Journal of Computational Physics 568 (2027) 115412

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Unified gas-kinetic scheme for unsteady multiscale flows with moving boundaries

Yue Zhang [a], Wenpei Long [a], Junzhe Cao [a], Kun Xu [a,b,c,*]

[a] Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[b] Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[c] Shenzhen Research Institute, Hong Kong University of Science and Technology, Shenzhen, China

ARTICLE INFO

Keywords:
Overset mesh
Implicit unified gas-kinetic scheme
Unsteady flow
Moving body problem

ABSTRACT

Simulating multiscale flows with moving boundaries, such as hypersonic multi-body separation and flows in micro-electro-mechanical systems (MEMS), requires robust numerical methods that handle large relative mesh motion together with complex flow physics. This paper presents an overlapping moving-mesh technique coupled with the unified gas-kinetic scheme (UGKS). An unsteady implicit dual-time UGKS is extended to overset meshes, with memory-efficient data handling and parallel computing optimizations. The method is assessed on MEMS-type moving-boundary problems, an oscillating-cylinder benchmark, and a three-dimensional multi-body demonstration. The results indicate that the proposed overset UGKS provides an accurate and practical tool for dynamic multiscale flow interactions in complex geometries.

1. Introduction

The development of near-space hypersonic vehicles faces a major bottleneck in multi-body separation, where transient and unsteady aerodynamic coupling severely impacts mission reliability. Similarly, the continued advancement of Micro-Electro-Mechanical Systems (MEMS) relies on the accurate characterization of moving components, such as actuators and sensors. A unifying challenge in both fields is that their operating environments involve complex multi-scale flow phenomena [1,2], encompassing both rarefied and continuum regimes. Therefore, it is essential to develop advanced numerical methods capable of simulating multi-scale flows with dynamic moving meshes.

Due to the significantly reduced gas density in near-space environments and the extremely small characteristic lengths in MEMS devices, non-equilibrium flow phenomena become highly pronounced. Accurately capturing these non-equilibrium physics requires additional degrees of freedom, which challenges the validity of the macroscopic Navier-Stokes equations and necessitates the use of gas kinetic theory. At the fundamental level, the Boltzmann equation governs these flows, resolving physics at the scales of the particle mean free path and mean collision time.

Traditionally, approaches to simulating non-equilibrium flows fall into two categories: stochastic and deterministic methods. Stochastic methods, most notably the direct simulation Monte Carlo (DSMC) method [3], employ representative particles to track the microscopic gas distribution function and simulate its evolution. By characterizing non-equilibrium physics via adaptive particles

* Corresponding author at: Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.
E-mail addresses: yzhangnl@connect.ust.hk (Y. Zhang), wlongab@connect.ust.hk (W. Long), jcaobb@connect.ust.hk (J. Cao), makxu@ust.hk (K. Xu).

https://doi.org/10.1016/j.jcp.2026.115412
Received 10 April 2026; Received in revised form 24 August 2026; Accepted 18 September 2026
Available online 22 September 2026
0021-9991/© 2026 Elsevier Inc. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

in a local velocity space, DSMC achieves high computational efficiency for high-speed rarefied flows. Furthermore, because particle free-streaming and collisions strictly adhere to conservation laws, the method is highly robust. However, its inherent stochastic nature introduces statistical noise. Consequently, for low-speed flow simulations, DSMC requires a prohibitively large number of particles or extensive statistical averaging to achieve an acceptable signal-to-noise ratio.

Conversely, deterministic approaches, such as the discrete velocity method (DVM) [4,5], solve the Boltzmann equation (or related kinetic models) based on a discrete velocity distribution function. Because the gas fluxes at cell interfaces are constructed within this deterministic framework, these methods yield accurate solutions entirely free of statistical noise. Nevertheless, discretizing the global velocity space incurs substantial memory consumption and computational costs, particularly for three-dimensional, high-speed flow simulations.

A common limitation of both traditional stochastic and deterministic methods is the use of operator splitting to separate free transport and collisions, which often introduces numerical dissipation proportional to the time step. Consequently, the spatial mesh size and time step are strictly constrained by the particle mean free path and mean collision time, respectively. To overcome these limitations, unified gas-kinetic methods have been developed in recent years. Prominent examples include the DVM-based unified gas-kinetic scheme (UGKS) [6,7], the discrete UGKS (DUGKS) [8], and the particle-based unified gas-kinetic wave-particle (UGKWP) method [9,10]. By utilizing the integral or finite-difference solution of the kinetic equation along characteristics, these schemes introduce the time step into the flux evaluation as a physical observation scale, effectively coupling particle collisions with free transport. As a result, the physical laws are directly modeled within the discretized space [11], and the unified preserving (UP) properties of these schemes have been rigorously proven [12]. These methods function as highly accurate Boltzmann solvers in the rarefied regime and seamlessly recover the Navier-Stokes equations in the continuum regime, bypassing the stringent constraints imposed by microscopic scales. These developments provide the kinetic foundation for the moving-mesh extension pursued in the present work.

Furthermore, to make the unified gas-kinetic scheme (UGKS) viable for large-scale engineering applications, this paper incorporates advanced acceleration and optimization techniques. Recently, substantial efforts have been dedicated to enhancing the computational performance and reducing the memory footprint of multi-scale algorithms like UGKS and DUGKS. A prominent memory-reduction strategy is the implementation of an unstructured discrete velocity space (DVS) [13,14], which drastically reduces the number of discrete velocity points required for three-dimensional problems without sacrificing accuracy. Similarly, the adaptive UGKS (AUGKS) [15] utilizes an adaptive velocity space decomposition to efficiently capture local non-equilibrium phenomena. By employing a discrete distribution function in highly non-equilibrium regions and a continuous distribution function in near-continuum regimes, AUGKS significantly conserves memory. This adaptive framework has been further extended to three-dimensional flows involving rotational and vibrational non-equilibrium [16], and analogous adaptive strategies have been successfully applied to DUGKS [17] and UGKWP [18,19].

In terms of temporal acceleration, the integration of implicit methods has proven highly effective. Implicit UGKS (IUGKS) formulations have been developed for both steady [20,21] and unsteady [22] flows. By coupling macroscopic predictions with microscopic implicit iterations, IUGKS achieves convergence rates one to three orders of magnitude faster than its explicit counterpart across all flow regimes. This implicit framework has also been expanded to simulate three-dimensional thermal non-equilibrium flows with internal degrees of freedom [23]. Additionally, the incorporation of multigrid methods has yielded an order-of-magnitude increase in computational speed [24]. Recently, these advancements culminated in the development of the implicit adaptive UGKS (IAUGKS) [25], which synthesizes both adaptive and implicit techniques. Another notable advancement is the general synthetic iterative scheme [26,27], which couples macroscopic equations with the Boltzmann equation to dynamically adjust the velocity distribution function for rapid convergence. These implicit and efficiency-oriented developments are adopted in the present solver; their coupling with overlapping meshes is detailed in Section 3.

Parallel to algorithmic accelerations, the simulation of multi-scale flows involving moving meshes has garnered significant attention. Chen et al. [28] extended the UGKS to moving-mesh problems by employing a reference-frame method coupled with a velocity-space adaptation technique. Wang et al. [2,29,30] developed an Arbitrary Lagrangian-Eulerian (ALE) based DUGKS to simulate transonic continuum and rarefied gas flows with moving boundaries. Similarly, Tao et al. [31,32] combined the immersed boundary (IB) method with DUGKS to model particle-fluid interactions and thermal rarefied gas flows in MEMS devices. More recently, Zeng et al. [33] introduced a GSIS-ALE approach for moving-boundary problems in rarefied flows using an overlapping-mesh technique; by leveraging synthetic iterative schemes, this method significantly improves computational efficiency over explicit solvers and removes the Courant-Friedrichs-Lewy (CFL) timestep restriction.

The present work does not introduce a new UGKS flux formulation; it builds on the unsteady implicit UGKS of Zhu et al. [22] and the memory-efficient parallelization of Zhang et al. [34]. The novelty is a kinetic overset coupling strategy for dual-time UGKS, rather than a new flux discretization or a standalone parallel implementation. Relative to fixed-mesh UGKS/IUGKS [6,20,22], the present method supports unstructured multi-body overset motion without mesh deformation. Compared with the moving-mesh UGKS of Chen et al. [28], which employs a reference-frame formulation, overlapping body-fitted zones are used here. Unlike ALE-DUGKS [29,30], the overset mesh avoids Lagrangian deformation and associated mesh-quality issues. In contrast to IB-DUGKS [31,32], near-body resolution is retained on body-fitted grids rather than through immersed boundaries. Finally, GSIS-ALE/overset [33] accelerates a different kinetic framework and does not employ the present dual-time UGKS residual/LU-SGS coupling of both $\mathbf{W}$ and f. The kinetic overset contribution comprises: wall-distance hole cutting with buffer layers on unstructured moving meshes; interpolation of both the macroscopic conservatives $\mathbf{W}$ and the velocity-dependent distribution f with identical gradient-corrected inverse-distance weights, so that $\mathbf{W}$–f consistency is preserved at the fringe; and overset exchange treated analogously to MPI communication inside residual evaluation and LU-SGS, with the overset connectivity frozen during all dual-time inner iterations of a physical step.

The remainder of this paper is organized as follows. Section 2 introduces the implicit unified gas-kinetic scheme (UGKS) for unsteady flows. Section 3 details the extension of this implicit solver to moving-mesh problems utilizing an overlapping-mesh technique. Section 4 presents the numerical results to demonstrate the performance of the proposed method. In particular, a Taylor–Green accuracy test is first used to assess the spatial order of the overset interpolation. Finally, Section 5 concludes the paper with a summary of the key findings.

2. Implicit unified gas-kinetic scheme

In this section, the implicit unified gas-kinetic scheme for unsteady flow [22] is introduced. In addition, mesh motion is accounted for in this scheme.

2.1. Unified gas-kinetic scheme

In this work, the Shakhov model is used in the UGKS. With mesh motion, it can be written as

$$\frac{\partial f}{\partial t} + \mathbf{w}\cdot\nabla f = \frac{f^+ - f}{\tau}, \tag{1}$$

where $f = f(\mathbf{x}, t, \mathbf{v}, \boldsymbol{\xi})$ is the distribution function for gas molecules at physical space location $\mathbf{x}$ with microscopic translation velocity $\mathbf{v}$ and internal velocity $\boldsymbol{\xi}$, $\mathbf{w} = \mathbf{v} - \mathbf{U}$ is the relative velocity between the gas molecules and the mesh velocity $\mathbf{U}$, τ is particle collision time, and f^+ is the modified equilibrium distribution function. The modified equilibrium distribution function is given by

$$f^+ = g\left[1 + (1-\mathrm{Pr})\mathbf{c}\cdot\mathbf{q}\left(\frac{c^2}{\mathrm{R}T} - 5\right)/(5pRT)\right] = g + g^+,$$

where g is the Maxwellian distribution, Pr is the Prandtl number, $\mathbf{c} = \mathbf{v} - \mathbf{V}$ is the random velocity, $\mathbf{V}$ is the macroscopic velocity, $\mathbf{q}$ is the heat flux, R is gas constant and T is the temperature. The Maxwellian distribution is

$$g = \rho\left(\frac{\lambda}{\pi}\right)^{\frac{K+D}{2}} e^{-\lambda((\mathbf{v}-\mathbf{V})^2 + \boldsymbol{\xi}^2)},$$

where ρ is density, $\lambda = m/(2k_B T)$, m is molecule mass, k_B is Boltzmann constant, D is the spatial dimension, K is the number of internal degrees of freedom and $\boldsymbol{\xi}^2 = \xi_1^2 + \xi_2^2 + \cdots + \xi_K^2$.

The collision terms satisfy the conservative constraint, or compatibility condition.

$$\int (f^+ - f)\Psi \,\mathrm{d}\mathbf{v}\,\mathrm{d}\boldsymbol{\xi} = 0,$$

where $\Psi = (1, \mathbf{v}, \frac{1}{2}(\mathbf{v}^2 + \boldsymbol{\xi}^2))^T$ is the collision invariants and $\mathrm{d}\boldsymbol{\xi} = \mathrm{d}\xi_1\,\mathrm{d}\xi_2\cdots\mathrm{d}\xi_K$. The macroscopic conservative variables and heat flux can be calculated via

$$\mathbf{W} = \begin{pmatrix}\rho \\ \rho\mathbf{V} \\ \rho E\end{pmatrix} = \int \Psi f \,\mathrm{d}\mathbf{v}\,\mathrm{d}\boldsymbol{\xi}, \qquad \mathbf{q} = \frac{1}{2}\int (\mathbf{v}-\mathbf{V})(|\mathbf{v}-\mathbf{V}|^2 + \boldsymbol{\xi}^2) f \,\mathrm{d}\mathbf{v}\,\mathrm{d}\boldsymbol{\xi}.$$

To reduce memory usage, two reduced distribution functions

$$h(\mathbf{x}, t, \mathbf{v}) = \int f \,\mathrm{d}\boldsymbol{\xi}, \qquad b(\mathbf{x}, t, \mathbf{v}) = \int \boldsymbol{\xi}^2 f \,\mathrm{d}\boldsymbol{\xi}$$

are introduced. The corresponding reduced Shakhov equations and equilibrium closures follow Zhu et al. [22]; in the following, f is used in place of h and b for notational simplicity.

Similar with the whole physical domain Ω discretized into small cells Ω_i, the whole velocity space is discretized into velocity space cells $\delta\mathbf{v}_k$. The governing equation of the distribution function at the discretized velocity points can be written as

$$\frac{\partial f_k}{\partial t} + \mathbf{w}_k\cdot\nabla f_k = \frac{f_k^+ - f_k}{\tau}, \tag{2}$$

where $f_k = f_k(\mathbf{x}, t) = f(\mathbf{x}, t, \mathbf{v}_k)$. By taking the integration of the above equation over the physical cell Ω_i and from time t^n to t^{n+1}, the distribution function at the cell interface can be updated by

$$f_{i,k}^{n+1} = f_{i,k}^n - \frac{1}{|\Omega_i|}\sum_{j\in N(i)} S_{ij}\mathcal{F}_{ij,k} + \int_{t^n}^{t^{n+1}} \frac{f_{i,k}^+ - f_{i,k}}{\tau_i}\,\mathrm{d}t,$$

where $f_{i,k}^n$ and $f_{i,k}^{n+1}$ are the average of distribution function over cell Ω_i and $\delta\mathbf{v}_k$ at time t^n and t^{n+1}, $|\Omega_i|$ denotes the volume of cell i, $N(i)$ is the set of all interface-adjacent neighboring cells of cell i and j is one of the neighboring cells of i. The interface between cells i and j is labeled ij and has area S_{ij}. And the microscopic flux $\mathcal{F}_{ij,k}$ is

$$\mathcal{F}_{ij,k} = w_{k,n}\int_{t^n}^{t^{n+1}} f_{ij,k}\,\mathrm{d}t, \tag{3}$$

where $w_{k,n} = \mathbf{w}_k \cdot \mathbf{n}_{ij}$ is the normal velocity via surface ij with normal direction $\mathbf{n}_{ij}$, and $f_{ij,k}$ is the average distribution function over cell $\delta\mathbf{v}_k$ at the center of cell interface ij.

By taking moments of the above equation and considering the compatibility condition, the governing equation of macroscopic conservative values can be obtained as

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{1}{\Omega_i} \sum_{j \in N(i)} S_{ij} \mathbf{F}_{ij}, \tag{4}$$

where $\mathbf{W}_i^n$ and $\mathbf{W}_i^{n+1}$ are cell-averaged conservative value at time t^n and t^{n+1}, and the macroscopic flux is defined by

$$\mathbf{F}_{ij} = \sum_k w_k \begin{pmatrix} \mathcal{H}_{ij,k} \\ \mathbf{v}_k \mathcal{H}_{ij,k} \\ \frac{1}{2}(\mathbf{v}_k^2 \mathcal{H}_{ij,k} + \mathcal{B}_{ij,k}) \end{pmatrix}, \tag{5}$$

where w_k is the weight of velocity space cell $\delta\mathbf{v}_k$ (set as the volume of the cell for unstructured velocity space in this work), and $\mathcal{H}_{ij,k}$ and $\mathcal{B}_{ij,k}$ are defined by

$$\mathcal{H}_{ij,k} = w_{k,n} \int_{t^n}^{t^{n+1}} h_{ij,k} \, \mathrm{d}t, \ \mathcal{B}_{ij,k} \quad = w_{k,n} \int_{t^n}^{t^{n+1}} b_{ij,k} \, \mathrm{d}t.$$

To construct the numerical fluxes at $\mathbf{x}_0 = (0,0,0)^T$, the integral solution along the characteristic line $\mathbf{x}' = \mathbf{x}_0 - \mathbf{w}(t - t')$ of BGK Eq. (2) gives

$$f_k(\mathbf{x}, t, \boldsymbol{\xi}) = \frac{1}{\tau} \int_{t_0}^{t} f_k^+(\mathbf{x}', t') e^{-(t-t')/\tau} \, \mathrm{d}t' + e^{-t/\tau} f_{0,k}(\mathbf{x} - \mathbf{w}_k t),$$

where $f_{0,k}(\mathbf{x})$ is the initial distribution function at the beginning of each step t_n, and $f^+(\mathbf{x}, t)$ is the effective equilibrium state distributed in space and time around $\mathbf{x}$ and t. The integral solution provides a multiscale model of the evolution from an initial non-equilibrium distribution f to an equilibrium distribution f^+ via collisions.

To achieve second-order accuracy, the initial distribution function $f_{0,k}(\mathbf{x})$ is approximated as

$$f_{0,k}(\mathbf{x}) = \begin{cases} f_k^l + \mathbf{x} \cdot \nabla f_k^l, & w_{k,n} > 0, \\ f_k^r + \mathbf{x} \cdot \nabla f_k^r, & w_{k,n} < 0, \end{cases}$$

where f_k^l and f_k^r are the reconstructed initial distribution functions at the left and right sides of the interface. The equilibrium state is approximated [35] as

$$f_k^+(\mathbf{x}, t) \approx g_{0,k} + g_{0,k}^+ + \mathbf{x} \cdot \nabla g_{0,k} + \frac{\partial g_{0,k}}{\partial t} t.$$

It is derived from the conservative variables contributed by all particles transported from both sides of the interface. Details of the distribution functions and derivatives are demonstrated in earlier work [6,11].

In summary, the distribution function at the cell interface is

$$f_k(\mathbf{0}, t, \boldsymbol{\xi}) = \begin{cases} c_1 f_k^l + c_2 \mathbf{w}_k \cdot \nabla f_k^l + c_3 (g_{0,k} + g_{0,k}^+) + c_4 \mathbf{w}_k \cdot \nabla g_{0,k} + c_5 \partial_t g_{0,k}, & w_{k,n} > 0, \\ c_1 f_k^r + c_2 \mathbf{w}_k \cdot \nabla f_k^r + c_3 (g_{0,k} + g_{0,k}^+) + c_4 \mathbf{w}_k \cdot \nabla g_{0,k} + c_5 \partial_t g_{0,k}, & w_{k,n} < 0, \end{cases} \tag{6}$$

and

$$\begin{aligned} c_1 &= e^{-t/\tau}, \\ c_2 &= -t e^{-t/\tau}, \\ c_3 &= 1 - e^{-t/\tau}, \\ c_4 &= t e^{-t/\tau} - \tau(1 - e^{-t/\tau}), \\ c_5 &= t - \tau(1 - e^{-t/\tau}). \end{aligned}$$

Far-field and other artificial boundaries are imposed with ghost cells: macroscopic states follow the corresponding Euler-type prescriptions, and the distribution is taken as the associated Maxwellian, with interface fluxes selected by the particle velocity direction. Solid walls use the isothermal Maxwell boundary condition of UGKS [6,22]: the reflected part is a wall Maxwellian $g_w(\rho_w, T_w, \mathbf{v}_w)$, with wall density ρ_w fixed by the no-penetration condition on the mass flux. From the resulting wall macroscopic flux $\mathbf{F}_w = (F_{w0}, \mathbf{F}_{mw}, F_{w4})^T$, the surface pressure, shear stress and heat flux are

$$p_w = \mathbf{F}_{mw} \cdot \mathbf{n}_w, \qquad \sigma_w = |\mathbf{F}_{mw} - p_w \mathbf{n}_w|, \qquad q_w = F_{w4}.$$

The source term can be discretized by the trapezoidal rule as

$$\int_{t^n}^{t^{n+1}} \frac{f_{i,k}^+ - f_{i,k}}{\tau_i} \, \mathrm{d}t = \frac{\Delta t}{2} \left(\frac{f_{i,k}^{+(n+1)} - f_{i,k}^{n+1}}{\tau_i^{n+1}} + \frac{f_{i,k}^{+(n)} - f_{i,k}^n}{\tau_i^n} \right),$$

where the $f^{+(n+1)}$ and τ^{n+1} can be obtained through the macroscopic conservative values $\mathbf{W}^{n+1}$.

2.2. Implicit unified gas-kinetic scheme for unsteady flows

The present solver follows the unsteady dual-time implicit UGKS of Zhu et al. [22]. With mesh motion retained through the relative velocity $\mathbf{w} = \mathbf{v} - \mathbf{U}$, the cell-averaged macroscopic and microscopic equations read

$$\frac{\partial \mathbf{W}_i}{\partial t} + \frac{1}{|\Omega_i|}\sum_{j\in N(i)} S_{ij}\hat{\mathbf{F}}_{ij} = 0,$$

$$\frac{\partial f_{i,k}}{\partial t} + \frac{1}{|\Omega_i|}\sum_{j\in N(i)} S_{ij}\hat{F}_{ij,k} = \frac{f^{+}_{i,k} - f_{i,k}}{\tau_i},$$

where the hat denotes UGKS fluxes averaged over a short explicit interval Δt_s (CFL = 0.5) so that the multiscale interface evolution is preserved. Physical time is discretized by the Crank–Nicolson scheme and advanced by dual-time iterations indexed by s, yielding the delta-form systems

$$\left(\frac{1}{\Delta t'} + \frac{1}{\Delta t}\right)\Delta \mathbf{W}_i^{n+1,s} + \frac{1}{2|\Omega_i|}\sum_{j\in N(i)} S_{ij}\Delta\hat{\mathbf{F}}_{ij}^{n+1,s} = \mathbf{R}_i^{n+1,s}, \tag{7}$$

$$\left(\frac{1}{\Delta t'} + \frac{1}{\Delta t} + \frac{1}{2\tau_i^{n+1,s}}\right)\Delta f_{i,k}^{n+1,s} + \frac{1}{2|\Omega_i|}\sum_{j\in N(i)} S_{ij}\Delta\hat{F}_{ij,k}^{n+1,s} = r_{i,k}^{n+1,s}, \tag{8}$$

with $\Delta\mathbf{W}_i^{n+1,s} = \mathbf{W}_i^{n+1,s+1} - \mathbf{W}_i^{n+1,s}$, $\Delta f_{i,k}^{n+1,s} = f_{i,k}^{n+1,s+1} - f_{i,k}^{n+1,s}$, and virtual time step $\Delta t'$ taken at CFL = 100. The residuals $\mathbf{R}_i^{n+1,s}$ and $r_{i,k}^{n+1,s}$ are the standard Crank–Nicolson residuals of [22]; in particular,

$$\hat{r}_{i,k}^{n} = -\frac{1}{|\Omega_i|}\sum_{j\in N(i)} S_{ij}\hat{F}_{ij,k}^{n} + \frac{f^{+,n}_{i,k} - f^{n}_{i,k}}{\tau_i^{n}}, \tag{9}$$

$$\hat{r}_{i,k}^{n+1,s} = -\frac{1}{|\Omega_i|}\sum_{j\in N(i)} S_{ij}\hat{F}_{ij,k}^{n+1,s} + \frac{f^{+,n+1,s}_{i,k} - f^{n+1,s}_{i,k}}{\tau_i^{n+1,s}}, \tag{10}$$

enter the microscopic right-hand side through $r_{i,k}^{n+1,s} = -(f_{i,k}^{n+1,s} - f_{i,k}^{n})/\Delta t + (\hat{r}_{i,k}^{n} + \hat{r}_{i,k}^{n+1,s})/2$. Left-hand flux variations are approximated by an Euler Riemann solver for $\Delta\hat{\mathbf{F}}$ and a first-order upwind scheme for $\Delta\hat{F}$, and the resulting linear systems are solved by LU-SGS [36]. Each physical step proceeds as in [22]: evaluate time-averaged fluxes and residuals at t^n and at the current dual-time state; update $\mathbf{W}$ then f by LU-SGS; correct $\mathbf{W}$ by the moment of $f - g$; and repeat the inner loop until the dual-time residual is driven down. The overset coupling in Section 3 inserts fringe interpolation into this residual / LU-SGS communication path, without changing the underlying dual-time formulation.

3. The algorithm of UGKS with overlapping mesh

In the current framework, the UGKS is extended to overlapping meshes through a systematic procedure involving overset mesh assembly and inter-grid data communication. The assembly process is facilitated by a grid zone decomposition and an efficient global-to-local parallel search strategy. Subsequently, the data exchange between overlapping zones is executed using a gradient-corrected distance-based weighted average interpolation method to ensure numerical consistency.

Compared with conventional CFD overset methods, which typically transfer a small set of macroscopic conservatives or primitives, the kinetic overset problem must also supply the full discrete distribution $f(\mathbf{v})$ of the whole velocity space. The main difficulties are therefore: (i) robust hole cutting and donor selection under large relative motion; (ii) consistent interpolation of both $\mathbf{W}$ and f without breaking $\mathbf{W} = \sum_k w_k \Psi_k f_k$; (iii) embedding this high-dimensional transfer in the dual-time residual / LU-SGS communication without rebuilding the fringe geometry at every inner iteration. The remedies adopted here are summarized below and detailed in the following subsections: wall-distance classification with buffer layers; ADT with neighbor-to-neighbor donor search (preferring comparable local mesh scales between receptors and donors); identical geometric weights for $\mathbf{W}$ and f with gradient-corrected IDW; and connectivity reassembled once per physical step but held fixed during dual-time iterations.

The whole algorithm is shown as follows:

Step 1 According to the Newton-Euler equation or given motion function, calculate the motion of the moving domain.
Step 2 Assemble the overset mesh.
Step 3 Evolve the solution according to the algorithm of unsteady UGKS introduced in Section 2.2.
Step 4 Go to step 1 for the next time step.

In the following, the details of the overset mesh assembly of Step 2 and interpolation in Step 3 are introduced.

3.1. Automated mesh assembly and classification

In the overset mesh, multiple zones are used to present the different moving forms. Compared with the arbitrary Lagrangian-Eulerian (ALE) mesh, the overset mesh does not require mesh deformation, which is more complicated and may lead to negative

Table 1
Classification and functional roles of cells in the overset mesh system.

Cell Type	Definition Criteria	Physical Evolution	Data Source
Active	All nodes are active	Full UGKS evolution	Local computation
Inactive	All nodes are inactive	None (Hole)	N/A
Boundary	Mixed active/inactive nodes	Full UGKS evolution	Local computation
Buffer	Inactive cells near boundary	N/A	Interpolated

volume and mesh quality issues. The overset mesh is assembled by the grid zone division and through interpolation between the zones to communicate the information between the zones.

In this paper, the unstructured mesh is used to assemble the overset mesh. The key step in overset mesh assembly is the classification of points and cells, determining which cells participate in calculation and interpolation. According to [37], the wall distance, defined as the minimum distance from a point to the wall, is used as the classification criterion. Points are classified into two types: active and inactive. For each point, if the wall distance in the current zone is less than the wall distances in other zones, the point is classified as active; otherwise, it is inactive. Cells are divided into three types: active, inactive, and boundary cells. A cell is active if all its nodes are active, inactive if all its nodes are inactive, and otherwise, it is a boundary cell. In addition, to achieve second-order accuracy, two buffer layers are defined as the two layers of cells within the inactive cell region that are adjacent to the boundary cells. These buffer layers serve as the transitional region for interpolation and calculation near the boundary cells. The classification and functional roles of cells in the overset mesh system are shown in Table 1. It is worth noting that boundary cells are not chosen as interpolation targets. This is to avoid the scenario where a boundary cell's donor cells in other domains are also classified as boundary cells, which could lead to numerical instability or cyclic dependency during data exchange.

3.2. Parallel searching strategy based on ADT and neighbor-to-neighbor search

The most critical and challenging part of the overset mesh assembly is the donor cell search process, which is involved in both the point classification and the final interpolation steps. The goal of this process is to determine which cell contains a given point within overlapping mesh zones. In this paper, the Alternating Digital Tree (ADT) algorithm [38] is adopted to efficiently locate candidate donor cells, which are then supplemented by a neighbor-to-neighbor search [39] for precise identification. In practice, receptor cells and their donor stencils are recommended to have comparable local mesh length scales in the overlap region, so that geometric mismatch at the fringe remains limited.

The ADT algorithm works by recursively dividing the entire computational domain along alternating spatial coordinates (e.g., x, y, z directions), building a hierarchical tree structure in which each non-leaf node represents a bounding box that encompasses its children. Each split partitions the domain along one coordinate direction, and the splitting direction alternates at each level of the tree, hence the name. In this way, the ADT organizes all cell bounding boxes into a binary tree, enabling fast exclusion of large regions unrelated to the point of interest. As a result, the search complexity for finding possible donor cells is reduced from $O(N)$ to $O(\log N)$, where N is the total number of cells. This greatly improves the efficiency of donor cell searching in large-scale unstructured meshes.

The neighbor-to-neighbor search algorithm starts from the center of the initial cell, which is selected by the ADT algorithm as a rough candidate. A ray is drawn from the center of this initial cell to the target point. By determining which face the ray first intersects, the search proceeds into the neighboring cell sharing that face. This process is repeated—casting a ray from the center of the current cell to the target point and entering the next neighboring cell via the intersected face—until the cell containing the target point is reached. However, the neighbor-to-neighbor search algorithm may fail if the ray passes through the interior of a moving object. As shown in Fig. 1, for the same target point A, starting from point B allows the algorithm to work, but starting from point C leads to failure. In this situation, it is necessary to determine a new face through which the ray can re-enter the computational domain. All boundary faces of the domain are collected, and all faces that the ray intersects are recorded. If the total number of intersected faces is even, it indicates the target point is outside the computational domain and should be marked as an inactive point. If the number is odd, it proves that the target point is located inside the physical object, and the face closest to the target point among the intersected faces should be selected as the new entry face for the ray to continue the search.

In this study, the search algorithm is parallelized using the Message Passing Interface (MPI). The physical computational mesh is partitioned with the METIS library to ensure balanced workload distribution across processors, regardless of domain boundaries. A global-to-local parallel search strategy, as described in [40], is employed. For a given target point, different processors may return different search outcomes. For example, the point could be located inside a cell (see Fig. 2a), cross the far-field or wall boundary (Fig. 2b), or enter another processor's subdomain through a process interface (Fig. 2c) or via a wall boundary (Fig. 2d). As illustrated in the figures, points found in a cell can be further classified as active or inactive based on their minimum wall distance. Distinct numerical markers are assigned to each case: active points are marked as −1, inactive as 0, points that exit the computational boundary as 1 (Fig. 2a), points entering other processors through a wall as 2 (Fig. 2c), and those entering through a process interface as 3 (Fig. 2d). The final classification of each point is determined by gathering and reducing the markers from all involved processors, typically by selecting the minimum value. The search for donor cells used in transplantation is handled in a similar parallel fashion.

The whole process of the overset mesh assembly is shown as follows:

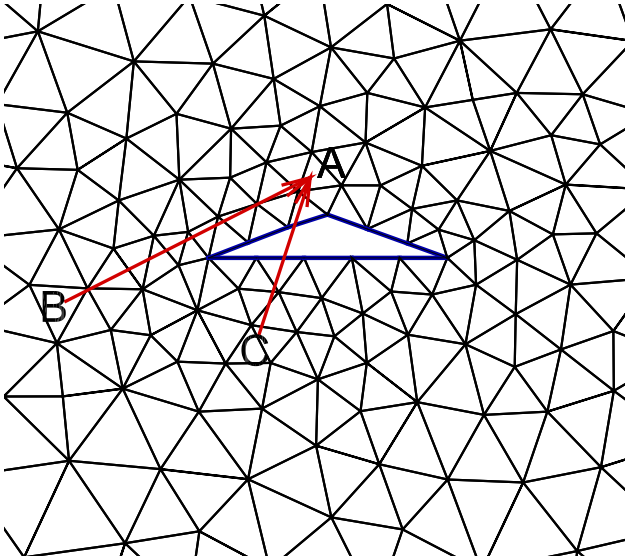


Fig. 1. The neighbor-to-neighbor search algorithm.

- The preparatory work for each grid zone, such as building the alternating digital tree (ADT) data structure, calculating the minimum distance to the wall, and constructing some necessary geometric and topology information.
- For each node, find its donor cells in other grid zones. Classify nodes as active or inactive based on the minimum distance to the wall. Active nodes are those within the minimum distance to the wall, or the distance is less than the minimum distance to the wall in the moving domain. The parallel search is implemented using the global-to-local parallel searching strategy.
- Cells are categorized into three types: active, inactive, and boundary cells. A cell is active if all its nodes are active, inactive if all its nodes are inactive, and otherwise, it is a boundary cell. To achieve second-order accuracy, a buffer layer consisting of neighboring cells of boundary and inactive cells is added to the boundary cells.
- Donor cells are found for each boundary cell and buffer layer cell in the overlap region.

The overset mesh assembly result is shown in Fig. 3. The different colors represent the different types of cells. Two layers of buffer layer cells are added to the boundary cells to achieve second-order accuracy.

3.3. Interpolation

To obtain the cell-averaged value of the interpolation cell, a distance-weighted averaging method is first introduced. Specifically, for a given interpolation cell i, the donor cell is located in another domain based on the centroid of cell i. Then, the face-neighboring cells of the donor cell are collected, forming the set $N(i)$. Only active cells are included in $N(i)$. The plain inverse-distance-weighted (IDW) interpolation reads

$$\phi_i^{\mathrm{IDW}} = \frac{\sum_{j\in N(i)} w_{ij}\,\phi_j}{\sum_{j\in N(i)} w_{ij}}, \quad w_{ij} = d_{ij}^{-2},$$

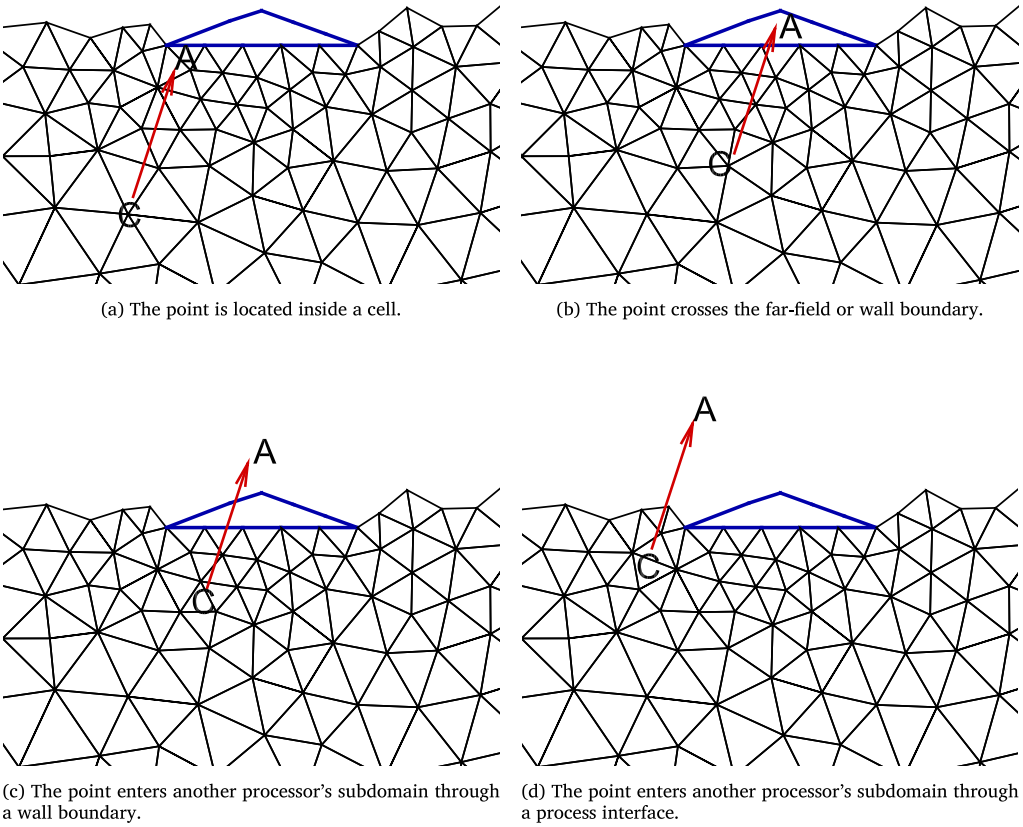
where ϕ_j denotes the variable (macroscopic variable or distribution function) on the active cell j, d_{ij} is the distance between the centroids of cell i and cell j, and w_{ij} is the distance-based weight. By including only active cells in the set, invalid cells are excluded from the interpolation.

The plain IDW formula is formally first-order accurate and may degrade the overall second-order accuracy of UGKS near overset interfaces. To recover a second-order transfer, a gradient-corrected IDW is adopted. A least-squares gradient $\nabla\phi$ is constructed about the donor cell using the same stencil $N(i)$, and the interpolated value is evaluated as

$$\phi_i = \frac{\sum_{j\in N(i)} w_{ij}\left[\phi_j + \nabla\phi\cdot\left(\mathbf{x}_i - \mathbf{x}_j\right)\right]}{\sum_{j\in N(i)} w_{ij}},$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the centroids of the receptor and donor-stencil cells, respectively. For a linear field this correction is exact; if the least-squares system is singular, the method falls back to plain IDW. Unless otherwise stated, the numerical examples in Section 4 employ the gradient-corrected interpolation. An exception is the three-dimensional TSTO case in Section 4.5, where plain IDW is retained for numerical stability under strong hypersonic discontinuities.

In UGKS, the macroscopic conservative variables and the microscopic distribution function must remain consistent, namely $\mathbf{W} = \sum_k w_k \Psi_k f_k$. At the overset interface, both $\mathbf{W}$ and f are interpolated with the same geometric weights $\{w_{ij}\}$ (and the same gradient



(a) The point is located inside a cell.

(b) The point crosses the far-field or wall boundary.

(c) The point enters another processor's subdomain through a wall boundary.

(d) The point enters another processor's subdomain through a process interface.

Fig. 2. The search results of the different processors. The C is the beginning point of the ray, A is the target point.

correction). Denoting the interpolation operator by L, one has

$$L(\mathbf{W}) = L\left(\sum_k w_k \Psi_k f_k\right) = \sum_k w_k \Psi_k L(f_k),$$

so the interpolated macroscopic variables remain the moments of the interpolated distribution function. In practice, $\mathbf{W}$ is not recomputed from the interpolated f; instead, $\mathbf{W}$ and f are interpolated independently with identical weights. This strategy preserves the microscopic–macroscopic consistency and avoids non-physical decoupling of $\mathbf{W}$ and f across the overset interface.

Regarding positivity, plain IDW is a convex combination and thus preserves $f \geq 0$ whenever all donor values are non-negative, whereas the gradient correction does not rigorously guarantee positivity. Any non-exact interface transfer also introduces additional local truncation error near the fringe.

3.4. Integration with implicit dual time-stepping solver

The dual-time flux evaluation of Section 2.2 is implemented in a memory-efficient manner [34]. Together with the overset interpolation, the whole algorithm is shown as Algorithm 1, which shows that the interpolation is needed in the same way as MPI communication.

In the dual-time algorithm, the coupling with the overset assembly is organized as follows. Within each physical time step, the mesh is first advanced by the physical time increment Δt, after which the overset topology is reassembled once: wall-distance classification, *i*Blank markers, donor search and interpolation maps are updated at the new geometry t^{n+1}. The dual-time (inner) iterations then solve the unsteady residual on this frozen geometry. Consequently, the overset connectivity—including donor cells and geometric weights—is held fixed during all inner iterations of that physical step; it is rebuilt only after the next mesh motion. During every residual evaluation and LU-SGS sweep of an inner iteration, however, the field values (and, when required, the residual fields) on fringe/receptor cells are refreshed by interpolating from the active donor stencil, in the same manner as an MPI halo exchange. Inactive cells are excluded from residual accumulation and from the LU-SGS updates. With the connectivity frozen at t^{n+1}, the overset interface does not introduce an additional geometric CFL restriction beyond that of the dual-time/LU-SGS scheme on active cells. In

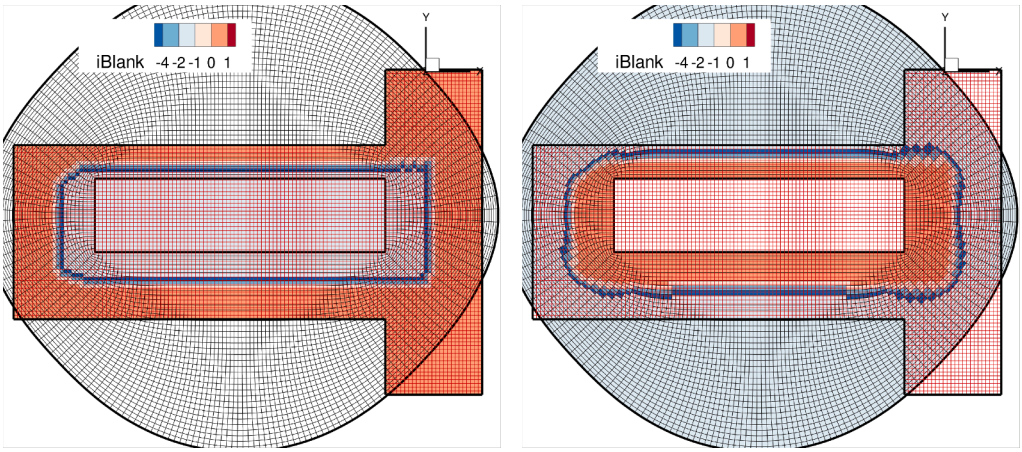


Fig. 3. The assembly of the overset mesh. 1 denotes the active cells, 0 denotes the inactive cells, −1 denotes the boundary cells, and −2 and −4 denotes the buffer layer cells.

practice, we recommend that receptor cells and their donor stencils have comparable local mesh length scales in the overlap, so that the fringe truncation error remains controlled; this matched-resolution setting is used in the numerical examples of Section 4.

Algorithm 1 Parallel algorithm for the UGKS flux calculation.

Calculate the time step Δt_s based on the CFL condition and compute heat flux for each cell.
Send macroscopic values, heat flux, and microscopic distribution function f_1.
Receive macroscopic values, heat flux, and microscopic distribution function f_1.
Interpolate macroscopic variables, heat flux and distribution functions f_1 into the overlapping mesh.
Reconstruct macroscopic values.
Reconstruct microscopic values for f_1 and store their gradients as ∇f_1.
Send gradients of macroscopic values, microscopic distribution function f_2, and gradients of microscopic values ∇f_1.
for $k = 1$ to $k_{\max}$ **do**
  Let $\&\nabla f_f \leftarrow (k \mod 2 == 1)\,?\,\nabla f_1 : \nabla f_2$ and $\&\nabla f_r \leftarrow (k \mod 2 == 1)\,?\,\nabla f_2 : \nabla f_1$.
  if $k == 1$ **then**
    Receive gradients of macroscopic values, distribution function f_2, and gradients of microscopic ∇f_f.
    Interpolate distribution functions f_2 into the overlapping mesh.
    Calculate equilibrium state and time coefficients for cell interfaces.
  else
    Receive distribution function f_{k+1} and gradients of microscopic ∇f_f.
    Interpolate distribution function f_{k+1} into the overlapping mesh.
  end if
  Reconstruct microscopic values for f_{k+1} and store their gradients as ∇f_r.
  Send microscopic distribution function f_{k+2} and gradients of microscopic values ∇f_r.
  Evolve flux $\mathcal{F}_k$ with f_k and ∇f_f using Eq. (3), and their contribution to macroscopic flux using Eq. (5).
  Interpolate f_k and their gradients into the boundary and calculate the incident part of the flux.
end for
Calculate the reflection part of the flux at the boundary.

Compared with the original algorithm in [34], several differences are introduced:

- The residuals have to be stored due to the implicit scheme.
- About four times more memory than the original algorithm is needed to store the gas distribution function and its residuals because the dual time-stepping scheme is used.
- The wall boundary flux calculation was split into the incident part and the reflection part so that the gas distribution function and its gradients do not need to be stored in the boundary cells. Only two scalers, incident flux and reflection flux of density, are needed to be stored in the boundary cells.

4. Numerical results

4.1. Accuracy test: Taylor–Green vortex on an overset mesh

To quantify the spatial accuracy of the overset communication, the two-dimensional Taylor–Green vortex with an analytical solution is considered. The periodic domain is $[0,1]^2$, and the exact velocity and pressure fields are

$$u(x,y,t) = -\frac{u_0}{A}\cos(Ax)\sin(By)\,e^{-\nu(A^2+B^2)t},$$
$$v(x,y,t) = \frac{u_0}{B}\sin(Ax)\cos(By)\,e^{-\nu(A^2+B^2)t},$$
$$p(x,y,t) = \rho_0 RT - \frac{\rho_0 u_0^2}{4}\left(\frac{\cos(2Ax)}{A^2}+\frac{\cos(2By)}{B^2}\right)e^{-2\nu(A^2+B^2)t},$$

with $A = B = 2\pi$, $u_0 = 1$, $\rho_0 = 1$, $\nu = \mu_{\mathrm{ref}}/\rho_0 = 10^{-2}$ and $RT = 100$. The relative L_2 error of the x-velocity on active cells is defined as

$$E(u) = \frac{\left\|\mathbf{u} - \mathbf{u}^{\mathrm{exact}}\right\|_2}{\left\|\mathbf{u}^{\mathrm{exact}}\right\|_2}.$$

Three resolutions $N = 16$, 32, and 64 are examined. For each N, a single-grid $N \times N$ computation on the periodic unit square provides the baseline. The overset configuration consists of a background $N \times N$ Cartesian mesh covering $[0,1]^2$ and a near-body $N \times N$ patch of the same spacing, centered at $(0.5, 0.5)$ with side length 1 and rotated by 45°. Periodic conditions are imposed on the background outer boundaries. The explicit second-order UGKS with MUSCL reconstruction is used, the CFL number is 0.5, and the solution is advanced to $t = 1$. Two overset interpolations are compared: plain inverse-distance weighting (IDW) and the gradient-corrected IDW of Section 3.3. Errors are evaluated only on exclusive active cells ($i\mathrm{Blank} = 1$).

Table 2 summarizes $E(u)$ for the single-grid, overset plain-IDW and overset gradient-corrected cases. On the single-grid meshes, $E(u)$ decreases from 0.694 to 0.025 with observed orders of about 2.1 ($16 \to 32$) and 2.7 ($32 \to 64$), consistent with the second-order spatial reconstruction. For the overset mesh, plain IDW gives $E(u) = 0.754$, 0.216 and 0.076 at $N = 16$, 32 and 64, with observed orders of about 1.8 and 1.5, indicating a clear degradation relative to the single-grid baseline at fine resolution. With the gradient correction, the corresponding errors become 0.719, 0.221 and 0.063, and the observed orders are about 1.7 and 1.8. Although the absolute overset error remains larger than the single-grid value by a factor of about 1.04, 1.38 and 2.5 at $N = 16$, 32 and 64, respectively, the gradient-corrected scheme reduces the $N = 64$ error by about 17% relative to plain IDW. These results confirm that gradient-corrected IDW significantly mitigates the fringe interpolation error of plain IDW and preserves a convergence rate close to the design order of the underlying UGKS scheme. Fig. 4 further illustrates the $N = 64$ gradient-corrected solution: the x-velocity field at $t = 1$ (Fig. 4a) and the u/u_0 profile along $x = 0.5$ at $t = 0.1$, 0.5 and 1.0 compared with the analytical solution (Fig. 4b).

4.2. Forced motion of a micro beam in confined geometries

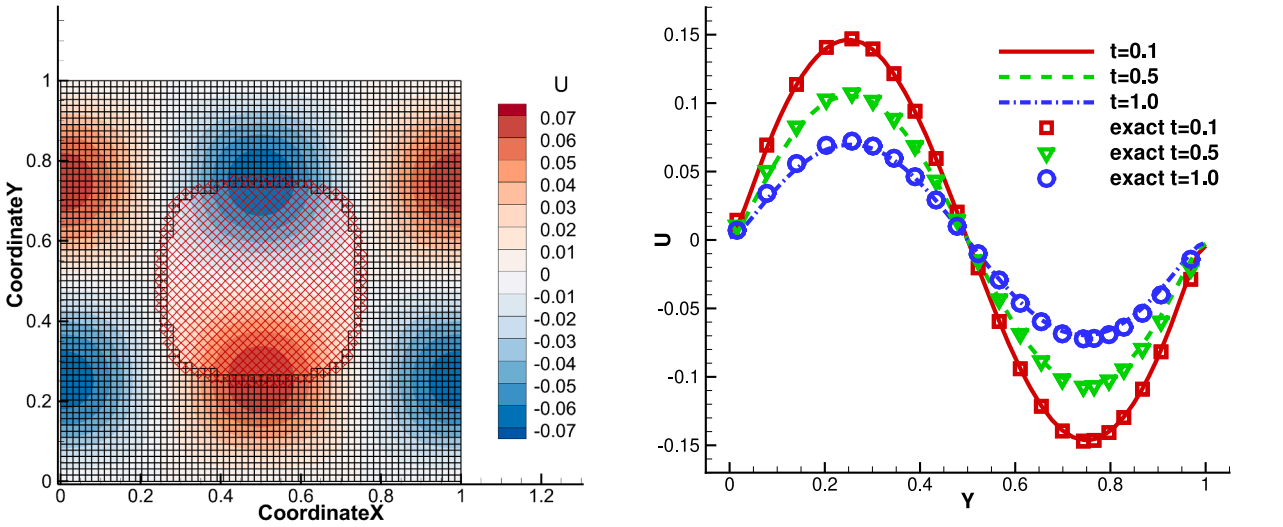
In this numerical example, we investigate the rarefied gas flow within an open cavity containing a micro-beam structure. This configuration simulates the damping effects observed in MEMS comb drive mechanisms, particularly between the shuttle and stator components [41]. The detailed configuration and dimensional specifications are presented in Fig. 5.

The shuttle (micro beam) is positioned centrally in the left cavity at the initial state. The oscillatory motion of the shuttle, governed by $v_c = \cos(2\pi f_0 t)$, induces fluid exchange through the upper and lower channels. Following the parameters established by Tiwari et al. [42], the oscillation operates at $f_0 = 176{,}000$Hz. The initial conditions specify a cavity pressure of $p_0 = 0.1$bar $= 10{,}000$Pa, fluid temperature of $T_0 = 293$K, and density of $\rho_0 = 0.1641\mathrm{kg/m^3}$. These parameters yield a Knudsen number of $\mathrm{Kn} = 0.2583$, referenced to the 4.2μm gap width. The wall maintains a constant temperature of $T_w = 293$K. The upper and lower channels implement free-flow conditions matching the initial state. The velocity space discretization spans $[-5\sqrt{RT_0}, 5\sqrt{RT_0}]^2$ with a uniform 49×49 grid. The time step is set to 1×10^{-8}s and 10 inner steps are used for the dual time-stepping scheme.

The computational results at $t = 1.2 \times 10^{-6}$s are displayed in Fig. 6. Fig. 6a shows the computational mesh structure, where red elements correspond to the micro beam region and black elements represent the background fluid domain. The interface between these regions exhibits smooth transitions. Fig. 6b reveals the x-velocity distribution, showing opposing flow patterns at the corners: positive velocities at top-right and lower-left, contrasting with negative velocities at top-left and lower-right. The upward shuttle motion compresses the upper cavity region, resulting in higher velocity magnitudes. This compression effect is further evidenced by

Table 2
Relative L_2 error $E(u)$ of the Taylor–Green vortex at $t = 1$ ($\nu = 10^{-2}$, active cells only). Observed orders are computed between successive grids.

N	Single grid	Order	Overset IDW	Order	Overset grad-IDW	Order
16	6.94×10^{-1}	–	7.54×10^{-1}	–	7.19×10^{-1}	–
32	1.60×10^{-1}	2.12	2.16×10^{-1}	1.80	2.21×10^{-1}	1.70
64	2.52×10^{-2}	2.67	7.61×10^{-2}	1.51	6.32×10^{-2}	1.81



(a) x-velocity at $t = 1$.

(b) u/u_0 versus y at $x = 0.5$.

Fig. 4. Taylor–Green vortex on the $N = 64$ overset mesh with gradient-corrected IDW: (a) x-velocity field at $t = 1$; (b) velocity profiles at $x = 0.5$ for $t = 0.1$, 0.5 and 1.0 (symbols) compared with the analytical solution (dashed lines).

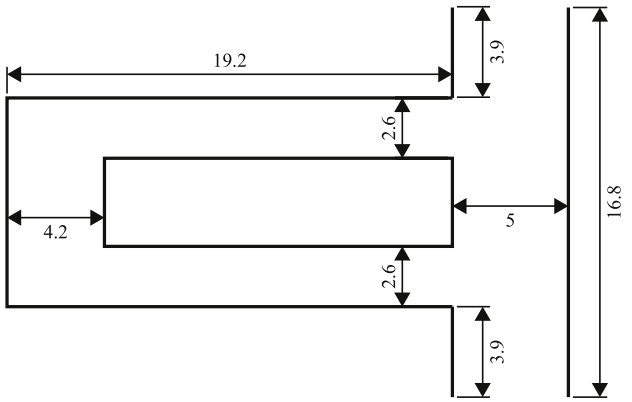


Fig. 5. Schematic representation of the micro beam oscillation setup in confined space (unit: μm).

the increased pressure and Mach number distributions shown in Fig. 6f and e. The velocity vector field in Fig. 6b highlights enhanced flow speeds near the top corners. The y-velocity contours demonstrate upward fluid motion along the shuttle's vertical walls, while downward flow occurs along the lateral sides. These flow characteristics align with the findings reported by Tiwari et al. [42].

4.3. Free motion of a micro particle in a lid-driven cavity

The investigation of particle-laden rarefied gas flows represents a significant research area, with the particle motion in a lid-driven cavity serving as a fundamental benchmark problem [31,33,42]. The computational setup, illustrated in Fig. 7a, consists of a square cavity with side length L containing a spherical particle of diameter $d = 0.14L$. The particle, having a density ratio of $\rho_c/\rho_f = 10$ relative to the surrounding gas, is initially positioned at the cavity's center. The flow is driven by the top wall moving horizontally at a constant speed of $U_0 = 30$m/s, while the particle is allowed to move freely under the influence of fluid forces. The simulation parameters are configured with a fluid density of $\rho_f = 1.103947 \times 10^{-6}$kg/m^3, uniform temperature $T_w = T_f = 273$K, and cavity length $L = 1$m, yielding a Knudsen number of 0.1. The computational domain employs a velocity-space discretization with 41×41 points spanning $[-5\sqrt{RT_f}, 5\sqrt{RT_f}]^2$. The spatial domain utilizes 108×108 elements for the background mesh, with 88 uniform elements along the particle circumference. A refined mesh spacing of 0.005 is implemented at the cavity boundaries and particle surface. The baseline dual time-stepping setting uses a physical time step $\Delta t = 2 \times 10^{-4}$s with $N_{\text{sub}} = 10$ inner iterations. To examine the temporal robustness of this choice, companion runs are performed with a refined time step $\Delta t = 1 \times 10^{-4}$s ($N_{\text{sub}} = 10$) and with an increased dual-time iteration count $N_{\text{sub}} = 20$ ($\Delta t = 2 \times 10^{-4}$s). The testing platform consists of an AMD Ryzen Threadripper 2990WX CPU

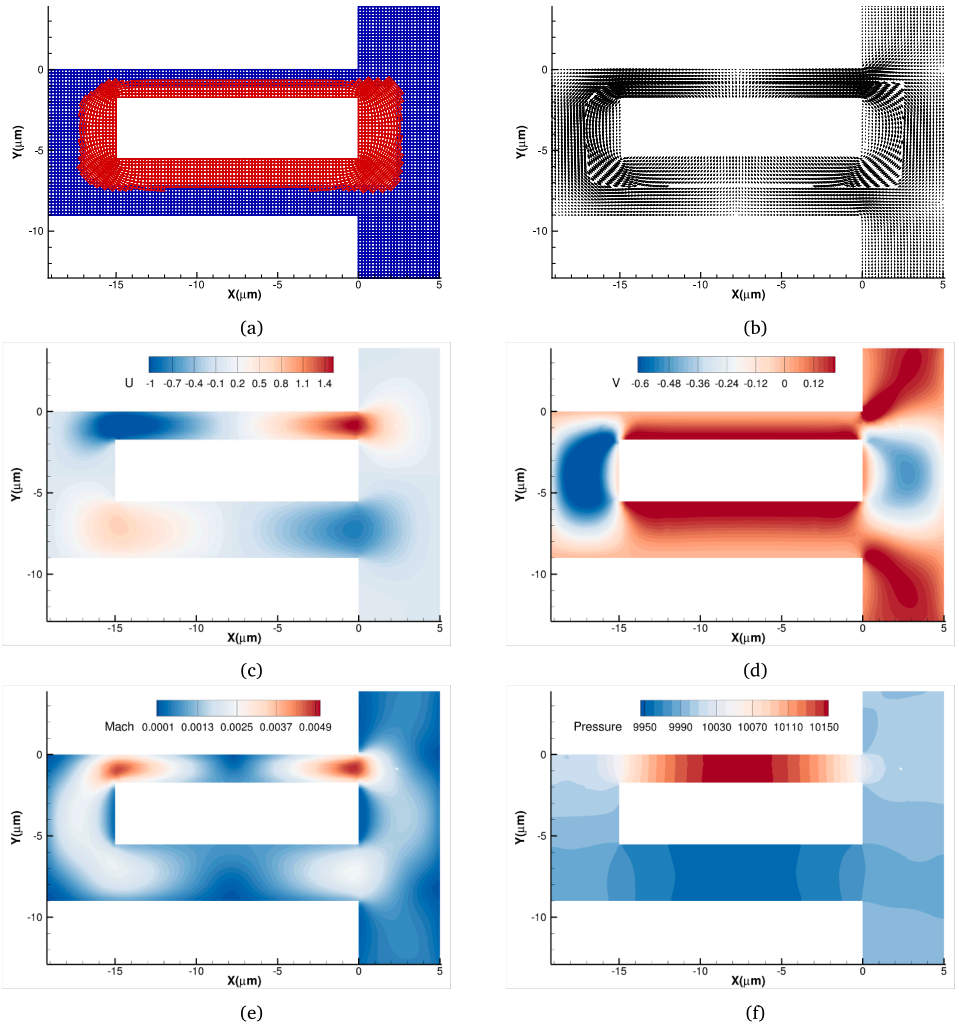


Fig. 6. Mesh distribution, velocity vector and contours at $t = 1.2 \times 10^{-6}$s in the forced motion of a micro beam in confined geometries (a) Mesh assembly result, (b) The velocity vector, (c) The x-direction velocity contour, (d) The y-direction velocity contour, (e) The Mach number contour, (f) The pressure contour.

operating at 4.2 GHz, featuring 32 cores and 128 GB of memory. The code is compiled with GCC 11.4.0 using the `-O2` optimization flag and linked with Open MPI 4.1.5. About 700 min are required for the entire simulation, with 2300 physical time steps, on 32 cores.

The numerical results at dimensionless time $t^* = t\sqrt{2RT_f}/L = 50$ are presented in Fig. 7b, depicting the mesh configuration and Mach number distribution as the particle approaches the cavity's upper region. The shearing action of the moving lid, together with the confinement of the stationary walls, generates a circulating flow that entrains the particle. Starting from the cavity center, the particle follows a clockwise path with a gradual outward drift rather than returning to its initial location.

Fig. 8 compares the particle-center trajectory and velocity histories obtained with the three dual-time settings against the reference data of Tiwari et al. [42]. The refined-Δt solution ($\Delta t = 1 \times 10^{-4}$s, $N_{\text{sub}} = 10$), the baseline ($\Delta t = 2 \times 10^{-4}$s, $N_{\text{sub}} = 10$) and the $N_{\text{sub}} = 20$ run nearly collapse over the plotted horizon, indicating that the particle motion is insensitive to the tested dual-time parameters. The present trajectories reproduce the overall spiral reported by Tiwari et al., with only a mild outward offset that is common to all three dual-time curves and therefore attributable to modeling or mesh differences rather than temporal under-resolution. The velocity components in Fig. 8b exhibit the same qualitative features: an early leftward acceleration of u_x, subsequent wall-induced reversals, and a rise-and-fall pattern of u_y.

Because the cavity is closed, the fluid mass on exclusive active cells (iBlank = 1) provides a check under continuous overset reassembly. For the baseline run, Fig. 9 shows that the relative mass variation remains within 2.4×10^{-3}. This variation arises mainly because the set of exclusive active cells changes as the particle moves and the overset classification is updated, rather than from a physical mass source or sink.

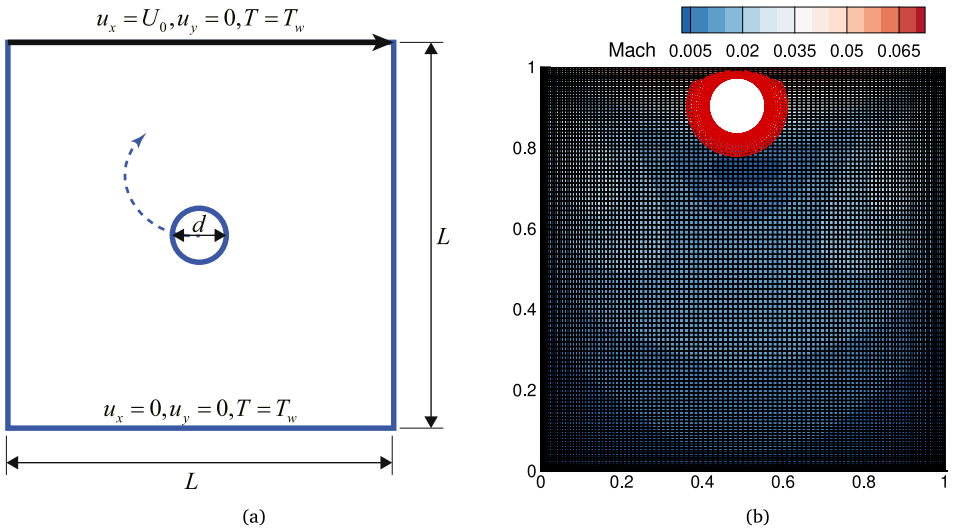


Fig. 7. The geometry, boundary conditions and mesh of the free motion of a micro particle in a lid-driven cavity (a) The geometry and boundary conditions (b) The mesh and Mach number contours at $t^* = 50$.

Table 3
Computational cost of dual-time versus explicit UGKS for the lid-driven cavity overset case on 32 cores. T_{100} and t_{100} are measured over the first 100 steps; $T(t = 1\,\mathrm{s}) = T_{100}/t_{100}$ is the extrapolated wall time to physical time $t = 1$ s.

Scheme	Δt [s]	N_{sub}	T_{100} [s]	t_{100} [s]	T_{100}/t_{100} [s/s]	$T(t = 1\,\mathrm{s})$ [h]	Rel.
Dual-time (baseline)	2.0×10^{-4}	10	801	2.00×10^{-2}	4.01×10^{4}	11.1	1.0
Dual-time	1.0×10^{-4}	10	799	1.00×10^{-2}	7.99×10^{4}	22.2	2.0
Dual-time	2.0×10^{-4}	20	1545	2.00×10^{-2}	7.73×10^{4}	21.5	1.9
Explicit (CFL = 0.5)	$\approx 1.44 \times 10^{-6}$	–	46	1.44×10^{-4}	3.20×10^{5}	88.8	8.0

To quantify the efficiency gain of dual-time integration relative to explicit UGKS, all runs use the same overset mesh, gas model and 32-core platform described above. For each scheme we record the wall-clock time T_{100} and the physical time t_{100} advanced over the first 100 time steps, and define the cost rate T_{100}/t_{100}. The wall time required to reach a fixed physical horizon $t = 1$ s is then estimated as

$$T(t = 1\,\mathrm{s}) = \frac{T_{100}}{t_{100}} \times 1\,\mathrm{s},$$

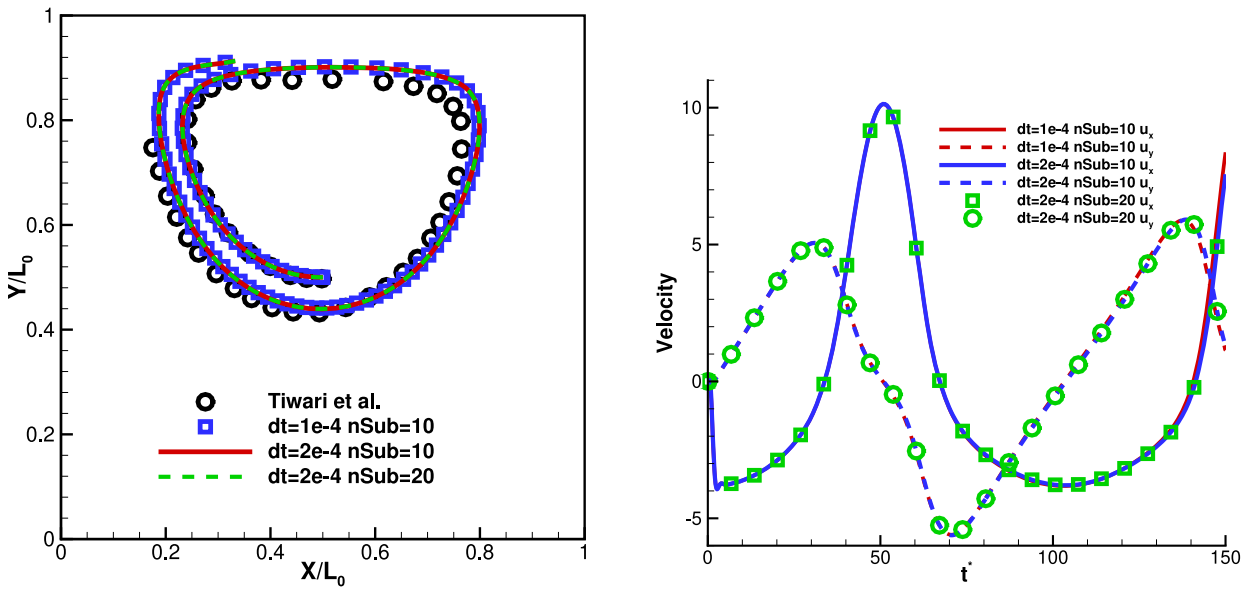
which is appropriate when the per-step cost remains approximately constant (as confirmed by longer dual-time logs up to $t = 0.4$ s). Table 3 lists Δt (or CFL), N_{sub}, the measured T_{100} and t_{100}, the cost rate, the projected $T(t = 1\,\mathrm{s})$, and the cost relative to the baseline dual-time run. The baseline ($\Delta t = 2 \times 10^{-4}$s, $N_{\mathrm{sub}} = 10$) projects to about 11.1 h, while explicit UGKS at CFL = 0.5 (mean physical step $\Delta t \approx 1.44 \times 10^{-6}$s) projects to about 88.8 h, i.e. an approximately eightfold wall-clock reduction. Halving Δt or doubling N_{sub} nearly doubles the cost, consistent with the dual-time workload, while the particle trajectories remain essentially unchanged (Fig. 8).

4.4. Flow around an oscillating circular cylinder

The movement of a circular cylinder in a fluid is a classic example of a moving boundary problem and is widely studied in offshore and civil engineering fields. In this study, we conduct simulations of a circular cylinder oscillating in a stationary fluid to assess our method's ability to handle moving boundaries. The geometry and the mesh assembly at the initial state are shown in Fig. 10. The computational domain is $[-80, 80] \times [-60, 60]$m and the cylinder with a radius of 0.5m is located at the origin in the initial state. 240 points are used to discretize the cylinder circumference, and the height of the first layer of the mesh is 4×10^{-3}m. The background mesh near the cylinder is refined to 0.04m. The total number of background meshes is 21,614, and the cylinder mesh has 12,960. Nitrogen is used as the working gas in this simulation. The background temperature and wall temperature are $T_f = T_w = 270$K, respectively. The motion of the cylinder is governed by the following equation:

$$x(t) = A \sin(2\pi f t), \quad y(t) = 0,$$

where A is the amplitude and f is the frequency. Three non-dimensional parameters, the Mach number Ma, the Reynolds number Re and the Keulegan–Carpenter number KC, are defined as $Ma = U_{\max}/c$, $Re = \rho U_{\max} d/\mu$ and $KC = U_{\max}/(fd)$, where $U_{\max}$ is the maximum velocity of the cylinder, d is the cylinder diameter and μ is the dynamic viscosity. In this test, the Mach number is



(a) (b)

Fig. 8. Particle-center trajectory and velocity history for the lid-driven cavity: (a) trajectory compared with Tiwari et al. [42] for $\Delta t = 1 \times 10^{-4}$s ($N_{\text{sub}} = 10$), the baseline $\Delta t = 2 \times 10^{-4}$s ($N_{\text{sub}} = 10$) and $N_{\text{sub}} = 20$; (b) corresponding u_x (solid) and u_y (dashed/symbols) histories with time normalized by $L/\sqrt{2RT_f}$.

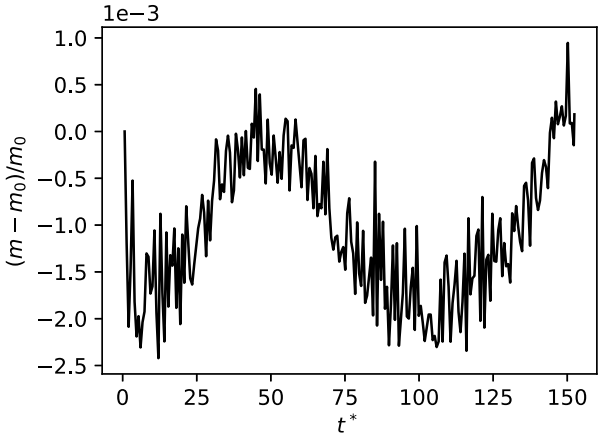


Fig. 9. Relative fluid-mass variation $(m - m_0)/m_0$ on exclusive active cells for the baseline dual-time run ($\Delta t = 2 \times 10^{-4}$s, $N_{\text{sub}} = 10$).

$Ma = 0.173$, the Reynolds number is $Re = 100$, and the Keulegan–Carpenter number is $KC = 5$. 180 time steps are used to discretize one period of the cylinder motion, and 20 inner steps are used for the dual time-stepping scheme. In each velocity direction, 32 unified elements are used to discretize the velocity space $[-5\sqrt{RT_f}, 5\sqrt{RT_f}]$.

Totally, three periods of the cylinder motion are simulated, and the last period is selected for the analysis. For the inline force D_x non-dimensionalized by $\rho U_{\max}^2 d$, the present results are compared with the results of Dütsch et al. [43] in Fig. 11. The present results are in good agreement with the results.

The vorticity contours with vector fields and the velocity profiles for phase positions 180°, 210°, and 330° are depicted in Fig. 12. From the vorticity contours, the flow state can be clearly observed at 180°, 210°, and 330°. At 180°, the cylinder reaches its maximum velocity to the right, presenting the largest positive velocity at the center and resulting in a compact wake structure. At 210° and 330°, the cylinder's instantaneous positions are the same, and the speed magnitudes are equal but with opposite directions: at 210°, the cylinder is moving leftward, while at 330°, it is moving rightward. Notably, for the leftward motion at 210°, the cylinder undergoes a process of acceleration followed by deceleration, leading to the formation of longer and more pronounced vortical structures in the

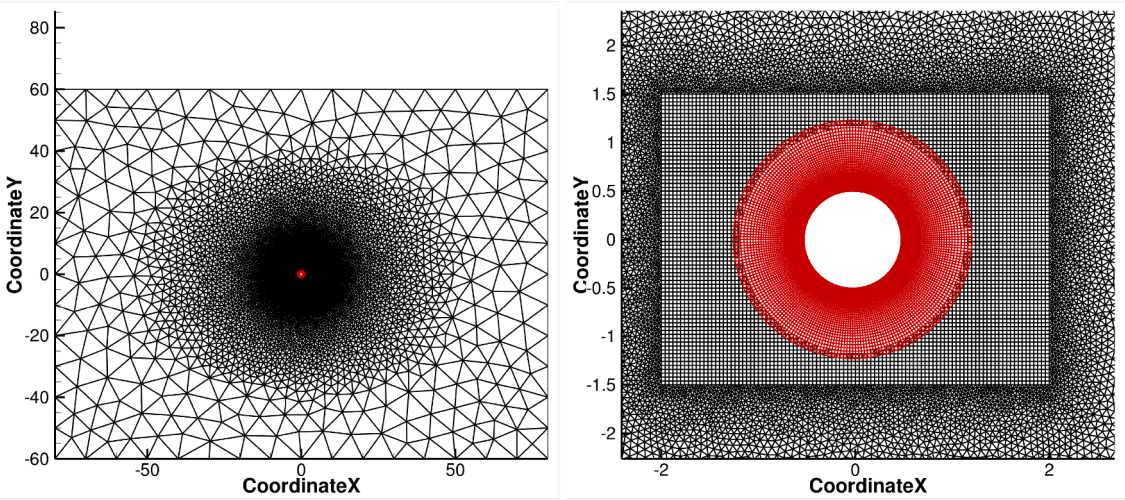


Fig. 10. The geometry and mesh assembly at the initial state of the flow around an oscillating circular cylinder with the Reynolds number $Re = 100$ and the Keulegan–Carpenter number $KC = 5$.

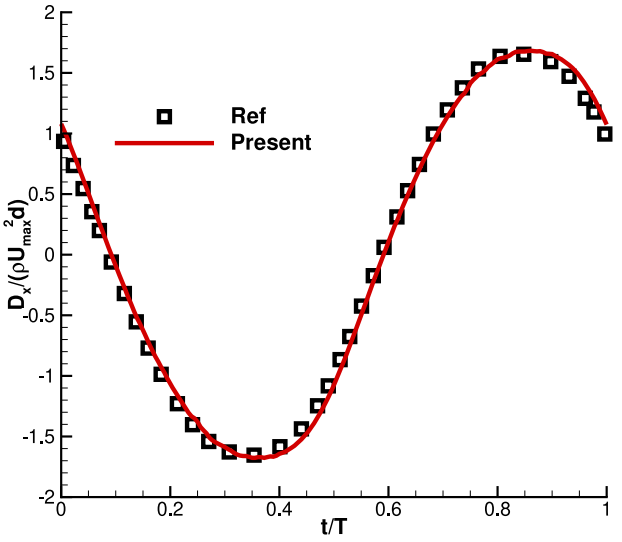


Fig. 11. Comparison of inline force D_x of the oscillating cylinder with the result of Dütsch et al. [43].

wake compared to the rightward case at 330°. The velocity profiles normalized by $U_{\max}$ at four vertical cross sections $x = -0.6, 0, 0.6, 1.2$ are also shown. Compared with the results of Dütsch et al. [43], the present results are in good agreement.

4.5. Two-stage-to-orbit (TSTO) hypersonic vehicles

This model is used to test the three-dimensional mesh motion and the overset mesh assembly. For numerical stability in this hypersonic multi-body separation, overset data transfer uses plain IDW rather than the gradient-corrected scheme of Section 3.3. The two-stage-to-orbit (TSTO) hypersonic vehicles are described in [44]. The geometry is shown in Fig. 13a. The transverse stage separation (TSS) scheme, in which the orbiter moves along the direction normal to the booster's upper surface, is studied in this paper. The initial angle of incidence is set as $\alpha = 8°$. The incoming Mach number is 8, and the Knudsen number is 0.1 according to the reference length $l_o = 0.162$m. Argon is used as the working gas. The incoming flow is uniform with temperature $T_\infty = 273$K, density $\rho_\infty = 5.3002 \times 10^{-6}$kg/m^3 and velocity $U_\infty = 2461.917$m/s. The mass of the orbiter is $m_o/\rho_\infty \cdot l_o^3 = 10$, and the moment of inertia are $I_{xx}/\rho_\infty \cdot l_o^5 = 6$, $I_{yy}/\rho_\infty \cdot l_o^5 = 12.5$, and $I_{zz}/\rho_\infty \cdot l_o^5 = 10$, respectively. The Newton-Euler equation is used to solve the motion of the orbiter. The time step is set to 5×10^{-6}s and 15 inner steps are used for the dual time-stepping scheme. At the initial state, the mesh is shown in Fig. 13b. Totally, $1,258,788$ elements are used for the background mesh, and $264,570$ for the orbit mesh. The height of the first layer of the mesh is 5×10^{-4}m. The DVS is discretized into $4,748$ cells in a sphere mesh with center at $(0.4U_\infty, 0, 0)$ a radius of $8\sqrt{\gamma RT_\infty}$. The velocity space near the zero and free stream velocity points is refined within a radius of $\sqrt{RT_\infty}$. The simulation was

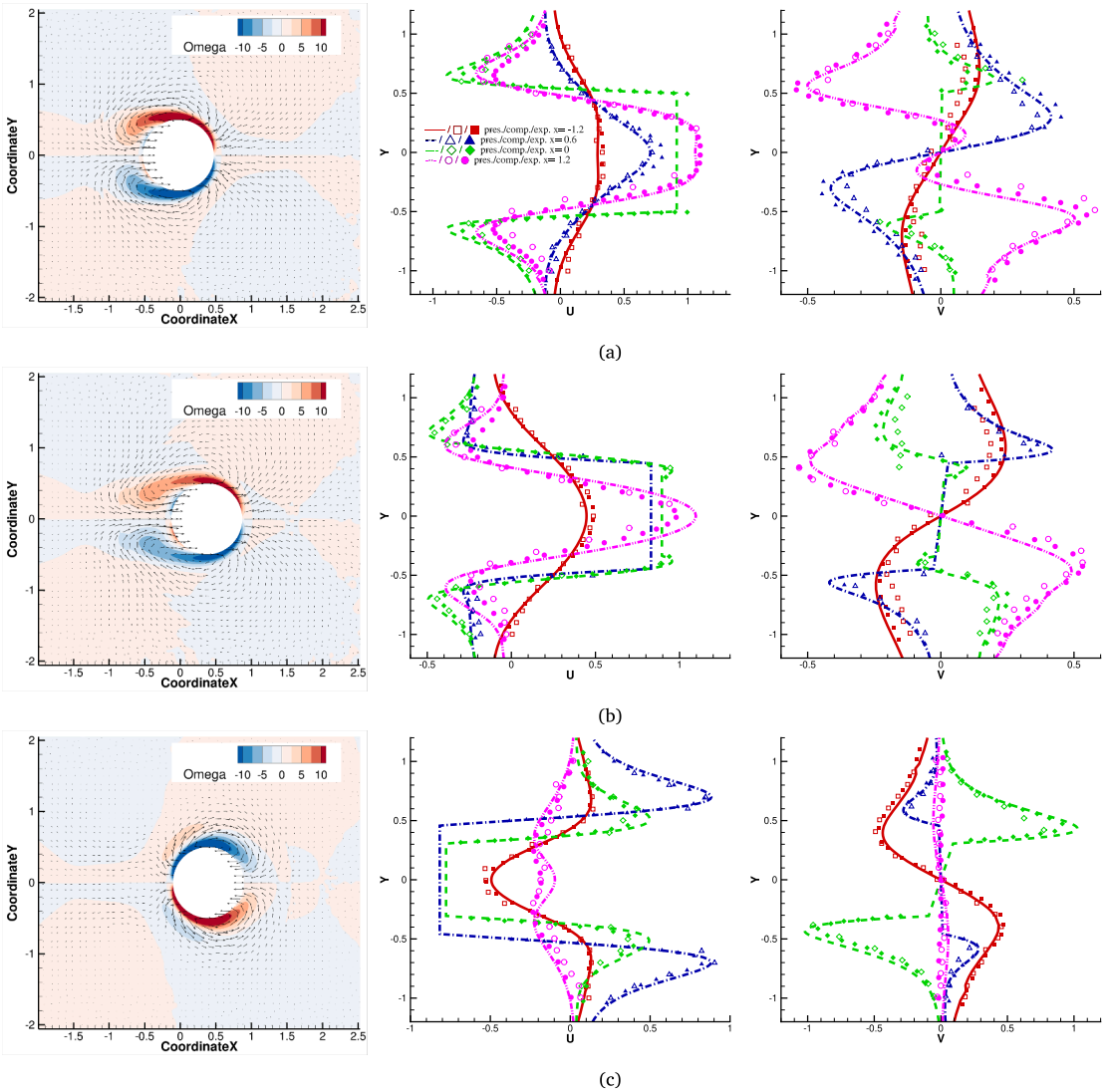
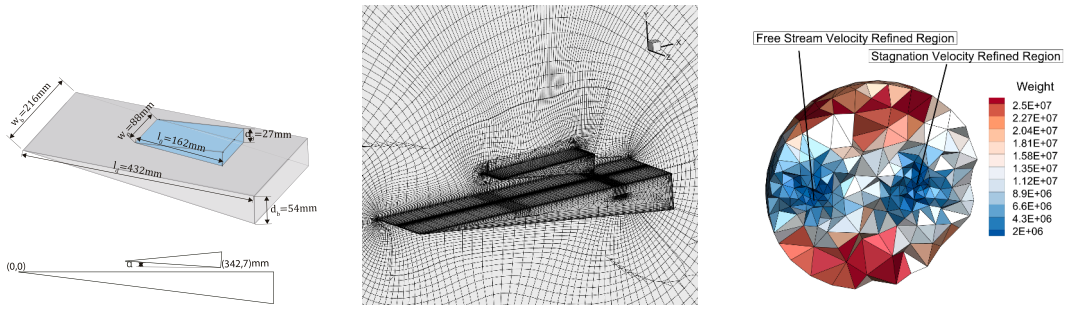


Fig. 12. The vorticity contour with the vector field and the velocity profile at four vertical cross sections $x = -0.6, 0, 0.6, 1.2$ of the flow around an oscillating circular cylinder for phase positions (a) 180°, (b) 210°, and (c) 330°. The lines are the present results and the symbols are the computational/experimental results of Dütsch et al. [43].

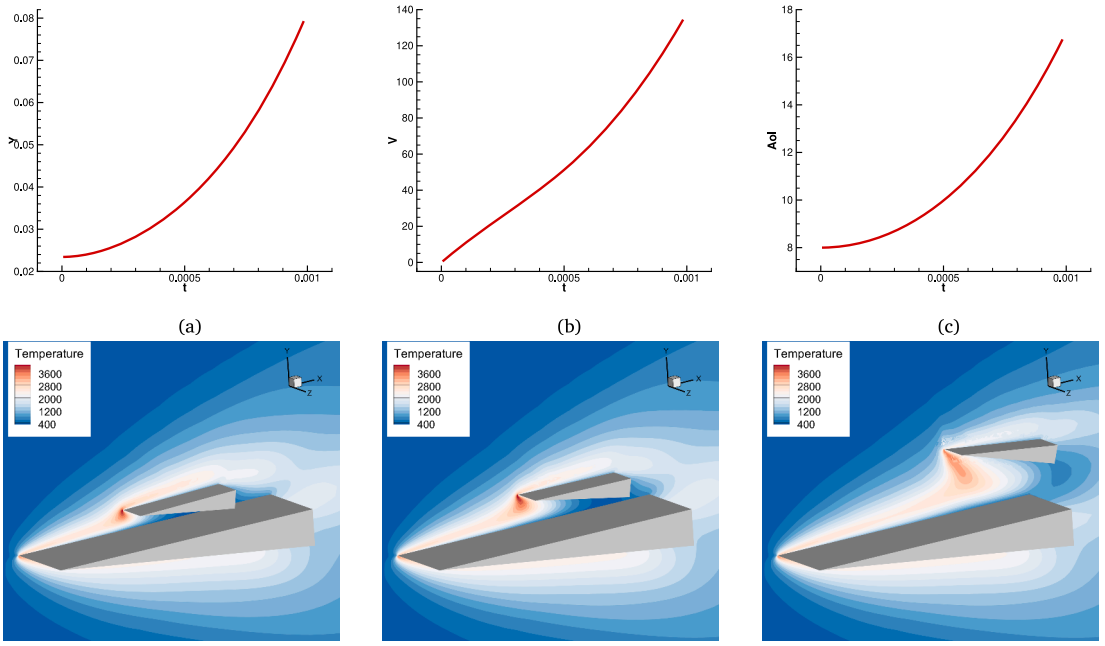
conducted on the SUGON computation platform using dual Hygon 7285H processors, each equipped with 32 cores, for a total of 64 cores at a base frequency of 2.5 GHz. The system architecture leverages the C86-2G microarchitecture, and the computation network is supported by a 100Gb InfiniBand (IB) connection. 256 cores are used for this computation, and about 4615 s are required for one physical time step.

The computational results are shown in Fig. 14. The evolution of the orbiter's center of mass position in the y-direction, as shown in Fig. 14a, exhibits a quadratic trend over time. As observed from the history of the y-direction velocity in Fig. 14b, the acceleration first decreases and then increases. This is attributed to the initial reduction caused by the orbiter's rearward motion, followed by an increase in acceleration in the y-direction due to the subsequent increase in the angle of incidence. The contour plots in Fig. 14d–f illustrate how changes in the relative positions of the two bodies lead to corresponding changes in the pressure distributions. The motion states show that the orbiter separates from the booster in the y-direction.



(a) (b) (c)

Fig. 13. The discretization of physical space and velocity space. The geometry (a) and initial mesh assembly (b) of the two-stage-to-orbit (TSTO) hypersonic vehicles. (c) The DVS mesh consisting of 4748 cells.



(a) (b) (c)

(d) (e) (f)

Fig. 14. The computational results of the two-stage-to-orbit (TSTO) hypersonic vehicles (a) The history position of orbiter's center of mass in the y-direction (b) The history velocity of orbiter's center of mass in the y-direction (c) The history angle of incidence. The pressure contours at different times (d) $t = 0$ (e) $t = 5 \times 10^{-4}$s (f) $t = 9 \times 10^{-4}$s.

5. Conclusion

In this work, an overlapping-mesh technique is coupled with the unsteady implicit UGKS to simulate complex flows with moving boundaries, without introducing a new UGKS flux formulation. The essential contribution is a kinetic overset strategy in which **W** and f are interpolated with identical gradient-corrected weights, and the overset connectivity is held fixed during dual-time inner iterations while fringe data are refreshed like MPI halos. Computational cost is further reduced by a dual time-stepping algorithm together with memory-efficient parallel implementation. A Taylor–Green accuracy test shows that gradient-corrected IDW preserves a convergence rate of about 1.7–1.8 on overset meshes and reduces the fine-grid fringe error compared with plain IDW, while the single-grid baseline exhibits nearly second-order decay ($p \approx 2.1$–2.7). The validity and robustness of the proposed method are demonstrated through a comprehensive series of test cases, ranging from two-dimensional micro-scale flows to complex three-dimensional vehicle separation problems. Overall, the numerical results support the accuracy and practical efficiency of the present approach for challenging multiscale gas dynamics with moving geometries; in particular, dual-time integration substantially relaxes the explicit CFL restriction on the physical time step, as quantified by the cavity timing comparison in Table 3 (about an eightfold wall-clock reduction versus explicit UGKS at CFL = 0.5).

CRediT authorship contribution statement

Yue Zhang: Writing – original draft, Visualization, Validation, Resources, Methodology; **Wenpei Long:** Writing – review & editing, Visualization, Validation, Methodology; **Junzhe Cao:** Writing – review & editing, Methodology; **Kun Xu:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 92371107), the National Key R & D Program of China (Grant No. 2022YFA1004500), and the Hong Kong Research Grants Council (Grant No. 16208324).

Data availability

Data will be made available on request.

References

[1] Z.-H. Li, A.-P. Peng, Q. Ma, L.-N. Dang, X.-W. Tang, X.-Z. Sun, Gas-kinetic unified algorithm for computable modeling of Boltzmann equation and application to aerothermodynamics for falling disintegration of uncontrolled Tiangong-No. 1 spacecraft, Adv. Aerodyn. 1 (1) (2019) 4.
[2] Y. Wang, S. Liu, C. Zhuo, C. Zhong, Investigation of nonlinear squeeze-film damping involving rarefied gas effect in micro-electro-mechanical systems, Comput. Math. Appl. 114 (2022) 188–209.
[3] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Vol. 1, Clarendon Press, (1994). https://books.google.com.hk/books?id=Bya5QgAACAAJ
[4] C.K. Chu, Kinetic-theoretic description of the formation of a shock wave, Phys. Fluids 8 (1) (1965) 12–22.
[5] J.Y. Yang, J.C. Huang, Rarefied flow computations using nonlinear model Boltzmann equations, J. Comput. Phys. 120 (2) (1995) 323–339.
[6] K. Xu, J.-C. Huang, A unified gas-kinetic scheme for continuum and rarefied flows, J. Comput. Phys. 229 (20) (2010) 7747–7764.
[7] Juan-Chen Huang, K. Xu, P. Yu, A unified gas-kinetic scheme for continuum and rarefied flows II: multi-dimensional cases, Commun. Comput. Phys. 12 (3) (2012) 662–690.
[8] Z. Guo, K. Xu, R. Wang, Discrete unified gas kinetic scheme for all Knudsen number flows: low-speed isothermal case, Phys. Rev. E-Stat. Nonlinear Soft Matter Phys. 88 (3) (2013) 033305.
[9] C. Liu, Y. Zhu, K. Xu, Unified gas-kinetic wave-particle methods I: continuum and rarefied gas flow, J. Comput. Phys. 401 (2020) 108977.
[10] Y. Zhu, C. Liu, C. Zhong, K. Xu, Unified gas-kinetic wave-particle methods. II. Multiscale simulation on unstructured mesh, Phys. Fluids 31 (6) (2019) 067105.
[11] K. Xu, Direct modeling for computational fluid dynamics: construction and application of unified gas-kinetic schemes, Advances in Computational Fluid Dynamics, Vol. 4, World Scientific, (2014).
[12] Z. Guo, J. Li, K. Xu, Unified preserving properties of kinetic schemes, Phys. Rev. E 107 (2) (2023) 025301.
[13] V.A. Titarev, Numerical modeling of high-speed rarefied gas flows over blunt bodies using model kinetic equations, Eur. J. Mech.-B/Fluids 64 (2017) 112–117.
[14] J. Chen, S. Liu, Y. Wang, C. Zhong, Conserved discrete unified gas-kinetic scheme with unstructured discrete velocity space, Phys. Rev. J. 100 (2019) 043305.
[15] T. Xiao, C. Liu, K. Xu, Q. Cai, A velocity-space adaptive unified gas kinetic scheme for continuum and rarefied flows, J. Comput. Phys. 415 (2020) 109535.
[16] Y. Wei, W. Long, K. Xu, Adaptive unified gas-kinetic scheme for diatomic gases with rotational and vibrational nonequilibrium, Comput. Phys. Commun. 305 (2024) 109324.
[17] L.M. Yang, L.C. Han, H. Ding, Z.H. Li, C. Shu, Y.Y. Liu, Adaptive partitioning-based discrete unified gas kinetic scheme for flows in all flow regimes, Adv. Aerodyn. 5 (1) (2023) 15.
[18] Y. Wei, J. Cao, X. Ji, K. Xu, Adaptive wave-particle decomposition in UGKWP method for high-speed flow simulations, Adv. Aerodyn. 5 (1) (2023) 25.
[19] J. Cao, Y. Wei, W. Long, C. Zhong, K. Xu, Adaptive criterion and modification of wave-particle decomposition in UGKWP method for high-speed flow simulation, Comput. Fluids 305 (2025) 106896.
[20] Y. Zhu, C. Zhong, K. Xu, Implicit unified gas-kinetic scheme for steady state solutions in all flow regimes, J. Comput. Phys. 315 (2016) 16–38.
[21] X. Xu, Y. Zhu, C. Liu, K. Xu, UGKS-based implicit iterative method for multiscale nonequilibrium flow simulations, SIAM J. Sci. Comput. 44 (4) (2022) B996–B1017.
[22] Y. Zhu, C. Zhong, K. Xu, An implicit unified gas-kinetic scheme for unsteady flow in all Knudsen regimes, J. Comput. Phys. 386 (2019) 190–217.
[23] R. Zhang, S. Liu, J. Chen, C. Zhuo, C. Zhong, A conservative implicit scheme for three-dimensional steady flows of diatomic gases in all flow regimes using unstructured meshes in the physical and velocity spaces, Phys. Fluids 36 (1) (2024) 016114.
[24] Y. Zhu, C. Zhong, K. Xu, Unified gas-kinetic scheme with multigrid convergence for rarefied flow study, Phys. Fluids 29 (9) (2017) 096102.
[25] W. Long, Y. Wei, K. Xu, An implicit adaptive unified gas-kinetic scheme for steady-state solutions of nonequilibrium flows, Phys. Fluids 36 (10) (2024) 106114.
[26] W. Su, L. Zhu, P. Wang, Y. Zhang, L. Wu, Can we find steady-state solutions to multiscale rarefied gas flows within dozens of iterations?, J. Comput. Phys. 407 (2020) 109245.
[27] W. Su, L. Zhu, L. Wu, Fast convergence and asymptotic preserving of the general synthetic iterative scheme, SIAM J. Sci. Comput. 42 (6) (2020) B1517–B1540.
[28] S. Chen, K. Xu, C. Lee, Q. Cai, A unified gas kinetic scheme with moving mesh and velocity space adaptation, J. Comput. Phys. 231 (20) (2012) 6643–6664.
[29] Y. Wang, C. Zhong, S. Liu, Arbitrary Lagrangian-Eulerian-type discrete unified gas kinetic scheme for low-speed continuum and rarefied flow simulations with moving boundaries, Phys. Rev. E 100 (6) (2019) 063310.
[30] Y. Wang, S. Liu, C. Zhuo, C. Zhong, Arbitrary Lagrangian-Eulerian-type conserved discrete unified gas kinetic scheme for the simulations of transonic continuum and rarefied gas flows with moving boundaries, Appl. Math. Model. 113 (2023) 545–572.
[31] S. Tao, H. Zhang, Z. Guo, L.-P. Wang, A combined immersed boundary and discrete unified gas kinetic scheme for particle–fluid flows, J. Comput. Phys. 375 (2018) 498–518.
[32] Q. He, S. Tao, G. Liu, L. Wang, Y. Ge, J. Chen, X. Yang, Thermal rarefied gas flow simulations with moving boundaries based on discrete unified gas kinetic scheme and immersed boundary method, Int. J. Heat Mass Transf. 226 (2024) 125508.
[33] J. Zeng, Y. Zhang, L. Wu, GSIS-ALE for moving boundary problems in rarefied gas flows, J. Comput. Phys. 525 (2025) 113761.
[34] Y. Zhang, Y. Wei, W. Long, K. Xu, An efficiency and memory-saving programming paradigm for the unified gas-kinetic scheme, Comput. Phys. Commun. 314 (2025) 109684.

[35] X. Xu, Y. Chen, C. Liu, Z. Li, K. Xu, Unified gas-kinetic wave-particle methods V: diatomic molecular flow, J. Comput. Phys. 442 (2021) 110496.
[36] S. Yoon, A. Jameson, Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations, AIAA J. 26 (9) (1988) 1025–1026.
[37] K. Nakahashi, F. Togashi, D. Sharov, Intergrid-boundary definition method for overset unstructured grid approach, AIAA J. 38 (11) (2000) 2077–2084.
[38] J. Bonet, J. Peraire, An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems, Int. J. Numer. Methods Eng. 31 (1) (1991) 1–17.
[39] R. Löhner, Robust, vectorized search algorithms for interpolation on unstructured grids, J. Comput. Phys. 118 (2) (1995) 380–387.
[40] X.H. Chang, R. Ma, N.H. Wang, Z. Zhao, L.P. Zhang, A parallel implicit hole-cutting method based on background mesh for unstructured Chimera grid, Comput. Fluids 198 (2020) 104403.
[41] A. Tsimpoukis, D. Valougeorgis, Linear harmonic oscillatory rarefied gas flow with arbitrary frequency in comb finger blocks, Sens. Actuators A: Phys. 331 (2021) 112997.
[42] S. Tiwari, A. Klar, G. Russo, Interaction of rigid body motion and rarefied gas dynamics based on the BGK model, Math. Eng. 2 (2020) 203–229.
[43] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, J. Fluid Mech. 360 (1998) 249–271.
[44] Y. Wang, Y. Wang, Z. Jiang, Numerical investigation of aerodynamic separation schemes for two-stage-to-orbit-like two-body system, Aerosp. Sci. Technol. 131 (2022) 107995.