INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **34**: 403–424

Development of highly accurate interpolation method for mesh-free flow simulations II. Application of CIVA method to incompressible flow simulations

Nobuatsu Tanaka*,[1]

Department of Mechanical Engineering, Ibaraki University, Naka-narusawa, Hitachi, Ibaraki, Japan

SUMMARY

This is the second report on the development of a highly accurate interpolation method, which is called cubic interpolation with volume/area (CIVA) co-ordinates, for mesh-free flow simulations. In this paper, the method of determining the c-parameter of CIVA using a constant curvature condition is first considered for the two- and three-dimensional cases. A computation of a three-dimensional passive scalar advection problem is performed for accuracy verification and for comparison with widely used methods. Then, an application algorithm of the CIVA method respecting incompressible fluid simulation is presented. As the incompressible condition based on Lagrangian approaches causes problems, in this paper we consider the condition based on the conventional Eulerian approach. The CIVA-based incompressible flow simulation algorithm enables a highly accurate simulation of many kinds of problems that have complicated geometries and involve complicated phenomena. To confirm the facts, numerical analyzes are executed for some benchmark problems, namely flow in a square cavity, free surface sloshing and moving boundary problems in complex geometries. The results show that the method achieves high accuracy and has high flexibility, even for the flows involving high Reynolds number, complicated geometries, moving boundaries and free surfaces. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: CIP; CIVA; computational fluid dynamics; incompressible fluid; mesh-free method; particle method

1. INTRODUCTION

In computational fluid dynamics (CFD), numerical accuracy and stability are the most important factors because they can have far-reaching effects on the results. However, it is difficult to be consistent with regard to these two factors. Accordingly, researchers have developed several kinds of consistent methods, for example, the high-order schemes (e.g. UTOPIA and K-K) in the finite difference method (FDM) and high-order elements in the

* Correspondence to: Department of Mechanical Engineering, Ibaraki University, 4-12-1, Naka-narusawa, Hitachi, Ibaraki 316-8511, Japan.
[1] E-mail: tanaka@mech.ibaraki.ac.jp

Received July 1999
Revised December 1999

finite element method (FEM). However, these methods are mesh-based and are not applicable to mesh-free approaches. Mesh-free approaches have been attracting the attention of fluid researchers because they do not require mesh generation, which generally requires a lot of time and care. The gridless method and the particle method are popular in CFD, but have drawbacks in that improvement of numerical accuracy and the application to high-Reynolds number flow are difficult.

Therefore, we have developed a new, highly accurate and stable interpolation algorithm, applicable to the mesh-free method [1,2]. To achieve high accuracy and stability, we extend the cubic-interpolated pseudo-particle (CIP) method [3,4], which in its original form executes interpolation with a rectangle or a rectangular parallelepiped mesh, for a triangle/tetrahedron and the local (natural) co-ordinates. In the FEM field, the local co-ordinates are frequently called the volume co-ordinates for three dimensions and the area co-ordinates for two dimensions. The new interpolation method, which we call cubic interpolation with volume/area (CIVA) co-ordinates, makes it possible to achieve highly accurate interpolation based on a tetrahedron/triangle in the case of the mesh-free method. From the computational results, it was confirmed that the CIVA method improved the accuracy of the gridless method and particle method [1]. We use the CIVA–particle method in this study, as it is flexible in the treatment of moving calculation points. By virtue of its flexibility, the CIVA–particle method makes possible the application of the augmented Lagrangian–Eulerian (ALE) [5] method and provides the following advantages [1]:

(a) It is easy to handle the inlet and outlet boundaries.
(b) The sparseness and denseness of particle distribution can be controlled easily (or adaptively).
(c) Fixed particles around boundaries make the accuracy better. (The full-Lagrangian algorithm in which the computing points always move according to the flow inevitably cause disturbance, i.e. the numerical viscosity.)

These points become serious problems in the conventional particle (full-Lagrangian) methods because of the difficulty of dealing effectively with them using those methods.

The CIVA method has a control parameter c and we must provide in advance the means of determining the value in order to close the system. In the previous study, we used the c-parameter given in References [6,7]. However, in this study we will first consider a determination method for the c-parameter with constant curvature conditions in two and three dimensions and investigate the validity in the case of a three-dimensional problem. Then, we apply the CIVA–particle method to an incompressible flow simulation. For the incompressible flow simulation with the particle method, the most difficult point is how to handle the incompressible condition (mass conservation law) derived from Lagrangian approaches. The full-Lagrangian incompressible flow simulation methods, such as the incompressible smoothed particle hydrodynamics (SPH) [8,9] method and the moving particle semi-implicit (MPS) [10,11] method, have been presented but they are subject to problems that will be discussed below. To avoid the problems, we treat the incompressible condition using the conventional Eulerian method. As the CIVA method can achieve high-order interpolation based on a triangle or a tetrahedron, Voronoi diagram and Delaunay triangulation can be utilized for automatic construction of mesh for the incompressible condition. Reflecting the fact that

Voronoi diagram and Delaunay triangulation are currently prominent topics in computational geometry, some application software programs for two and three dimensions are freely available. Some incompressible fluid simulation methods based on the Voronoi polygon have been developed (e.g. see Reference [12]), but the methods generally use a collocated grid arrangement, which defines both velocity and pressure at a point, so that the pressure field is prone to oscillation, especially in high-Reynolds number (*Re*) flow. To solve the problem, the Delaunay triangle is used instead of the Voronoi polygon as a control volume in this study. Then, velocity is defined at the vertex points of the triangle and pressure at the barycenter. This arrangement is called P1P0-type in FEM and is known to prevent the oscillation of pressure field. This method is not a completely mesh-free method, as it generates a global mesh for the Poisson equation. However, we can consider it to be a pseudo-mesh-free method because the user is not conscious of mesh generation. In other words, this approach is considered to be a method for improving accuracy of methods based on an unstructured mesh system.

Finally, to confirm the accuracy and flexibility of our incompressible method, numerical analyses are executed for some benchmark problems, namely flow in a square cavity, free surface sloshing and moving boundary problems in complex geometries.

2. THE CIVA ALGORITHM

In order to evaluate the advection or convection phenomena in this study, the CIVA–particle method based on the Lagrangian rearrangement interpolation (LRI) algorithm [1] is applied. The LRI algorithm is considered to be a mesh-free version of the ALE method. In fact, we can confirm that the LRI algorithm is equivalent to the ALE method, where the convection flux in the rezoning phase is evaluated by the upwind method (see Appendix A). However, the LRI method is more flexible than the ALE method because it does not need any mesh-based discretization.

2.1. Determination of the c-parameter by constant curvature conditions

The CIVA method has a control parameter c, whose value it is necessary to provide in advance in order to close the system. Thus, we will consider in this section the determination method of the c-parameter of CIVA in two and three dimensions.

In two dimensions, scalar profile in a triangle is assumed to be the following third-order polynomial, using area co-ordinates (L_1, L_2, L_3):

$$\tilde{f}(L_1, L_2, L_3) = \sum_{i=1}^{3} \alpha_i L_i + \sum_{j,k=1\ (j \neq k)}^{3} \beta_{jk}[L_j^2 L_k + cL_1L_2L_3] \qquad (1)$$

where c is the control parameter. The unknown coefficients of Equation (1), α_i, β_{jk}, can be determined independently of c and directly without solving simultaneous linear equations

$$\alpha_i = f_i, \qquad \beta_{jk} = f_j - f_k + (x_k - x_j)f_j^x + (y_k - y_j)f_j^y \qquad (2)$$

We must provide in advance the means of determining the value of c because the term $L_1L_2L_3$ in Equation (1) and the first-order spatial differential values become zero at every vertex of a triangle; therefore, c cannot be determined from the known information of the vertices. This set-up of c is an interesting problem and a subject for further investigation [2]. The use of another calculation point and the application of least-squares approximation are among the candidate approaches for accomplishing this. Here, we apply the constant curvature condition (CCC in this paper) [6,7] to the set-up. The condition means that the cubic function expressed by Equation (1) gives arbitrary curvatures in all areas within a triangle, i.e. the parameter c should be determined to satisfy

$$S_3^c \supset S_2 \tag{3}$$

where S_3^c is the function space spanned by the (incomplete) cubic polynomial (1) and S_2 is the function space spanned by the complete second-order polynomial that is expressed, using area co-ordinates, by

$$g(L_1, L_2, L_3) = \sum_{i=1}^{3} \alpha_i L_i + \sum_{j,k=1\ (k>j)}^{3} \beta_{jk} L_j L_k \tag{4}$$

To make the derivatives at the vertices calculated from Equation (1) equal to those calculated from Equation (4), the following relations should be established:

$$f_i^x = g_i^x, \quad f_i^y = g_i^y, \quad i = 1, 2, 3 \tag{5}$$

Substituting relations (5) into Equation (1) and rearrangement by β_{jk} results in

$$\begin{aligned} f(L_1, L_2, L_3) = \sum_{i=1}^{3} \alpha_i L_i &+ \beta_{12} L_1 L_2 (L_1 + L_2 + 2cL_3) + \beta_{23} L_2 L_3 (2cL_1 + L_2 + L_3) \\ &+ \beta_{13} L_1 L_3 (L_1 + 2cL_2 + L_3) \end{aligned} \tag{6}$$

By taking into consideration the characteristic of area co-ordinates, $L_1 + L_2 + L_3 = 1$, we can find that iff c is equal to $\frac{1}{2}$, the cubic polynomial (1) recovers the complete second-order polynomial (4) and satisfies the CCC of Equation (3).

In the three-dimensional case, the CIVA method utilizes a tetrahedron for interpolation and the volume co-ordinates. The following three-dimensional cubic polynomial using volume co-ordinates (L_1, L_2, L_3, L_4) in a tetrahedron, whose polynomial is in more general form than that in previous works [1], can be assumed in this paper

$$\tilde{f}(L_1, L_2, L_3, L_4) = \sum_{i=1}^{4} \alpha_i L_i + \sum_{j,k=1\ (j\neq k)}^{4} \beta_{jk} \left[L_j^2 L_k + \sum_{l=1}^{4} c_l \frac{L_1 L_2 L_3 L_4}{L_l} \right] \tag{7}$$

where the coefficients can be calculated independently of the parameters c_l without solving the linear system as follows:

$$\alpha_i = f_i, \qquad \beta_{jk} = f_j - f_k + (x_k - x_j)f_j^x + (y_k - y_j)f_j^y + (z_k - z_j)f_j^z \tag{8}$$

We have to specify c_l in advance because the terms $L_1L_2L_3L_4/L_l$ $(l = 1, 2, 3, 4)$ and the first-order spatial differential values become zero at every vertex of the tetrahedron and their coefficients cannot be determined from the known information at the vertices. For example, the CCC can be useful for determining the parameter c_l.

The complete second-order polynomial using volume co-ordinates is given by

$$g(L_1, L_2, L_3, L_4) = \sum_{i=1}^{4} \alpha_i L_i + \sum_{j,k=1\ (j>k)}^{4} \beta_{jk} L_j L_k \tag{9}$$

By the same consideration as in two dimensions, it is necessary and sufficient for satisfying the CCC that Equation (7) be transformed into

$$f(L_1, L_2, L_3, L_4) = \sum_{i=1}^{4} \alpha_i L_i + \frac{1}{2} \sum_{j,k=1\ (j\neq k)}^{4} \beta_{jk} L_j L_k (1 + L_j - L_k) \tag{10}$$

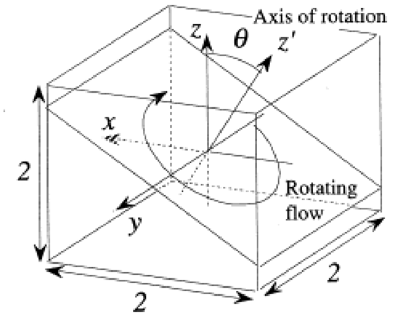
Function (10) is the interpolating function of CIVA with the CCC for three dimensions, and all the coefficients can be calculated from Equations (8). Therefore, the CCC makes it possible for a scalar value and the spatial derivatives in a tetrahedron to be interpolated with the only known information (the Cartesian co-ordinate values, the scalar values and the spatial derivatives) at the vertices from Equations (8) and (10).

2.2. Numerical analysis of a three-dimensional advection problem

To check the validity and accuracy of the CIVA method with the above-mentioned c-parameter determination of CCC (interpolating function (10)), a three-dimensional passive scalar advection problem shown in detail in Figure 1 is considered. (The two-dimensional verification has already been done in Reference [1].) This problem was used as a benchmark problem when Kawamoto *et al.* [13] checked the accuracy of the high-order finite difference schemes. The governing equation for advection of scalar f by flow $\mathbf{u}$ is

$$\frac{\partial f}{\partial t} + (\mathbf{u}\cdot\nabla)f = 0 \tag{11}$$

The flow circulates uniformly in the region as angular velocity of $\omega = 2\pi$ (rad s^{-1}). The calculation is performed over 1000 steps, with a time step of 0.001 s (1 rotation) on the uniformly fixed points of $x \times y \times z = 30 \times 30 \times 30$.



○Angular velocity: ω=2π (rad s^{-1})
○1000 step with Δt=0.001 (s)
○Inclination of axis of rotation: θ=0°, 45°
○Calculation points:
x×y×z=30×30×30 (Uniform)
○Initial scalar profile:
$$\begin{cases} f=1/2(\cos(\pi r/0.3)+1) & (0 \leq r \leq 0.3) \\ f=0 & (0.3>r) \end{cases}$$
r: Distance from the point (0,1/2,0)

Figure 1. Benchmark problem of three-dimensional passive scalar advection.

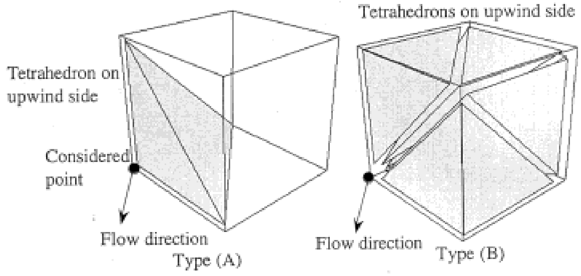


Figure 2. Choice of a tetrahedron on the upwind side.

We considered two approaches for constructing a tetrahedron on the upwind side, as shown in Figure 2. The comparison of peak values of the scalar is shown in Table I, which includes the results obtained with other methods given in Reference [13]. The table indicates that the results obtained with the two approaches for constructing a tetrahedron are virtually the same. We can confirm that CIVA with CCC can achieve the accuracy between the third-order upwind scheme and the fourth-order central difference scheme.

Table I. Maximum value of scalar after 1 rotation.

	$\theta = 0°$	$\theta = 45°$
QUICK [13]	0.664	0.577
Third-order upwind [13]	0.631	0.532
Fourth-order central [13]	0.928	0.929
CIVA(A)	0.817	0.753
CIVA(B)	0.825	0.722

3. INCOMPRESSIBLE FLOW SIMULATION

3.1. Problems of the incompressible condition of the particle method

In view of the computing point movement, the governing equations are as follows:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u} \tag{12}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = 0 \tag{13}$$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\nabla p + \frac{1}{Re}\Delta\mathbf{u} \tag{14}$$

For incompressible flow simulation with the particle method, the most difficult point is how to handle the mass conservation law. The full-Lagrangian approaches, such as smoothed particle hydrodynamics (SPH) and moving particle semi-implicit (MPS), use a predictor–corrector method (projection method), such as the marker and cell (MAC) or simplified MAC (SMAC) methods, to handle the condition. In the MAC method, the following Poisson equation of pressure is solved:

$$\nabla^2 p_i^{n+1} = f_{\mathrm{ex}} \tag{15}$$

where the right-hand side term is calculated explicitly. In incompressible SPH [8,9], the left-hand side of Equation (15) is discretized using the kernel function, W, as

$$\nabla^2 p_i^{n+1} \equiv 2 \sum_{j\neq i} \frac{m_j}{\rho_j} \frac{(p_j - p_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} (\mathbf{x}_j - \mathbf{x}_i)\cdot\nabla_i W(|\mathbf{x}_j - \mathbf{x}_i|) \tag{16}$$

or in MPS [10] as

$$\nabla^2 p_i^{n+1} \equiv \frac{2d}{n^0\lambda} \sum_{j\neq i} (p_j - p_i) W(|\mathbf{x}_j - \mathbf{x}_i|) \tag{17}$$

The discretized form of Equation (15) is solved to correct the velocity **u** and position **x** of particles

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t \nabla p^{n+1} \tag{18}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \frac{\Delta t}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n) \tag{19}$$

where the tilde ($\sim$) denotes a predictor and particle movements are evaluated by a Crank–Nicolson scheme. From these equations, the relationship between pressure and position is

$$\mathbf{x}^{n+1} = \mathbf{x}^{n} + \frac{\Delta t}{2}(\mathbf{u}^{n} + \tilde{\mathbf{u}}) - \frac{\Delta t^2}{2}\nabla p^{n+1} \tag{20}$$

This algorithm is deemed to contain problems. The first problem is that, in full-Lagrangian algorithms, the discretized system of Equation (15) becomes a non-linear system, as discretization of Laplacian operator depends on the particle position. For example, the discretized Laplacian operator in incompressible SPH is more exactly expressed as

$$\nabla^2 p_i^{n+1} \equiv 2\sum_{j\neq i}\frac{m_j}{\rho_j}\frac{(p_j^{n+1}-p_i^{n+1})}{|\mathbf{x}_j^{n+1}-\mathbf{x}_i^{n+1}|^2}(\mathbf{x}_j^{n+1}-\mathbf{x}_i^{n+1})\cdot\nabla_i W(|\mathbf{x}_j^{n+1}-\mathbf{x}_i^{n+1}|) \tag{21}$$

By considering that the kernel function is generally non-linear with respect to **x**, it is clear that Equations (20) and (21) become a non-linear system. It is necessary to solve the system by using a suitable non-linear system solver, whereas the discretized system of Poisson equation in Euler approaches becomes a linear system and is solvable by conventional matrix solvers, such as successive overrelaxation (SOR) and conjugate gradient (CG). This is the disadvantage of the full-Lagrangian method.

The second problem is the local minimum problem of the particle positions. The original MPS method satisfies the incompressible condition by keeping the ‘particle number density’ constant, i.e.

$$f_{\text{ex}} \equiv -\frac{\rho}{\Delta t^2}\frac{\langle\tilde{n}\rangle_i - n^0}{n^0} \tag{22}$$

where n^0 is the initial particle density and $\langle n\rangle_i$ denotes the number of particles in a unit volume. Equation (22) represents the deviation of the particle number density from the constant value. The particle number density corresponds to the particle position and if the particle positions are in a local minimum state; the particle would not be able to move anywhere. Some troubles caused by this local minimum problem will be presented.

Artificial viscosity (of velocity)

Let us consider a two-dimensional cavity-driven flow in a square cavity, as is shown in Figure 3, which is one of the popular benchmark problems. We assume the case of no viscosity. In the case, it is clear that there is no flow in the cavity at any time. When this problem is solved with the MPS method under the initial condition of uniformly arranged particles (particles initially have a constant particle number density), the movement of boundary particles causes a flow in the cavity in spite of no viscosity. At the first time step, the particles in the moving wall move in $\Delta t \times v$ distance toward the right-hand side (see Figure 3). To satisfy the incompressible condition of the MPS method, i.e. the particle number, density has to be kept constant in the cavity region, e.g. the particles near the moving wall will move to a certain position that is different from the original position. In the MPS method, the movement of particles means that they have a velocity. This fact contradicts the physical behavior of fluid.

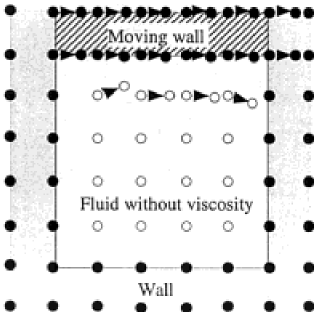


Figure 3. Artificial viscosity effect in square cavity problem.

Δt paradox

When Δt becomes small (of course Δt should be bigger than the significant digits of the computer used), the time-integral accuracy of the discrete methods, such as FDM and FEM, is generally improved. However, the MPS method is an exception to this rule.

For example, we consider a two-dimensional thermal driven flow in a square cavity, as is shown in Figure 4, which is also one of the popular benchmark problems. We assume the case of a certain viscosity. When we solve this problem with the MPS method using uniformly arranged particles initially and using a relatively small time step, it causes no flow in the cavity. The particles near the hot and cold walls receive the buoyancy force by which particles near a hot wall move upward and particles near a cold wall move downward. However, when the time step is smaller than a certain threshold, to make the particle number density constant, the moved particles are forced to move back to the original position. This means that the fluid in the cavity does not move forever. The reason is that the initial particle position is at the local minimum, which satisfies the incompressible condition and there is no other local minimum in the vicinity of the original position.

Discrete position and velocity

Two-dimensional flow in a channel shown in Figure 5 is considered here. When we solve this problem with the MPS method using the initial condition of uniformly arranged particles, the

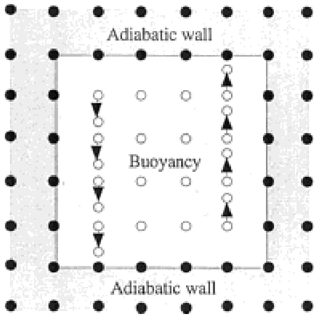


Figure 4. Δt paradox in thermal cavity problem.

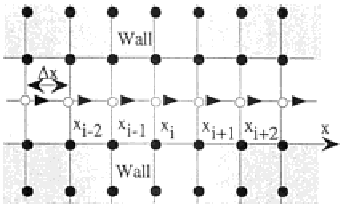


Figure 5. Discrete position and velocity in a channel flow.

situation occurs in which particles exist only at the grid position, x_i. This fact means that the particle position and velocity take only discrete values. For example, $\Delta t = 1$ and all the initial particle velocity is a certain value between $\Delta x/2$ and $3 \times \Delta x/2$. To satisfy the incompressible condition, the particle position after one time step will be $+\Delta x$-shifted, independently of the initial particle velocity.

One of the reasons for this problem may be the small number of particles. However, in a complex problem it is possible for these problems to occur. To prevent the above-mentioned local minimum problems, in the case of using the revised MPS, there is no choice but to adopt the inconsistent algorithm in which the constant condition of particle number density is not consistent with the Poisson equation. However, the algorithm causes the drawback in that the condition for particle number density is not strictly satisfied in each time step. A similar situation occurs when the methods based on the non-staggered mesh do not always satisfy the divergence-free condition. The reason is a similar inconsistency of the discretized equations of pressure gradient and velocity divergence with the discretized Poisson equation.

To prevent the problems that affect the accuracy of the results, we consider the incompressible condition by the conventional Eulerian method based on an automatic Voronoi/Delaunay tessellation.

3.2. Incompressible algorithm

3.2.1. Basic algorithm. Instead of Equation (13) we use the following governing equation for the incompressible condition from the Eulerian viewpoint

$$\nabla \cdot \mathbf{u} = 0 \tag{23}$$

The basic algorithm to solve Equations (12), (14) and (23) is based on the MAC method outlined in Figure 6. Equation (12) and the convection terms of Equation (14) are evaluated by Lagrangian movement of the particles. The other terms are discretized by the FVM based on the control volumes, which are automatically generated with the Voronoi/Delaunay tessellation method. Delaunay tessellation is in duality with Voronoi tessellation, and Voronoi tessellation can be obtained from Delaunay tessellation (see Figure 7).

In this study, we deal with the incompressible condition by the following algorithm. At first, the considered region is automatically divided into a mesh based on the computing points using Voronoi/Delaunay tessellation. Based on the mesh with FVM, the Poisson equation

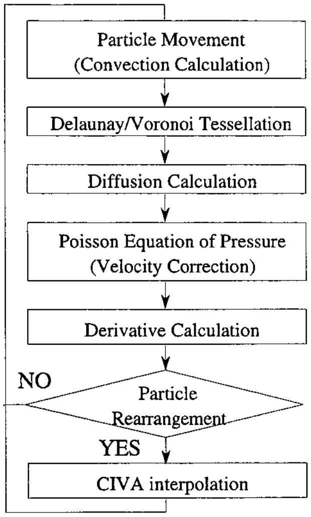


Figure 6. MAC-based algorithm.

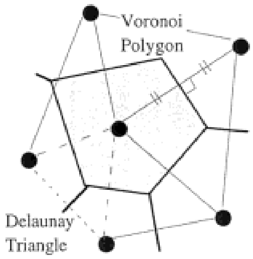


Figure 7. Voronoi/Delaunay tessellation.

derived from Equation (23) is discretized into a linear system. When the MAC and FVM methods are combined, the integration of the Poisson equation of pressure over the control volume becomes the following:

$$\int_{\partial\Omega} \nabla p \cdot d\mathbf{n} = \frac{1}{\Delta t} \int_{\partial\Omega} \tilde{\mathbf{u}} \cdot d\mathbf{n} \tag{24}$$

where $\int_{\partial\Omega} d\mathbf{n}$ means the integration on a boundary $\partial\Omega$. The linear system derived from Equation (24) is solved by a suitable matrix solver, such as SOR or incomplete Cholesky conjugate gradient (ICCG).

Equations (12), (14) and (23) in the case of low-Reynolds number flow can be solved with the control volume of the Voronoi polygon (Figure 8(a) is the example of the cavity flow of

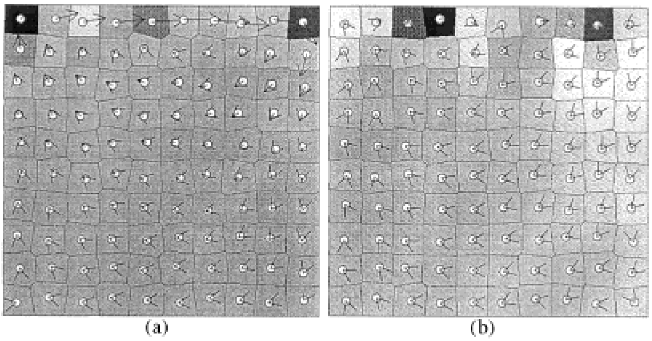


Figure 8. Particle velocity by Voronoi-based algorithm; (a) $Re = 10^2$, (b) $Re = 10^3$.

$Re = 10^2$). However, this is difficult in the case of high *Re* flow because pressure distribution oscillates in a 'checkerboard' manner, as shown in Figure 8(b). The oscillation is caused by the collocated grid points that define both velocity and pressure at the same position. To solve the problem with the collocated points, we need to introduce the stabilizing algorithms, which are studied in the FEM field. To make matters worse, it is not easy to provide the boundary conditions for variant values, as the velocity is not defined on the boundary.

On the other hand, to stabilize the pressure field, we use the control volume of the Delaunay triangle instead of the Voronoi polygon for mass conservation, as is shown in Figure 9. Furthermore, velocity is defined at the vertex points of the triangle and pressure is defined at the barycenter. The pressure definition point is not common to every time step and appears only after the triangulation, and so we call it the virtual pressure point (VPP). The arrangement of velocity and pressure is the P1P0-type of FEM and is also similar to the 'staggered' mesh in FDM. Such arrangements are known to be able to suppress the oscillation of pressure field.

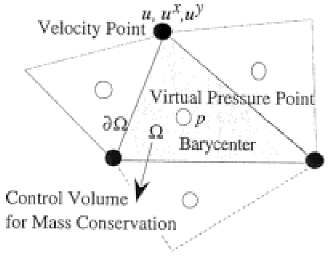


Figure 9. Control volume for mass conservation.

3.2.2. Particle rearrangement algorithm. When particles move according to the fluid behavior, they gather or scatter at certain points. Therefore, it is necessary to rearrange particles in order to control the particle density. A simple algorithm restores the particle position at every time step. The algorithm is efficient because it requires only one tessellation, but is not able to treat moving boundaries. Another example is to keep the particle number density constant [10,11]. In this study, we introduce a new algorithm using the charged particle model, which balances by moving the particle by means of the repulsive force of electric charge provided. This algorithm can be utilized in initial particle distribution and also in adaptive particle distribution.

In the particle rearrangement procedure, an interpolation algorithm is required. As high-order interpolation is needed to reduce the error, we adopt the CIVA algorithm. The c-parameter is given by CCC, i.e. $c=\frac{1}{2}$. The boundary condition for the derivatives in the normal direction of the wall surface was computed from the distribution of the scalar value of the fluid.

3.3. Numerical analyses

In this section, we analyze some two-dimensional benchmark problems to confirm the validity of our approaches. The problems considered here are a driven flow in a two-dimensional square cavity, a free surface flow and a fluid flow with moving and complicated boundaries.

3.3.1. Driven flow in a two-dimensional square cavity. At first, in order to check the accuracy, we analyze a driven cavity flow in a two-dimensional square cavity, which is one of the most popular benchmark problems. The number of particles, including boundary particles, is 121. In this study, time step is chosen to be a tenth of the Courant condition, because it was found that explicit evaluation of particle movements causes large numerical error in particle methods [1]. The computation is for $Re=10^2$, 10^3, 10^4. The above-mentioned charged particle model is used for particle rearrangement algorithm.

Figure 10 shows the comparison of velocity profile in the center of each axis direction with the popular benchmark solution [14]. In the case of $Re=10^2$, the results agree well with the benchmark. As Re increases, however, the difference can be observed in the region where the velocity gradient is large. The main reason is considered to be the small amount of particles. Therefore, in order to investigate the effect of the number of particles, we performed the simulation of $Re=10^4$ in which the number of particles is changed to 441. The result is shown in Figure 11. Although the velocity gradients near the walls approach the benchmark solution by increasing the number of particles, overshooting of the velocity profile remains. The reason for the overshooting is considered to be the effect of the method on the derivative boundary condition. Further examination and improvement are required concerning these points.

Figure 12 shows the pressure and particle velocity distribution without utilizing the spatial derivatives for $Re=10^2$ and 10^4 and Figure 13 shows the detailed velocity distribution interpolated with CIVA for $Re=10^3$. In comparison with Figure 8(b), it can be confirmed that even in the higher Re case, the oscillation of pressure is well suppressed by the VPP method. Because the particles are rearranged at every time step by the charged particle model and the time step is relatively small, we scarcely confirm any particle movements. This is the same phenomenon as that pointed out in Section 3.1 and the reason is that the particle positions are

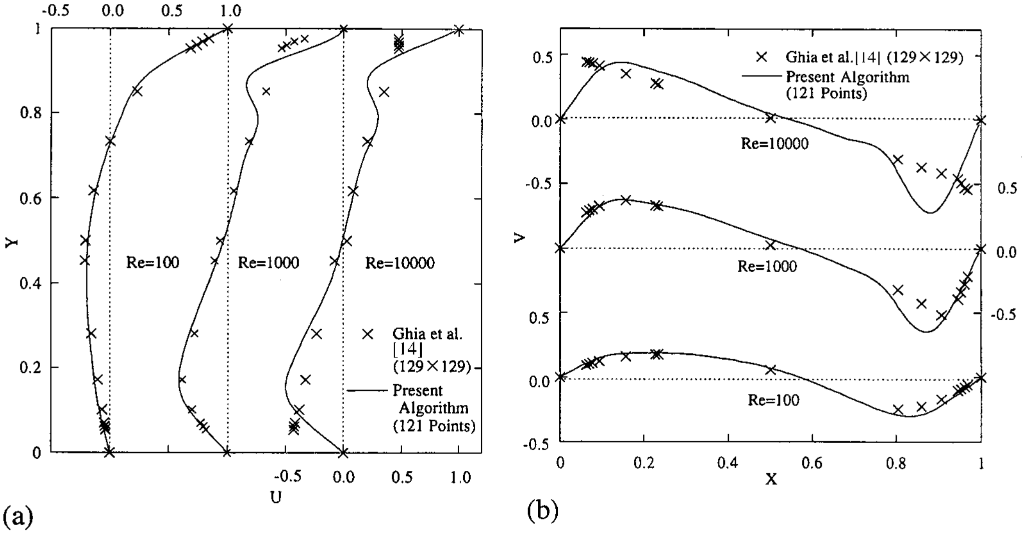


Figure 10. Comparison of velocity profile (with CIVA interpolation). (a) Horizontal velocity at vertical center; (b) vertical velocity at horizontal center.

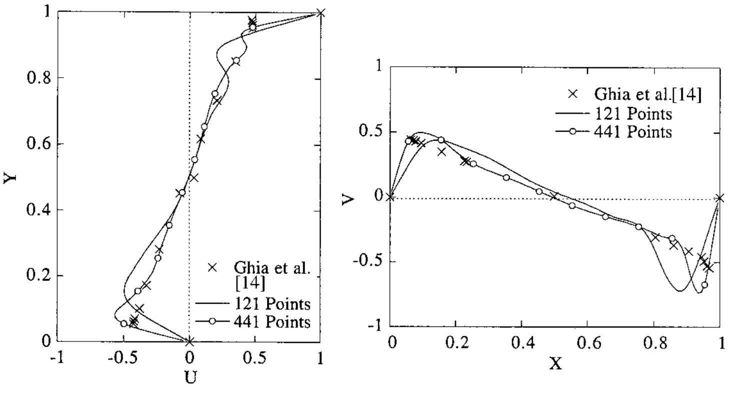


Figure 11. Effect of the variation in particle number ($Re = 10^4$). (a) Vertical velocity at horizontal center; (b) horizontal velocity at vertical center.

at the local minimum. Thus, we executed a simulation of $Re = 10^3$ without particle rearrangement. The result is shown in Figure 14, which indicates that the particles move according to the fluid flow and the movement reproduces well the fluid behavior. However, without particle rearrangement, some particles gather or scatter at certain points or are captured by the wall. Therefore, it is necessary to rearrange or rezone particles in order to control the particle density at a suitable time.

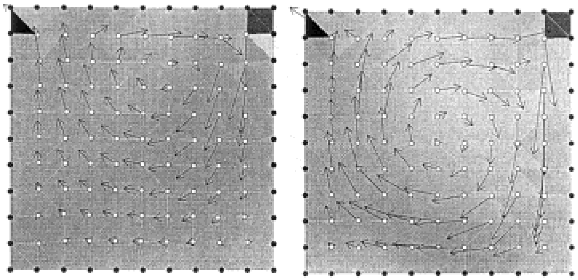


Figure 12. Particle velocity by Delaunay-based algorithm with VPP (left: $Re = 10^2$, right: $Re = 10^4$).

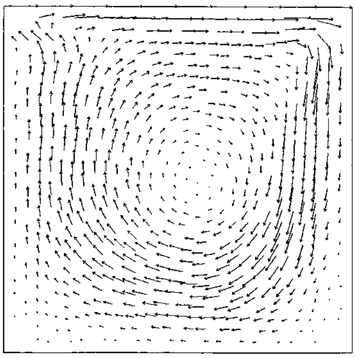


Figure 13. CIVA interpolated velocity field ($Re = 10^3$).

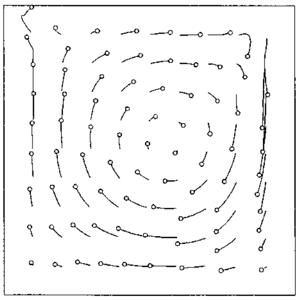


Figure 14. Tracing particles without rearrangement ($Re = 10^3$).

3.3.2. Free surface flow. The next computation is for an oscillating fluid flow with free surface in the case of no viscosity ($Re = \infty$ in Equation (14)). Initial conditions are that the free surface height is $y(x) = 1 + 0.1 \sin \pi(x - 0.5)$ and the fluid is quiescent (Figure 15). The condition for free surface is given by pressure at the surface equal to 0 and slip condition for velocities in the direction parallel to the surface. In this calculation, we do not use the charged particle model, but suppose that particles can only be movable vertically and fixed horizontally.

The time histories of wall-surface water levels are shown in Figure 16. The result based on FDM–VOF using a 100 × 150 uniform mesh was also included for comparison. Although calculation points are few for the CIVA–particle method, the results are in good agreement. As there is no viscosity in this case, the oscillating period is analytically calculable with the velocity potential method. Thus, we compare the oscillating period with the three methods as

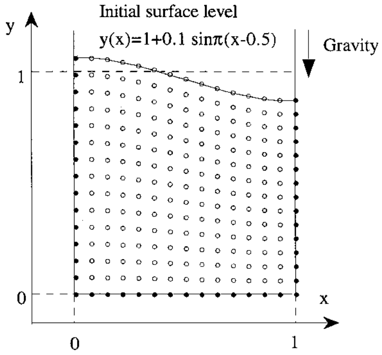


Figure 15. Free surface sloshing problem and the initial condition.

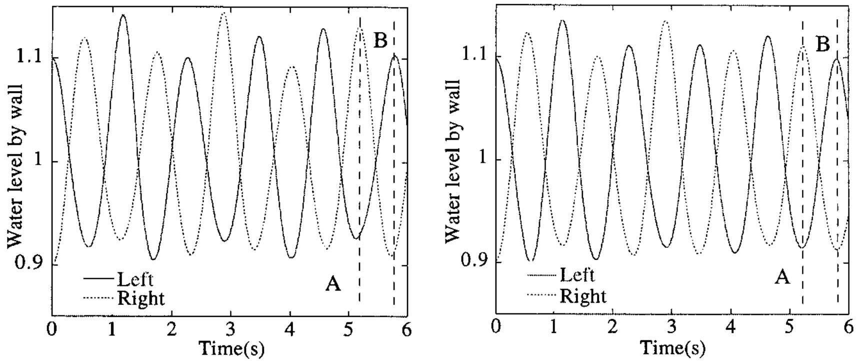


Figure 16. Time history of surface level for left and right wall. (a) FDM–VOF (100 × 150 grid points); (b) CIVA–particle (225 particles).

is shown in Table II. As the velocity potential method is based on a linear theory, the period by the velocity potential method is slightly different from the results by the FDM–VOF or the CIVA–particle method. The period by the CIVA–particle method with few calculation points agrees well with the result by the FDM–VOF method with fine mesh. Flow pattern and pressure distribution at the times A and B in Figure 16 are shown in Figure 17. The behaviors of free surface flow are well simulated.

3.3.3. Example of computation of fluid flow with moving and complicated boundaries. The last computation is an example of simulation of fluid flow with moving boundaries in complex geometries, as shown in Figure 18. The problem considered is a flow in a glasses-like geometry having rotating wings. The spacing of particles is also kept constant by using the above-mentioned charged particle model.

Table II. Comparison of sloshing frequency.

	Vector potential	FDM–VOF	CIVA–particle
Frequency	0.882	0.869	0.862

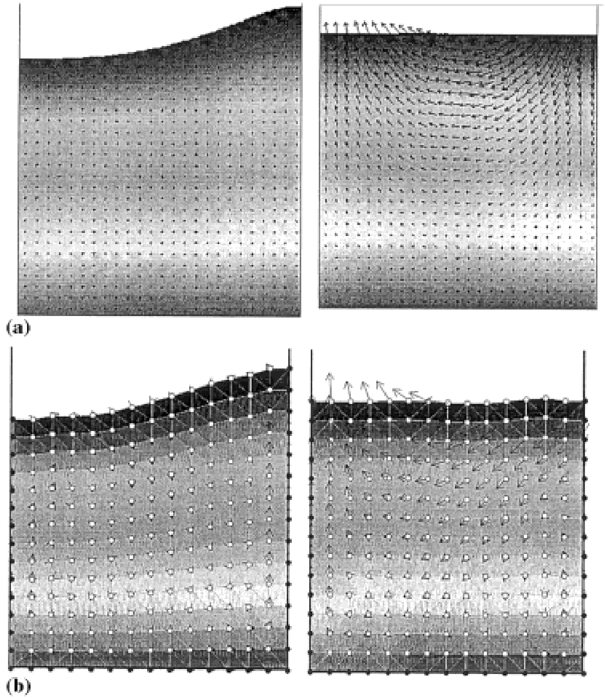


Figure 17. Flow vector and pressure distribution. (a) FDM–VOF (left: at time A, right: at time B); (b) CIVA–particle (left: at time A, right: at time B).

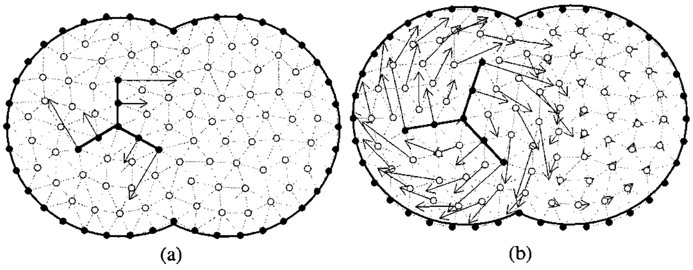


Figure 18. Particle velocity of a flow around rotating wings ($Re = 10^3$). (a) Initial condition; (b) steady state.

The initial condition and the computed result are indicated in Figure 18. We can confirm that there exists a secondary rotating flow in the right-hand side region, which is driven by the rotating flow around the wings in the left-hand side region. Thus, with the CIVA–particle method, we can treat flexibly a flow with complicated boundaries and moving boundaries.

4. CONCLUSION

Regarding the new interpolation algorithm, CIVA, the method of determining the c-parameter in two and three dimensions using CCC was considered. A verification of the accuracy and comparison with the other high-order methods, such as the third-order upwind scheme, were also performed with a three-dimensional passive scalar advection problem. Consequently, it has been clarified from the results that the CIVA method with CCC can achieve the medium accuracy between the third-order upwind and the fourth-order central difference schemes. Next, an incompressible fluid simulation was considered. As the incompressible condition based on full-Lagrangian approaches causes problems, the condition in this study was satisfied by the conventional Eulerian method. Numerical analyses for some benchmark problems were executed with a new incompressible fluid simulation method based on the CIVA–particle method, and from the results we confirmed that the method achieves high accuracy and has high flexibility, even for high-Reynolds number flow, complicated geometries, moving boundaries and free surface.

ACKNOWLEDGMENTS

The author would like to thank Associate Professor S. Kosizuka of Tokyo University and N. Shirakawa of Toshiba Corporation for reviewing the paper and giving valuable advice in regard to its revision.

APPENDIX A. RELATION BETWEEN CIVA-PARTICLE METHOD AND ALE METHOD

Here, we consider the relation of the LRI method in a CIVA–particle method and the ALE method. As shown in Figure 19, co-ordinate systems are established as space (fixed) co-ordinates: $\mathbf{x}$ (x, y), material co-ordinates: $\mathbf{X}$ (X, Y) and ALE (moving) co-ordinates: $\boldsymbol{\chi}$ (χ, ξ).

An advection equation is expressed as follows by the above-mentioned co-ordinates:

Fixed co-ordinates

$$\left.\frac{\partial f}{\partial t}\right|_{x} + (\mathbf{u}\cdot\nabla)f = 0 \tag{25}$$

Material co-ordinates

$$\left.\frac{\partial f}{\partial t}\right|_{X} = 0 \tag{26}$$

ALE co-ordinates

$$\left.\frac{\partial f}{\partial t}\right|_{\chi} + (\mathbf{c}\cdot\nabla)f = 0 \tag{27}$$

where $\mathbf{c} = \mathbf{u} - \mathbf{w}$ and $\mathbf{w}$ is the velocity of co-ordinate movement in ALE co-ordinates.

A.1. LRI method

Each phase of the LRI method in the CIVA–particle method will perform the following evaluation (see Figure 20):

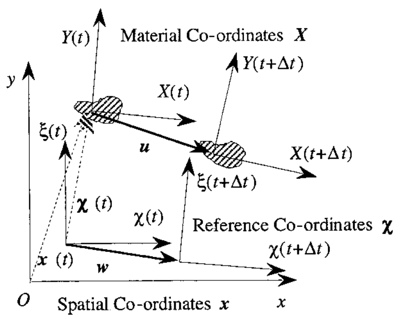


Figure 19. Co-ordinate system.

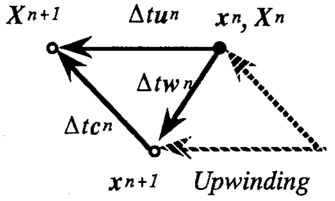


Figure 20. Co-ordinates in LRI.

(C.1) *Lagrangian phase:* Equation (26) is evaluated and a time step is set forward to $n \rightarrow n+1$

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t\mathbf{u} = \mathbf{x}^n + \Delta t\mathbf{u}$$

$$f(t^{n+1}, \mathbf{X}^{n+1}) = f(t^n, \mathbf{X}^n) \tag{28}$$

(C.2) *Rearrangement phase:* calculation points are rearranged to determine $\mathbf{x}^{n+1}$ at the $(n+1)$th time step

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\mathbf{w} = \mathbf{X}^{n+1} - \Delta t\mathbf{c} \tag{29}$$

(C.3) *Interpolation phase:* in order to determine f at $\mathbf{x}^{n+1}$ and at t^{n+1}, interpolation is processed

$$f(t^{n+1}, \mathbf{x}^{n+1}) = f(t^{n+1}, \mathbf{X}^{n+1} - \Delta t\mathbf{c}) \tag{30}$$

By using Equation (28), Equation (30) is transformed to

$$f(t^{n+1}, \mathbf{x}^{n+1}) = f(t^n, \mathbf{X}^n - \Delta t\mathbf{c}) \tag{31}$$

When $\mathbf{X}^n = \mathbf{x}^n$ is taken into consideration, an LRI method evaluates the following equations by the procedure (C.1)–(C.3):

$$f(t^{n+1}, \mathbf{x}^{n+1}) = f(t^n, \mathbf{x}^n - \Delta t\mathbf{c}) \tag{32}$$

A.2. ALE method

In the case of explicitly evaluating the advection, the ALE method consists of the following two phases [5]:

(A.1) Lagrangian phase
(A.2) Rezoning or convective flux calculation phase

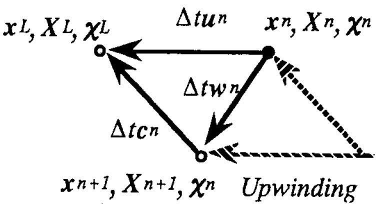


Figure 21. Co-ordinates in ALE.

The ALE method advances the time step by the fractional time step method and evaluates the following (see Figure 21):

(A.1) *Lagrangian phase:* while Equation (26) is evaluated, the time step is not completely advanced but is taken as the value in mid-phase L

$$\mathbf{x}^L = \mathbf{x}^n + \Delta t\mathbf{u}$$

$$f(t^L, \mathbf{X}^L) = f(t^n, \mathbf{X}^n) \tag{33}$$

(A.2) *Rezoning phase:* perform the rezoning of mesh and decide the position $\mathbf{x}^{n+1}$ of the calculation point at time $(n+1)$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\mathbf{w} = \mathbf{x}^L - \Delta t\mathbf{c} \tag{34}$$

Equation (27) is evaluated in order to determine the value of the calculation point at $(n+1)$th time (convection flux evaluation). This evaluation is generally done by discretizing Equation (27), but here we evaluate Equation (27) by the upwind method. In this case, the following equations are evaluated in consideration of the movement of co-ordinates:

$$f(t^{n+1}, \chi^{n+1}) = f(t^L, \chi^L - \Delta t\mathbf{c}) \tag{35}$$

Since the following equation holds:

$$f(t^L, \chi^L) = f(t^L, \mathbf{X}^L) \tag{36}$$

Equation (35) can be deformed as follows in consideration of Equation (33):

$$f(t^{n+1}, \chi^{n+1}) = f(t^L, \mathbf{X}^L - \Delta t\mathbf{c}) = f(t^n, \mathbf{X}^n - \Delta t\mathbf{c}) \tag{37}$$

Therefore, if $\chi^{n+1} = \mathbf{x}^{n+1}$ and $\mathbf{X}^n = \mathbf{x}^n$ are taken into consideration when convection flux is evaluated by the upwind method, the phase of (A.1) and (A.2) evaluates the following equations:

$$f(t^{n+1}, \mathbf{x}^{n+1}) = f(t^n, \mathbf{x}^n - \Delta t \mathbf{c}) \tag{38}$$

From Equations (32) and (38), it is confirmed that the LRI method is equivalent to the ALE method, which evaluates the convection flux in the rezoning phase by the upwind method.

REFERENCES

1. Tanaka N. Development of highly accurate interpolation method for mesh-free flow simulations I. Integration of gridless, particle and CIP methods. *Journal of Numerical Methods in Fluids* 1999; **30**(8): 957–976.
2. Tanaka N. Stability and accuracy analysis of CIVA-based convection evaluating algorithm. *Proceedings of Japan Society of Mechanical Engineers 11th Computational Mechanics Conference* 1998; **98**(2): 153–154 (in Japanese).
3. Yabe T, Takei E. A new high-order Godunov method for general hyperbolic equations. *Journal of the Physics Society of Japan* 1988; **57**: 2598–2601.
4. Yabe T, Ishikawa T, Wang PY. A universal solver for hyperbolic equations by cubic-polynomial interpolation II. Two- and three-dimensional solvers. *Computational Physics Communications* 1991; **66**: 233–242.
5. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speed. *Journal of Computational Physics* 1974; **14**: 227–253.
6. Zienkiewicz OC. *The Finite Element Method* (3rd edn). McGraw-Hill: New York, 1977.
7. Bazeley GP, Cheung YK, Irons BM, Zienkiewicz OC. The triangular elements in bending-conforming and non-conforming solutions. In *Proceedings of a Conference on Matrix Methods in Structural Mechanics*, Air Force Institute of Technology, Wright Patterson AFB, OH, 1965.
8. Cummins SJ, Rudman MJ, Monaghan JJ. Projection method and SPH. Applied Mathematical Report and Preprints 97/22, 1997.
9. Cummins SJ, Rudman MJ. Truly incompressible SPH. Proceedings of ASME Fluid Engineering Division Summer Meeting. FEDSM98-4923 (CD-ROM), 1998.
10. Koshizuka S, Tamako H, Oka Y. A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal* 1995; **4**(1): 29–46.
11. Koshizuka S, Oka Y. Moving particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science Technology* 1996; **123**: 421–434.
12. Taniguchi N, Arakawa C, Kobayashi T. Construction of a flow-simulating method with finite volume based on Voronoi diagrams. *Transactions of the Japan Society of Mechanical Engineers (B)* 1989; **55**(513): 1324–1328 (in Japanese).
13. Kawamoto S, Iwase H, Tanahashi T. High-accuracy of three-dimensional advection equation using finite difference methods. *Japan Society of Mechanical Engineers International Journal*, *Series II* 1992; **35**(4): 536–542.
14. Ghia U, Ghia KN, Shin CT. High-Re solution for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**: 387–411.