Journal of Computational Physics 242 (2013) 809–827

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

An efficient parallel simulation of interacting inertial particles in homogeneous isotropic turbulence

Ryo Onishi [a,b,*], Keiko Takahashi [a], J.C. Vassilicos [b]

[a] Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology, 3173-25 Showa-machi, Kanazawa-ku, Yokohama, Kanagawa 236-0001, Japan
[b] Department of Aeronautics, Imperial College London, SW7 2AZ, UK

ARTICLE INFO

Article history:
Received 25 April 2012
Received in revised form 10 January 2013
Accepted 10 February 2013
Available online 27 February 2013

Keywords:
Parallel computing
Particle collision
Hydrodynamic interaction
Homogeneous isotropic turbulence

ABSTRACT

This study has conducted parallel simulations of interacting inertial particles in statistically-steady isotropic turbulence using a newly-developed efficient parallel simulation code. Flow is computed with a fourth-order finite-difference method and particles are tracked with the Lagrangian method. A binary-based superposition method has been developed and implemented in the code in order to investigate the hydrodynamic interaction among many particles. The code adopts an MPI library for a distributed-memory parallelization and is designed to minimize the MPI communication, which leads to a high parallel performance. The code has been run to obtain collision statistics of a monodisperse system with $St = 0.4$ particles, where St is the Stokes number representing the particle relaxation time relative to the Kolmogorov time. The attained Taylor-microscale based Reynolds number R_λ ranges from 54.9 to 527. The largest simulation computed the flow on 2000^3 grids and 1000^3 (one billion) particles. Numerical results have shown that the collision kernel increases for $R_\lambda<100$ then decreases as R_λ increases. This Reynolds dependency is attributed to that of the radial distribution function at contact, which measures the contribution of particle clustering to the collision kernel. The results have also shown that the hydrodynamic interaction for $St = 0.4$ particles decreases both the radial relative velocity and radial distribution function at contact, leading the collision efficiency less than unity. The collision efficiency increases from 0.65 to 0.75 as R_λ increases for $R_\lambda<200$ and then saturates.

1. Introduction

Several mechanisms have been proposed in the literature to explain what causes the fast size-broadening of cloud droplets, which could result in quick rain initiation at the early stage of cloud development. Examples are enhanced collision rate of cloud droplets by turbulence [11,14], turbulence entrainment [6,17], giant cloud condensate nuclei [46,37] and turbulent dispersions of cloud droplets [34]. The most intensely discussed is the first mechanism; enhanced collision rate by turbulence. This has initiated extensive research on particle collisions in turbulence ([36,43,33,25,9, and references therein]).

There are several collision models that predict collision rates of particles in turbulence. Saffman and Turner [32] analytically derived a collision model for particles with zero or very small St ($=\tau_p/\tau_\eta$, where τ_p is the particle relaxation time and τ_η the Kolmogorov time), while Abrahamson [1] derived a model for particles with large St. There is yet no widely accepted model for finite inertial particles, although water droplets in cumulus clouds have finite inertia; cloud droplets have

* Corresponding author at: Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology, 3173-25 Showa-machi, Kanazawa-ku, Yokohama, Kanagawa 236-0001, Japan.
E-mail address: onishi.ryo@jamstec.go.jp (R. Onishi).

0021-9991/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.02.027

$St = O(10^{-2\sim0})$ and rain drops $St = O(10^{0\sim2})$. One difficulty arises from the preferential motion of inertial particles. Inertial particles preferentially cluster in regions of low vorticity and high strain if $St \ll 1$ [20], and cluster in a way to mimic the clustering of zero-acceleration points by the sweep-stick mechanism if $1 \lesssim St \lesssim \tau_p/T$, where T is the integral time scale of the turbulence [7]. This matters because clustering increases the mean collision rate [36]. The clustering effect prevents the construction of a fully-analytical model for finite-inertial particles, and requires several empirical parameters in collision models [49,43,48,22,12]. Those parameters are usually determined by direct numerical simulation (DNS) data. Data from laboratory experiments [33,19] would of course help, but available data are very much limited.

One serious problem is that no collision data is available for high Reynolds number flows. The Taylor-microscale based Reynolds number R_λ ($= u'l_\lambda/\nu$, where u' is the rms of velocity fluctuations, l_λ the Taylor microscale and ν the kinematic viscosity) for collision statistics attained by DNS has been at most $R_\lambda \sim 100$. This value is much smaller than those in cloud turbulence, in which R_λ ranges from 10^3 (shallow cumulus clouds) to 10^5 (deep cumulus clouds). Nevertheless, there are several studies where collision models were used in cloud simulations to investigate the impact of enhanced collisions of cloud droplets [24,45,40,26]. They simply extrapolated their collision models to high R_λ, without justification. A simple solution is to obtain collision statistics for high R_λ flows for justifying models, which requires high-performance computing.

Code parallelization is indispensable for high-performance computing. The parallelization is classified into two types. One is the shared-memory parallelization (openMP and auto parallelization libraries are commonly used), and the other the distributed-memory parallelization (message-passing interface, MPI, is commonly used). In the shared-memory parallelization, all processors operate independently but share the same memory resources, i.e., global memory. The global memory concept provides a user-friendly programming perspective to memory. However, shared-memory computers cannot scale very well. Most of them have only ten or fewer processors. In contrast, memory is scalable with number of processors in the distributed-memory parallelization, which therefore is preferable in massively-parallel simulations. Processors have their own local memory and there is no concept of global memory space across all processors. When a processor needs access to data in another processor, data must be communicated through network connecting inter-processor memory, which has much narrower band than that between processor and local memory. Therefore the key to success of massively-parallel simulations with distributed-memory parallelization is in reducing the amount of data communications.

There have been numerous DNS codes for colliding particles in turbulence [43,29,13,5,39,44,25]. One may notice that most of them adopt pseudo-spectral models (PSMs). Unfortunately, few of the PSM codes for particle collisions are designed for the distributed-memory parallel simulations (the only one exception to the authors' knowledge is the very recent work by Rosa et al. [31]), and therefore the attained R_λ has been limited. Furthermore, a parallel PSM code faces major difficulties in massively parallel computing: PSM requires all-to-all data communication for the Fourier transformation. This prevents the PSM from maintaining good parallel efficiency for massively-parallel simulations. Another difficulty is imposed when coupling the PSM with particle calculations. The flow is computed in wavespace in PSM, but Lagrangian particles are in physical space. Code developers therefore need to consider domain decompositions in both wavespace and physical space. These two difficulties could be major reasons why there are few distributed-memory codes for colliding particles employing PSM.

Recently, Onishi et al. [23] developed a finite-difference model (FDM) with an efficient large-scale forcing scheme named reduced-communication forcing (RCF) for statistically-stationary isotropic turbulence. The FDM employs the three-dimensional domain decomposition leading to high parallel efficiency. They also confirmed good reliability of their FDM, which employs a conservative fourth-order finite difference scheme [21]. FDM requires less communications than PSM, and it is, therefore, suitable for massively-parallel computing. Coupling the particle calculation with the FDM can be an alternative to the coupling with the PSM for simulations of inertial particles in high Reynolds number flows.

One important physical process, which has often been neglected due to its high computational cost, is hydrodynamic interactions between particles. These interactions cause particles tend to avoid collisions, thus often leading to a collision efficiency $E_c < 1$. There are several studies which observed that turbulence increases the collision efficiency ([27,39]). However, these studies did not provide data for high Reynolds flows, leaving the Reynolds number dependency of collision efficiency unclear.

Recently, Ayala et al. [3] (hereafter referred to as AGW07) proposed a numerical scheme to consider the hydrodynamic interaction among colliding particles in three-dimensional turbulence. Their scheme is based on the superposition method [28] and adopts the Gauss–Seidel method to iteratively solve a large linear system. It is reportedly feasible to perform a three-dimensional simulation using the scheme, which still requires a huge computational cost for calculating the hydrodynamic interaction. It is preferable to employ a computationally lighter scheme for larger-size computations.

This study aims to develop an MPI parallel code for interacting particles in homogeneous isotropic turbulence (PIPIT, Parallel code for Interacting Particles in homogeneous Isotropic Turbulence) based on FDM coupled with Lagrangian particle calculations, and run the code to obtain data for high-Reynolds flows. The data obtained is used to investigate the Reynolds number dependencies of collision statistics of inertial particles. An efficient scheme, named binary-based superposition method (BiSM), for the hydrodynamic interaction calculation is also proposed and implemented in PIPIT. BiSM is based on the superposition method but is more accurate than the original superposition method [28] and is more computationally efficient than the scheme by Ayala et al. [3]. The cell-index method [2] is adopted in PIPIT for efficient detection of neighboring pairs.

In the following section, we describe numerical procedures for collision detection (subSection 2.1) and hydrodynamic interaction (subSection 2.2). The main frame of PIPIT is then introduced in Section 3, where the algorithm for efficient parallel simulations is described. Numerical results and discussion are mostly presented in Section 4, which consists of

the following subsections. At first, the performance and an optimal setting of PIPIT are discussed in subSections 4.1 and 4.2. Collision statistics for different Reynolds number flows is then presented in subSection 4.3. This study is concluded in Section 5.

2. Particle interactions

2.1. Geometric collisions

2.1.1. Collision detection

There are several ways in dealing with collision events. One of the colliding pair of droplets may be removed immediately after collision (*Scheme 1*), or droplets may be allowed to overlap (ghost-particle condition) (*Scheme 2*). Scheme 1 is more realistic because the collision-coalesced droplet will form a particle of larger size and will disappear from the original size group. Scheme 2 is suitable for discussing the so-called spherical form (refer to Eq. (3)), where the effect of clustering is clear. In order to involve a discussion on the clustering effect this study employs Scheme 2.

A collision is judged from the trajectories of a pair of droplets assuming linear particle movement for a time interval Δt. The linear particle movement leads to linear change of $S(t)$ for Δt, where $S(t)$ is the separation distance between a pair of particles selected for collision detection. There are then two situations when a collision must be counted: (i) If $S(t^{n-1}) > R$, where R ($= 2r$ in a monodisperse system) is the collision radius, and $S(t^n) \leqslant R$, a collision must occur. (ii) If $S(t^{n-1}) > R$ and $S(t^n) > R$, a collision could occur if $S(t) \leqslant R$ for $t^{n-1} < t < t^n$.

After the background airflow has reached a statistically stationary state, monodispersed water droplets are introduced into the flow. After a period exceeding three times the eddy-turnover time $T_0 = L_0/U_0$, collision detection is then started. Each run with collision detection lasted for a time T_0 and statistical uncertainties were calculated from three or more runs. The particle volume fraction was so dilute -typically order $10^{-6\sim-5}$, except for a massive-number-of-particle case in Section 4.1, that only binary collisions were considered.

The collision rate between monodispersed particles per unit volume and time, N_c, is given by

$$N_c = \frac{1}{2} K_c n_p^2, \tag{1}$$

where K_c is the collision kernel and n_p ($=N_p/V_d$, where N_p is the total number of particles and V_d the volume of the computational domain) the droplet number concentration. The collision rate at nth timestep N_c^n is calculated from the number of collision pairs $N_{col.pair}^n$ detected in the domain for a time interval Δt as $N_c^n = N_{col.pair}^n/(V_d \Delta t)$. Thus, the collision kernel at the nth time step, K_c^n, is obtained as

$$K_c^n = 2\frac{N_{col.pair}^n}{n_p^2 V_d \Delta t}. \tag{2}$$

The mean collision kernel, $\langle K_c \rangle$, was calculated by time averaging the collision kernels at each time step.

Wang et al. [42] formulated the collision kernel based on the spherical formulation as

$$\langle K_c(r_1, r_2) \rangle = 2\pi R^2 \langle |w_r(x = R)| \rangle g(x = R), \tag{3}$$

where $\langle \cdots \rangle$ denotes the ensemble average, $R = r_1 + r_2$ the collision radius, $\langle |w_r(x = R)| \rangle$ (simply $\langle |w_r| \rangle$ hereafter) the radial relative velocity at contact, and $g(x = R)$ (simply $g(R)$ hereafter) the radial distribution function, RDF, at contact. The term $g(R)$ represents the clustering effect, which is equal to unity when particles are uniformly distributed.

2.1.2. Code validation for collision statistics

Intense validation tests of the code for collision statistics have been conducted. Firstly conducted is a comparison with the theoretical collision kernel derived by Saffman and Turner [32] for collisions among droplets with no inertia, that is, droplets that follow the flow perfectly and act as fluid tracers. The model of Saffman and Turner [32] gives $K_c(r_1, r_2)/\lambda R^3 = 1.294$, where λ ($=\sqrt{\epsilon/\nu}$, where ϵ is the energy dissipation rate) is the local shear rate. The calculated mean collision kernel for zero-inertia particles with $r_1 = r_2 = 30\ \mu$m for a low Reynolds number flow (refer to case N64 in Table 2 in Section 4)

Table 1
Cases BI and TRI correspond to Figs. 1(a) and (b), respectively. $L_{pq}(= d_{pq}/R$, where $R = 2r$ is the collision radius) in second column denotes the normalized distance between particles p and q in Fig. 1. Errors and number of floating point operations (FPO) shows relative values based on the results from ItrSM.

Case	(L_{12}, L_{13}, L_{23})	Method	$\lvert\mathbf{u}^{[1]}\rvert/V_\infty$	Estimated relative error	Computed relative error	No. of FPO
BI	$(2, -, -)$	ItrSM	0.270	–	–[reference]	1.00[reference]
		OrgSM	0.369	0.333(Eq. (18))	0.368	0.09
		BiSM	0.270	0	0	0.28
TRI	$(2, 30, 30.5)$	ItrSM	0.261	–	–	1.00
		OrgSM	0.345	0.335(Eq. (25))	0.352	0.09
		BiSM	0.262	0.022(Eq. (24))	0.019	0.28

Table 2
Case configurations and typical turbulent statistics. $Re = U_0 L_0/\nu$, u' the rms of flow velocity fluctuation, $k_{max}(=N/2)$ the maximum wavenumber, l_η the Kolmogorov scale, λ the local shear rate and R_λ the Taylor-microscale based Reynolds number. The standard deviation for the analysis period is shown by ±value. The number of droplets were set to $N_p = (N/2)^3$ and the Stokes number was set to 0.4 for all the cases.

	N^3	$L_0[\times 10^{-2}$ m]	Re	u'	$k_{max} l_\eta$	R_λ	N_p	St
N64	64^3	0.500	66.7	1.02	1.76	54.9±0.6	32^3	0.4
N128	128^3	1.00	143	1.02	2.00	81.3±0.8	64^3	0.4
N256	256^3	2.00	360	0.98	2.06	126±1.8	128^3	0.4
N512	512^3	4.00	908	1.01	2.03	207±2.0	256^3	0.4
N1000	1000^3	7.81	2220	1.02	2.00	323±1.7	500^3	0.4
N2000	2000^3	15.6	5590	1.00	2.00	527±5.4	1000^3	0.4

was $\langle K_c \rangle / \lambda R^3 = 1.277$, which gives an error of 1.3%. This result compares well with the 1% numerical uncertainty obtained by Wang et al. [41].

The radial relative velocity at contact $\langle |w_r| \rangle$, and the RDF at contact $g(R)$ are calculated based on the algorithm by Wang et al. [43]. The collision kernel directly obtained from Eq. (2) and the calculated collision kernel from the spherical formulation, i.e., Eq. (3), are compared. For the low Reynolds flow, where R_λ=54.9, the error for St = 0.4 was 2.7%, which is comparable to 1.5% for St=0.4 reported in [43] for $R_\lambda = 45$. Interestingly the error becomes smaller as Reynolds number increases. In the highest Reynolds number flow field (refer to case N2000 in Table 2 in Section 4), where R_λ=527, the error was as small as 0.55%. The reason why the error becomes smaller as Reynolds number increases is not clear yet. Relation (3) is based on the assumption that $\langle |w_r| \rangle$ and $g(R)$ are uncorrelated, something which may not be fully valid when R_λ is small and the flow effectively consists of single-sized eddies. [43] investigated, in their low Reynolds DNS, the origin of the error and found a slightly larger inward particle-flux than outward one at contact distant although the two fluxes should match in statistically steady state. This mismatch may become smaller in higher Reynolds number flows.

Turbulent collision kernels for different St particles at R_λ=54.9 were obtained and compared with the data for $R_\lambda = 54.3$ in [25], where a pseudo-spectral method was employed for flow. The comparison has confirmed good agreements in both mean values and standard deviations (not shown).

2.2. Hydrodynamic interaction (HI)

2.2.1. Physical description

While moving in a flow medium, a particle induces a flow disturbance in its neighborhood. The disturbance may intervene between particles for the so-called hydrodynamic interaction (HI). The particle Reynolds number based on the gravitational settling velocity for cloud droplets in the atmosphere is of the order of 0.01–1.0. It is a good start to assume the disturbance flow to be a Stokes flow. The disturbance flow at $\mathbf{x}$ due to a droplet located at $\mathbf{y}$ can be written as

$$\mathbf{u}_{St}(\mathbf{x}; r, \mathbf{U}_{rel}(\mathbf{y})) = \tag{4}$$

$$\begin{cases} \left[-\frac{3}{4}\frac{r}{d} + \frac{3}{4}\left(\frac{r}{d}\right)^3\right](\mathbf{U}_{rel}(\mathbf{y}) \cdot \mathbf{d})\frac{\mathbf{d}}{d^2} - \left[\frac{3}{4}\frac{r}{d} + \frac{1}{4}\left(\frac{r}{d}\right)^3\right]\mathbf{U}_{rel}(\mathbf{y}) & \text{for } d > r, \\ 0 & \text{for } d \leqslant r, \end{cases} \tag{5}$$

where r is the particle radius, $d(=|\mathbf{d}| = |\mathbf{x} - \mathbf{y}|)$ the distance from the particle center, and $\mathbf{U}_{rel}(\mathbf{y}) = \mathbf{U}(\mathbf{y}) - \mathbf{V}(\mathbf{y})$, where $\mathbf{U}(\mathbf{y})$ is the air flow velocity and $\mathbf{V}(\mathbf{y})$ the particle velocity at the same point $\mathbf{y}$.

Since the Stokes disturbance flows above are governed by a linear equation, they can be superimposed to satisfy the same Stokes equation locally. The superposition method ([28]) stands on this basis. However, Wang et al. [38] pointed out that the original superposition method (OrgSM, hereafter) does not satisfy the no-slip boundary conditions for multiple particles in the system. Ayala et al. [3] (AGW07) then developed an iterative superposition method (ItrSM, hereafter). ItrSM is more reliable but computationally expensive due to its iteration procedure. For example, it was reported that about 95% of the computational time was consumed for the ItrSM in a simulation for a system of 200,000 monodisperse particles in a turbulent flow on a 64^3 grid.

2.2.2. Binary-based superposition method (BiSM)

This study proposes an intermediate method between OrgSM and ItrSM in terms of both computational cost and reliability. The present method is named the binary-based superposition method (BiSM). Firstly we briefly revisit OrgSM and ItrSM, then introduce BiSM.

The disturbance flow field at point $\mathbf{x}$ in a system containing N_p particles is written as

$$\mathbf{u}(\mathbf{x}) = \sum_{k=1}^{N_p} \mathbf{u}_{St}^{[k]}(\mathbf{x}), \tag{6}$$

where

$$\mathbf{u}_{St}^{[k]}(\mathbf{x}) \equiv \mathbf{u}_{St}\left(\mathbf{x}; r^{[k]}, \mathbf{U}_{rel}^{[k]}\right), \tag{7}$$

where superscript k indicates the disturbance due to kth particle. In OrgSM, the particle velocity relative to the air flow is calculated simply as $\mathbf{U}_{rel}^{[\xi]} = \mathbf{U}_{bg}(\mathbf{y}^{[\xi]}) - \mathbf{V}(\mathbf{y}^{[\xi]}) (\equiv \mathbf{U}^*(\mathbf{y}^{[\xi]}))$, where $\mathbf{U}_{bg}$ is the background air flow. In ItrSM, it is instead as $\mathbf{U}_{rel}^{[\xi]} = \mathbf{U}^*(\mathbf{y}^{[\xi]}) + \mathbf{u}(\mathbf{y}^{[\xi]})$ yielding

$$\begin{aligned}\mathbf{u}^{[\xi]} &= \sum_{k\neq\xi}^{N_p} \mathbf{u}_{St}\left(\mathbf{y}^{[\xi]}; r^{[k]}, \mathbf{U}^*(\mathbf{y}^{[k]}) + \mathbf{u}(\mathbf{y}^{[k]})\right) \\ &= \sum_{k\neq\xi}^{N_p} \mathbf{u}_{St}^{(\xi)[k]}, \text{ for } \xi = 1, 2, \cdots, N_p,\end{aligned} \tag{8}$$

where we use an abbreviation as

$$\mathbf{u}_{St}\left(\mathbf{y}^{[\xi]}; r^{[k]}, \mathbf{U}^*(\mathbf{y}^{[k]}) + \mathbf{u}(\mathbf{y}^{[k]})\right) \equiv \mathbf{u}_{St}^{(\xi)}\left(\mathbf{U}^*(\mathbf{y}^{[k]}) + \mathbf{u}(\mathbf{y}^{[k]})\right) \equiv \mathbf{u}_{st}^{(\xi)[k]}. \tag{9}$$

The linearity brings, e.g., $\mathbf{u}_{St}^{(\xi)}\left(\mathbf{U}^*(\mathbf{y}^{[k]}) + \mathbf{u}(\mathbf{y}^{[k]})\right) = \mathbf{u}_{St}^{(\xi)}\left(\mathbf{U}^*(\mathbf{y}^{[k]})\right) + \mathbf{u}_{St}^{(\xi)}\left(\mathbf{u}(\mathbf{y}^{[k]})\right)$. Eq. (8) is a linear system of dimension $3N_p$ (unknown variables). It is not feasible to directly solve, i.e., to calculate the inverse matrix of, this system for large N_p. [3] adopted the Gauss–Seidel method and iteratively solved the system.

In a binary particle case, i.e., in case the system contains only two particles, the disturbance flows are written as

$$\mathbf{u}^{[1]} = \mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[2]} + \mathbf{u}^{[2]}\right), \tag{10}$$

$$\mathbf{u}^{[2]} = \mathbf{u}_{St}^{(2)}\left(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]}\right), \tag{11}$$

which form a set of linear system with six unknown variables. Substitution of Eq. (10) into Eq. (11) yields three equations for the three components of $\mathbf{u}^{[2]}$, which can be mathematically solved with ease. The solutions $\mathbf{u}_{1\leftrightarrow2}^{[1]}$ and $\mathbf{u}_{1\leftrightarrow2}^{[2]}$ satisfy

$$\begin{cases}\mathbf{u}_{1\leftrightarrow2}^{[1]} = \mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[2]}\right) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{1\leftrightarrow2}^{[2]}\right), \\ \mathbf{u}_{1\leftrightarrow2}^{[2]} = \mathbf{u}_{St}^{(2)}\left(\mathbf{U}^{*[1]}\right) + \mathbf{u}_{St}^{(2)}\left(\mathbf{u}_{1\leftrightarrow2}^{[1]}\right).\end{cases} \tag{12}$$

BiSM is based on this solution, assuming that ***interactions via three or more particles are negligible***. Eventually, the solution from BiSM for multiple particles is given by

$$\mathbf{u}^{[\xi]} = \sum_{k\neq\xi}^{N_p} \mathbf{u}_{\xi\leftrightarrow k}^{[k]}. \tag{13}$$

One well-known problem of the superposition method is that it does not account for the lubrication effect, which becomes significant for small separation as $d_{12}/(r_1 + r_2) < 1.1$, where d_{12} is the separation between particles 1 and 2. Very recently, [30] proposed a parameterization based on [16], who derived exact solutions of forces acting on a pair of droplets. The parameterization is for binary systems, and, therefore, it can be adopted in BiSM. If we adopt this parameterization instead of Eq. (12), it may lead to an improvement on BiSM in accuracy. This will be a next step of this study.

2.2.3. Error analysis

Two same-sized particles

Let us think of the binary-particle system shown in Fig. 1(a). The analytical solution of disturbance flow on particle-1 is

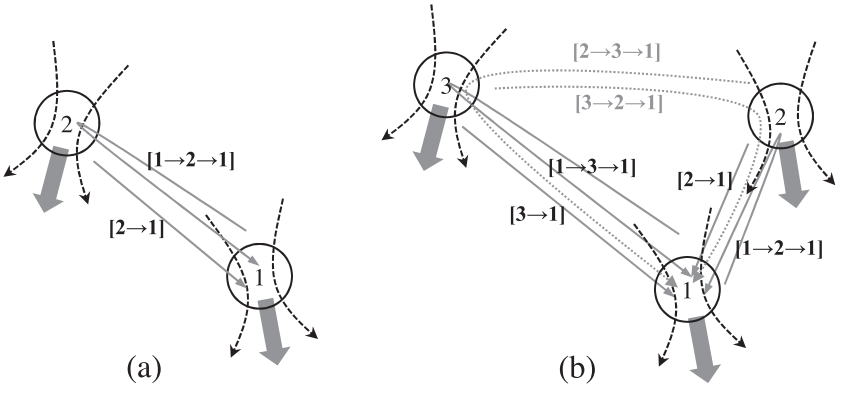

Fig. 1. (a) binary-particle system and (b) triplet-particle system.

$$\mathbf{u}^{[1]} = \mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[2]}\right) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{1\leftrightarrow 2}^{[2]}\right), \tag{14}$$

where $\mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[2]}\right)$ is indicated by the arrow $[2 \to 1]$ and $\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{1\leftrightarrow 2}^{[2]}\right)$ by $[1 \to 2 \to 1]$ in Fig. 1(a). The solution is directly obtained by BiSM and iteratively obtained by ItrSM. Indeed, there is no inherent difference between BiSM and ItrSM in case of binary-particle system: The difference is only in the procedure to obtain the solution, a direct procedure in BiSM but an iterative one in ItrSM. Therefore there is no error in BiSM compared to ItrSM, i.e., $Err^{(BiSM)} = 0$.

There is, however, a significant difference between OrgSM and the other two. The solution from OrgSM is

$$\mathbf{u}^{[1](OrgSM)} = \mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[2]}\right) \tag{15}$$

meaning the arrow $[1 \to 2 \to 1]$ is ignored. Comparison with Eq. (14) shows that the error level of OrgSM on $\mathbf{u}^{[1]}$ is

$$Err^{(OrgSM)}(\mathbf{u}^{[1]}) = \frac{\left|\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{1\leftrightarrow 2}^{[2]}\right)\right|}{\left|\mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[2]}\right) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{1\leftrightarrow 2}^{[2]}\right)\right|}. \tag{16}$$

Roughly speaking, Eq. (4) leads to

$$|\mathbf{u}_{St}(r, \mathbf{v})| \sim A(v/L), \tag{17}$$

where $v = |\mathbf{v}|, L_{12} = d_{12}/R$ and A a positive coefficient with order of 1, i.e., $O(1)$. Considering $\mathbf{u}_{1\leftrightarrow 2}^{[2]} \sim A(\left|\mathbf{U}^{*[2]}\right|/L_{12})$, Eq. (16) becomes

$$Err^{(OrgSM)}(\mathbf{u}^{[1]}) = \frac{A\left(A(\left|\mathbf{U}^{*[2]}\right|/L_{12})/L_{12}\right)}{A(\left|\mathbf{U}^{*[2]}\right|/L_{12}) + A\left(A(\left|\mathbf{U}^{*[2]}\right|/L_{12})/L_{12}\right)} \simeq 1/(1 + L_{12}). \tag{18}$$

This indicates that the error of OrgSM is negligible only for large separation, i.e., $L_{12} \gg 1$.

Three same-sized particles

An example of a system containing three particles is shown in Fig. 1(b). Eq. (8) for this system becomes

$$\begin{aligned}\mathbf{u}^{[1]} &= \sum_{k\neq 1}^{3}\mathbf{u}_{St}^{(1)[k]} = \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[3]} + \mathbf{u}^{[3]}) + \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[2]} + \mathbf{u}^{[2]}) = \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[3]}) + \mathbf{u}_{St}^{(1)}\left(\sum_{k\neq 2}^{3}\mathbf{u}_{St}^{(2)[k]}\right) + \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[2]}) + \mathbf{u}_{St}^{(1)}\left(\sum_{k\neq 3}^{3}\mathbf{u}_{St}^{(3)[k]}\right)\\ &= \underbrace{\mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[3]})}_{[3\to 1]} + \underbrace{\mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[2]})}_{[2\to 1]} + \underbrace{\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(2)}(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]})\right)}_{[1\to 2\to 1]} + \underbrace{\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(3)}(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]})\right)}_{[1\to 3\to 1]} + \underbrace{\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(2)}(\mathbf{U}^{*[3]} + \mathbf{u}^{[3]})\right)}_{[3\to 2\to 1]}\\ &\quad + \underbrace{\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(3)}(\mathbf{U}^{*[2]} + \mathbf{u}^{[2]})\right)}_{[2\to 3\to 1]},\end{aligned} \tag{19}$$

where square brackets under the terms in the RHS correspond to the arrows in Fig. 1(b). BiSM ignores interactions via three or more particles, that is, the arrows $[2 \to 3 \to 1]$ and $[3 \to 2 \to 1]$ in Fig. 1(b), leading to

$$\mathbf{u}^{[1](BiSM)} = \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[3]}) + \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[2]}) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(2)}(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]})\right) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(3)}(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]})\right) \tag{20}$$

$$= \mathbf{u}_{1\leftrightarrow 2}^{[1]} + \mathbf{u}_{1\leftrightarrow 3}^{[1]}. \tag{21}$$

OrgSM solves

$$\mathbf{u}^{[1](OrgSM)} = \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[2]}) + \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[3]}). \tag{22}$$

Comparison between Eqs. (19) and (20) reveals the error of BiSM for a monodisperse system ($r_1 = r_2 = r_3$ and therefore $\left|\mathbf{U}^{*[1]}\right| \sim \left|\mathbf{U}^{*[2]}\right| \sim \left|\mathbf{U}^{*[3]}\right|$) as

$$\begin{aligned}Err^{(BiSM)}(\mathbf{u}^{[1]}) = \Big|&\mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(2)}(\mathbf{U}^{*[3]} + \mathbf{u}^{[3]})\right) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(3)}(\mathbf{U}^{*[2]} + \mathbf{u}^{[2]})\right)\Big| \div \Big|\mathbf{u}_{St}^{(1)}\left(\mathbf{U}^{*[3]}\right) + \mathbf{u}_{St}^{(1)}(\mathbf{U}^{*[2]}) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(2)}(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]})\right)\\ &+ \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(3)}(\mathbf{U}^{*[1]} + \mathbf{u}^{[1]})\right) \quad + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(2)}(\mathbf{U}^{*[3]} + \mathbf{u}^{[3]})\right) + \mathbf{u}_{St}^{(1)}\left(\mathbf{u}_{St}^{(3)}(\mathbf{U}^{*[2]} + \mathbf{u}^{[2]})\right)\Big|\end{aligned} \tag{23}$$

$$\sim \frac{1/(L_{12}L_{23}) + 1/(L_{13}L_{23})}{1/L_{12} + 1/L_{13} + 1/L_{12}^2 + 1/L_{13}^2 + 1/(L_{12}L_{23}) + 1/(L_{13}L_{23})}, \tag{24}$$

where $\mathbf{u}^{[k]} \ll \mathbf{U}^{*[k]}$ and $A^2 \sim A \sim 1$ are assumed. Similarly, the error of OrgSM is estimated as

$$Err^{(OrgSM)}(\mathbf{u}^{[1]}) \sim \frac{1/L_{12}^2 + 1/L_{13}^2 + 1/(L_{12}L_{23}) + 1/(L_{13}L_{23})}{1/L_{12} + 1/L_{13} + 1/L_{12}^2 + 1/L_{13}^2 + 1/(L_{12}L_{23}) + 1/(L_{13}L_{23})}. \tag{25}$$

2.2.4. Validation of BiSM

We have calculated the disturbance flow due to same-sized particles in a binary system and triplet system. (Table 1) shows results on the disturbance flow at the position of 1-particle. Assuming the results from ItrSM to be true, errors of OrgSM and ItrSM can be estimated from the equations in the previous section. For ItrSM, a convergence criteria must be imposed. In this study we follow [3] and set:

$$\frac{\left|u_i^{(k)^{l+1}} - u_i^{(k)^l}\right|}{u_{charact}} \leqslant 10^{-5}, \tag{26}$$

where $u_{charact}$ is a characteristic velocity and $u_i^{(k)^l}$ is the i-component of the disturbance flow velocity at the location of the kth droplet at lth iteration. $u_{charact}$ was set to the gravitational settling velocity for a monodisperse system, while to the differential settling velocity for a bidisperse system.

In the binary case (case-BI), the separation between two particles was set to twice the diameter. BiSM should be, and is indeed, identical to ItrSM in that the two methods adopt different solver but solve the same linear system. OrgSM has 36.8% error in $\left|\mathbf{u}^{[1]}\right|/V_\infty$, which is comparable with the estimate from Eq. (18).

In the triplet case (case-TRI), two of the three particles are closely located and their separation is again twice the diameter. The other particle is separated from the two particles by approximately 30 times the diameter, which is comparable with a mean separation of droplets with 30 μm in radius in atmospheric clouds.

Fig. 2 shows the normalized disturbance flow field, $|\mathbf{u}|/V_\infty$, in case-TRI calculated from (a) ItrSM, (b) OrgSM and (c) BiSM. General shapes of the three results are similar. Closer look, however, reveals that the result from OrgSM is different from the other two. For example, the contour area for $|\mathbf{u}|/V_\infty = 0.05$ in (b) is smaller than those in (a) and (c). BiSM has only 2% error in $\left|\mathbf{u}^{[1]}\right|/V_\infty$, while OrgSM more than 30%. This large error in OrgSM is inevitable even in case-TRI in that it originates from a binary system as in case-BI. ItrSM conducted 12 iterations to obtain the converged solutions in both cases, requiring about four times larger number of floating-point operations (FPO), which is a measure of computational cost, than BiSM. If particles are located closer, the number of required iterations becomes larger. In a many particle system, where some of the particles are in contact with each other, ItrSM requires much larger iterations than the present cases. Then the cost difference between ItrSM and BiSM becomes larger.

The parameter used to measure the HI effect on particle collision is the collision efficiency E_c. For the case of two isolated particles settling in a stagnant flow, E_c is defined as

$$E_c = \frac{y_c^2}{R^2}, \tag{27}$$

where y_c is the far-field, off-center horizontal separation of the approaching trajectory of a smaller particle relative to a larger particle. In most of the cases, the HI tends to make y_c smaller than $R(= r_1 + r_2)$, resulting in $E_c < 1$. In numerical simulations, the trajectories of two particles falling under gravity in a stagnant flow are numerically integrated to obtain E_c from Eq. (27). It should be noted that this definition of E_c is not applicable to the collision efficiency among same-sized droplets, for which an approaching axis cannot be defined since their settling velocities are identical.

In a general approach, a large number of particles are simultaneously considered with many-body interactions, and E_c is then the ratio of number of collisions with the HI to the number of collisions when the HI is ignored, interpreted as

$$E_c = \frac{\langle K_c \rangle [HI]}{\langle K_c \rangle [NoHI]}, \tag{28}$$

where [HI] indicates that the HI is considered and [NoHI] not. This formulation is used in subsection (4.3), in which collision efficiency among same-sized droplets in different Reynolds number flows is discussed.

Fig. 3 shows the collision efficiency E_c, obtained based on Eq. (27), between r_1 and r_2 particles in a stagnant flow. The solid line is from AGW07, which adopts ItrSM. For comparison purpose, the physical constants were set to be consistent with those used in AGW07 for this figure. The results from OrgSM tend to produce larger values than AGW07 and BiSM. The error

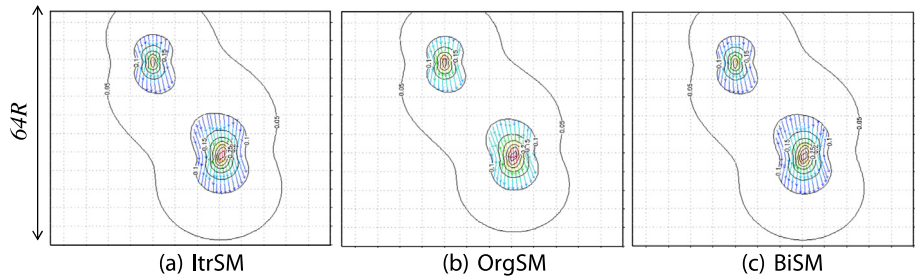

Fig. 2. Disturbance flow normalized by the particle settling velocity in case-TRI obtained from (a) ItrSM, (b) OrgSM and (c) BiSM. Arrow shows the flow direction, and color of the arrow and contour and the magnitude of the disturbance flow.

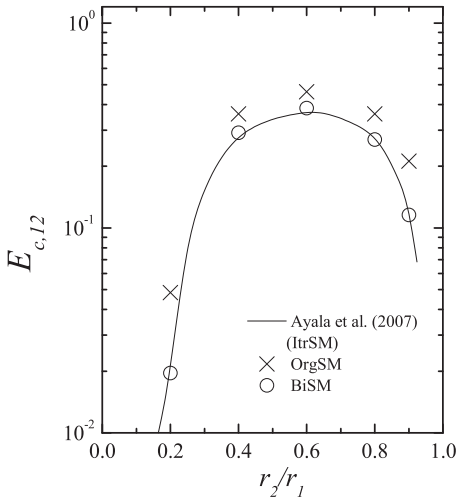

Fig. 3. Collision efficiency between r_1 and $r_2 (< r_1)$ particles in stagnant flow. The larger particle is a water droplet of r_1=25 μ m. The solid line is from Fig. 9 in[3].

is attributed to the > 30% errors shown in Table 1. The consistency among AGW07 and BiSM confirms the reliability of BiSM for collision efficiency calculations in a stagnant flow.

In order to confirm the reliability of BiSM in turbulent flows, this study has conducted two comparisons: (i) comparison between BiSM and ItrSM for the exact same turbulent flow, (ii) comparison with ItrSM in [39].

(i) A collision efficiency for monodispersed particles with St = 0.4 for R_λ=54.9 was obtained based on Eq. (28). The system had a particle volume fraction of $\phi_v = 9.1 \times 10^{-6}$, which is comparable to the typical value of 10^{-5} in atmospheric clouds. The mean collision efficiency $\langle E_c \rangle$ from ItrSM was 0.652, while that from BiSM was 0.650. This confirms that BiSM is as reliable as ItrSM under a typical dilute condition for clouds. With regards to the computational cost, ItrSM required 28 times larger CPU time for the HI calculation than BiSM in this colliding-particle system, while it required only about 4 times larger CPU time in the above cases BI and TRI, where particles are not colliding. The large difference is due to the difference in the number of iterations. In the colliding-particle system the average number of iterations was 40, while it was 12 in the non-colliding systems. When a pair of droplets is colliding, i.e., the separation is very small, a large number of iterations are required by ItrSM. A major problem is that a large number of iterations are required even when there is only one pair of colliding droplets out of many pairs in a many-particle system. BiSM, which does not require iterations, is obviously free from this problem.

(ii) Wang et al. [39] measured collision efficiencies using the iterative approach. For example, collision efficiencies were measured for an isotropic turbulent flow with $R_\lambda = 43$ and $\epsilon = 400$ cm^2/s^3 with bidisperse particles. The collision efficiency of $r_1 - r_1$ collisions for $r_1 = 20$ μm and r_2/r_1=0.9, where the volume fraction $\phi_v = 9.97 \times 10^{-6}$, was (Ec (ItrSM)=)0.812 ± 0.082. We have conducted a comparison by assuming that the $r_2/r_1 = 0.9$ case is virtually a monodisperse case. Computational settings were adjusted to obtain a similar condition with [39]. An isotropic turbulent flow with $R_\lambda = 55$ and $\epsilon = 400$ cm^2/s^3 was obtained using a 64^3 numerical grid. The number of monodisperse droplets with $r_1 = 20$ μm was set to 98,304, which lead to a volume fraction of $\phi_v = 1.03 \times 10^{-5}$(∼$9.97 \times 10^{-6}$ in [39]). The collision efficiencies obtained from BiSM an OrgSM were (Ec (BiSM)=)0.857 ± 0.064 and (Ec (OrgSM)=)1.08 ± 0.08, respectively, where ±value shows the standard deviation obtained from 6 runs, with each run for $30T_0$. This long run duration of $30T_0$ was needed to obtain reliable data with a small statistical error from such a system with small number of particles. Ec (BiSM) agrees with Ec (ItrSM) within the statistical error, while Ec (OrgSM) does not. This further confirms the reliability of BiSM for a turbulent flow.

These two comparisons have confirmed the reliability of BiSM and its advantage in computational cost for turbulent flow cases under dilute conditions as seen in atmospheric clouds.

3. Parallel code for interacting particles in homogeneous isotropic turbulence (PIPIT)

3.1. Flow phase

3.1.1. Governing equations and numerical methods

We solve the three-dimensional continuity and Navier–Stokes equations for incompressible flows;

$$\nabla \cdot \mathbf{U} = 0, \tag{29}$$

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\mathbf{U} = -\nabla P + \frac{1}{Re}\nabla^2 \mathbf{U} + \mathbf{F}(\mathbf{x}, t). \tag{30}$$

Here, Re is the Reynolds number defined as $Re = U_0 L_0/\nu$, where U_0 is a representative velocity, L_0 a representative length, and ν the kinematic viscosity. We consider the case of zero mean flow, and can therefore consider the velocities in the above equations as velocity fluctuations. The last term in the RHS represents the external forcing for achieving a statistically steady state. This study employs the reduced-communication forcing (RCF) ([23]), which is suitable for massively-parallel finite-difference model (FDM), to maintain the energy of motion with $|\mathbf{k}| < 2.5$, where $\mathbf{k}$ is a wavevector.

In our FDM, spatial derivatives are calculated using fourth-order central differences. We employed the conservative scheme of [21] for the advection term, and the second-order Runge–Kutta scheme for time integration. To solve the velocity–pressure coupling we used the HSMAC scheme ([15]), iterating until the RMS of the velocity divergence became smaller than δ/Δ, where Δ is the grid spacing and δ was chosen to be 10^{-3}. The governing equations were discretized on a cubic domain of length $2\pi L_0$, and periodic boundary conditions applied in all three directions. The flow cube was discretized uniformly into N^3 gridpoints, resulting in $\Delta = 2\pi L_0/N$.

3.2. Particle phase

3.2.1. Governing equations and numerical methods

Water droplets are considered as Stokes particles with inertia, governed by the equation

$$\frac{d\mathbf{V}}{dt} = -\frac{1}{\tau_p}(\mathbf{V} - (\mathbf{U}(\mathbf{x}, t) + \mathbf{u}(\mathbf{x}, t))) + \mathbf{g},$$

where $\mathbf{V}$ is the particle velocity, $\mathbf{U}$ the air velocity at a droplet position, $\mathbf{u}$ the disturbance flow velocity due to surrounding droplets, $\mathbf{g} = (-g, 0, 0)$ the gravitational acceleration and τ_p the particle relaxation time defined as $\tau_p = (2/9)(\rho_p/\rho_f)(r^2/\nu)$, where r is the particle radius and ρ_p/ρ_f the ratio of the density of the particle material to that of the fluid. ρ_p/ρ_f was set to 10^3 for a comparison with AGW07 in subSection 2.2, otherwise to 8.43×10^2 at 1 atm and 298 K. The gravity was considered in subSection 2.2, where influence of hydrodynamic interaction on gravitational collision is discussed, but not in Section 4. The neglect of gravity is justified for collisions of monodispersed small water droplets without hydrodynamic interactions ([25]). Although it has not been justified for hydrodynamically-interacting particles, this study uses it for simplicity. The Stokes number, $St = \tau_p/\tau_\eta$, where τ_η is the Kolmogorov time, was set to 0.4. The gravitational acceleration g was set to 9.8 m s^{-2} for the with-gravity case. The second-order Runge–Kutta method was used for time integration. The flow velocity at a droplet position was linearly interpolated from the adjacent grid values. The adoption of the simple linear interpolation has been justified by the comparison with the cubic Hermitian, cubic Lagrangian and fifth order Lagrangian interpolations from ([35]). The PIPIT itself can consider the nonlinear drag if needed. This study, however, employed the linear drag model since the droplet considered was at most 29.6 μm in radius, whose particle Reynolds number Re_p (=$2rV_\infty/\nu$, where $V_\infty(= g\tau_p)$ is the particle settling velocity) is smaller than unity. Turbulence modulation by droplets was assumed negligible because of the high droplet dilution.

3.2.2. Cell-index method

As in Fig. 4, the domain is divided into M_{cell}^3 equal cells and a list of particles in each cell is created ([2]). A particle in (l, m, n)-cell has a chance to collide with the particles inside the neighboring 27 cells, i.e., $([l, l \pm 1], [m, m \pm 1], [n, n \pm 1])$-cells. When sequentially checking the neighboring pairs, the number of cells to be checked for the particles in (l, m, n)-cell is reduced to 14 to avoid duplicated checks. The average number of particles in each cell is $\langle N_{p,cell}\rangle = N_p/M_{cell}^3$, where N_p is the total number of particles. For a particle in a cell, the number of collision pairs with particles inside the same cell is $(\langle N_{p,cell}\rangle - 1)/2$, and the number of collision pairs with particles in the surrounding 13 cells is $13\langle N_{p,cell}\rangle$. The total number of possible combinations to be checked is $M_{cell}^3 \times \langle N_{p,cell}\rangle \times ((\langle N_{p,cell}\rangle - 1)/2 + 13\langle N_{p,cell}\rangle) = N_p(27\langle N_{p,cell}\rangle - 1)/2$. If $\langle N_{p,cell}\rangle$ is kept constant by increasing M_{cell}^3 proportional to N_p, the number of combinations stays of order $O(N_p)$, which is significantly smaller than searching through all possible combinations of particles, which is $N_p(N_p - 1)/2 \sim O(N_p^2)$. This means that the appropriate setting for the cell-index method can, ideally, suppress the increase of cost for particle interactions and make it proportional to $O(N_p)$ rather than $O(N_p^2)$. In reality, however, the overhead for the cell-index method leads the cost being proportional to $O(N_p^\sigma)$, where $1 < \sigma < 2$. This is further discussed later in subSection 4.2.

Decrease of $\langle N_{p,cell}\rangle \left(= N_p/M_{cell}^3 = N_p(\Delta_{cell}/2\pi)^3\right)$ reduces the number of possible combinations to be checked as obvious from the previous paragraph. There are, however, certain restrictions on the setting for $\langle N_{p,cell}\rangle$: (i) Smaller $\langle N_{p,cell}\rangle$ leads to larger overhead, and the overhead becomes not negligible for $\langle N_{p,cell}\rangle \leqslant 1$ as will be discussed in subSection 4.1. (ii) Smaller $\langle N_{p,cell}\rangle$, i.e., larger M_{cell}, requires larger computational memory. Particularly, it becomes a significant problem when M_{cell} is larger than N. (iii) Δ_{cell} should be chosen so that it is greater than the distance that a particle can travel within a time interval in order to ensure that all possible collisions are taken into account. (iv) For hydrodynamic interactions, Δ_{cell} should be larger than the influential length scale of the interaction, which is several tens of times larger than the particle radius (refer to Fig. 2 in AGW07).

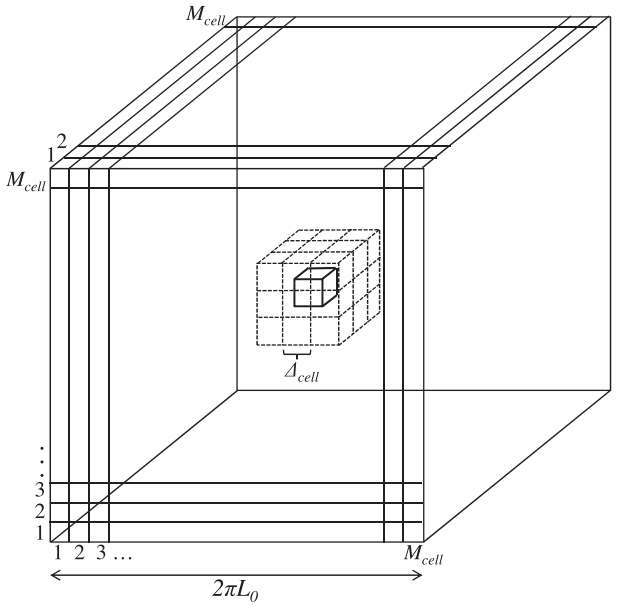

Fig. 4. Illustrative sketch of the cell-index method. Computational domain is divided into fractions of subdomains.

3.3. Distributed-memory parallelization for three-dimensional domain decomposition

3.3.1. Outlook of data communication procedure

The cubic domain of length $2\pi L_0$ is three-dimensionally decomposed as in Fig. 5, which shows the case for $M_x \times M_y \times M_z$-process parallelization. Each process is assigned an MPI (message-passing interface) domain with $n_x \times n_y \times n_z$ flow-grids, where $n_x = N/M_x$, $n_y = N/M_y$ and $n_z = N/M_z$. Note that M_{cell} for the cell-index method is necessarily greater than or equal to max $(M_x,\ M_y,\ M_z)$ for straightforward coding, eventually each MPI process should handle multiple cells.

Data communication among processes should be explicitly coded for the distributed-memory parallelization using the MPI library. In this study, where particle phase in addition to flow phase and furthermore particle interactions -geometric collisions and hydrodynamic interactions (HI)- are considered, we have to code the communication for flow motion, particle motion and particle interactions:

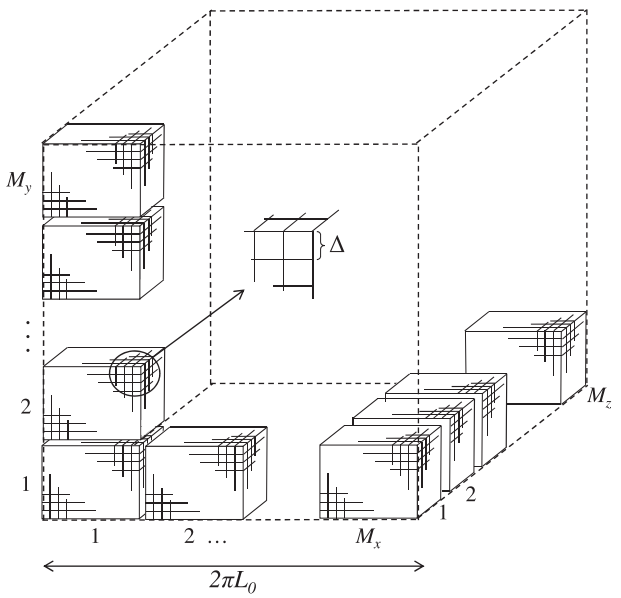

Fig. 5. Three-dimensional domain decomposition for a distributed-memory parallelization.

(i) **Communication for Flow Motion:** Communication of flow velocities and pressure in halos of MPI domains for stencils to be used by finite-difference schemes
(ii) **Communication for Particle Motion:** Transfer of particles moved out of a MPI domain to neighboring MPI domains
(iii) **Communication for Particle Interactions:** Communication for *shadow* particles, beside the boundaries of MPI domains

The *shadow* particle appearing in communication (iii) is the particle used only for particle interactions and its motion is not calculated. Each process calculates the motions of assigned particles, referred to as *real* particles. The solid definition of the *shadow* and *real* particles are described below.

For example, process $a, Proc^a$, calculates the motions of particles inside its domain D^a. Those particles are expressed as $P_i^a = \left\{ P_i | \vec{x}(P_i) \subset D^a \right\}$, where $\vec{x}$ denotes the position vector in a global domain. When a particle P_α^a is in the vicinity of a domain boundary, i.e.,$\vec{x}(P_\alpha^a) \subset D^{a|a\perp b}$, where $D^{a|a\perp b}$ denotes the fraction of area in D^a next to the boundary between D^a and D^b, it may be so close to a particle P_β^b of which $\vec{x}(P_\beta^a) \subset D^{b|a\perp b}$ that they may collide or hydrodynamically-interact. To check the possibility of the interactions between P_α^a and P_β^b , $Proc^a$ needs to know the particles $P_j^b = \left\{ P_j | \vec{x}(P_j) \subset D^b \right\}$. This requires the communication (iii). $Proc^b$ sends the copies of P_j^b, which become the *shadow* particles $P_j^{b\to a}$ for $Proc^a$. The halo region noted in communication (i) contains 4 stencils because PIPIT adopts fourth-order finite-difference schemes for flow calculation. This means each process has flow information for the halo region for extra. Therefore, the halo region for flow can be considered as a buffer region for particles. Even though a particle moves out from an MPI region assigned to a process, the particle tracking can be continued by the same process until the particle moves out from the halo region. This can relax the frequency of the communications (ii) and (iii). That is, it is not necessary to perform the two communications every timestep. In PIPIT, the communications (ii) and (iii) are done every $N_{p,INT}$ timesteps. Finally, the *real* particles for $Proc^a$ are defined as the particles that locate inside D^a, i.e., $P_i | \vec{x}(P_i) \subset D^a$ just before the communication (ii) (Definition of the *real* particle).

3.3.2. Procedures for parallel computing

PIPIT employs the second-order Runge–Kutta (R–K) method. The subroutine for the HI calculation is called at both first and second steps of the R-K method, while that for collision detection only after the second step. The procedures for collision detection are summarized below. The HI calculation additionally requires Procedures (d) and (e) during the (a) R–K steps for particle motion.

1. make a list of neighboring cells of each cell for the cell-index method,
2. Time integration: istep = istep + 1:
   (a) R–K steps for particle motion (each particle R–K step follows each flow R–K step)
   (b) if (mod (istep,$N_{p,INT}$)==0.or. istep==istart) then
      i. remove shadow particles
      ii. exchange real particles crossing MPI boundaries (Communication (ii))
      iii. create a list of particles in each cell
      iv. create a list of shadow particles to be send based on the list
   (c) endif
   (d) copy the listed shadow particles (Communication (iii))
   (e) detect collisions using the cell-index method
3. End of time integration

After Procedure (ii), it is ensured that all the real particles are inside each MPI domain, i.e., no real particles are in halo regions. The neighboring list must be created for the real particles in this state.

In time steps when the if-sentence in Procedure (b) is false, some real particles may move on halo regions. Due to the fourth-order finite-difference method, each process have flow information in halo regions, whose width is 3.5Δ (not 4Δ since PIPIT adopts a staggered grid system). In order to reduce the overhead for the cell-index method, it is better to have a large $N_{p,INT}$. However, there is a restriction that Procedure (ii) should be undertaken before some particles move out from the halo region. Assuming the large-scale particle motion follows the flow, maximum travel length of particles for $N_{p,INT}$ time steps are $u_{max}N_{p,INT}\Delta t$. This must not exceed the width of the halo region, i.e., $N_{p,INT}u_{max}\Delta t < 3.5\Delta$. This yields $N_{p,INT} < 3.5/CFL_{max}$, where $CFL_{max} = \Delta/(u_{max}\Delta t)$ is the flow CFL number. This study typically sets CFL_{max} and $N_{p,INT}$ as 0.3 and 8, respectively.

4. Numerical results and discussion

4.1. Optimal computational conditions for particle interactions

There is a flexibility in setting the total number of particles in the domain N_p and the size of cell, Δ_{cell}. Larger N_p would be better in terms of smaller number of time steps required for collision statistics in that the number of collisions is

proportional to N_p^2, while the simulation cost is ideally proportional only to N_p as far as $\langle N_{p,cell}\rangle$ is kept small. However, finite computer resource limits the available N_p. Moreover, smaller $\langle N_{p,cell}\rangle$, i.e., smaller Δ_{cell} and larger M_{cell}, requires larger overhead for the cell-index method and larger computational memory. In short, larger N_p and smaller $\langle N_{p,cell}\rangle$ are ideally preferable but there are indeed restrictions for them.

In order to seek a preferable computational setting for simulations of interacting particles, we measured the elapsed time for the simulations in different settings for particles under a fixed setting for flow. The elapsed time was divided into four tasks;

1. time integration of flow motion (T_{flow}),
2. time integration of particle motion ($T_{p.motion}$),
3. collision detection ($T_{p.col}$), and
4. hydrodynamic interaction calculation ($T_{p.HI}$).

The time for the overhead for cell-index method and accompanying data communications were equally distributed into $T_{p.col}$ and $T_{p.HI}$. The times for initialization, file input and output were discarded. Since computational setting for flow was fixed, T_{flow} can be considered as constant although it had relative deviations of about 3% due to measuring errors, which seems inevitable. The domain was discretized onto 256^3 grid points and decomposed into 4^3 MPI domains. The elapsed time for 100 time steps was measured for different combinations of number of particles N_p and $\langle N_{p,cell}\rangle$.

Fig. 6(a) shows the elapsed time for the time integration of particle motion, $T_{p.motion}$, normalized by T_{flow}, in different N_p and $\langle N_{p,cell}\rangle$. As anticipated, $T_{p.motion}$ increases proportional to N_p and it is insensitive to $\langle N_{p,cell}\rangle$. Fig. 6(b) and (c) show the elapsed times for collision detection and HI calculation, i.e., $T_{p.col}$ and $T_{p.HI}$, respectively. The estimate in subSection 3.2.2 anticipates that smaller $\langle N_{p,cell}\rangle$ leads to smaller $T_{p.col}$ and $T_{p.HI}$. However, Fig. 6(b) and (c) show that the balancing with larger overhead for smaller $\langle N_{p,cell}\rangle$ results in an optimal $\langle N_{p,cell}\rangle$ between 0.1 and 1.

4.2. Computational performance

Figs. 7(a) and (b) show the actual CPU times for the four tasks for different number of particles on the same number of flow grids. For the cell-index method, Δ_{cell} was fixed to 4Δ in Fig. 7(a), while $\langle N_{p,cell}\rangle$ was fixed to unity in Fig. 7(b). Comparison between Figs. 7(a) and (b) confirms that the required CPU time for particle motion is independent of the setting for the cell-index method. In Fig. 7(a) the required CPU times for collision detection and HI increases proportional to N_p^2, while those in (b) to N_p^σ, where $1<\sigma<2$. This suggests that the fixed $\langle N_{p,cell}\rangle$ is better than fixed Δ_{cell} in terms of the required CPU time for increasing number of particles. Fig. 13 in AGW07 showed the HI calculation required a CPU time proportional to N_p^2, implying that AGW07 might have adopted fixed Δ_{cell}.

According to Fig. 12 in AGW07, the HI calculation in an ItrSM simulation for 200,000 particles on 64^3 flow grids takes 95% of the total CPU time. A simulation for 256^3 particles on 256^3 flow grids can be compared with the case in AGW07 since roughly speaking $200,000/64^3 \sim 256^3/256^3$. The HI calculation by BiSM took only 65% of the total CPU time. This indicates that the HI calculation by ItrSM in AGW07 required about 10 times larger CPU time than that by BiSM as far as the rest of the calculation was assumed to require the same CPU time. This estimate roughly matches with the observation in subSection 2.2.4 that ItrSM required 27 times larger CPU time for the HI calculation. Therefore, it can be concluded that BiSM requires $O(10)$ times less calculation than ItrSM.

Fig. 8 shows elapsed time per time step for different number of flow grids with different number of particles. The simulations included both the collision detection and hydrodynamic interaction calculation. An SGI Altix4700 system was used for two 256^3-grid calculations, an SGI ICEX system for a 512^3-grid calculation, and the Earth Simulator 2 for a $1{,}024^3$-grid calculation. The peak performances per core of these systems are 6.4 GFLOPS, 20.8 GFLOPS and 102.4GFLOPS, respectively. In all the cases, the total elapsed-time decreases linearly with increasing number of cores. This confirms good linearities of PIPIT on multiple supercomputer systems, exhibiting the inherent good parallel performance of PIPIT. It should also be noted that, as shown in the previous subsection, most of the time was spent for flow calculation in the case for 256^3 grids with 64^3 particles, while mostly for particle interactions in the case for 256^3 grids with 256^3 particles. The observed good parallel performance in both cases implies good parallel performances of PIPIT in both flow and particle calculations. Note that the so-called superlinearity is observed at several points, for example, 128→256 cores in the case for 256^3 grids with 64^3 particles and and 1024→2048 cores in the case for 512^3 grids with 512^3 particles. The superlinearity is thought to be due to, and dependent on, cache hierarchies and network performances of the systems (e.g., [18]).

4.3. Reynolds dependencies on collision statistics

4.3.1. Computational setting

Table 2 shows the typical turbulence statistics as well as the configurations for flow calculations. The Taylor-microscale based Reynolds number, R_λ, obtained in this study ranges from 54.9 to 527. This wide range of R_λ enables us to discuss the Reynolds dependency of collision statistics. The standard deviation for the analysis period indicated by ±value in the R_λ column represents that the flow statistics deviate by approximately 1% relative to the mean value.

Considering the discussion in subSection 4.1, the number of droplets were set to $N_p = (N/2)^3$, and the cell size was set to twice the grid size for $\langle N_{p,cell}\rangle$ to be unity.

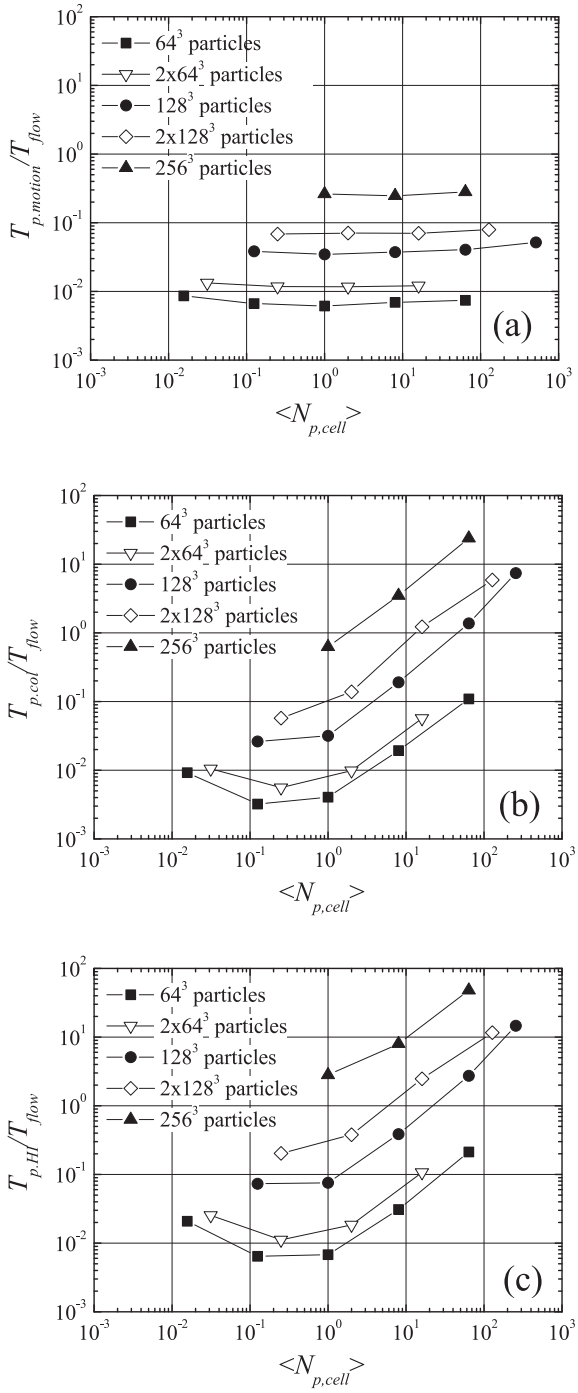

Fig. 6. Relative computational time for 256^3 grids domain on 64 processors; Elapsed time for (a) particle motion, (b) collision detection and (c) hydrodynamic interaction (BiSM) relative to that for flow motion.

The same time interval dt was used for flow and particle motions. This study determined dt so that the flow CFL number was kept below 0.3, and dt/τ_p ranged from 0.0065 (N2000) to 0.063 (N64), which also satisfied the restriction $dt \leqslant 0.15\tau_p$ for HI calculation ([3]).

4.3.2. Collision kernel and efficiency among same-sized droplets

Fig. 9(a) shows the mean collision kernel normalized by λR^3 obtained from the present code, PIPIT, together with some other DNS data and model predictions in literature. The error bars show ±one standard deviation. On the whole the deviation

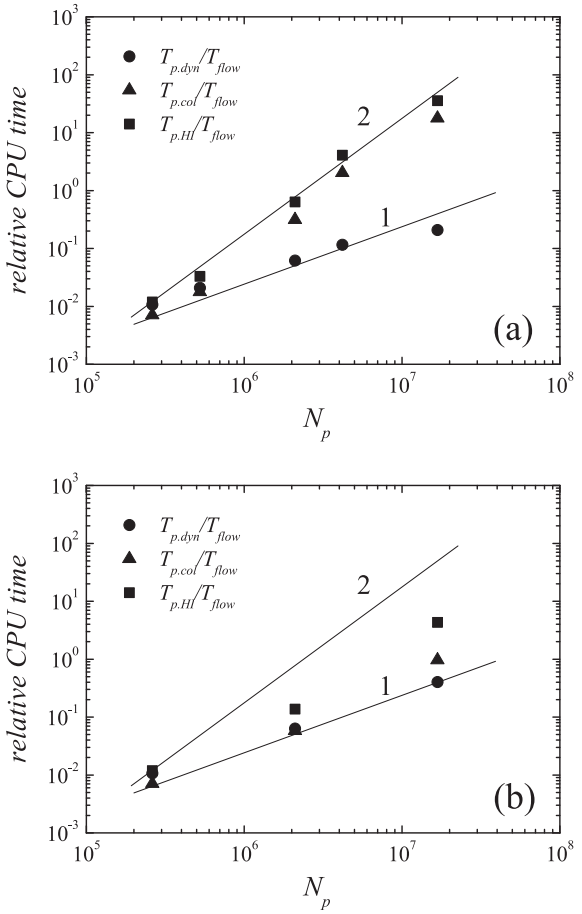

Fig. 7. Actual CPU times for the different computational tasks for 256^3-flow-grid simulations on 64 processors containing different number of particles. For the cell-index method, Δ_{cell} was fixed to 4Δ in (a), while $\langle N_{p.cell} \rangle$ was fixed to unity in (b). The lines with gradient 1 in (a) and (b) are identical. So are the lines with gradient 2.

becomes smaller as R_λ increases because larger number of droplets were calculated for larger R_λ. The deviation, however, does not monotonically decrease due to the fact that some deviation in flow statistics is inevitable as shown in the column of R_λ in Table 2. The mean values from PIPIT agree with the other DNS data in the low R_λ range. The predictions by models, except for the model by [32] for $St \ll 1$, are comparable to the present result on the whole. These facts provide extra confidence in PIPIT.

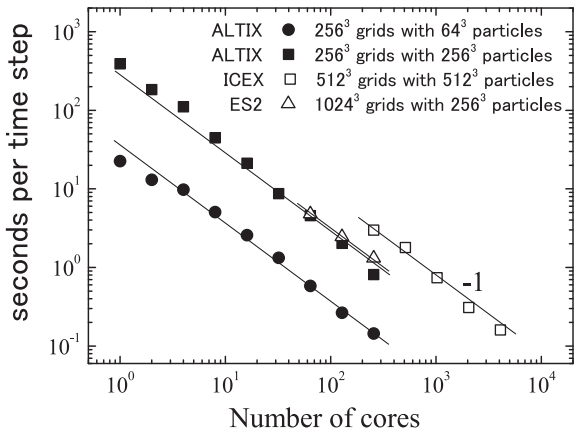

Fig. 8. Wall clock time versus number of cores for different number of particles and flow grids on three different supercomputer systems.

It should be noted that the present result shows that the normalized collision kernel increases for $R_\lambda < 100$ and then decreases as R_λ increases for $R_\lambda > 100$. Many authors ignore the Reynolds dependency and assume a constant collision kernel irrespective of R_λ ([32,10,47]) or assume a convergence ([4]). However, the present result shows a clear R_λ dependence. This may result in a significant difference in atmospheric clouds, where R_λ ranges from 10^3 to 10^5. We have confirmed that the collision kernel for $St = 0.1$ particles converges at around $R_\lambda \sim 80$ and then become constant for larger R_λ (not shown). That is, the assumption of constant collision kernel against R_λ is valid only for very small St.

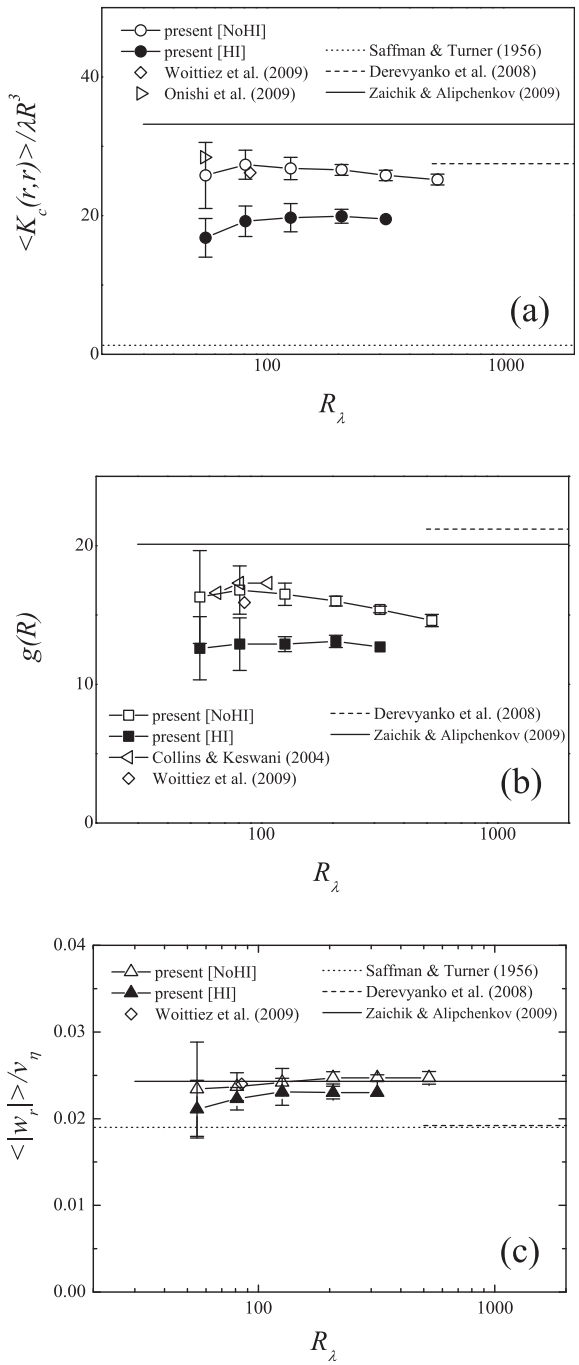

Fig. 9. (a) Collision kernel, (b) radial distribution function and (c) radial relative velocity at contact for $St = 0.4$ plotted against the Taylor-microscale based Reynolds number R_λ. White plots are for the cases without hydrodynamic interaction (HI) and solid plots with HI. Model predictions (lines) ([32,10,47]) and DNS results (plots) ([8,25,44]) in literature are also drawn for comparison. The error bars show ±one standard deviation from more than three runs. The plot for N1000[HI] does not show a error bar in that only one run was performed for it.

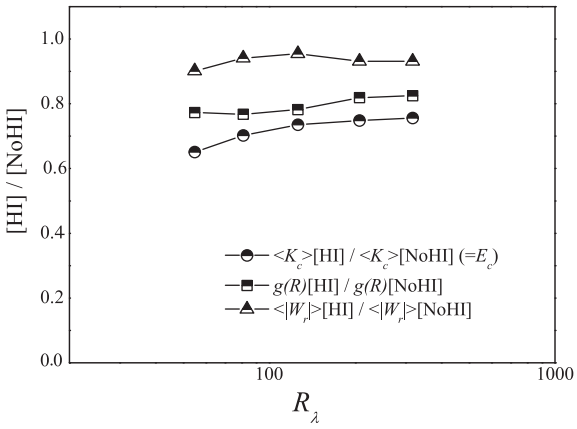

Fig. 10. Ratios of collision kernel, radial distribution function and radial relative velocity at contact, K_c, $g(R)$ and $\langle|w_r|\rangle$ respectively, with hydrodynamic interactions (HI) to those without HI for $St = 0.4$. $\langle K_c\rangle[HI]/\langle K_c\rangle[NoHI]$ represents the collision efficiency E_c. Data are calculated from the mean values in Fig. 9.

Figs. 9(b) and (c) show the radial distribution function and radial relative velocity at contact; $g(R)$ and $\langle|w_r|\rangle$, respectively, defined in Eq. (3), as function of R_λ. For atmospheric relevance, we assume that the Reynolds number is increased by increasing the integral scale so as to keep l_η and R as the same values as R_λ increases. The radial relative velocity is normalized by the Kolmogorov velocity $v_\eta(=(\epsilon\nu)^{1/4})$. Reynolds dependency of $g(R)$ is very similar to that of the collision kernel. In contrast, $\langle|w_r|\rangle$ shows little Reynolds dependency. Therefore, the decrease in collision kernel for $R_\lambda > 100$ is attributable to the decrease in $g(R)$. The decrease could be explained by introducing the concept of 'locality'. The St is normally a globally-averaged value, i.e., the value averaged over the whole domain and time. The 'local' St deviates in space and time and the deviation becomes larger for larger intermittency. If the 'local' $g(R)$ is simply proportional to the log of the local St, the global $g(R)$ will stay the same even for larger intermittency. But if not, the global $g(R)$ can be different for different intermittency. Of course, we anticipate that the latter happens in our data. We address the decrease in $g(R)$ with increasing Reynolds number in detail in a future report.

Figs. 9 includes the results from the cases with HI. In those cases, the flow disturbance due to a particle is ignored when the distance is larger than a truncation radius H, i.e., $u_{St}^{[\xi]}(\mathbf{x}) = 0$ for $|\mathbf{x} - \mathbf{y}^{[\xi]}|/r^{[\xi]} > H$, where H was set to 30 according to AGW07. The cases with HI all show smaller values than those without HI, implying the hydrodynamic interactions prevent collision events.

Fig. 10 shows the ratios of $\langle K_c\rangle$, $g(R)$ and $\langle|w_r|\rangle$ with HI to those without HI. The ratio of $\langle K_c\rangle$ with HI to that without HI represents the collision efficiency E_c. The radii of $St = 0.4$ droplets in this study range from 25 to 30 μm: For example, according to Fig. 3 the collision efficiency between $r_1 = 25$ and $r_2 = 22.5$ μm droplets in a stagnant flow is about 0.1. The collision efficiency in Fig. 10 is, in contrast, over 0.65. This supports the suggestion that the turbulence increases the collision efficiency among same-sized droplets. The collision efficiency increases from 0.65 to 0.75 as R_λ increases for $R_\lambda < 200$ and then saturates. The saturation is also seen in both the ratios of $\langle|w_r|\rangle[HI]/\langle|w_r|\rangle[NoHI]$ and $g(R)[HI]/g(R)[NoHI]$.

Lastly, we would like to emphasize that the Reynolds dependency for high R_λ has been discussed for the first time here because the data obtained in this study straddle wide range of R_λ from 54.9 up to 527.

5. Concluding remarks

This study has developed a parallel code for the simulation of interacting droplets in stationary homogeneous isotropic turbulence -we name the code PIPIT (Parallel code for Interacting Particle in homogeneous Isotropic Turbulence). Air turbulence is calculated using a fourth-order finite-difference scheme, while droplet motions are tracked by the Lagrangian method. PIPIT employs the MPI (message-passing interface) library for distributed-memory parallelization and can decompose the domain in three directions. It is designed to minimize data communications for efficient parallel computing through adopting (i) the finite-difference model (FDM) for flow, (ii) the reduced-communication forcing (RCF) ([23]) for achieving statistically-steady flows and (iii) the halo region for FDM as a buffer for particle calculation leading to a dramatic relaxation of data communication frequency.

The FDM with RCF does not require all-to-all data communication, which is required in spectral models, and, therefore, can achieve high-parallel efficiencies in flow simulations ([23]). Moreover the halo region required for finite-differenciation in FDM can be used as the buffer region for particle calculations, i.e., even though a particle moves out from an MPI domain, the particle can be tracked by the same MPI process as far as it stays within the halo region. This can reduce the data communication frequency for particle calculations. Indeed, the communication was required only once in eight time steps in this study. Notable schemes in order to cut the computational cost for particle interaction simulations have also been implemented in PIPIT; (a) the cell-index method ([2]) and (b) the binary-based superposition method (BiSM), which has been developed in this study. The BiSM is based on the concept of the superposition method ([28]), but with more reliability

comparable to the iterative superposition method (ItrSM) by Ayala et al. [3]. Under a typical dilute condition as in atmospheric clouds, the computational cost of BiSM is smaller by order of 10 than that of ItrSM, whereas an error of BiSM compared to ItrSM is insignificant. Coupling of the cell-index method with BiSM can reduce the computational cost for particle interactions to $O(N_p^\sigma)$, where N_p is the total number of particles and $1 < \sigma < 2$, from $O(N_p^2)$. PIPIT is therefore a promising tool for investigating particle interactions in high Reynolds number flows.

We have confirmed the anticipated features of PIPIT: It has been confirmed that PIPIT has a good parallel efficiency, i.e. elapsed time linearly decreases with increasing number of processors. We have found a good performance while keeping the average number of droplets per cell-box in the order of $O(10^{-1\sim0})$. To obtain collision statistics with less elapsed-time, the larger total number of particles N_p is preferable in that number of collisions increases proportionally to N_p^2 while computational cost only proportionally to N_p^σ with $1 < \sigma < 2$. However, the limited computer memory restricts the available number of droplets. We found that $N_p = (N/2)^3$, where N^3 is the number of grid points, is a good compromise for water droplet simulations in air turbulence. The memory size required for particle calculations was then similar to those for flow calculations. If we had used larger number of droplets, we would have had to worry not only about triplet collisions but also about the huge memory demand for particle interaction calculations, which would have largely limited the attainable Reynolds number.

The largest simulation in this study computed the flow on 2000^3 grids with 1000^3 (=one billion) Lagrangian inertial droplets. The attained Taylor-microscale based Reynolds number, R_λ, ranges from 54.9 up to 527. This wide range of R_λ has enabled us to investigate the Reynolds dependency of collision statistics. This study has targeted $St = 0.4$ particles, which corresponds to cloud droplets with $25 \sim 30$ μm in radius in atmosphere. The PIPIT results have revealed that the collision kernel increases for R_λ<100 and then starts to decrease for higher R_λ as R_λ increases. This Reynolds dependency is attributed to the Reynolds dependency of clustering effect, which shows a similar trend.

In addition, the collision efficiency among same-sized droplets with $St = 0.4$ has been obtained using the newly developed binary-based superposition method (BiSM). It has been confirmed that the collision efficiency is increased by turbulence. The data for this paper's wide range of R_λ has shown that the collision efficiency increases from 0.65 to 0.75 as R_λ increases for R_λ<200 and then saturates.

Many authors have assumed a constant collision kernel irrespective of Reynolds number ([32,10,47]) or its convergence to an asymptotic constant value ([4]). However, the present data reveal a Reynolds number dependence of the radial distribution function at contact.

Acknowledgments

The large-size numerical simulations presented were carried out on the Earth Simulator 2 supercomputer in the Japan Agency for Marine-Earth Science and Technology. This research was conducted while Ryo Onishi was receiving the JSPS postdoctoral fellowship for research abroad.

Appendix A

List of symbols

d	Distance from a particle center ($= \|\mathbf{d}\|$) (m)
d_{ij}	Separation between particles i and j (m)
D^a	Computational domain assigned to the MPI process a
E_c	Collision efficiency
ϵ	Energy dissipation rate (m^2s^{-3})
g	Gravitational acceleration ($m\ s^{-2}$)
$g(R)$	Radial distribution function at contact ($= g(x = R)$)
K_c	Collision kernel (m^3s^{-1})
l_λ	Taylor microscale (m)
L_0	Representative length scale (m)
L_{ij}	Separation between particles i and j normalized by R
M_{cell}^3	Number of cells in the computational domain
M_x, M_y, M_z	Number of processes
n_p	Particle number density (m^{-3})
n_x, n_y, n_z	Number of flow grids assigned to each MPI process
N^3	Number of grids in the computational domain
N_c	Collision frequency ($m^{-3}s^{-1}$)
N_p	Total number of particles
$N_{p,cell}$	Average number of particles in each cell
$N_{p,INT}$	Interval of main MPI communication

(*continued on next page*)

r_i	Particle radius i (m)
R	Collision radius $(= r_1 + r_2)$ (m)
R_λ	Taylor-microscale based Reynolds number $(= u' l_\lambda / \nu)$
S	Separation distance between a pair of particles selected for collision detection (m)
St	Stokes number $(= \tau_p / \tau_\eta)$
t	Time (s)
T	Integral time scale of turbulence (s)
T_0	Representative time scale (s)
T_{flow}	Elapsed time for the simulation of flow motion (s)
$T_{p.col}$	Elapsed time for collision detection (s)
$T_{p.HI}$	Elapsed time for hydrodynamic interaction calculation (s)
$T_{p.motion}$	Elapsed time for the simulation of particle motion (s)
u'	Root-mean-square of velocity fluctuations (m s^{-1})
$\mathbf{u}$	Disturbance flow field (m s^{-1})
$\mathbf{u}_{St}$	Stokes disturbance flow (m s^{-1})
$\mathbf{u}^{[i]}_{i \hookrightarrow j}$	Analytical disturbance flow at the position of i-particle in a binary system (i and j-particles) (m s^{-1})
U_0	Representative velocity scale (m s^{-1})
$\mathbf{U}$	Air flow velocity (m s^{-1})
$\mathbf{U}^*$	Relative velocity between the background air flow and particle $(= \mathbf{U}_{bg} - \mathbf{V})$ (m s^{-1})
$\mathbf{U}_{bg}$	Background air flow velocity at a particle center position (m s^{-1})
$\mathbf{U}_{rel}$	Relative flow velocity to particle velocity $(= \mathbf{U} - \mathbf{V})$ (m s^{-1})
$\mathbf{V}$	Particle velocity (m s^{-1})
V_d	Volume of the computational domain (m^3)
V_∞	Particle settling velocity (m s^{-1})
v_η	Kolmogorov velocity $\left(= (\epsilon\nu)^{1/4}\right)$ (m s^{-1})
w_r	Radial relative velocity at contact $(= w_r(x = R))$ (m s^{-1})
y_c	Far-field, off-center horizontal separation of the approaching trajectory of a smaller particle relative to a larger particle (m)

Greek symbols and special notations

Δ_{cell}	Size of the cell (m)
Δt	Time interval (s)
ϵ	Energy dissipation rate ($m^2 s^{-3}$)
λ	Local shear rate $(= \sqrt{\epsilon/\nu})$ (s^{-1})
ν	Kinematic viscosity (m s^{-2})
ρ_p	Particle mass density (kg m^{-3})
ρ_f	Fluid (air) mass density (kg m^{-3})
τ_p	Particle relaxation time (s)
τ_η	Kolmogorov time $(= \sqrt{\nu/\epsilon})$ (s)
[HI]	with hydrodynamic interaction
[NoHI]	without hydrodynamic interaction
$\langle \cdots \rangle$	Mean value

Acronyms

AGW07	[3]
BiSM	Binary-based Superposition Method
DNS	Direct Numerical Simulation
FDM	Finite-Difference Model
FPO	Floating-Point Operations
GFLOPS	Giga FLoating-point Operations Per Second
HI	Hydrodynamic Interaction
ItrSM	Iterative Superposition Method (=the improved superposition method in [3])
MPI	Message-Passing Interface
OrgSM	Original Superposition Method ([28])
PIPIT	Parallel code for Interacting Particles in homogeneous Isotropic Turbulence
PSM	Pseudo-Spectral Model
RCF	Reduced Communication Forcing ([23])
R–K	Runge–Kutta

References

[1] J. Abrahamson, Collision rates of small particles in a vigorously turbulent fluid, Chem. Eng. Sci. 30 (11) (1975) 1371–1379.
[2] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Oxford University Press, 1987.
[3] O. Ayala, W.W. Grabowski, L.-P. Wang, A hybrid approach for simulating turbulent collisions of hydrodynamically-interacting particles, J. Comput. Phys. 225 (1) (2007) 51–73.
[4] O. Ayala, B. Rosa, L.-P. Wang, Effects of turbulence on the geometric collision rate of sedimenting droplets. Part 2: Theory and parameterization, New J. Phys. 10 (7) (2008) 075016.
[5] O. Ayala, B. Rosa, L.-P. Wang, W.W. Grabowski, Effects of turbulence on the geometric collision rate of sedimenting droplets. Part 1. Results from direct numerical simulation, New J. Phy. 10 (7) (2008) 075015.
[6] A.M. Blyth, Entrainment in cumulus clouds, J. Appl. Meteorol. 32 (1993) 626–641.
[7] S.W. Coleman, J.C. Vassilicos, A unified sweep-stick mechanism to explain particle clustering in two- and three-dimensional homogeneous, isotropic turbulence, Phys. Fluids 21 (11) (2009) 113301.
[8] L.R. Collins, A. Keswani, Reynolds number scaling of particle clustering in turbulent aerosols, New J. Phys. 6 (1) (2004) 119.
[9] V. Dallas, J.C. Vassilicos, Rapid growth of cloud droplets by turbulence, Phys. Rev. E 84 (4) (2011) 1–5.
[10] S. Derevyanko, G. Falkovich, S. Turitsyn, Evolution of non-uniformly seeded warm clouds in idealized turbulent conditions, New J. Phys. 10 (2008) 075019.
[11] G. Falkovich, A. Pumir, Sling effect in collisions of water droplets in turbulent clouds, J. Atmos. Sci. 64 (12) (2007) 4497–4505.
[12] C.N. Franklin, P.A. Vaillancourt, M.K. Yau, Statistics and parameterizations of the effect of turbulence on the geometric collision kernel of cloud droplets, J. Atmos. Sci. 64 (3) (2007) 938–954.
[13] C.N. Franklin, P.A. Vaillancourt, M.K. Yau, P. Bartello, Collision rates of cloud droplets in turbulent flow, J. Atmos. Sci. 62 (7) (2005) 2451–2466.
[14] W.W. Grabowski, L.-P. Wang, Diffusional and accretional growth of water drops in a rising adiabatic parcel: effects of the turbulent collision kernel, Atmos. Chem. Phys 9 (2009) 2335–2353.
[15] C.W. Hirt, J.L. Cook, Calculating three-dimensional flows around structures and over rough terrain, J. Comp. Phys. 10 (1972) 324–340.
[16] D.J. Jeffrey, Y. Onishi, Calculation of the resistance and mobility functions for two unequal rigid spheres in low-reynolds-number flow, J. Fluid Mech. 139 (1984) 261–290.
[17] S.K. Krueger, C.W. Su, P.A. McMurtry, Modeling entrainment and finescale mixing in cumulus clouds, J. Atmos. Sci. 54 (23) (1997) 2697–2712.
[18] S. Laizet, N. Li, Incompact3d: a powerful tool to tackle turbulence problems with up to o(10^5) computational cores, Int. J. Numer. Methods Fluids 67 (2011) 1735–1757.
[19] J. Lu, H. Nordsiek, E.W. Saw, R.A. Shaw, Clustering of charged inertial particles in turbulence, Phys. Rev. Lett. 104 (18) (2010) 184505.
[20] M.R. Maxey, The gravitational settling of aerosol particles in homogeneous turbulence and random flow fields, J. Fluid Mech. 174 (1987) 441–465.
[21] Y. Morinishi, T. Lund, O. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, J. Comput. Phys. 143 (1) (1998) 90–124.
[22] R. Onishi, Numerical simulations of chermical reaction and droplet growth in environmental turbulent flows, Ph.D. thesis, Kyoto University, 2005.
[23] R. Onishi, Y. Baba, K. Takahashi, Large-scale forcing with less communication in finite-difference simulations of stationary isotropic turbulence, J. Comput. Phys. 230 (10) (2011) 4088–4099.
[24] R. Onishi, H. Takagi, K. Takahashi, S. Komori, Turbulence effects on cloud droplet collisions in mesoscale convective clouds, Turbulence Heat Mass Trans. 5 (2006) 709–712.
[25] R. Onishi, K. Takahashi, S. Komori, Influence of gravity on collisions of monodispersed droplets in homogeneous isotropic turbulence, Phys. Fluids 21 (2009) 125108.
[26] R. Onishi, K. Takahashi, S. Komori, High-resolution simulations for turbulent clouds developing over the ocean, Gas Trans. Water Surf. 2010 (6) (2011) 582–592.
[27] M. Pinsky, A. Khain, M. Shapiro, Collisions of small drops in a turbulent flow. Part I: Collision efficiency. problem formulation and preliminary results, J. Atmos. Sci. 56 (15) (1999) 2585–2600.
[28] H.R. Pruppacher, J.D. Klett, Microphysics of Clouds and Precipitation, Springer, 1998.
[29] W.C. Reade, L.R. Collins, Effect of preferential concentration on turbulent collision rates, Phys. Fluids 12 (2000) 2530.
[30] B. Rosa, L-P. Wang, M.R. Maxey, W.W. Grabowski, An accurate and efficient method for treating aerodynamic interactions of cloud droplets, J. Comput. Phys. 230 (2011) 8109–8133.
[31] B. Rosa, H. Parishani, O. Ayala, L.-P. Wang, W. Grabowski, High-resolution simulation of turbulent collision of cloud droplets, Lecture Notes in Computer Science, 7204, 2012, 401–10.
[32] P.G. Saffman, J.S. Turner, On the collision of drops in turbulent clouds, J. Fluid Mech. 1 (01) (1956) 16–30.
[33] E.W. Saw, R.A. Shaw, S. Ayyalasomayajula, P.Y. Chuang, A. Gylfason, Inertial clustering of particles in high-reynolds-number turbulence, Phys. Rev. Lett. 100 (21) (2008) 214501.
[34] R.S.R. Sidin, R.H.A. Ijzermans, M.W. Reeks, A Lagrangian approach to droplet condensation in atmospheric clouds, Phys. Fluids 21 (10) (2009) 106603.
[35] S. Sundaram, L. Collins, Numerical considerations in simulating a turbulent suspension of finite-volume particles, J. Comput. Phys. 124 (2) (1996) 337–350.
[36] S. Sundaram, L.R. Collins, Collision statistics in an isotropic particle-laden turbulent suspension. Part 1: Direct numerical simulations, J. Fluid Mech. 335 (1997) 75–109.
[37] S.C. Van Den Heever, W.R. Cotton, Urban aerosol impacts on downwind convective storms, J. Appl. Meteor. Climatol. 46 (6) (2007) 828–850.
[38] L.-P. Wang, O. Ayala, W.W. Grabowski, Improved formulations of the superposition method, J. Atmos. Sci. 62 (4) (2005) 1255–1266.
[39] L.-P. Wang, O. Ayala, B. Rosa, W.W. Grabowski, Turbulent collision efficiency of heavy particles relevant to cloud droplets, New J. Phys. 10 (7) (2008) 075013.
[40] L.-P. Wang, B. Rosa, H. Gao, G. He, G. Jin, Turbulent collision of inertial particles: Point-particle based, hybrid simulations and beyond, Int. J. Multiphase Flow 35 (9) (2009) 854–867.
[41] L.-P. Wang, A.S. Wexler, Y. Zhou, On the collision rate of small particles in isotropic turbulence I. Zero-inertia case, Phys. Fluids 10 (1998) 266–276.
[42] L.-P. Wang, A.S. Wexler, Y. Zhou, Statistical mechanical descriptions of turbulent coagulation, Phys. Fluids 10 (1998) 2647–2651.
[43] L.-P. Wang, A.S. Wexler, Y. Zhou, Statistical mechanical description and modelling of turbulent collision of inertial particles, J. Fluid Mech. 415 (2000) 117–153.
[44] E.J.P. Woittiez, H.J.J. Jonker, L.M. Portela, On the combined effects of turbulence and gravity on droplet collisions in clouds: a numerical study, J. Atmos. Sci. 66 (7) (2009) 1926–1943.
[45] Y. Xue, L.-P. Wang, W.W. Grabowski, Growth of cloud droplets by turbulent collision–coalescence, J. Atmos. Sci. 65 (2) (2008) 331–356.
[46] Y. Yin, Z. Levin, T.G. Reisin, S. Tzivion, The effects of giant cloud condensation nuclei on the development of precipitation in convective clouds – a numerical study, Atmos. Res. 53 (1–3) (2000) 91–116.
[47] L. Zaichik, V. Alipchenkov, Statistical models for predicting pair dispersion and particle clustering in isotropic turbulence and their applications, New J. Phys. 11 (2009) 3018.
[48] L. Zaichik, O. Simonin, V. Alipchenkov, Two statistical models for predicting collision rates of inertial particles in homogeneous isotropic turbulence, Physics of Fluids 15 (2003) 2995–3005.
[49] Y. Zhou, A.S. Wexler, L.-P. Wang, Modelling turbulent collision of bidisperse inertial particles, J. Fluid Mech. 433 (2001) 77.