and $\tilde{\boldsymbol{B}}_0 = \boldsymbol{I}$) before the driving pressure is imposed. For comparison, two kinds of simulations are performed: one is based on the SMAC algorithm, in which the Fast Fourier Transform (FFT) in x and z directions and the Tri-Diagonal Matrix Algorithm (TDMA) in wall-normal (y) direction are applied to exactly solving the pressure Poisson equation (10) in a finite difference form, while the other is based on the present ACM.

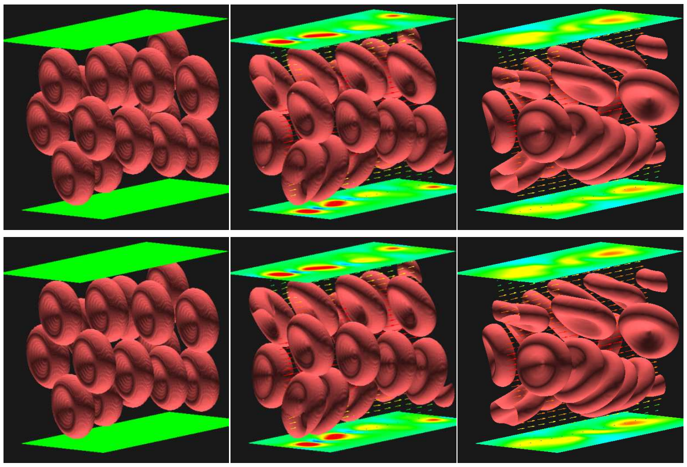


Fig. 2 The snapshots of hyperelastic particles in a three-dimensional Poiseuille flow. The computational extent is $L_x \times L_y \times L_z = 21.12\mu\text{m} \times 21.12\mu\text{m} \times 21.12\mu\text{m}$, and the number of grid points is $M_x \times M_y \times M_z = 128 \times 128 \times 128$. The material properties and the driving pressure gradient are scaled using the density ρ, and set to $\mu/\rho = 1\ (\mu\text{m})^2/\mu\text{s}$, $G/\rho = 2\times10^{-2}\ (\mu\text{m})^2/(\mu\text{s})^2$ and $-\Delta P/(\rho L_x) = 2\times10^{-4}\ \mu\text{m}/(\mu\text{s})^2$. The top and bottom panels show the results based on the SMAC method using FFT-TDMA and those on the present ACM, respectively. The left, middle, and right panels show the results at $t = 0$, $t = 0.64$ms, and $t = 6.4$ms, respectively. The colors on the walls indicate the magnitude of the shear stress.

The particle position and orientation are shown in Fig. 2. There are no significant discrepancies in the particle position and shape between the results based on the accurate SMAC algorithm and the present ACM.

3.3 Discretization, implementation and parallelization

The basic equations are numerically solved by means of a finite-difference method. We follow a conventional staggered grid arrangement. 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 scheme [37,38] is applied in order to avoid numerical instability and to suppress the dissipative error [19]. To integrate the equations in time, we employ the first-order Euler scheme.

We adopt an OpenMP-MPI hybrid programming model, and carry out all the computations in a double precision. The kernel code is written in Fortran and C with OpenMP directives, and implemented into an object-oriented framework V-Sphere [39], which includes class libraries to facilitate the software development especially for MPI parallel programs. We employ a domain decomposition over the whole computational domain given as a set of regularly-divided rectangular domains, that minimizes the computational-load imbalance, and is suited to the network topology of the Tofu system. The schedule of the MPI communications together with the computational procedures in each computational time step is schematically illustrated in Fig. 3. Since the spatial derivative of any quantity is discretized in the finite difference manner, an adjacent communication at six boundary surfaces of each decomposed domain is essential for referring to data at the contiguous compute node. The sleeve width in the adjacent communication for each boundary surface corresponds to three grid points for 10 quantities (ϕ_s^{N+1} (×1), $\tilde{\boldsymbol{B}}^{N+1}$ (×6) and $\boldsymbol{v}^{N+1}$ (×3)) and to one grid point for five quantities ($\boldsymbol{v}^*$ (×3), D^* (×1) and p^N (×1)). To reduce the time and frequency of the adjacent communication, we allocate buffer arrays, and pack/unpack the data to/from the contiguous buffer. The number of *Allreduce* operations is six, corresponding to the number of the volume averaged quantities in Eqs. (6) and (13).

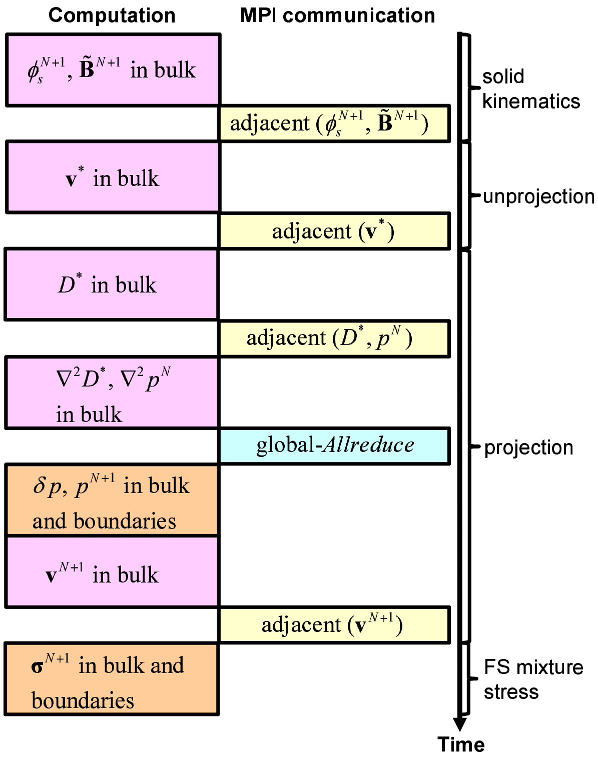

Computation
MPI communication
ϕ_s^{N+1}, $\tilde{\mathbf{B}}^{N+1}$ in bulk
adjacent (ϕ_s^{N+1}, $\tilde{\mathbf{B}}^{N+1}$)
$\mathbf{v}^*$ in bulk
adjacent ($\mathbf{v}^*$)
D^* in bulk
adjacent (D^*, p^N)
$\nabla^2 D^*$, $\nabla^2 p^N$ in bulk
global-*Allreduce*
δp, p^{N+1} in bulk and boundaries
$\mathbf{v}^{N+1}$ in bulk
adjacent ($\mathbf{v}^{N+1}$)
$\boldsymbol{\sigma}^{N+1}$ in bulk and boundaries
solid kinematics
unprojection
projection
FS mixture stress
Time

Fig. 3 Schematic diagram of computational procedures and MPI communications

4. Performance analyses

4.1 Performance in a serial job

Using a compute node (8 cores) without MPI communication, we here perform FSI simulations in a channel as demonstrated in Fig. 2, and examine the dependence of the performance upon the problem size $M = M_x \times M_y \times M_z$ up to $M = 8,388,608$. A performance profiler of the K computer provides the performance data accurately by means of a hardware counter measurement. A three-dimensional loop operation in the code is written in a form:

```
do  k=1,Mz
   do  j=1,My
      do  i=1,Mx
         (operations)
      enddo
   enddo
enddo
```

To demonstrate the efficiency in vector processing at the instruction level, Fig. 4(a) shows a plot of the number ratio of SIMD instructions to total instructions for the overall computation versus the innermost loop size M_x for various outer loop sizes M_y and M_z. Since the plotted symbols collapse onto a single curve, the SIMD ratio is dependent solely upon M_x irrespective of M_y or

M_z. As shown in Fig. 4(b) that reports the ratio of the sustained performance to the peak performance of the K computer, the ratio is dependent mainly upon M_x, indicating the relevance of the SIMDization to the performance enhancement. Inspecting Fig. 4(b) carefully, we find weak but obvious M_y dependence. (The level of the M_z dependence was confirmed to be comparable to that of the M_y dependence). We confirmed that the performance for fixed M_x was positively correlated with M_y and M_z in cases of $M_x \leq 128$, whereas it did not always increase with increasing M_y or M_z in cases of $M_x \geq 256$. Note that the highest sustained performance was achieved with a $512 \times 128 \times 128$ mesh.

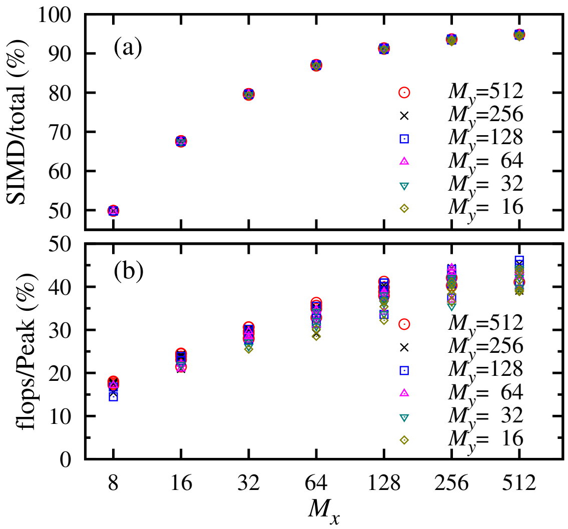


Fig. 4 (a) the number ratio of SIMD instructions to total instructions versus the innermost loop size M_x. (b) the performance ratio versus M_x.

Table 1 Computational time, performance, memory throughput and SIMD ratio in a serial job (using 8 cores) for various number M of grid points: $M_1 = 64 \times 64 \times 64$, $M_2 = 128 \times 128 \times 128$ and $M_3 = 512 \times 128 \times 128$

(a) overall

	M_1	$M2$	M_3
time/step	0.031 s	0.21 s	0.73 s
Gflops (/peak (%))	39.9 (31.2)	52.4 (40.9)	59.6 (46.6)
Mem. Thr (GB/s)	16.2	18.5	20.9
SIMD/total (%)	87.0	91.4	95.0

(b) advection terms $(\boldsymbol{v} \cdot \nabla) f$ in Eqs. (2), (4) and (5)

	M_1	$M2$	M_3
time/total (%)	65.5	71.6	73.1
Gflops (/peak (%))	58.7 (45.8)	66.5 (51.9)	74.1 (57.9)
Mem. Thr (GB/s)	12.5	14.1	16.1
SIMD/total (%)	90.8	94.4	97.3

(c) Oldroyd rate term $\boldsymbol{L} \cdot \tilde{\boldsymbol{B}} + \tilde{\boldsymbol{B}} \cdot \boldsymbol{L}^T$ in Eq. (5)

	M_1	$M2$	M_3
time/total (%)	15.2	13.4	11.2
Gflops (/peak (%))	17.0 (13.3)	22.6 (17.6)	30.7 (23.4)
Mem. Thr (GB/s)	11.4	15.5	21.8
SIMD/total (%)	76.5	82.3	87.3

(d) stress tensor components in Eq. (3)

	M_1	$M2$	M_3
time/total (%)	4.9	4.8	5.4
Gflops (/peak (%))	12.7 (9.9)	14.4 (11.3)	14.6 (11.4)
Mem. Thr (GB/s)	40.6	42.9	42.1
SIMD/total (%)	68.1	70.2	71.8

(e) divergence of stress tensor $\nabla \cdot \boldsymbol{\sigma}$ in Eq. (2)

	M_1	$M2$	M_3
time/total (%)	2.2	2.3	2.6
Gflops (/peak (%))	12.4 (9.7)	13.7 (10.7)	14.3 (11.2)
Mem. Thr (GB/s)	43.2	45.1	45.1
SIMD/total (%)	70.3	77.0	82.3

To further clarify the code feature in view of the performance, some statistical data for the overall computation and for four of the most time-consuming procedures are listed in Table 1. As in Table 1(b), 65% or more time is spent for evaluating the advection terms, and the ratio to the peak performance is as high as 45.8%, 51.9% and 57.9% for $64 \times 64 \times 64$, $128 \times 128 \times 128$ and $512 \times 128 \times 128$ meshes, respectively. Such a high performance dominates the overall performance in Table 1(a). As in Table 1(c), the performance for evaluating the Oldroyd rate term is strongly dependent upon the problem size, and that reaches 23.4% for the $512 \times 128 \times 128$ mesh. As in Table 1(d),(e), the performance in evaluating the stress tensor and the stress divergence is much lower than that for the advection terms since it is limited by the memory bandwidth (that is in fact confirmed by the memory throughput values close to the effective peak ($\approx$ 46GB/s) of the K computer). Nevertheless, these procedures do not much impact on the overall performance since each process is not much expensive.

It should be emphasized that the algorithm based on the present ACM makes it possible to achieve such a high performance in Table 1(a) and high parallel performance as will be demonstrated in §4.2 and §4.3. When one uses the SMAC method, corresponding to a standard approach for incompressible fluid flow problems, most of the computational time is spent for solving the pressure Poisson equation (instead of Eqs. (11), (12) and (13)) in massively parallel computing. The Poisson equation solvers include iterative methods such as the Jacobi, Gauss-Seidel, successive over-relaxation, and multi-grid ones, and a direct method using Fast Fourier Transform (FFT) and tridiagonal matrix algorithm. Although each iterative method has advantage or disadvantage in terms of efficiency in operation, convergence rate, parallel scalability and so on, the performance in whichever method is limited by the memory bandwidth. In consideration of the B/F value of the K computer and the Himeno benchmark tests [40] thereon, the ratio to the peak performance in each iterative procedure would be no more than 10%, and hence the overall performance approaches such a low value. Further, the performance of the FFT is degraded considerably with increasing the number of compute nodes due to the expensive global communication traversing all the compute nodes even though it is efficient in a serial job.

4.2 Weak scaling

To analyze the impact of the number N of compute nodes on the parallel performance, we conduct weak scaling tests in a range between $N = 1$ and $N = 82,944$ (i.e., 8 and $663,552$ cores) of the K computer. We perform FSI simulations in the channel for three different problem sizes per each compute node: $M/N = 64 \times 64 \times 64$, $128 \times 128 \times 128$ and $512 \times 128 \times 128$ (here, M denotes the total number of grid points). The system includes one RBC per $32 \times 64 \times 64$ mesh domain. In the maximum size problem ($M = 24,576 \times 6,912 \times 4,096 \approx 7 \times 10^{11}$), the number of RBCs is $5,308,416$. The sustained performance in flops and its ratio to the peak performance are reported in Fig. 5(a) and Fig. 5(b), respectively. Likewise the single node performance in §4.1, the parallel

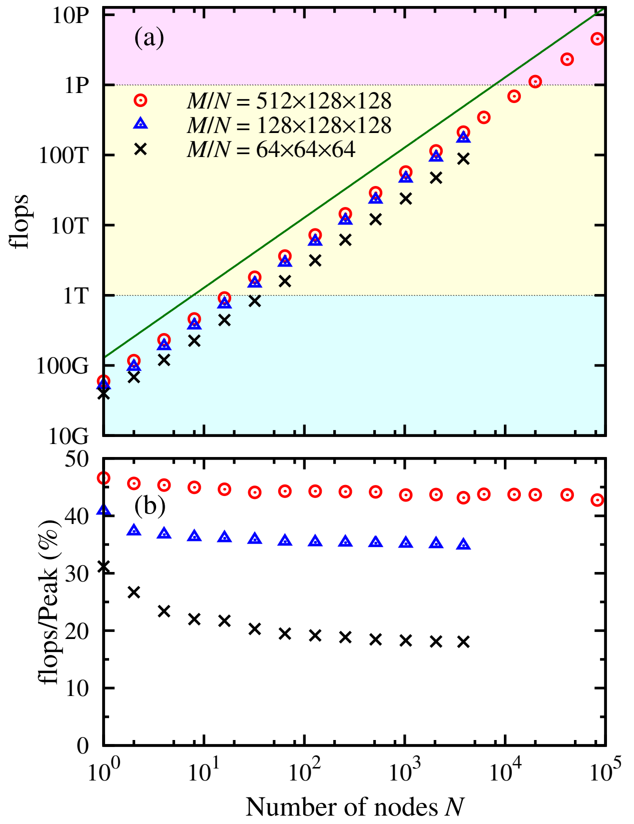


Fig. 5 Weak scaling performance as a function of the number N of compute nodes. The circles, triangles and crosses indicate the measured results with $512 \times 128 \times 128$, $128 \times 128 \times 128$ and $64 \times 64 \times 64$ meshes per compute node, respectively. (a) sustained performance in flops. The solid line indicates the peak performance of the K computer. (b) ratio of the sustained performance to the peak performance.

performance is dependent mainly upon M/N. In Fig. 5(a), all the plots for respective M/N exhibit excellent scalability as compared with the slope of the peak performance. Notably, for the maximum problem size per each node ($M/N = 512 \times 128 \times 128$) in Fig. 5(b), the performance ratio is 45.6%, 43.7% and 42.7% for $N = 2$, $N = 12,288$ and $N = 82,944$, respectively, and hence the parallel efficiency (based on the single node performance ratio 46.6% in Table 1(a)) is more than 91.6%. For this size per node, the sustained performance of 4.54 Pflops is achieved at $N = 82,944$. Even for the smaller size per node, the sustained performance is found to be high enough: the performance ratio at $N = 3,840$ is 18.1% and 34.8% for $M/N = 64 \times 64 \times 64$ and $M/N = 128 \times 128 \times 128$ meshes, respectively. Both the efficiency and scalability shown in Fig. 5 are unprecedentedly high among numerical simulations of incompressible fluid and/or structure dynamics performed on state-of-the-art scalar-type supercomputers owing to the novel scheme and algorithm described in §3.

4.3 Strong scaling

Here, we conduct strong scaling tests. We perform FSI simulations for three different meshes: $M = 1,024 \times 256 \times 256$, $2,048\times512\times512$ and $4,096\times1,024\times1,024$, with which the systems involve 512, 4,096 and 32,768 RBCs, respectively. The sustained performance in flops is reported in Fig. 6(a). The plots for the respective problem sizes exhibit good scalability even though the communication-to-computation time ratio is likely to be large using an efficient code. The adjacent and global communication time ratios to the total elapsed time are reported in Fig. 6(b) and Fig. 6(c), respectively. More than 20% of the elapsed time is spent for communication when N is increased by a factor of 16 or more. Even in such a situation, nevertheless, the sustained performance is found to be high enough: the performance ratios are larger than 28% and 20% when N is increased by factors of 16 and 32, respectively (see Fig. 5(b), and the inset of Fig. 6(a) showing the parallel efficiency). The drop in the parallel efficiency (the inset of Fig. 6(a)) with increasing N/N_{base} is not much dependent upon the overall problem size M owing to the excellent weak scaling as demonstrated in §4.2. Thus, not only the change in the communication time but also the efficiency in vector processing dependent upon the problem size per compute node (as examined in §4.1) are considerably reflected on the strong scalability in Fig. 6(a).

5. Blood flow simulation

We apply the developed code to blood flow simulations with RBCs and platelets in a capillary vessel. The diameter of the capillary vessel is set to $D = 104\mu$m, which is comparable to that of a cerebral arteriole. The initial RBC shape is discoid biconcave with 8μm diameter, and the initial platelet shape is spheroid with diameters of $2\mu\text{m} \times 2\mu\text{m} \times 1.6\mu\text{m}$ [16, 36]. We shall shed light on the role of the RBC on the fluctuation of the platelet, which would afford insight into the dynamic interaction in an initial process of thrombus formation. To clarify this point, we vary the RBC volume fraction (i.e., the hematocrit Ht) from 0% (no RBC) to 20%. In the present study, the driving pressure gradient is fixed at $-\Delta P/(\rho L_x) = 4 \times 10^{-5}\ \mu\text{m}/(\mu\text{s})^2$. The blood vessel and the platelet are treated as hyperelastic. Thus, not only the fluid-RBC-platelet motion but also the deformable motion of the blood vessel are computed to be coupled.

Figure 7 visualizes the shape of the RBCs, platelets and vessel wall at $Ht = 20\%$ for three time instants. At $t = 0$ (Fig. 7(a)), the RBCs and platelets are seeded almost regularly in space but with a small random perturbation to promote the symmetry breaking. The RBCs are mixed over time. The RBC subjected to a high shear rate near the vessel wall highly deforms and generates the velocity disturbance around it when translating in the downstream. To demonstrate the relevance of such an agitated fluid flow to the platelet motion, Fig. 8 reports radial positions of some platelets in time. Even in the absence of the RBCs (Fig. 8(a)), the radial position is slightly fluctuated due to the hydrodynamic interactions among the platelets and wall. More remarkably, however, the platelet in the presence of the RBCs is much more dispersed (Fig. 8(b)), since it moves in a flow undergoing a larger fluctuation induced by the RBCs. It should be noted that the present results are consistent with the experimental result of the enhanced thrombus formation in the presence of the RBCs [42] presumably because the platelets get increased chance to approach to the vessel wall.

Figure 9 shows the radial profiles of the root-mean-square velocity $\langle v_r^2 \rangle_p^{1/2}$. It is also seen the enhancement of the platelet fluctuation due to the presence of the RBCs. For $0\% \le Ht \le 15\%$, the

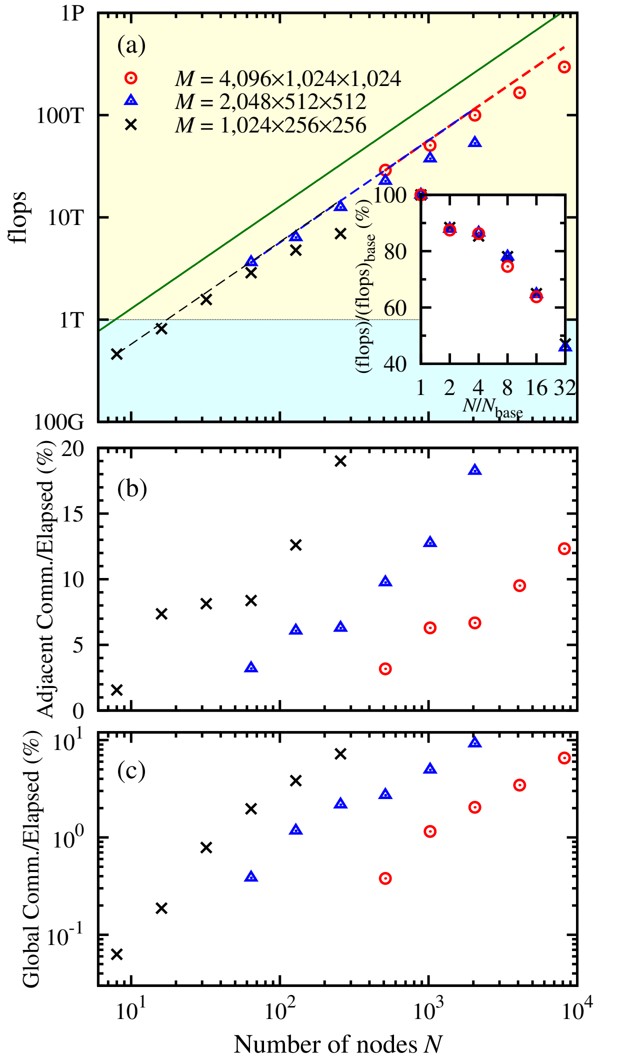


Fig. 6 Strong scaling performance and communication time as a function of the number N of compute nodes. The crosses, triangles and circles indicate the measured results with the number M of total grid points, corresponding to $M = 1,024 \times 256 \times 256$, $M = 2,048 \times 512 \times 512$ and $M = 4,096 \times 1,024 \times 1,024$, respectively. (a) sustained performance in flops. The solid line indicates the peak performance of the K computer. The dashed lines with different colors indicate the theoretical performance based on the sustained performance with N_{base}(= the smallest N) nodes (with a 512×128×128 mesh per compute node) for respective M conditions. The inset shows the parallel efficiency, i.e. plots of the sustained performance normalized by that with N_{base} nodes versus N/N_{base}. (b) ratio of the adjacent communication to the total elapsed time. (c) ratio of the global communication to the total elapsed time.

fluctuating intensity $\langle v_r^2\rangle_p^{1/2}$ becomes higher with increasing Ht, whereas for $Ht = 20\%$, the intensity $\langle v_r^2\rangle_p^{1/2}$ is lower than those for $Ht = 5$, 10 and 15 (%). Such a non-monotonous change in $\langle v_r^2\rangle_p^{1/2}$ implies that there exist competitive effects on the platelet fluctuation with increasing Ht. The effects may include not only a positive contribution, corresponding to the increase in the number of agitation sources, but also negative ones such as the reduction in the fluid volume, in which the platelet can freely move, and the reduction in the flow rate. Interestingly, in the presence of the RBCs ($Ht \geq 5\%$), the fluctuating intensities of the platelets increase from $r/(D/2) \approx 0.4$ to ≈ 0.75, and then they steeply decrease near the wall. Such a peaky profile is consistent with the finding in [15] that the radial drifts of the laterally-located platelets are restricted by the RBCs performing the tank-treading motion near the wall.

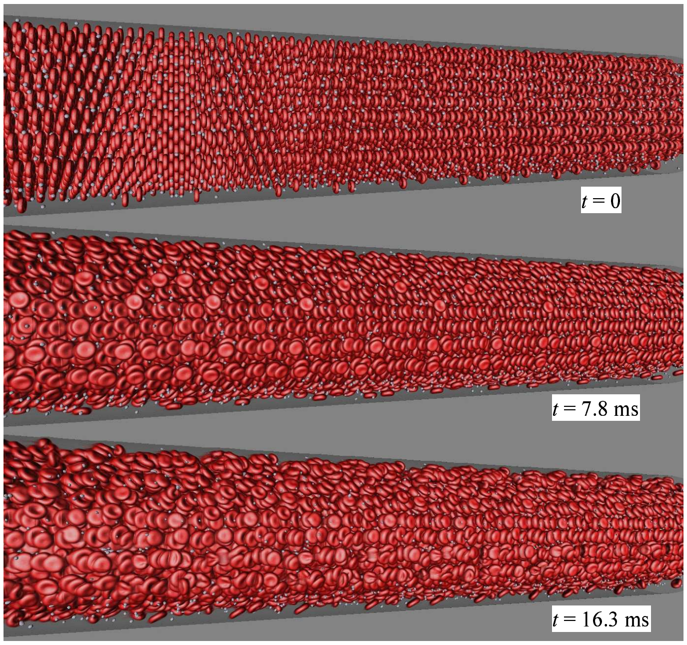


Fig. 7 Snapshots of the blood flow in a capillary vessel at $Ht = 20\%$ with a $3,072 \times 640 \times 640$ mesh. The numerical simulation was performed using 4,800 compute nodes (38,400 cores) of the K computer. The red and white surfaces indicate the interface of the RBC and that of the platelet, respectively, which are visualized using a distributed-parallel large-data visualization system LSV [41]. Top (a), middle (b) and bottom (c) panels show the results at $t = 0$, $t = 7.8$ms and $t = 16.3$ms, respectively.

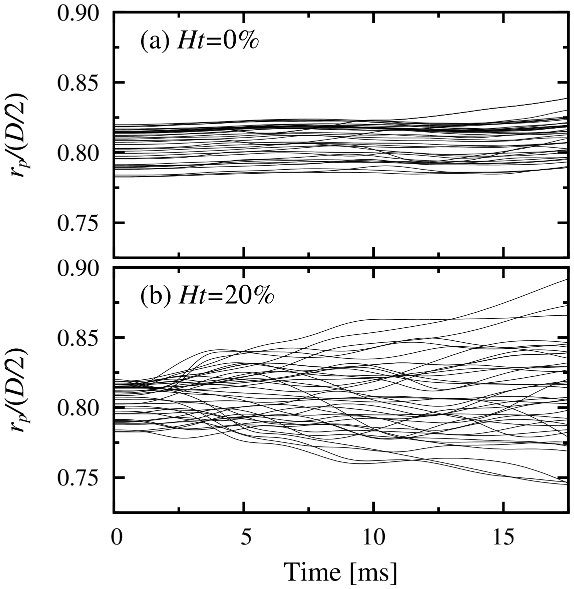


Fig. 8 Temporal evolution of trajectories of part of platelets, which are initially located at $r_p/(D/2) \approx 0.8$. Here, r_p denotes the distance between the particle centroid and the axis. (a) $Ht = 0\%$ (no RBC); (b) $Ht = 20\%$.

6. Conclusion and perspectives

We have developed the efficient and scalable numerical method for fluid-structure interaction problems, which is useful for a wide variety of biomedical applications. We focused on microchannel and capillary flows including a large number of flexibly de-

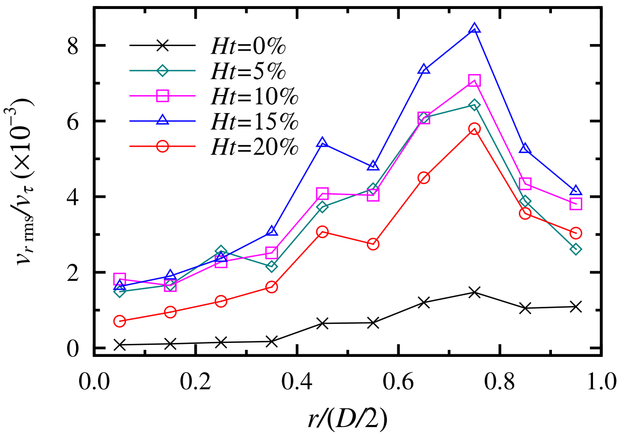


Fig. 9 Radial profiles of the radial root-mean-square velocity component sampled over the platelets and in a range between t = 4ms and t = 17.5ms for various hematocrit of Ht = 0, 5, 10, 15 and 20 (%).

formable bodies such as blood flows. We have put our primary effort into redesigning the basic equation set for facility in the solution and the algorithm for a significant boost in performance on state-of-the-art scalar-type supercomputers. The full Eulerian method [19] facilitates treatment of a time-dependent complex geometry in practical applications. Furthermore, it was confirmed to exploit the system performance owing to the promotion of SIMDization and the minimization of the computational-load imbalance. The novel artificial compressibility method avoids iterative procedures for implicitly solving the pressure Poisson equation, and thus to considerably reduce the amount of the memory access and adjacent communication. This algorithm would pave a new way for a wide range of applications of incompressible fluid and structure dynamics because it has a potential to provide the best trade-off in terms of computational efficiency, accuracy and time steps in massively parallel computing.

Both the weak and strong scaling tests showed good scalability on the K computer. In particular, the weak scaling test using a 512 × 128 × 128 mesh per compute node demonstrated the parallel efficiency of 91.6% for 82, 944 compute nodes based on the single node performance (46.6% of peak). Both the efficiency and scalability were unprecedentedly high among numerical simulations of fluid/structure dynamics performed on the recent scalar-type supercomputers. Notably, the sustained performance of 4.54 Pflops (42.7% of peak) was achieved for a microchannel flow simulation with 5,308,416 deformable bodies using 24, 576 × 6, 912 × 4, 096 grids on 82,944 compute nodes. We applied the developed method to large-scale computations of blood flows including Red Blood Cells (RBCs) and platelets in a capillary vessel of about 100μm diameter, which is comparable to the arteriole size in a brain, using 4,800 compute nodes. The usefulness of the method was confirmed by demonstrating the relevance of the RBC to the platelet dispersion.

In order to extend the applicability of the code to more practical problems in thrombus formation, it is important to incorporate ligand-receptor interactions between the von Willebrand factor on the injured wall and the glycoprotein on the platelet surface, corresponding to molecular scale phenomena. It is a challenging task to overcome the multi-scale/physics difficulty irrespective of the further increase in computational performance toward Exascale computing. A stochastic modeling based on a kinetic Monte Carlo method and the implementation of a robust time advancement during the coupling are the ongoing subjects of the authors. To be compared with the experiment in a microchannel with 100μm height [42], the simulation may require several times larger number of compute nodes than the present simulations of the blood flow in the capillary vessel because of the wider extent in the streamwise and spanwise directions.

Acknowledgments

The authors are grateful to Prof. Shinya Goto, Prof. Taku Ohwada, Prof. Kengo Nakajima, Dr. Kenji Ono, Dr. Noriko Tamura and Dr. Naoto Yamamura for fruitful discussion, and to Dr. Yosuke Ohno, Dr. Gen Masumoto, Mr. Aki Hasegawa and Mr. Hikaru Inoue for their assistance in using the K computer at the RIKEN AICS. K.S., Y.K. and S.N. thank Mr. Hiroyuki Yoshikawa for his effort in handling the software LSV [41]. The authors thank two reviewers 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. The authors utilized the RIKEN Integrated Cluster of Clusters (RICC) when developing the code. Part of the results was obtained by early access to the K computer at the RIKEN AICS.

References

[1] Pozrikidis, C. (Ed.): *Modeling and Simulation of Capsules and Biological Cells*, Chapman & Hall/CRC, Boca Raton (2003).

[2] Bungartz, H.-J. and Schäfer, M. (Eds.): *Fluid-Structure Interaction – Modelling, Simulation, Optimisation*, *Lecture Notes in Comput. Sci. Engrg.*, Vol. 53 (2006).

[3] Pozrikidis, C. (Ed.): *Computational Hydrodynamics of Capsules and Biological Cells*, CRC Press, Boca Raton (2011).

[4] Rahimian, A., Lashuk, I., Veerapaneni, S.K., Chandramowlishwaran, A., Malhotra, D., Moon, L., Sampath, R., Shringarpure, A., Vetter, J., Vuduc, R., Zorin, D. and Biros, G.: Petascale direct numerical simulation of blood flow on 200K cores and heterogeneous architectures, *Proc. of 2010 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis*, doi: 10.1109/SC.2010.42 (2010).

[5] Bernaschi, M., Bisson, M., Endo, T., Fatica, M., Matsuoka, S., Melchionna, S. and Succi, S.: Petaflop biofluidics simulations on a two million-core system, *Proc. of 2011 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis*, doi: 10.1145/2063384.2063389 (2011).

[6] Grinberg, L., Morozov, V., Fedosov, D., Insley, J.A., Papka, M.E., Kumaran, K. and Karniadakis, G.E.: A new computational paradigm in multiscale simulations: Application to brain blood flow, *Proc. of 2011 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis*, doi: 10.1145/2063384.2063390 (2011).

[7] Stoltz, J.F., Singh, M. and Riha, P.: *Hemorheology in Practice*, IOS, Amsterdam (1999).

[8] Wootton, D.M. and Ku, D.N.: Fluid mechanics on vascular systems, diseases, and thrombosis, *Annu. Rev. Biomed. Eng.*, Vol. 1, pp. 299–329 (1999).

[9] Popel, A.S. and Johnson, P.C.: Microcirculation and and hemorheology, *Annu. Rev. Fluid Mech.*, Vol. 37, pp. 43–69 (2005).

[10] Goto, S., Ikeda, Y., Saldívar, E. and Ruggeri, Z.M.: Distinct mechanisms of platelet aggregation as a consequence of different shearing flow conditions, *J. Clin. Invest.*, Vol. 101, pp. 479–486 (1998).

[11] Andrews, D.A. and Philip, S.: Role of red blood cells in thrombosis, *Cur. Opinion Hematology*, Vol. 6, pp. 76–82 (1999).

[12] Fogelson, A.L. and Guy, R.D.: Platelet-wall interactions in continuum models of platelet thrombosis: formulation and numerical solution, *Math. Med. Biology*, Vol. 21, pp. 293–334 (2004).

[13] Nesbitt, W.S., Westein, E., Tovar-Lopez, F.J., Tolouei, E., Mitchell, A., Fu, J., Carberry, J., Fouras, A. and Jackson, S.P.: A shear gradient-dependent platelet aggregation mechanism drives thrombus formation, *Nature Med.*, Vol. 15, pp. 665–675 (2009).

[14] Mody, N.A. and King, M.R.: Platelet adhesive dynamics. Part I: characterization of platelet hydrodynamic collisions and wall effects, *Biophys. J.*, Vol. 93, pp. 2539–2555 (2008).

[15] Crowl, L. and Fogelson, A.L.: Analysis of mechanisms for platelet near-wall excess under arterial blood flow conditions, *J. Fluid Mech.*, Vol. 676, pp. 348–375 (2011).

[16] Ii, S., Sugiyama, K., Takagi, S. and Matsumoto, Y.: A computational blood flow analysis in a capillary vessel including multiple red blood cells and platelets, *J. Biomech. Sci. Engrg.*, Vol. 7, pp. 72–83 (2012).

[17] Sugiyama, K., Ii, S., Takeuchi, S., Takagi, S. and Matsumoto, Y.: Full Eulerian simulations of biconcave neo-Hookean particles in a Poiseuille flow, *Comput. Mech.*, Vol. 46, pp. 147–157 (2010).

[18] Nagano, N., Sugiyama, K., Takeuchi, S., Ii, S., Takagi, S. and Matsumoto, Y.: Full Eulerian finite-difference simulation of fluid flow in hyperelastic wavy channel, *J. Fluid Sci. Tech.*, Vol. 5, pp. 475–490 (2010).

[19] Sugiyama, K., Ii, S., Takeuchi, S., Takagi, S. and Matsumoto, Y.: A full Eulerian finite difference approach for solving fluid-structure coupling problems, *J. Comput. Phys.*, Vol. 230, pp. 596–627 (2011).

[20] Ii, S., Sugiyama, K., Takeuchi, S., Takagi, S. and Matsumoto, Y.: An implicit full Eulerian method for the fluid-structure interaction problem, *Int. J. Numer. Meth. Fluids*, Vol. 65, pp. 150–165 (2011).

[21] Takagi, S. Sugiyama, K., Ii, S. and Matsumoto, Y.: A review of full Eulerian methods for fluid structure interaction problems, *J. Appl. Mech.*, Vol. 79, 010911 (2012).

[22] Ii, S., Gong, X., Sugiyama, K., Wu, J., Huang, H. and Takagi, S.: A full Eulerian fluid-membrane coupling method with a smoothed volume-of-fluid approach, *Comm. Comput. Phys.*, Vol. 12, pp. 544–576 (2012).

[23] Shimokawabe, T., Aoki, T., Takaki, T., Endo, T., Yamanaka, A., Maruyama, N., Nukada, A. and Matsuoka, S.: Peta-scale phase-field simulation for dendritic solidification on the TSUBAME 2.0 supercomputer, *Proc. of 2011 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis*, doi: 10.1145/2063384.2063388 (2011).

[24] Oliker, L., Carter, J., Wehner, M., Canning, A., Ethier, S., Mirin, A., Parks, D., Worley, P., Kitawaki, S. and Tsuda, Y.: Leading computational methods on scalar and vector HEC platforms, *Proc. of 2005 ACM/IEEE Conf. on Supercomputing*, doi: 10.1109/SC.2005.41 (2005).

[25] Ohwada, T. and Asinari, P.: Artificial compressibility method: asymptotic numerical method for incompressible Navier-Stokes equations, *J. Comput. Phys.*, Vol. 229, pp. 1698–1723 (2010).

[26] Chorin, A.J.: A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.*, Vol. 2, pp. 12–26 (1967).

[27] Rogers, S.E. and Kwak, D.: Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations, *AIAA J.*, Vol. 28, pp. 253–262 (1990).

[28] Glowinski, R., Pan, T.-W., Hesla, T.I., Joseph, D.D. and Périaux, J.: A ficticious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, *J. Comput. Phys.*, Vol. 169, pp. 363–426 (2001).

[29] Kajishima, T. and Takiguchi, S.: Interaction between particle clusters and particle-induced turbulence, *Int. J. Heat and Fluid Flow*, Vol. 23, pp. 639–646 (2002).

[30] Zhang, Z. and Prosperetti, A.: A second-order method for three-dimensional particle simulation, *J. Comput. Phys.*, Vol. 210, pp. 292–324 (2005).

[31] Yonezawa, A., Watanabe, T., Yokokawa, M., Sato, M. and Hirao, K.: Advanced Institute for Computational Science (AICS) – Japanese national high-performance computing research institute and its 10-Petaflops supercomputer "K", *Proc. of SC '11 State of the Practice Reports*, doi: 10.1145/2063348.2063366 (2011).

[32] Maruyama, T., Yoshida, T., Kan, R., Yamazaki, I., Yamamura, S., Takahashi, N., Hondou, M. and Okano, H.: SPARC64 VIIIFX: a new generation octocore processor for Petascale computing, *IEEE Micro*, Vol. 30, pp. 30–40 (2010).

[33] Ajima, Y., Sumimoto, S. and Shimizu, T.: Tofu: A 6D mesh/torus interconnect for exascale computers, *Computer*, Vol. 42, pp. 36–40 (2009).

[34] Amsden, A.A. and Harlow, F.H.: A simplified MAC technique for incompressible fluid flow calculations, *J. Comput. Phys.*, Vol. 6, pp. 322–325 (1970).

[35] Sugiyama, K., Kawashima, Y., Koyama, H., Noda, S., Ii, S., Takagi, S., Matsumoto, Y. and Himeno, R.: Development of explicit Eulerian finite difference solver for large-scale fluid-structure interaction systems, *Proc. of High Performance Computing Symp. 2012*, IPSJ-HPCS2012056 (2012).

[36] Gong, X., Sugiyama, K., Takagi, S. and Matsumoto, Y.: The deformation behavior of multiple red blood cells in a capillary vessel, *J. Biomech. Engrg.*, Vol. 131, 074504 (2009).

[37] Jiang, G.-S. and Shu, C.-W.: Efficient implementation of weighted ENO scheme, *J. Comput. Phys.*, Vol. 126, pp. 202–228 (1996).

[38] Osher, S. and Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*, Sect. 3, Springer-Verlag, New York (2003).

[39] Ono, K., Tamaki, T. and Yoshikawa, H.: Development of a framework for parallel simulators with various physics and its performance, *Lecture Notes in Comput. Sci. Engrg.*, Vol. 67, pp. 9–18 (2009).

[40] The Himeno benchmark, `http://accc.riken.jp/HPC_e/himenobmt_e.html`

[41] LSV, `http://www.csrp.riken.jp/application_h_e.html#H2`

[42] Goto, S. and Tamura, N. (private communication).