JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 79-92 (2021)
DOI: 10.6688/JISE.202101_37(1).0006

A Dynamic Parallel Meshless Method for the Problems with Large-Scale Movable and Deformable Boundary

LIANG WANG[1], RUI XUE[1], NING CAI[1,*], PAN CHEN[1], XIAOBO CUI[1], WEI WU[2], MIAOMIAO NIU[1], DONGLIANG ZHANG[1], ZHAO ZHANG[1] AND XIAOSONG ZHANG[3]

[1]*School of Energy and Power Engineering*
Nanjing Institute of Technology
Nanjing, 211167 P.R. China
[2]*Beijing Aerospace Technology Research Institute*
Beijing, 100074 P.R. China
[3]*School of Energy and Environment*
Southeast University
Nanjing, 210096 P.R. China
E-mail: cnnjit@126.com

This paper puts forward a dynamic parallel meshless computing algorithm that efficiently solves flow fields with largescale motions of movable and deformable boundaries. The partition boundary is updated, as the moving boundary moves across the material interface. Meanwhile, the point clouds near the moving boundary are reconstructed. Our algorithm also solves the workload balance between nodes and information exchange in each region of the computational field, using the governing equations in the arbitrary Lagrangian-Eulerian (ALE) form. The AUFS scheme is extended to calculate the numerical convective flux. Take the interaction between a helium bubble and a shockwave as an example. Our algorithm is applied to compute the flow field with different numbers of discrete points (33,044 and 66,089) and partitions (2 and 4). The results show that our algorithm achieves an efficiency of over 80%, and captures the interaction between shockwaves and the bubble accurately. Hence, our parallel algorithm is suitable for solving problems with largescale motions of deformation boundaries. The research results shed new light on the calculation speed for similar problems.

Keywords: dynamic parallel algorithm, meshless method, large-Scale movable boundary, parallel efficiency, arbitrary Lagrangian-Eulerian (ALE) form

1. INTRODUCTION

Computational fluid dynamics (CFD) has provided solutions to complex engineering problems like the structural reconstruction near movable boundaries, the combustion of combustible gases or liquids, and partial differential equations with nonlinear terms [1-5]. Meanwhile, there is a growing need for accurate numerical simulation in practical cases, arising from computing methods for turbulence, namely, direct numerical simulation (DNS), large eddy simulation (LES), and detached eddy simulation (DES). To realize accurate simulation, it is necessary to employ a large number of grid points, which may reach hundreds of millions and even billions.

Received March 24, 2020; revised May 30, 2020; accepted June 14, 2020.
Communicated by Mahmoud Abdel-Aty.
[*] This work was supported by the National Natural Science Foundation of China (Grant No. 51806095 and No. 51906100), the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20181022 and No. BK20191015) and the research fund of Nanjing Institute of Technology (Grant No. YKJ201710 and No. ZKJ201702).

Currently, parallel computing, *i.e.* the simultaneous use of multiple central processing units (CPUs), has gained popularity among scholars, due to the limited computing power of a single CPU. There are several different forms of parallel computing: problem parallelism and algorithm parallelism [6-11]. In parallel computing, the parallel algorithm geometrically decomposes the computational domain into *n* regions, and each single processor takes care of one region [12, 13]. In this way, less time is spent on numerical calculations, leading to a higher computing efficiency. However, the parallel algorithm increases the processing amount, and thus the computing scale.

Recent years has seen the proliferation of the meshless method in CFD, thanks to its strong ability to discretize the computational domains near complex boundaries [14-19]. A noteworthy feature is that the meshless methods does not consider connectivity between points, since they do not use traditional structured/unstructured mesh topologies but employ flexible clouds of points, which are basically composed of a center point and several satellites, to discretize the flow domain. But this method faces problems like inefficiency and low accuracy. Compared with mesh-based methods, the meshless method involves lots of flux calculations per computational unit, ranging from matrix inversions to matrix multiplications [20]. In this paper, the meshless method is combined with the parallel algorithm to fully display the merits of both approaches.

With the development of computer technology, parallel meshless algorithms have become a research hotspot [21-24]. For instance, Yagaw *et al.* [25] and Fujisawa [26] integrated parallel computing with free mesh methods (FMM) for fluid flow analysis with largescale calculations. Nonetheless, this integrated approach is not completely meshless and limited in application scope. Danielson [27] was the first to establish a parallel meshless regenerative kernel particle algorithm for fluid dynamics. Singh [28] adopted the parallel element-free Galerkin method to solve the fluid flow equations, but did not specify the whole solving process. Zeng *et al.* [29] combined the Galerkin method and the Petrov-Galerkin method with parallel algorithms into a novel parallel meshless method, and employed the method to compute the relevant problems in elastic dynamics. Based on the least squares (LS) meshless method, Zhou *et al.* [30] established a parallel computing system by connecting several CPUs through high-speed network, which greatly improves the computing efficiency for the muzzle flow field induced by a projectile. On modern high-performance graphic processing units (GPUs), Ma *et al.* [22] proposed a meshless dynamic cloud method to carry out the simulation of steady compressible flows unsteady flows past oscillatory aerofoils. But the moving boundaries have relatively little displacements.

In fluid mechanics, there are two dominant parallel computing environments, namely, parallel virtual machine (PVM) and message passing interface (MPI). Both environments provide the standard language required for parallel communication, and support multiple programs. The distributed CPU cluster model offers a convenient strategy of parallel computing. Under this model, multiple CPUs transfer information through high-speed and universal networks. Hence, the resources of each CPU are fully utilized, which greatly enhances the efficiency of parallel computing. The distributed CPU cluster model is easier to build and adjust than market-specific servers with multiprocessors [28, 31].

This paper develops a parallel meshless method based on distributed CPU cluster model and the MPI environment. The proposed method is suitable to the simulation of largescale boundary motions, and extends the applicable scope of meshless methods in the CFD field.

2. GOVERNING EQUATIONS

This paper adopts the governing equations in the arbitrary Lagrangian-Eulerian (ALE) form [22]:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=0 \tag{1}$$

$$\mathbf{U}=(\rho \quad \rho u \quad \rho v \quad \rho E)^T \tag{2}$$

$$\mathbf{F}=(\rho u' \quad \rho uu'+p \quad \rho vu' \quad \rho Eu'+pu)^T \tag{3}$$

$$\mathbf{G}=(\rho v' \quad \rho uv' \quad \rho vv'+p \quad \rho Ev'+pv)^T \tag{4}$$

where, ρ is density; p is pressure; E is the total energy per unit mass; u and v are the x-axis and y-axis components of fluid motion velocity, respectively; $u' = u - u_w$ (u_w is the velocity of grid point along the x-axis); $v' = v - v_w$ (v_w is the velocity of grid point along the y-axis).

The above governing equations are constrained by the following state equation [30]:

$$p=(\gamma-1)(\rho E-p_\infty)-p_\infty \tag{5}$$

where, γ and p_∞ are the specific heat ratio and pressure expansion coefficient of gas/liquid, respectively. If the material is air, $\gamma=1.4$ and $p_\infty=0$; if the material is water, $\gamma=7.0$ and $p_\infty=3.03975\times10^8$ Pa.

3. SOLVING PROCEDURE

The local LS fitting is adopted to approximate the spatial derivatives of each point based on its own points cloud.

Several meshless simulation methods are available for deriving the inviscid convective flux, including advection upstream splitting method (AUSM+-up) [32], artificially upstream flux vector splitting scheme (AUFS) [33] and Harten-Lax-van Leer-contact (HLLC) [34]. Among them, the AUFS is a simple, robust and accurate upwind approach proposed by Sun *et al.* [33] to solve the Euler equations. In this paper, the AUFS is modified to solve the inviscid convective flux of the ALE equations:

$$\mathbf{F}=(\rho u \quad \rho u^2+p \quad \rho uv \quad \rho Eu+pu)^T-u_w\mathbf{U}, \tag{6}$$

$$\mathbf{G}=(\rho v \quad \rho uv \quad \rho v^2+p \quad \rho Eu+pv)^T-v_w\mathbf{U}. \tag{7}$$

The following relationship can be obtained from the research of Sun *et al.* [33]:

$$n_x\mathbf{F}+n_y\mathbf{G}=\mathbf{T}^{-1}\mathbf{F}(\mathbf{TU})-(n_xu_w+n_yv_w)\mathbf{U}=\mathbf{T}^{-1}[\mathbf{F}(\mathbf{TU})-\mathbf{T}(n_xu_w+n_yv_w)\mathbf{U}] \tag{8}$$

$$\boldsymbol{T}=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & n_x & n_y & 0\\ 0 & -n_y & n_x & 0\\ 0 & 0 & 0 & 1\end{pmatrix} \tag{9}$$

where, ***T*** is the transformation matrix; $n_x = b_{ij} \Big/ \sqrt{b_{ij}^2 + c_{ij}^2}$; $n_y = c_{ij} \Big/ \sqrt{b_{ij}^2 + c_{ij}^2}$.

From the above formula, the initial intermediate flux $\mathbf{W}_{ij}^{\mathrm{AUFS}}$ between central point *i* and its neighboring point *j* can be derived as:

$$\mathbf{W}_{ij}^{\mathrm{AUFS}} = (1 - M)\mathbf{W}_1 + M\mathbf{W}_2 \tag{10}$$

$$\mathbf{W}_1 = 0.5(\mathbf{P}_i + \mathbf{P}_j) + \delta\mathbf{U},\ \mathbf{W}_2 = \mathbf{U}^{\alpha}(\tilde{u}^{\alpha} - \tilde{u}_w^{\alpha} - s_2) + \mathbf{P}^{\alpha} \tag{11}$$

where $M = s_1/(s_1 + s_2)$, $\tilde{u}^{\alpha} = n_x u_{\alpha} + n_y v_{\alpha}$, $\tilde{u}_w^{\alpha} = n_x u_{w\alpha} + n_y v_{w\alpha}$.

$$\alpha = \begin{cases} i & s_1 > 0 \\ j & s_1 \le 0 \end{cases},\quad \mathbf{P} = \begin{pmatrix} 0 \\ pn_x \\ pn_y \\ p\tilde{u} \end{pmatrix} \tag{12}$$

$$\delta\boldsymbol{U} = \frac{1}{2\overline{c}} \begin{pmatrix} p_i - p_j \\ p_i u_i - p_j u_j \\ p_i v_i - p_j v_j \\ \dfrac{\overline{c}^2}{\gamma - 1}\left(p_i - p_j\right) + \dfrac{1}{2}\left(p_i q_i^2 - p_j q_j^2\right) \end{pmatrix} \tag{13}$$

where, $q^2 = u^2 + v^2$. Intermediate sound speed $\overline{c}$ and wave speed s_1 are simply set to be their algebraic averages:

$$s_1 = (u_i + u_j)/2,\ \ \overline{c} = (c_i + c_j)/2 \tag{14}$$

and s_2 can be computed from,

$$s_2 = \begin{cases} Min\left(0, u_i - u_w - c_i, u^* - c^*\right) & s_1 > 0 \\ Max\left(0, u_j - u_w + c_j, u^* + c^*\right) & s_1 \le 0 \end{cases}. \tag{15}$$

The exact solution for the speed u^* and the sound speed c^* between two isentropic waves can be found:

$$u^* = \frac{\left(u_i + u_j\right)}{2} + \frac{\left(c_i - c_j\right)}{\gamma - 1} - u_w,\ \ c^* = \frac{\left(c_i + c_j\right)}{2} + \frac{(\gamma - 1)\left(u_i - u_j\right)}{4}. \tag{16}$$

In addition, the fourth-order Runge-Kutta method is applied to march the governing equations in time. For compressible multi-material flows, the ghost fluid method (GFM) [35] is adopted to process the interface points and set up ghost fluid points. On the interface, the flow parameters of both interface points and ghost fluid points are corrected by the solver of the local Riemann problem.

Point movement strategy is one of the challenging issues in unsteady movable and deformable boundaries problems. In this paper, the displacement of point clouds caused by the interface deformation is solved by local reconstruction [30].

4. PARALLEL COMPUTING

In parallel computing, each CPU is treated as a computing node or process, each of which is responsible for processing a region of the computational domain of the engineering problem. Each region must have a suitable workload. Several computing nodes are connected in series through other devices. Many parallel computing methods adopt the explicit solution. The computing nodes do not interfere with each other except for transferring a small amount of information near the partition boundaries.

For simplicity, our parallel computing model (Fig. 1) was constructed by the single program, multi data (SPMD) method, which is widely used for numerical calculation of complex engineering fluid flows.

When a boundary moves from one region to another, the movable boundary will move towards, intersect and then move away from the partition boundary. Therefore, our parallel meshless method should be able to transfer the information on the movable boundary between the adjacent regions, and judge whether the movable boundary intersects the partition boundary during the movement. If an intersection is possible, the partition boundary changed by point cloud reconstruction should be transferred, together with the location and flow field of the deleted and added nodes. In addition, the communication between computing nodes must be controlled to ensure the efficiency of parallel computing.

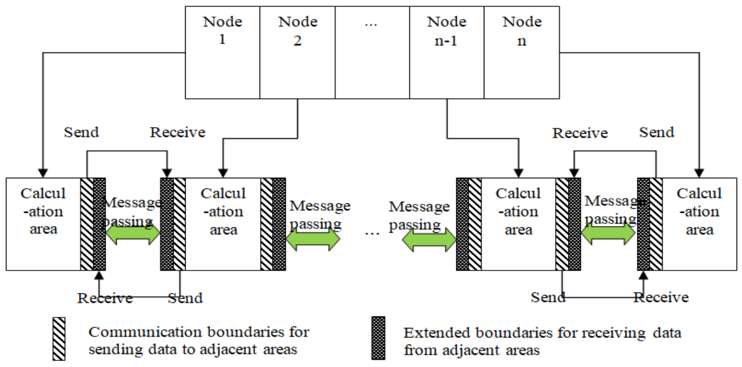


Fig. 1. Our parallel computing model.

4.1 Partitioning Method

Load balancing is the key difficulty of parallel computing. Because of the movable and deformable boundary, the area of point cloud reconstruction will be changed in our research, pushing up the computing load. This calls for a dynamic load balancing program that adjusts the workload of each computing node by changing the partition boundaries.

Through the analysis, our parallel computing program should cover three parts: flow field calculation in each partition; deletion of deformed point clouds and addition of new point clouds; transfer of computational communication between computing nodes. The second part directly bears on the workloads.

The computational domain is mainly partitioned based on geometric structure or graph theory. The geometry-based method, which is simple and easy-to-understand, divides the domain by point coordinates, argument angles, *etc.* Meanwhile, the graph theory-based method meshes the domain based on the graph of points and edges. The latter method can achieve a good quality, but requires a long time.

The geometry-based method is selected for our research. The point cloud of each region is pre-refined to clarify the features of the flow field. During the application of the said method, the workload of each computing node is controlled at the same level. Suppose the flow field contains a total of N_{total} points at time $t=0$, and m parallel computing nodes are employed. The portioning strategy can be implemented in the following steps:

Step 1: Read the information of all points in each computing node of the flow field, and set up the interval $[a, b]$ of a coordinate of the computing domain (set up the interval of the statistical angle, if the partitioning criterion is the argument angle).

Step 2: Let β (≥ 1, generally equals 1) be the calculation coefficient of each point, and $[1, 6]$ be the empirical interval of the coefficient. Then, total workload of the flow field can be quantified as $W_{\text{total}} = \sum_{i=1}^{N_{\text{total}}} \beta_i$. Hence, the mean workload assigned to each computing node is $W_{ave}=W_{total}/m$. If the point falls on the material interface, β equals 5.0; if the point falls on the moving boundary, β equals 2.5.

Step 3: According to the geometric features of the flow field and coordinate interval $[a-\delta, b+\delta]$, divide the computational domain into m regions. Note that δ is a small positive number.

Step 4: Let $(x_{i-1}, x_i]$ be the coordinate space of each region. Perform the following judgement from the first region to the last region:

$$\eta = \frac{W_i - W_{ave}}{W_{ave}} \tag{17}$$

where W_i is the workload of the ith partition. If $\eta \geq \eta_0$, reduce the value of x_i; otherwise, increase the value of x_i. In general, $\eta_0 = 3\%$ is relatively good.

Step 5: Through the above steps, the information on the point clouds of the region corresponding to the computing node is saved, and the number of points of each process is obtained as W_{total}/m.

Step 6: Point cloud is the basic calculation unit of the meshless method. If the center point of a point cloud falls on the material interface, some neighboring points must belong to the adjacent process. Therefore, the points on the material interface should be taken as communication points, in addition to saving the information of points in the current region. As shown in Fig. 2, the node information is sent from blue points 1-11 on the left side to the corresponding nodes and the red points 12-23 on the right side.

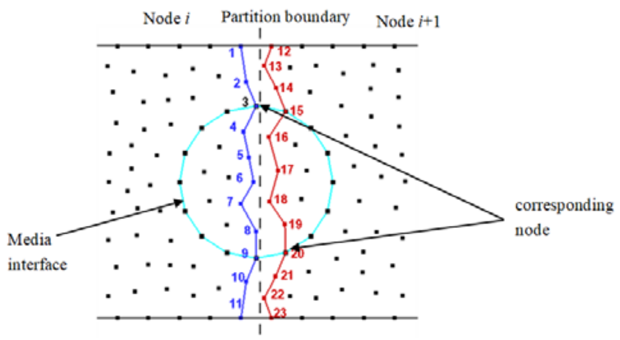


Fig. 2. Partition boundary in parallel computing.

4.2 Steps of Our Parallel Algorithm

Our parallel algorithm was designed to simulate the largescale motions of deformation boundary. The specific steps of our algorithm are as follows:

Step 1: Initialize the nodes required for parallel computing.
Step 2: Judge whether to calculate the flow field from zero moment or breakpoint. If the calculation starts from zero moment, the program reads the information on point clouds from the original mesh file, divides the computational domain, and establishes the material interface. If the calculation continues from a breakpoint, the program reads the parameter information of the entire flow field.
Step 3: Perform flow field calculation separately for each partition. Then, judge whether the point cloud structure near the movable boundary is abnormal. If yes, reconstruct the point cloud structure. After that, judge whether the deleted and added points are in the link list of communication points. If yes, the program reconstructs a new communication link. Note that only point clouds on the same side are constructed at a time, aiming to avoid confusion in the communication between adjacent regions.
Step 4: The adjacent regions transmit flow field information to each other, including the points and parameters in the link list, the movable boundary, and the point position.
Step 5: Re-determine the partition attributes of each point, and delete the points that do not belong to in the current region.
Step 6: The program outputs the data of the flow field, and then terminates all processes. Otherwise, execute Steps 3-6 again.

The parallel based meshless solver for unsteady flows was exclusively coded in Fortran 90.

5. VERIFICATION OF PARALLEL EFFICIENCY

This section attempts to verify the efficiency of our parallel algorithm. For this purpose, a cylindrical helium bubble was placed in a rectangular channel. On the right side of the channel, a planar shockwave was generated, which propagates to the left. Once hit by the shockwave, the bubble acquires the speed of traveling to the left, and subjects to force on the right side. The moving speed on the right side is faster than that on the left side. The speed difference induces shape change to the bubble. The interaction between the shockwave and the bubble is illustrated in Fig. 3, where the computational domain is 325×89 in size, the bubble diameter is 50, the bubble center falls on the axis of symmetry, the distance between the bubble and the right side is 150, and the distance between the shockwave propagating to the left and the right side is 25. As shown in Fig. 3, the computational domain consists of three parts. The parameters are as follows:

I: $\rho = 1$, $u = 0.0$, $v = 0.0$, $p = 1$, $\gamma = 1.4$, $P_\infty = 0.0$.
II: $\rho = 0.138$, $u = 0.0$, $v = 0.0$, $p = 1.0$, $\gamma = 1.67$, $P_\infty = 0.0$.
III: $\rho = 1.3764$, $u = -0.394$, $v = 0.0$, $p = 1.5698$, $\gamma = 1.4$, $P_\infty = 0.0$.

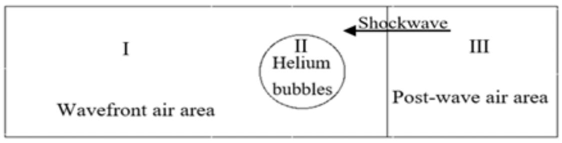


Fig. 3. The interaction between the shockwave and the bubble.

The boundary conditions are configured as follows:

The upper and lower boundaries are assumed as impermeable, the left inlet boundary as a free-stream boundary, and the right outlet boundary as a nonreflecting boundary based on Riemann invariants.

In this paper, the numerical calculation of the interaction process is carried out by using four CPUs. As shown in Fig. 4, the material interface deformed in motions lies across the second and the third regions, and intersects the movable parallel computing interfaces. The distribution of point clouds at the intersection is provided in the enlarged subgraph. The total number of points was initially 33,044.

Fig. 5 presents the calculation results of our parallel algorithm based on the AUFS scheme. It can be showed that the bubble intersects the horizontal axis of symmetry of the rectangular channel at two points.

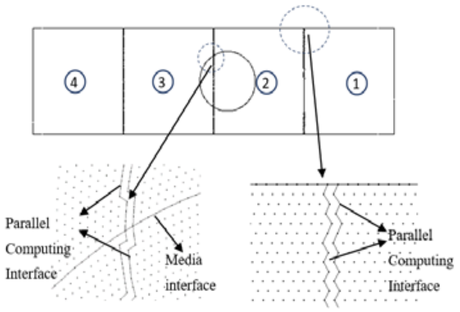


Fig. 4. Four-CPU parallel partitioning.

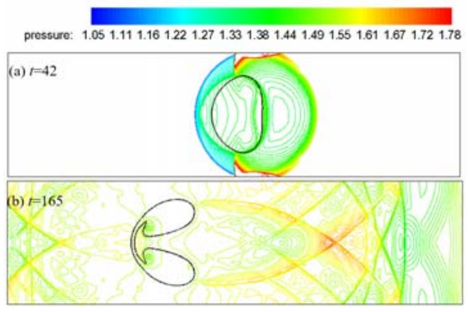


Fig. 5. Pressure contour obtained by our algorithm.

Fig. 6 compares the numerical results on the distance *l* between the two intersections obtained by our algorithm based on the AUFS scheme and Reference [34] with experimental results [36]. Within the tolerance range, the numerical results agree well with the experimental results. In this paper, the following calculations are calculated using the AUFS scheme. The shape and position of the bubble and its surrounding flow field changes constantly with the motion of the shockwave, which is consistent with the experimental phenomena. Firstly, the shock wave travels to the right side of the bubble, forcing the latter to deform and move to the left. The interaction creates a reflected shockwave and a transmitted shockwave. The internal and external shockwaves have a speed difference, which turns the bubble into a blow. The transmitted shockwave passes through the right side of the bubble, producing two reflected shockwaves at the upper and lower boundaries. The reflected shockwaves act on the bubble again, causing further shape changes to the bubble. Finally, the bubble assumes the shape of a broad bean and eventually ruptures.

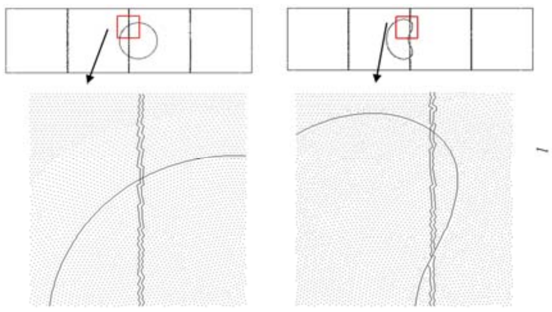


Fig. 6. Comparison of experimental and numerical results.

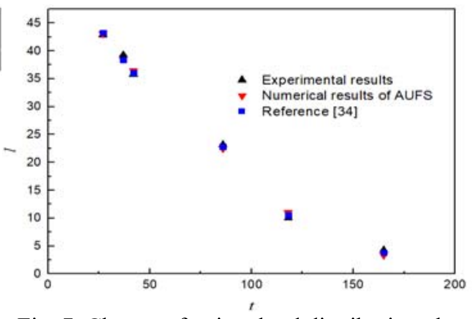


Fig. 7. Change of point cloud distribution obtained by our algorithm.

Fig. 7 records the point cloud distribution of the bubble from the initial moment to the moment when the entire bubble just enters the third region. It can be seen that our parallel algorithm captures the crossing of movable boundary over the partition boundary. The position and shape of the partition boundary changed with those of the moving boundary. Our algorithm fully demonstrated the change of complex shockwaves and the deformations of the material interface. The results show that our algorithm is feasible to solve the flow problems with largescale motions of deformable boundary.

To further verify its computing efficiency, our parallel algorithm is applied to simulate the flow field with 33,044 and 66,089 discrete points, respectively. Two computing nodes are used in parallel, and then four. The simulated results and speedup data are listed in Table 1. We define the calculation efficiency as:

$$\xi = \frac{t_1}{t_2 \times m} \tag{18}$$

where ξ is the calculation efficiency, t_1 the calculation time of single computer node, and t_2 is the parallel computing time, and m is the number of computing nodes.

Table 1. Speedup and efficiency of our algorithm.

Discrete nodes	Number of computing nodes	Speedup	ξ
33,044	2	1.728	86.4%
33,044	4	3.092	77.3%
66,089	2	1.780	89.0%
66,089	4	3.248	81.2%

It can be seen that our parallel algorithm is workable but not highly efficient. The efficiency is just over 80% on adopting four nodes. Under the same computing nodes, the parallel efficiency of the meshless method in Reference [30] is above 90%. The lack of efficiency is attributable to the following facts: (1) The moveable boundary intersects the material interface, making it difficult to balance the workload of each node; (2) The amount of information exchange between adjacent partitions is relatively large, including the information of neighboring points, the reconstruction of point clouds, repartition, *etc*.

6. CONCLUSIONS

In real-world scenarios, engineering problems are very complicated, especially when the boundaries are movable and deformable. It often takes a long time to numerically simulate such problems with a single CPU. This paper proposes a dynamic parallel meshless computing algorithm that fully displays the merits of meshless method in solving flow fields with movable and deformable boundaries. The parallel computing strategy and partitioning strategy are explained in detail. Take the interaction between a helium bubble and a shockwave as an example. Our algorithm is applied to compute the flow field with different numbers of discrete points and partitions. The results show that our algorithm achieves an efficiency of over 80%, an evidence of the feasibility of our solution. This paper provides a new method to improve the calculation speed for similar problems.

REFERENCES

1. O. Al-Abbasi, B. Sarac, and T. Ayhan, “Experimental investigation and CFD modeling to assess the performance of solar air humidifier,” *International Journal of Heat and Technology*, Vol. 37, 2019, pp. 357-364.
2. E. Adaze, A. Al-Sarkhi, H. M. Badr, and E. Elsaadawy, “Current status of CFD modeling of liquid loading phenomena in gas wells: a literature review,” *Journal of Petroleum Exploration and Production Technology*, Vol. 9, 2019, pp. 1397-1411.
3. A. Bazgir and N. Nabhani, “Computational fluid dynamics comparison of separation performance analysis of uniform and nonuniform counter-flow Ranque-Hilsch Vortex Tubes (RHVTs),” *International Journal of Heat and Technology*, Vol. 36, 2018, pp. 643-656.
4. S. Chourasiya, P. Shrivastava, and P. Jain, “A review on experimental and CFD analysis on a hybrid cooler,” *Journal of Energy Environment and Carbon Credits*, Vol. 9, 2020, pp. 1-5.
5. A. Marreiro, F. Beaumont, R. Taïar, and G. Polidori, “Application of infrared ther-

mal imaging and computational fluid dynamics techniques to whole body cryotherapy (WBC)," *Instrumentation Mesure Métrologie*, Vol. 16, 2017, pp. 11-32.

6. J. J. Chen, "Unstructured mesh generation and its parallelization," Ph.D. Thesis, Department of Engineering and Scientific Computation, Zhejiang University, 2006.
7. N. Chrisochoides and D. Nave, "Parallel Delaunay mesh generation kernel," *International Journal for Numerical Methods in Engineering*, Vol. 58, 2003, pp. 161-176.
8. D. Nave, N. Chrisochoides, and L. P. Chew, "Guaranteed-quality parallel Delaunay refinement for restricted polyhedral domains," *Computational Geometry*, Vol. 28, 2004, pp. 191-215.
9. A. Afzal, Z. Ansari, A. R. Faizabadi, and M. K. Ramis, "Parallelization strategies for computational fluid dynamics software: state of the art review," *Archives of Computational Methods in Engineering*, Vol. 24, 2017, pp. 337-363.
10. S. Jaiswal, R. Reddy, R. Banerjee, S. Sato, D. Komagata, M. Ando, and J. Okada, "An efficient GPU Parallelization for arbitrary collocated polyhedral finite volume grids and its application to incompressible fluid flows," in *Proceedings of the 23rd International Conference on High Performance Computing Workshops*, 2016, pp. 81-89.
11. P. Ding and Z. Liu, "Accelerating phase-field modeling of solidification with a parallel adaptive computational domain approach," *International Communications in Heat and Mass Transfer*, Vol. 111, 2020, pp. 1-12.
12. L. Linardakis and N. Chrisochoides, "Delaunay decoupling method for parallel guaranteed quality planar mesh refinement," *SIAM Journal on Scientific Computing*, Vol. 27, 2006, pp. 1394-1423.
13. Y. Liang, J. Chen, L. Chen, and Y. Zheng, "Parallel planar Delaunay mesh generation," *Journal of Zhejiang University* (*Engineering Science*), Vol. 42, 2008, pp. 558-564.
14. T. Jiang, Z. C. Chen, W. G. Lu, J. Y. Yuan, and D. S. Wang, "An efficient split-step and implicit pure mesh-free method for the 2D/3D nonlinear gross-Pitaevskii equations," *Computer Physics Communications*, Vol. 231, 2018, pp. 19-30.
15. S. Idelsohn, E. Onate, N. Calvo, and F. del Pin, "The meshless finite element method," *International Journal for Numerical Methods in Engineering*, Vol. 58, 2003, pp. 893-912.
16. S. Idelsohn, E. Onate, and F. del Pin, "A Lagrangian meshless finite element method applied to fluid-structure interaction problems," *Computers & Structures*, Vol. 81, 2003, pp. 655-671.
17. C. Y. Ku, C. Y., Liu, W. Yeih, C. S. Liu, and C. M. Fan, "A novel space-time meshless method for solving the backward heat conduction problem," *International Journal of Heat and Mass Transfer*, Vol. 130, 2019, pp. 109-122.
18. I. Ahmad, M. Riaz, M. Ayaz, M. Arif, S. Islam, and P. Kumam, "Numerical simulation of partial differential equations via local meshless method," *Symmetry*, Vol. 11, 2019, p. 257.
19. M. Depolli and G. Kosec, "Assessment of differential evolution for multi-objective optimization in a natural convection problem solved by a local meshless method," *Engineering Optimization*, Vol. 49, 2017, pp. 675-692.
20. M. Hashemabadi and M. Hadidoolabi, "Efficient gridless method using constrained weights optimization for two-dimensional unsteady inviscid flows at low angles of

attack," *Journal of Aerospace Engineering*, Vol. 30, 2017, No. 04017052.

21. J. Slak and G. Kosec, "Parallel coordinate free implementation of local meshless method," in *Proceedings of the 41st International Convention on Information and Communication Technology*, *Electronics and Microelectronics*, 2018, pp. 0172-0178.
22. Z. H. Ma, H. Wang, and S. H. Pu, "A parallel meshless dynamic cloud method on graphic processing units for unsteady compressible flows past moving boundaries," *Computer Methods in Applied Mechanics and Engineering*, Vol. 286, 2015, pp. 146-165.
23. G. Kosec, M. Depolli, A. Rashkovska, and R. Trobec, "Super linear speedup in a local parallel meshless solution of thermo-fluid problems," *Computers & Structures*, Vol. 133, 2014, pp. 30-38.
24. C. Moulinec, R. Issa, J. Marongiu, and D. Violeau, "Parallel 3-D SPH simulations," *Computer Modeling in Engineering and Sciences*, Vol. 25, 2008, pp. 133-147.
25. G. Yagawa and M. Shirazaki, "Parallel computing for incompressible flow using a nodal-based method," *Computational Mechanics*, Vol. 23, 1999, pp. 209-217.
26. T. Fujisawa, M. Inaba, and G. Yagawa, "Parallel computing of high-speed compressible flows using a node-based finite-element method," *International Journal for Numerical Methods in Engineering*, Vol. 58, 2003, pp. 481-511.
27. K. T. Danielson, S. Hao, W. K. Liu, R. A. Uras, and S. Li, "Parallel computation of meshless methods for explicit dynamic analysis," *International Journal for Numerical Methods in Engineering*, Vol. 47, 2000, pp. 1323-1341.
28. I. V. Singh, "Parallel implementation of the EFG method for heat transfer and fluid flow problems," *Computational Mechanics*, Vol. 34, 2004, pp. 453-463.
29. Q. H. Zeng, "Parallel algorithms and parallel implementation of meshless numerical simulation," Ph.D. Thesis, Department of Mechanics and Mechanical, University of Science and Technology of China, 2006.
30. X. Zhou, "The research on Gridless method for complex unsteady flows involving moving boundaries," Ph.D. Thesis, Department of Computer Science, Nanjing University of Science and Technology, 2012.
31. G. Chen, J. Wang, Z. Zhang, Z. Tian, L. Li, H. Kang, and Y. Jin, "Distributed-parallel CFD computation for all fuel assemblies in PWR core," *Annals of Nuclear Energy*, Vol. 141, 2020, pp. 1-12.
32. M. S. Liou, "A sequel to AUSM, Part II, AUSM+-up for all speeds," *Journal of Computational Physics*, Vol. 214, 2006, pp. 137-170.
33. M. Sun and K. Takayama, "An artificially upstream flux vector splitting scheme for the Euler equations," *Journal of Computational Physics*, Vol. 189, 2003, pp. 305-329.
34. B. Wang, "Numerical simulation technique research for multi-material flows on unstructural moving grids," Ph.D. Thesis, Department of Computer Science, Nanjing University of Science and Technology, 2008.
35. R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," *Journal of Computational Physics*, Vol. 152, 1999, pp. 457-492.
36. J. F. Haas and B. Sturtevant, "Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities," *Journal of Fluid Mechanics*, Vol. 181, 1987, pp. 41-76.

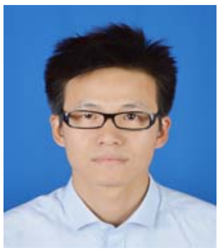
Liang Wang (王亮) received his Ph.D. degree in Engineering Thermophysics from Nanjing University of Science and Technology in 2016. He is currently a Lecturer at Nanjing Institute of Technology. His research interests include computational fluid dynamics, multi-medium flow with deformable boundary, and gas combustion.

Rui Xue (薛锐) received his Ph.D. degree in Engineering Thermophysics from Nanjing University of Science and Technology in 2016. He is currently an Associate Professor at Nanjing Institute of Technology. His research interests include pyrotechnic combustion mechanism and application research, computational fluid dynamics.

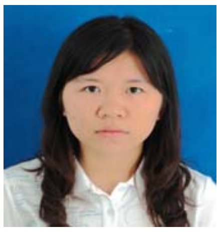
Ning Cai (蔡宁) received his Ph.D. degree in Thermal Energy Engineering from University of Shanghai for Science and Technology in 2012. He is currently a Lecturer at Nanjing Institute of Technology. His research interests include building energy efficiency, computational fluid dynamics, and enhanced heat and mass transfer.

Pan Chen (陈磐) received her Ph.D. degree in Power Engineering and Engineering Thermophysics from Southeast University in 2012. She is currently a Lecturer at Nanjing Institute of Technology. Her research interests include numerical simulation of gas and solid two-phase flow, computational fluid dynamics.

Xiaobo Cui (崔晓波) received his Ph.D. degree in Power Engineering and Engineering Thermophysics from Southeast University in 2015. He is currently a Lecturer at Nanjing Institute of Technology. Her research interests include modeling and optimizing control of energy systems, computational fluid dynamics.



Wei Wu (吴伟) received his Ph.D. degree in Engineering Thermophysics from Nanjing University of Science and Technology in 2015. His research interests include computational fluid dynamics, enhanced heat and mass transfer.



Miaomiao Niu (牛淼淼) received her Ph.D. degree in Power Engineering and Engineering Thermophysics from Southeast University in 2015. She is currently a Lecturer at Nanjing Institute of Technology. Her research interests include high value energy conversion and utilization of solid waste, enhanced heat and mass transfer.

Dongliang Zhang (张东亮) received her Ph.D. degree in Thermal Energy Engineering from Tongji University in 2012. She is currently an Associate Professor at Nanjing Institute of Technology. Her research interests include building energy efficiency, computational fluid dynamics, and enhanced heat and mass transfer.

Zhao Zhang (张钊) received his Ph.D. degree in Ergonomics and Environmental Engineering from Nanjing University of Aeronautics and Astronautics in 2018. He is currently a Lecturer at Nanjing Institute of Technology. His research interests include algorithm design and analysis, computational fluid dynamics.

Xiaosong Zhang (张小松) received her Ph.D. degree in Power Engineering and Engineering Thermophysics from Southeast University. He is a doctoral tutor and Distinguished Professor of Southeast University. His research interests include refrigeration and air conditioning, solar energy utilization and building energy saving.