



# Development of a particle collision algorithm for discontinuous Galerkin simulations of compressible multiphase flows

Eric J. Ching\*, Matthias Ihme

Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA

## ARTICLE INFO

### Article history:

Available online 26 March 2021

### Keywords:

Discontinuous Galerkin method  
Lagrangian particle tracking  
Particle-laden flow  
Hypersonic flow  
Curved elements  
Dusty flow

## ABSTRACT

This paper discusses the development of algorithms for simulating compressible, four-way coupled, particle-laden flows with discontinuous Galerkin methods. Specifically, the algorithmic developments focus on the treatment of hard-sphere collisions. First, we aim to reduce the computational effort devoted to inspecting pairs of particles for collisions during the given time step. This is often one of the most expensive stages of four-way coupled particle-laden flow simulations. The proposed algorithm exploits information about relative particle positions provided by the geometric mapping of a physical element to a reference element commonly employed in discontinuous Galerkin and other finite-element-based solvers. Second, we develop an enhanced particle-wall collision methodology that better accounts for finite particle size than previously developed schemes. A particular feature of the particle-particle and particle-wall collision algorithms is compatibility with arbitrary, curved, unstructured elements. Finally, we apply the resulting framework to a number of test cases, including hypersonic dusty flow over an entry capsule and surface erosion by sandblasting.

© 2021 Elsevier Inc. All rights reserved.

## 1. Introduction

Numerical simulations of fluid flows laden with solid particles are frequently performed to improve understanding and prediction of many engineering applications, ranging from dust-plume interaction in moon and Mars landings [1,2] to fluidized bed reactors [3,4] and grain harvesting and storage [5,6]. The disparate spatiotemporal scales in these physical processes typically render fully resolved simulations computationally intractable. As such, a number of modeling approaches have been developed to enable large-scale simulations of acceptable fidelity. In this work, we focus on the Euler-Lagrange method, in which the disperse phase is described as a distribution of points that evolves within the Eulerian mesh. This differs from the other common approach, known as the Euler-Euler method, in which both phases are treated in an Eulerian manner as continua [7].

For very dilute particle-laden flows, the reverse coupling of the particles to the carrier phase can often be ignored [7]. This is referred to as one-way coupling since only the effect of the carrier phase on the disperse phase is taken into account. Two-way coupling, in which the reverse coupling is included, is needed for moderately dilute flows. Consideration of even higher concentrations necessitates four-way coupling, in which particle-particle interactions are accounted for. In this study,

\* Corresponding author.

E-mail address: [eching@stanford.edu](mailto:eching@stanford.edu) (E.J. Ching).

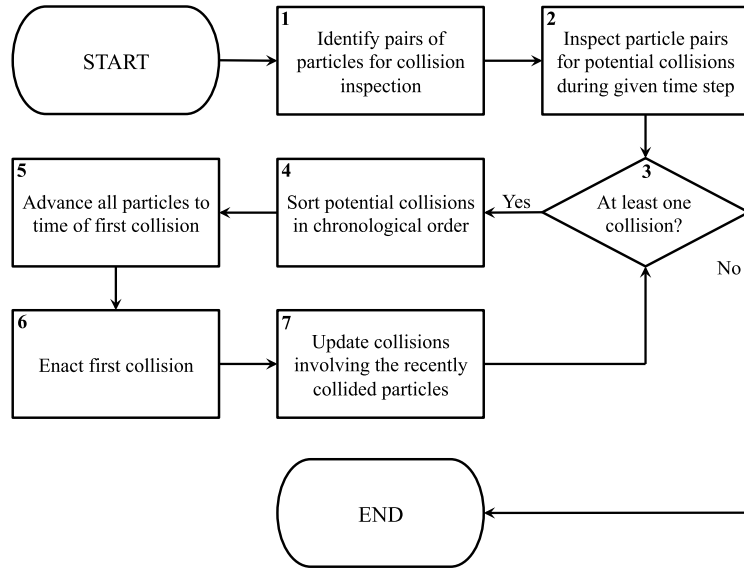


Fig. 1. Flowchart describing the procedure for handling hard-sphere interparticle collisions.

we target interparticle collisions based on the hard-sphere model. This model is more appropriate than the alternative soft-sphere model for the high-speed flows considered here due to its laxer time stepping constraints.

Regardless of the collision model, the computational cost associated with handling particle-particle collisions can be extremely high, especially for large volume fractions. A schematic of the individual steps for dealing with hard-sphere collisions is given in Fig. 1. The second step, in which pairs of particles are inspected for potential collisions, is often the main contributor to the high computational cost. In a brute-force approach where all possible pairs of particles in the entire domain are checked, the cost scales with the square of the total number of particles. Several strategies have been developed to significantly reduce this cost by (prior to this step) localizing the pairs of particles to be inspected. One common approach is to employ *particle neighbor lists* [8,9]. In this method, for a given particle, all other particles within a prescribed separation distance are stored. The given particle is checked for potential collisions with only the stored particles, instead of all other particles in the domain. However, this approach can be very memory-intensive, and there is some overhead associated with constructing the particle neighbor lists, which must be updated periodically. Furthermore, it is not always clear how often the lists should be updated, and previous studies have concluded that this approach is not very efficient for hard-sphere collisions [10,11]. Another common approach relies instead on *element neighbor lists* [12,11,13]. Specifically, a given particle is inspected for potential collisions with only other particles located in the neighborhood of mesh elements, i.e. the host element of the given particle and the node-sharing elements. For a standard hexahedral mesh, this corresponds to examining 27 elements, which can still result in many superfluous particle-pair inspections. Unnecessary inspections can also occur when a large number of particles reside in each element. A related approach is based on *bin neighbor lists*, in which the domain is partitioned into an auxiliary Cartesian grid of bins [14,10,15]. Note that here, we make a distinction between the *elements* of the primary Eulerian mesh and the *bins* of the secondary Cartesian mesh. Just as in the element-neighbor-list approach, in 3D calculations, the 27 bins comprising the neighborhood of a given particle are examined. This strategy is very popular for soft-sphere collisions, in which the bin size is simply equal to the maximum particle diameter. However, for polydisperse flows with wide size distributions, a significant number of unnecessary inspections can occur. Moreover, for hard-sphere collisions, the choice of bin size is less straightforward. If the bins are large enough to contain the volume of influence swept by a particle during a given time step, this can be very inefficient, especially for a wide distribution of particle velocities. This is because the volume of influence typically only comprises a small fraction of its host bin, which in general is cube-shaped. On the other hand, if the bin size is equal to the particle diameter, strict restrictions must be placed on the time step size; otherwise, collisions may be ignored [16].

Many variants of the above approaches have been proposed to overcome their deficiencies and further reduce the number of particle-pair inspections. Yao et al. [17] combined ideas from particle neighbor lists and bin neighbor lists to reduce the computational cost of updating the particle neighbor lists, but this can still be very memory-intensive. Breuer and Alletto [18] devised a dual-bin algorithm that uses two auxiliary Cartesian grids with minimal overlap. In addition, Ogarko and Luding [19] and Krijgsman et al. [20] developed a bin-based strategy that utilizes a multi-level grid, which is very efficient for soft-sphere collisions among particles of different sizes. Sigurgeirsson et al. [16] modified the bin-neighbor-list approach to be more appropriate for hard-sphere collisions. Specifically, to remove the aforementioned restriction of the time step size, they keep track of all bins traversed by the particle within a given time step and update the bin neighbor lists accordingly.

For certain particle-laden flows with hard-sphere collisions, the algorithm by Sigurgeirsson et al. [16] represents a good choice. However, if particle velocities range from near-zero to hundreds or thousands of meters per second (as is the case for the flows considered in this study), the number of bins traversed by a particle may be substantial, dramatically increasing the computational overhead of identifying those bins and updating the corresponding neighbor lists. This is in part due to Cartesian nature of the auxiliary grid, in which bins are uniform in size throughout the domain. In the element-neighbor-list approach, on the other hand, the potential variation in element size, orientation, and anisotropy can be useful for reducing computational cost. For instance, in particle-laden flow through a pipe or over a flat plate, the relatively large mesh spacing in the streamwise direction and relatively small spacing in the wall-normal direction align well with the volumes of influence swept by the particles within a time step, making the mesh elements very appropriate for localizing potential collision partners. The element-based approach is also appealing since it avoids the need to map bins to processors in parallel simulations. Although the bin-to-processor map can be constructed in the beginning of the simulation, it would need to be periodically reconstructed in the case of dynamic load balancing or mesh motion. Nevertheless, since the mesh size is not directly tied to particle trajectories, many superfluous particle-pair inspections can still occur in an element-based technique. As such, our main objective in this work is to develop an efficient variant of the element-neighbor-list approach that can further reduce the number of inspections without omitting collisions. This is done in the context of discontinuous Galerkin schemes, which have received considerable attention in recent years. Desirable properties of discontinuous Galerkin methods include arbitrarily high order of accuracy, geometric flexibility, a compact stencil, and an affinity for modern computing systems, such as graphics processing units. We previously developed a two-way coupled Euler-Lagrange methodology under a discontinuous Galerkin framework [21]. The back-coupling of the particles to the carrier fluid is weighed to the Eulerian mesh using Dirac delta functions. Tracking particles on high-order curved elements, which are prevalent in discontinuous Galerkin simulations, is much more difficult than on straight-sided elements. Special strategies are employed to deal with such elements. The basic tracking procedure leverages information on particle positions provided by the geometric mapping of a physical element to a reference element.

Building on the standard element-neighbor-list approach, we require the desired collision detection algorithm to satisfy the following: (i) it should be easy to implement; (ii) the operations to condense the element neighbor lists should be simple and incur minimal computational cost; and (iii) it should be compatible with arbitrary, curved, unstructured elements. To achieve the above, the geometric mapping is again relied upon. Here, it is further exploited to obtain information on the relative positions between a given pair of particles during their trajectories in order to further narrow potential collision partners. The proposed methodology can also be applied to other finite-element-based solvers, which frequently rely on the geometric mapping. The developed collision detection technique is presented as an alternative to bin-based approaches; it is not necessarily preferred for all multiphase flow configurations involving hard-sphere collisions, but its aforementioned advantages are especially evident when the mesh elements align well with the particle trajectories. Another important objective of this work is to improve on the approach developed in Ref. [21] for dealing with particle-wall collisions on curved, high-aspect-ratio elements. Specifically, said approach assumes infinitesimally small particles [21], while here, we aim to precisely account for finite particle size in computing the intersection between the trajectory of a given particle and the wall.

The remainder of this paper is organized as follows. Section 2 briefly describes the governing equations of the carrier fluid and their discontinuous Galerkin discretization. Section 3 focuses on the physical model of the disperse phase and the associated numerical methods. The improved particle-wall collision algorithm and the interparticle collision detection strategy are discussed in detail. In Section 4, we apply the developed collision algorithms to three test cases. In the first case, we verify the methodology by drawing comparisons to kinetic theory. The second case concerns the erosive action of sandblasting, and the final test involves hypersonic dusty flow over an entry capsule. We conclude the paper with a discussion of the main findings.

## 2. Carrier phase

In this section, we discuss the governing equations of the carrier phase and their discrete form upon applying the discontinuous Galerkin method.

### 2.1. Governing equations

The behavior of the carrier phase is described by the compressible Navier-Stokes equations, written in vector form as

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}_S = \nabla \cdot \mathbf{F}_V + \mathbf{S}, \quad (1)$$

where  $\mathbf{U}(\mathbf{x}, t) : \mathbb{R}^{N_d} \times \mathbb{R} \rightarrow \mathbb{R}^{N_U}$  is the conservative state vector,  $\mathbf{F}_S(\mathbf{U}) : \mathbb{R}^{N_U} \rightarrow \mathbb{R}^{N_U \times N_d}$  is the inviscid flux vector,  $\mathbf{F}_V(\mathbf{U}, \nabla \mathbf{U}) : \mathbb{R}^{N_U} \times \mathbb{R}^{N_U \times N_d} \rightarrow \mathbb{R}^{N_U \times N_d}$  is the viscous flux vector, and  $\mathbf{S}(\mathbf{U}, \nabla \mathbf{U}) : \mathbb{R}^{N_U} \times \mathbb{R}^{N_U \times N_d} \rightarrow \mathbb{R}^{N_U}$  is the source term vector that accounts for the effect of the disperse phase on the carrier phase. In addition,  $\mathbf{x} \in \mathbb{R}^{N_d}$  is the spatial coordinate vector,  $t \in \mathbb{R}^+$  is the time,  $N_U$  is the number of state variables, and  $N_d$  is the number of spatial dimensions. The terms in Eq. (1) can be expanded as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{bmatrix}, \quad \mathbf{F}_s = \begin{bmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + P \mathbb{I} \\ \mathbf{u}(\rho E + P) \end{bmatrix}, \quad \mathbf{F}_v = \begin{bmatrix} 0 \\ \boldsymbol{\tau} \\ \mathbf{u} \cdot \boldsymbol{\tau} - \mathbf{q} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} S_\rho \\ \mathbf{S}_m \\ S_e \end{bmatrix}, \quad (2)$$

where  $\rho$  is the fluid density,  $\mathbf{u}$  is the velocity,  $P$  is the pressure,  $E$  is the total specific energy, and  $\mathbb{I}$  is the identity matrix.  $S_\rho$ ,  $\mathbf{S}_m$ , and  $S_e$  are the terms to account for the back-coupled mass, momentum, and energy transfer, respectively. Pressure is computed from the ideal gas law as  $P = \rho RT$ , where  $R$  is the specific gas constant and  $T$  is the temperature.

For calorically perfect gases,  $T$  is computed in analytic form as  $T = (E + |\mathbf{u}|^2/2)/c_v$ , where  $c_v$  is the specific heat capacity at constant volume. For higher temperatures, we employ the thermally perfect gas model, with  $T$  computed iteratively from the NASA polynomial fits for thermodynamic properties [22]. Sutherland's law is used to compute the dynamic viscosity, and thermal conductivity is calculated based on the assumption of a constant Prandtl number. Note that Eqs. (1) neglect the volume fraction of the disperse phase, which is a reasonable assumption given the overall relatively dilute flows considered here. Nevertheless, future work will entail incorporating volume averaging [7,23] or filtering [24,25] to enhance the physical modeling fidelity. In addition, only interphase momentum and energy transfer are considered in this study, i.e. interphase mass transfer is ignored ( $S_\rho = 0$  in Eq. (2)).

## 2.2. Discontinuous Galerkin discretization

We consider the problem to be posed on the computational domain  $\Omega$  with boundary  $\partial\Omega$ .  $\Omega$  is partitioned into  $N_e$  discrete elements such that  $\Omega = \cup_{e=1}^{N_e} \Omega_e$ .  $\partial\Omega_e$  denotes the boundary of element  $\Omega_e$ . The local solution,  $\mathbf{U}^e$ , is approximated as

$$\mathbf{U}^e(\mathbf{x}, t) \approx \mathbf{U}_h^e(\mathbf{x}, t) = \sum_{n=1}^{N_b} \tilde{\mathbf{U}}_n^e(t) \phi_n(\mathbf{x}), \quad (3)$$

where  $\tilde{\mathbf{U}}_n^e(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^{N_u}$  is the  $n$ th vector of basis coefficients,  $\phi_n$  is the  $n$ th basis function, and  $N_b$  is the number of basis functions. In this work, we use a Lagrange polynomial basis. The global solution approximation can then be written as

$$\mathbf{U}_h = \oplus_{e=1}^{N_e} \mathbf{U}_h^e. \quad (4)$$

To solve for the basis coefficients on  $\Omega_e$ , we require  $\tilde{\mathbf{U}}^e(t)$  to satisfy

$$\sum_{n=1}^{N_b} d_t \tilde{\mathbf{U}}_n^e(t) \mathcal{M}_{mn}^e + \int_{\Omega_e} \phi_m \nabla \cdot \mathbf{F}_s d\Omega = \int_{\Omega_e} \phi_m \nabla \cdot \mathbf{F}_v d\Omega + \int_{\Omega_e} \phi_m \mathbf{S} d\Omega \quad \forall \phi_m, \quad (5)$$

where  $\mathcal{M}_{mn}^e = \int_{\Omega_e} \phi_m \phi_n d\Omega$  represents the element-local mass matrix.

For the calorically perfect gas model, we use the original Roe solver [26] and the BR2 scheme [27] to treat the inviscid and viscous fluxes, respectively. For the thermally perfect gas model, we employ the Roe solver by Yee et al. [28] and incorporate necessary modifications to the fourth-order homogeneity tensor resulting from derivatives of the viscous flux to the gradient of the state variables, i.e.  $\partial \mathbf{F}_d / \partial \nabla(\mathbf{U})$ . To evaluate integrals, we use standard Gauss-Legendre quadrature with an order of accuracy no less than  $2p + 1$ , where  $p$  is the user-prescribed order of the Lagrange polynomials. To capture shocks, intraelement pressure variations are used for shock detection while smooth artificial viscosity is used for stabilization [29].

## 3. Disperse phase

This section first briefly summarizes the physical model of the disperse phase and the particle method developed in Ref. [21]. The updated particle-wall collision algorithm and the proposed approach for identifying particle pairs for collision inspection are then discussed in detail.

### 3.1. Physical model

Lagrangian particle tracking is used to compute the behavior of the disperse phase in the flow field. To simplify the calculations, individual particles are treated as smooth, solid, inert spheres with fixed diameters that exchange only momentum and energy with the carrier phase. It is assumed that the particle temperature is uniform over the entire particle. Furthermore, particles interact with each other only through hard-sphere collisions. The particle state is computed using the following set of ordinary differential equations [7]:

$$\frac{d\mathbf{x}_d}{dt} = \mathbf{u}_d, \quad (6a)$$

$$m_d \frac{d\mathbf{u}_d}{dt} = \mathbf{F} = \mathbf{F}_{qs} + \mathbf{F}_M + \mathbf{F}_{\text{thermo}}, \quad (6b)$$

$$I_d \frac{d\boldsymbol{\omega}_d}{dt} = \mathbf{T} = \mathbf{T}_{qs}, \quad (6c)$$

$$m_d c_d \frac{dT_d}{dt} = Q = Q_{qs}, \quad (6d)$$

where the carrier and disperse phases are denoted by the subscripts “c” and “d”, respectively,  $\mathbf{x}_d$  is the location of the particle centroid,  $\mathbf{u}_d$  is the particle velocity,  $m_d = \rho_d \frac{\pi}{6} D^3$  is the particle mass (with  $\rho_d$  the particle density and  $D$  the particle diameter),  $\boldsymbol{\omega}_d$  is the angular velocity of the particle,  $I_d = \frac{1}{10} m_d D^2$  is the moment of inertia,  $c_d$  is the specific heat of the particle, and  $T_d$  is the particle temperature. Note that particle rotations, which are important for four-way coupled systems, were ignored in our previous work [21]. Equations (6) are valid during the free motion of the particle. The particle state can also change as a result of collisions with surfaces and other particles, which will be discussed later in this section.  $\mathbf{F}_{qs}$  and  $Q_{qs}$  denote the quasi-steady drag force and heating rate, respectively, defined as

$$\mathbf{F}_{qs} = \frac{\pi}{8} D^2 \rho_c (\mathbf{u}_c - \mathbf{u}_d) |\mathbf{u}_c - \mathbf{u}_d| C_D, \quad (7a)$$

$$Q_{qs} = \pi D \kappa_c (T_c - T_d) \text{Nu}, \quad (7b)$$

where  $\kappa_c$  is the thermal conductivity of the carrier fluid,  $C_D$  is the drag coefficient, and Nu is the Nusselt number. In this work, we employ the drag correlation by Henderson [30] and the Nusselt number correlation by Fox et al. [31].  $\mathbf{F}_M$ , the Magnus force, and  $\mathbf{T}_{qs}$ , the quasi-steady torque, take the forms proposed by Volkov [32]:

$$\mathbf{F}_M = -\frac{\pi}{16} D^3 \rho_c |\boldsymbol{\omega}_c - \boldsymbol{\omega}_d| |\mathbf{u}_c - \mathbf{u}_d| C_L \sin \Theta \hat{\mathbf{c}}, \quad (8a)$$

$$\mathbf{T}_{qs} = \frac{\pi}{64} D^5 \rho_c |\boldsymbol{\omega}_c - \boldsymbol{\omega}_d|^2 (C_{Ta} \hat{\mathbf{a}} + C_{Tb} \hat{\mathbf{b}}), \quad (8b)$$

where  $\boldsymbol{\omega}_c = \frac{1}{2} \nabla \times \mathbf{u}_c$  is the angular velocity of the carrier phase,  $C_L$  is the lift coefficient, and  $C_{Ta}$  and  $C_{Tb}$  are the torque coefficients [32].  $\hat{\mathbf{a}}$ ,  $\hat{\mathbf{b}}$ , and  $\hat{\mathbf{c}}$  are orthogonal unit vectors given by

$$\hat{\mathbf{a}} = \frac{\mathbf{u}_c - \mathbf{u}_d}{|\mathbf{u}_c - \mathbf{u}_d|}, \quad (9a)$$

$$\hat{\mathbf{b}} = \hat{\mathbf{c}} \times \hat{\mathbf{a}}, \quad (9b)$$

$$\hat{\mathbf{c}} = \frac{\hat{\mathbf{a}} \times (\boldsymbol{\omega}_c - \boldsymbol{\omega}_d)}{|\boldsymbol{\omega}_c - \boldsymbol{\omega}_d| \sin \Theta}, \quad (9c)$$

where  $\Theta$  is the angle between  $(\mathbf{u}_c - \mathbf{u}_d)$  and  $(\boldsymbol{\omega}_c - \boldsymbol{\omega}_d)$ .  $\mathbf{F}_{\text{thermo}}$  in Eq. (6b) is the thermophoretic force, for which the model by Loth [33] is employed. This force is in the direction of the negative temperature gradient and is caused by higher-energy molecular collisions on the “hot” side of the particle than on the “cold” side. The thermophoretic force can be non-negligible in high-speed dusty flows over blunt bodies, particularly in the thermal boundary layer due to the strong temperature gradients [34]. Further details on the physical model of the particles, including additional contributions to momentum and energy transfer not considered here, can be found in Ref. [7].

### 3.2. Reverse coupling

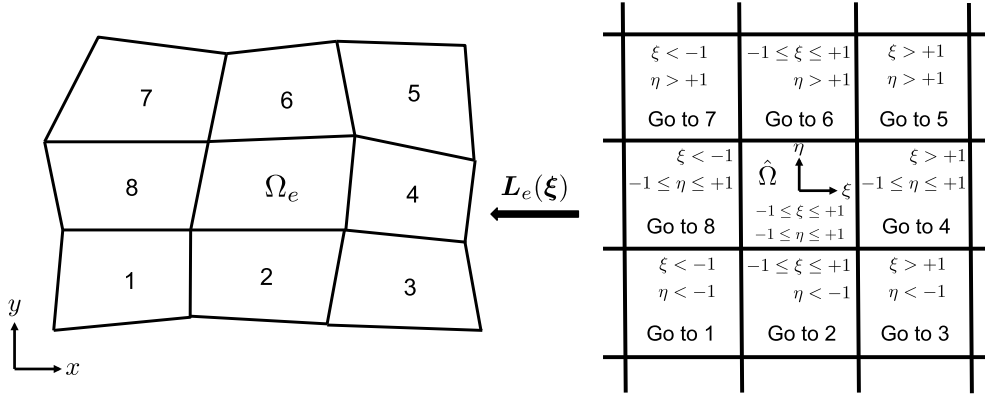
The back-coupling of the particles to the fluid is carried out via the source term on the RHS of Eqs. (1). Since interphase mass transfer is neglected,  $S_\rho = 0$ .  $\mathbf{S}_m$  and  $S_e$  are written as

$$\mathbf{S}_m = - \sum_{i=1}^{N_p} \mathbf{F}_i \delta(\mathbf{x} - \mathbf{x}_{d,i}), \quad (10a)$$

$$S_e = - \sum_{i=1}^{N_p} (Q_i + \mathbf{u}_{d,i} \cdot \mathbf{F}_i) \delta(\mathbf{x} - \mathbf{x}_{d,i}), \quad (10b)$$

where  $N_p$  is the total number of particles,  $\delta(\mathbf{x} - \mathbf{x}_{d,i})$  is the shifted Dirac delta function centered on the  $i$ th particle, and the second term on the RHS Eq. (10b) represents the work done by the drag force. The final integral on the RHS of Eqs. (5) can then be evaluated as

$$\int_{\Omega_e} \phi_m \mathbf{S} d\Omega = \sum_{i=1}^{N_p^e} \phi_m(\mathbf{x}_{d,i}) [0, \mathbf{F}_i, Q_i + \mathbf{u}_{d,i} \cdot \mathbf{F}_i]^T, \quad (11)$$



**Fig. 2.** Procedure for choosing the next element to search based on the reference position,  $\xi = [\xi, \eta]^T$ .

where  $N_p^e$  is the number of particles located in  $\Omega_e$ . The use of delta functions is very efficient and works adequately for the numerical tests considered in this study. For cases with stronger reverse coupling and/or particle diameters that approach the element size [12], smooth distribution functions [35] may be necessary to mitigate numerical instabilities and maintain accuracy.

### 3.3. Particle search-locate procedure

We proceed to review the search-locate algorithm presented by Allievi and Bermejo [36], which is used to identify the host cell of a given particle and map its physical position,  $\mathbf{x}_d$ , to its reference position,  $\xi_d$ , in the physical element.  $\xi_d$  is needed to interpolate the state of the carrier phase to the particle position (using the local polynomial approximation as in Eq. (3)) in evaluating the RHS of Eqs. (6). Note that in Ref. [21], we proposed a simplified variant of the original procedure by Allievi and Bermejo [36]. Here, we show the original procedure due to its greater relevance to the interparticle collision algorithm discussed in Section 3.5. We focus specifically on  $q = 2$  2D quadrilaterals, where  $q$  is the order of the geometry interpolation. The general concepts apply similarly to other element shapes, orders, and dimensions. Additional details can be found in Ref. [36].

The reference element is denoted by  $\hat{\Omega}$ , given as

$$\hat{\Omega} = \left\{ \xi = [\xi, \eta]^T \mid -1 \leq \xi, \eta \leq 1 \right\}. \quad (12)$$

The geometric mapping,  $L_e : \hat{\Omega} \rightarrow \Omega_e$ , is defined as

$$L_e(\xi) \equiv \begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^{N_n} \Phi_i(\xi) \begin{bmatrix} x_i^e \\ y_i^e \end{bmatrix}, \quad (13)$$

where  $[x_i^e, y_i^e]^T$  are the physical coordinates of the  $i$ th geometric node of  $\Omega_e$ ,  $\Phi_i$  is the  $i$ th geometric basis function, and  $N_n$  is the number of nodes. To determine whether the centroid of a given particle is located in  $\Omega_e$ , the following equation is solved for  $\xi_d$ :

$$\mathbf{G}(\xi_d) \equiv \mathbf{x}_d - L_e(\xi_d) = 0. \quad (14)$$

We employ Newton's method to solve Eq. (14) since in general, there is no analytical solution. If  $-1 \leq \xi_d, \eta_d \leq 1$ , then the particle is inside  $\Omega_e$ , otherwise, the next element to search is chosen according to Fig. 2.

### 3.4. Particle-wall collisions

This section summarizes the key components of the hard-sphere particle-wall collision algorithm developed previously and then presents the enhancements that more effectively account for finite particle sizes.

We first review the geometric mapping from the reference line segment to a given boundary face in physical space,  $\Gamma_f$ .  $\hat{\Gamma}$  denotes the reference line segment, defined as

$$\hat{\Gamma} = \{ \zeta \mid -1 \leq \zeta \leq 1 \}, \quad (15)$$

and  $\mathbf{H}_f : \hat{\Gamma} \rightarrow \Gamma_f$ , where  $\mathbf{H}_f$  is the geometric mapping, is given by

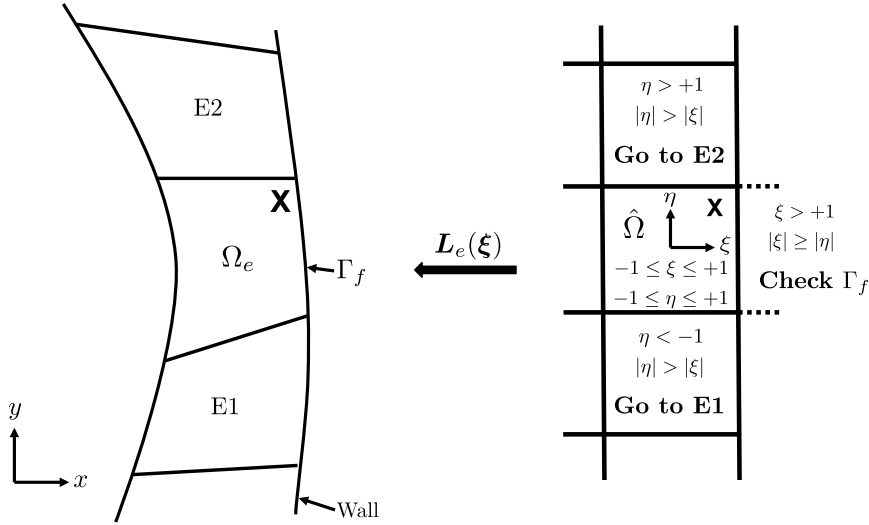


Fig. 3. Extension of the particle search-locate procedure to take into account particle-wall collisions based on the reference position,  $\xi = (\xi, \eta)$ .

$$\mathbf{H}_f(\zeta) \equiv \begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^{N_m} \psi_i(\zeta) \begin{bmatrix} x_i^e \\ y_i^e \end{bmatrix}, \quad (16)$$

where  $N_m$  is the number of geometric nodes of  $\Gamma_f$  and  $\psi_i$  is the  $i$ th geometric basis function of  $\Gamma_f$ .

The particle trajectory in a given time step is approximated as a line segment, parametrized as

$$\mathbf{r}^n(s) = \mathbf{x}_d^{n-1} + \mathbf{a}^n s, \quad (17)$$

where  $n$  indexes the time step,  $s \in [0, 1]$ , and  $\mathbf{a}^n = \mathbf{x}_d^n - \mathbf{x}_d^{n-1}$ . The following equation is solved for  $\zeta_d$  and  $s_d$ :

$$\mathbf{I} \equiv \mathbf{H}_f(\zeta_d) - \mathbf{r}^n(s_d) = 0. \quad (18)$$

Just as with Eq. (14), we employ Newton's method to solve Eq. (18). If  $-1 \leq \zeta_d \leq 1$  and  $0 < s_d \leq 1$ , then the particle collides with the wall and the post-collision trajectory can be computed. This procedure can be extended to arbitrary element shapes and orders.

Additional difficulties must be addressed in the case of curved elements, i.e.  $q \geq 2$ . For instance, a particle can bounce off a given boundary face multiple times in one time step. In addition, a particle can both exit the computational domain at a given boundary face and reenter it at the same boundary face in one time step, resulting in two intersections between the particle trajectory and said face. In general, no more than two such intersections occur within a time step. With such pathological cases in mind, we developed two methods that govern the execution of the aforementioned Newton search for solving Eq. (18) [21]. In *Method A*, the particle search-locate procedure described in Section 3.3 is extended as in Fig. 3 to account for particle-wall collisions. Although this approach is very efficient, it neglects the previously discussed scenario in which a particle trajectory exits and reenters the domain at a specific boundary face. In *Method B*,  $\Gamma_f$  is partitioned into arc segments, for which bounding boxes are computed. The Newton search is invoked on a given arc segment if any portion of the particle trajectory is inside the corresponding bounding box. This can result in extraneous Newton searches, but takes into consideration the aforementioned pathological scenario. Penalty-based approaches can also be incorporated to help ensure the correct intersection point is found [37]. These strategies for handling particle-wall collisions are discussed in detail in Ref. [21].

The above methodology for handling hard-sphere particle-wall collisions on non-planar faces, along with other relevant algorithms in the literature [38,39], assume the particles to be infinitesimally small. While appropriate for particles considerably smaller than the near-wall elements, accuracy is degraded for larger particles since the particle diameter is not explicitly incorporated into the formulation. As such, we now present enhancements to *Method A* and *Method B* that account for the finite size of particles. Here, we assume that the particle size is smaller than the element size, which is true for the flows considered in this study and in typical Euler-Lagrange simulations.

#### Method A

The nodes of  $\Gamma_f$  are shifted by a distance equal to the particle radius,  $R = D/2$ , such that the following modified face mapping is given:



$$\mathbf{H}_f^*(\zeta) \equiv \begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^{N_m} \psi_i(\zeta) \begin{bmatrix} x_i^e - R\hat{n}_1(\zeta_i) \\ y_i^e - R\hat{n}_2(\zeta_i) \end{bmatrix}, \quad (19)$$

where  $\zeta_1$  is the location of the  $i$ th node in reference space and  $\hat{n}_1$  and  $\hat{n}_2$  are the components of the outward-pointing unit normal vector, given by

$$\hat{\mathbf{n}}(\zeta) = \begin{bmatrix} \hat{n}_1(\zeta) \\ \hat{n}_2(\zeta) \end{bmatrix} = \frac{\frac{\partial H_{f2}}{\partial \zeta} \hat{\mathbf{e}}_x - \frac{\partial H_{f1}}{\partial \zeta} \hat{\mathbf{e}}_y}{\left| \frac{\partial \mathbf{H}_f}{\partial \zeta} \right|_\zeta}, \quad (20)$$

with  $\hat{\mathbf{e}}_x$  and  $\hat{\mathbf{e}}_y$  the standard basis vectors in the  $x$ - and  $y$ -directions, respectively. Equation (18) is then modified as

$$\mathbf{I}^* \equiv \mathbf{H}_f^*(\zeta_d) - \mathbf{r}^n(s_d) = 0. \quad (21)$$

The element mapping is adjusted in the same fashion, with the modified form denoted  $\mathbf{L}_e^*$ . Note that only the nodes associated with  $\Gamma_f$  are modified. With this, the element/face selection scheme described in Fig. 3 can be directly applied, resulting in an efficient particle-wall collision algorithm with minimal extraneous Newton searches. The modified face and element mappings ( $\mathbf{H}_f^*$  and  $\mathbf{L}_e^*$ , respectively) are utilized only in the tracking and localization of the particle, not, for example, in the evaluation of the state of the carrier fluid.

#### Method B

Here, the face mapping is modified as

$$\mathbf{H}_f^\dagger(\zeta) \equiv \begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^{N_m} \psi_i(\zeta) \begin{bmatrix} x_i^e \\ y_i^e \end{bmatrix} - R\hat{\mathbf{n}}(\zeta), \quad (22a)$$

$$= \mathbf{H}_f(\zeta) - R\hat{\mathbf{n}}(\zeta), \quad (22b)$$

and Eq. (18) is modified as

$$\mathbf{I}^\dagger \equiv \mathbf{H}_f^\dagger(\zeta_d) - \mathbf{r}^n(s_d) = 0. \quad (23)$$

Unlike in *Method A*, the element mapping,  $\mathbf{L}_e$ , is unchanged. The bounding boxes for each arc segment must be expanded in a similar fashion to take into account the finite particle size.  $\mathbf{H}_f^*$  in *Method A* is equivalent to  $\mathbf{H}_f^\dagger$  for  $q = 1$ . On the other hand,  $\mathbf{H}_f^*$  remains a polynomial for  $q \geq 2$ , while  $\mathbf{H}_f^\dagger$  in general is not. Due to its increased complexity,  $\mathbf{H}_f^\dagger$  is less appropriate for *Method A* since the element mapping should be modified in the same manner, which would significantly complicate the search-locate procedure described in Section 3.3. Note that  $\mathbf{H}_f^*$  can also be employed for *Method B*, which would simplify the determination of the particle-wall intersection, but we present  $\mathbf{H}_f^\dagger$  here for completeness.

For the flows computed in this work, the aforementioned pathological scenario in which a particle exits and reenters the domain at the same boundary face is very rare. Therefore, we elect to use the lower-cost *Method A*. Nevertheless, *Method B* is useful for cases in which the elements are significantly curved and the near-wall particle velocities are relatively high and isotropic. Furthermore, by directly modifying the element mapping in *Method A*, the determinant of the resulting geometric Jacobian can become negative, specifically for very high-aspect-ratio curved elements.

To compute the post-collision velocity of a given particle, we employ the theoretical model developed by Tsirkunov et al. [40] and extended by Volkov and Tsirkunov [41]. For reference, a schematic is provided in Fig. 4. The velocity at the contact point, denoted by the subscript “w,” just prior to impact is given as

$$\mathbf{u}_w^{(0)} = \mathbf{u}_d^{(0)} + R\boldsymbol{\omega}_d^{(0)} \times \hat{\mathbf{n}}_w, \quad (24)$$

where the superscript “(0)” indicates pre-collision. The unit tangential vector is obtained as

$$\hat{\mathbf{t}}_w = \frac{\mathbf{u}_w^{(0)} - u_{w,n}^{(0)} \hat{\mathbf{n}}_w}{|\mathbf{u}_w^{(0)} - u_{w,n}^{(0)} \hat{\mathbf{n}}_w|}, \quad (25)$$

where  $u_{w,n}^{(0)} = \mathbf{u}_w^{(0)} \cdot \hat{\mathbf{n}}_w$  is the normal component of the contact-point velocity. Letting  $\hat{\mathbf{k}}_w = \hat{\mathbf{n}}_w \times \hat{\mathbf{t}}_w$ , the components of the post-collision linear velocity,  $\mathbf{u}_d^{(1)}$ , are computed as



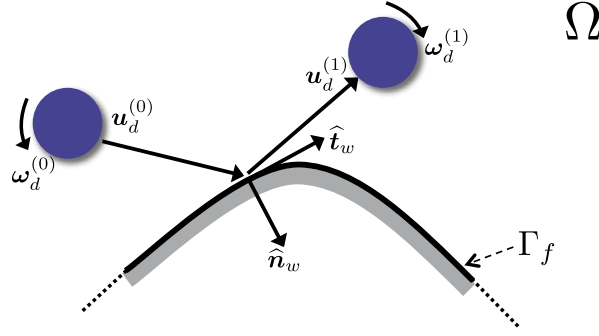


Fig. 4. Schematic of particle-wall collision. Vectors are not to scale.

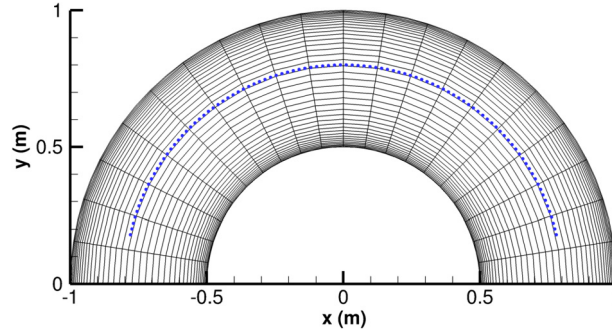


Fig. 5. Mesh for testing enhanced particle-wall collision algorithm, superimposed with initial particle locations (in blue). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$u_{w,n}^{(1)} = -e_{w,n} u_{w,n}^{(0)}, \quad (26a)$$

$$u_{w,t}^{(1)} = \begin{cases} e_{w,t} u_{w,t}^{(0)} - (1 - e_{w,t}) \frac{D}{2} \omega_{w,k}^{(0)}, & e_{w,t} > \frac{5}{7}, \\ e_{w,t} u_{w,t}^{(0)} - \frac{D}{7} \omega_{w,k}^{(0)}, & e_{w,t} \leq \frac{5}{7}, \end{cases} \quad (26b)$$

$$u_{w,k}^{(1)} = u_{w,k}^{(0)}, \quad (26c)$$

where  $e_{w,n}$  and  $e_{w,t}$  are the coefficients of restitution in the normal and tangential directions, respectively. The components of the post-collision angular velocity,  $\omega_d^{(1)}$ , are calculated as

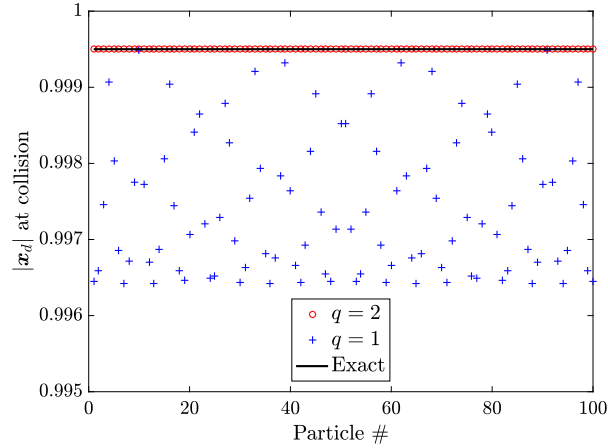
$$\omega_{w,n}^{(1)} = \omega_{w,n}^{(0)}, \quad (27a)$$

$$\omega_{w,t}^{(1)} = \omega_{w,t}^{(0)}, \quad (27b)$$

$$\omega_{w,k}^{(1)} = \begin{cases} \frac{5e_{w,t} - 3}{2} \omega_{w,k}^{(0)} + \frac{5}{4D} (e_{w,t} - 1) u_{w,t}^{(0)}, & e_{w,t} > \frac{5}{7}, \\ \frac{2}{7} \omega_{w,k}^{(0)} - \frac{2}{D} e_{w,t} u_{w,t}^{(0)}, & e_{w,t} \leq \frac{5}{7}. \end{cases} \quad (27c)$$

$e_{w,n}$  and  $e_{w,t}$  are obtained using the semiempirical correlations developed by Stasenko [42], which are functions of material properties, incident angle, and impact velocity. Eqs. (26) and (27) are derived from the laws of mechanics, along with realistic physical assumptions [40]. They account for the fact that most correlations to compute  $e_{w,n}$  and  $e_{w,t}$  for high-speed impacts are based on solely the pre- and post-collision linear velocities. These equations then represent modeling extensions to incorporate the angular velocities.

To test the enhanced particle-wall collision algorithm, we consider a configuration similar to that in Ref. [21] in which particles bounce off an annular boundary. The inner and outer annular boundaries are semicircles centered on the origin with diameters of one and two meters, respectively. Note the high near-wall-element aspect ratio. We initialize 100 particles with  $D = 0.001$  m in an annular profile with uniform speed in the outward radial direction. The mesh, superimposed with the initial particle locations, is given in Fig. 5. The carrier fluid is quiescent, and the back-coupling of the disperse phase is neglected. Fig. 6 displays the distance between the particle centroid and the origin at the point of collision for all particles in the top half of the domain. Results for both  $q = 1$  and  $q = 2$  are displayed. According to the exact solution,



**Fig. 6.** Distance from the origin at the point of collision for all particles in the top half of the domain. Results for both  $q = 1$  and  $q = 2$  are shown.

with  $D = 0.001$  m, the centroids of all particles should be located at exactly 0.9995 m when they strike the boundary. The  $q = 2$  results capture this very well due to the good representation of the circular boundary. On the other hand, the  $q = 1$  results are significantly worse since the circular boundary is approximated as a union of line segments and  $\hat{n}$  is uniform for each boundary face. These results support the notion established in Ref. [21] that curved meshes can significantly improve predictions of particle trajectories, especially when surface collisions are important. In addition, the proposed particle-wall collision algorithm is demonstrated to accurately account for finite particle size.

### 3.5. Interparticle collisions

This section describes the treatment of particle-particle collisions. First, we summarize the hard-sphere model for binary collisions, followed by the criterion for determining whether a pair of particles collides during a time step. We then propose an algorithm for selecting particle pairs for collision inspection.

#### 3.5.1. Hard-sphere model

Under the hard-sphere model, particle deformation is neglected and collisions are assumed to be instantaneous. Only binary collisions are considered. The impulse equations governing the collision are written as [7,43,44]

$$m_{d,i}(\mathbf{u}_{d,i}^{(1)} - \mathbf{u}_{d,i}^{(0)}) = \mathbf{J}, \quad (28a)$$

$$m_{d,j}(\mathbf{u}_{d,j}^{(1)} - \mathbf{u}_{d,j}^{(0)}) = -\mathbf{J}, \quad (28b)$$

$$I_{d,i}(\boldsymbol{\omega}_{d,i}^{(1)} - \boldsymbol{\omega}_{d,i}^{(0)}) = R_i \hat{\mathbf{n}}_{ij} \times \mathbf{J}, \quad (28c)$$

$$I_{d,j}(\boldsymbol{\omega}_{d,j}^{(1)} - \boldsymbol{\omega}_{d,j}^{(0)}) = R_j \hat{\mathbf{n}}_{ij} \times \mathbf{J}, \quad (28d)$$

where the subscripts “ $i$ ” and “ $j$ ” represent the two collision partners, the superscripts “(0)” and “(1)” denote pre- and post-collision quantities, respectively,  $\mathbf{J}$  is the impulse force, and  $\hat{\mathbf{n}}_{ij}$  is the unit vector from particle  $i$  to particle  $j$ , given by

$$\hat{\mathbf{n}}_{ij} = \frac{\mathbf{x}_{d,j} - \mathbf{x}_{d,i}}{|\mathbf{x}_{d,j} - \mathbf{x}_{d,i}|}. \quad (29)$$

The relative velocity between the collision partners at the contact point is defined as

$$\mathbf{v}_d = v_{d,n} \hat{\mathbf{n}}_{ij} + v_{d,t} \hat{\mathbf{t}}_{ij} = \mathbf{u}_{d,i} - \mathbf{u}_{d,j} + R_i \boldsymbol{\omega}_{d,i} \times \hat{\mathbf{n}}_{ij} + R_j \boldsymbol{\omega}_{d,j} \times \hat{\mathbf{n}}_{ij}, \quad (30a)$$

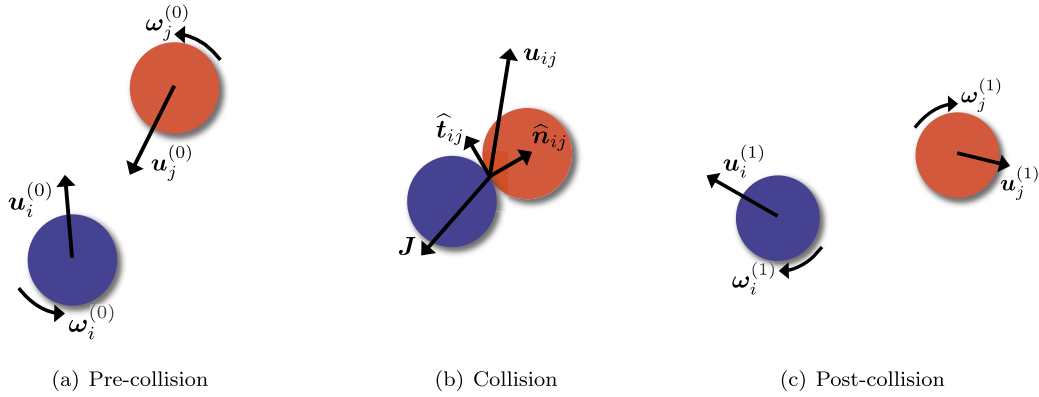
$$v_{d,n} \hat{\mathbf{n}}_{ij} = (\mathbf{v}_d \cdot \hat{\mathbf{n}}_{ij}) \hat{\mathbf{n}}_{ij}, \quad (30b)$$

$$v_{d,t} \hat{\mathbf{t}}_{ij} = \mathbf{v}_d - v_{d,n} \hat{\mathbf{n}}_{ij}, \quad (30c)$$

where  $v_{d,n}$  and  $v_{d,t}$  are the normal and tangential components and  $\hat{\mathbf{t}}_{ij}$  is the unit tangential vector. A schematic of the pre-collision, collision, and post-collision stages is provided in Fig. 7.

$\mathbf{J}$  is computed based on three parameters. The first is the coefficient of normal restitution,

$$e_n = -\frac{v_{d,n}^{(1)}}{v_{d,n}^{(0)}}, \quad (31)$$



**Fig. 7.** Pre-collision, collision, and post-collision stages under the hard-sphere model. Vectors are not to scale.

with  $0 \leq e_n \leq 1$ . The second parameter is the coefficient of tangential restitution,

$$e_t = -\frac{v_{d,t}^{(1)}}{v_{d,t}^{(0)}}, \quad (32)$$

with  $-1 \leq e_t \leq 1$ . The friction coefficient is the final parameter, given as

$$\mu_f = -\frac{|\hat{\mathbf{n}}_{ij} \times \mathbf{J}|}{\hat{\mathbf{n}}_{ij} \cdot \mathbf{J}}, \quad (33)$$

where  $\mu_f \geq 0$ .

Combining Eqs. (28) and (31) gives the normal components of  $\mathbf{J}$  as

$$J_n = -m_{ij}(1 + e_n)v_{d,n}^{(0)}, \quad (34)$$

where  $J_n \leq 0$  and  $m_{ij}$  is the reduced mass, given by

$$m_{ij} = \frac{m_{d,i}m_{d,j}}{m_{d,i} + m_{d,j}}. \quad (35)$$

The tangential component,  $J_t \leq 0$ , is obtained as

$$J_t = \max \left\{ \mu_f J_n, -\frac{2}{7}(1 + e_t)m_{ij}v_{d,t}^{(0)} \right\}. \quad (36)$$

In the first scenario, i.e.  $J_t = \mu_f J_n$ , a sliding collision occurs. The alternative scenario corresponds to a sticking collision. The post-collision velocities can then be computed from Eqs. (28). Note that  $e_n = 1$  and  $e_t = -1$  correspond to perfectly elastic collisions.

Following the model presented by Volkov and Tsirkunov [41], we assume that the kinetic energy lost during inelastic collisions is converted to thermal energy and absorbed by the particles. Specifically, the post-collision temperatures are computed as

$$T_{d,i}^{(1)} = T_{d,i}^{(0)} + \frac{\Delta K_{d,ij}}{2m_{d,i}c_d(T_{d,i}^{(0)})}, \quad (37a)$$

$$T_{d,j}^{(1)} = T_{d,j}^{(0)} + \frac{\Delta K_{d,ij}}{2m_{d,j}c_d(T_{d,j}^{(0)})}, \quad (37b)$$

where the change in specific heat due to the temperature rise is neglected. This is valid since the temperature change is typically small, and it simplifies the calculation of the post-collision temperatures.  $\Delta K_{d,ij} = \Delta K_{d,i} + \Delta K_{d,j}$  is the kinetic energy lost during the collision, where

$$\Delta K_{d,i} = K_{d,i}^{(0)} - K_{d,i}^{(1)}, \quad (38a)$$

$$\Delta K_{d,j} = K_{d,j}^{(0)} - K_{d,j}^{(1)}, \quad (38b)$$

with

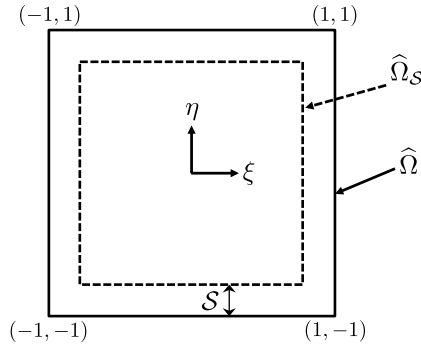


Fig. 8. Original reference element,  $\hat{\Omega}$ , and subelement,  $\hat{\Omega}_S$ .

$$K_{d,i} = \frac{1}{2}m_{d,i}|\mathbf{u}_{d,i}|^2 + \frac{1}{2}I_{d,i}|\boldsymbol{\omega}_{d,i}|^2, \quad (39a)$$

$$K_{d,j} = \frac{1}{2}m_{d,j}|\mathbf{u}_{d,j}|^2 + \frac{1}{2}I_{d,j}|\boldsymbol{\omega}_{d,j}|^2. \quad (39b)$$

### 3.5.2. Collision inspection

When a pair of particles is inspected for collision during a given time step, the following equation is solved for  $\lambda$ :

$$|\mathbf{y}_d^{n-1} + \lambda(\mathbf{y}_d^n - \mathbf{y}_d^{n-1})|^2 = (R_i + R_j)^2, \quad (40)$$

where  $\mathbf{y}_d = \mathbf{x}_{d,i} - \mathbf{x}_{d,j}$ .

If there exist two real positive values for  $\lambda$  and the smaller value,  $\lambda_1$ , is less than or equal to unity, then a collision occurs at  $t = t^{n-1} + \lambda_1 \Delta t$ , where  $\Delta t$  is the time step size. Otherwise, no collision occurs during the given time step.

### 3.5.3. Particle-pair selection algorithm

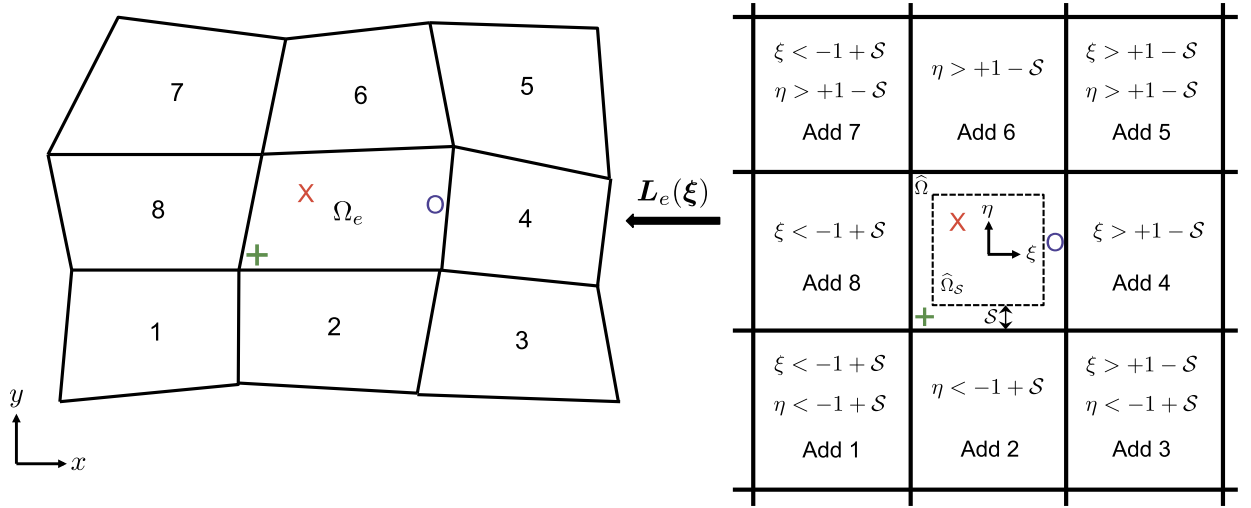
We proceed to discuss the proposed algorithm for identifying pairs of particles for collision inspection. This corresponds to the first step in the flowchart in Fig. 1, responsible for reducing the number of times Eq. (40) is solved. It is critical to decrease the quantity of collision inspections since these are often the cause for the substantial computational cost of treating particle-particle collisions. In our discussion of the algorithm, we again focus on  $q = 2$  quadrilaterals; the extension to 3D and to other shapes and orders is straightforward.

We introduce a subelement in reference space, defined as

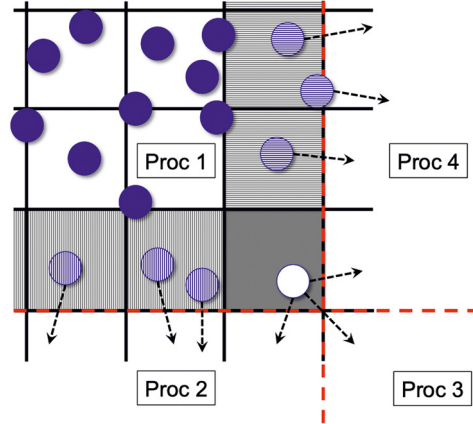
$$\hat{\Omega}_S = \left\{ \boldsymbol{\xi} = [\xi, \eta]^T \mid -1 + S \leq \xi, \eta \leq 1 - S \right\}, \quad (41)$$

where  $S$  is a parameter that controls the size of the subelement. A schematic of  $\hat{\Omega}_S$ , along with  $\hat{\Omega}$  (the bi-unit reference square), is shown in Fig. 8. For a given particle, information about its position in reference space,  $\boldsymbol{\xi}_d$ , relative to  $\hat{\Omega}_S$  is used to construct an *element proximity list*. The particle is inspected for collision with only other particles located inside the elements that are part of the list. Element proximity lists are therefore akin to the element neighbor lists described in Section 1, but the key difference is in how the element proximity lists are populated. In both cases, the host element of the particle is added to the list. However, instead of indiscriminately adding all node-sharing elements, we populate the list according to the schematic in Fig. 9. Only if the particle centroid is outside  $\hat{\Omega}_S$  (and therefore “close” to one or more faces of the host element) is a neighboring element added to the element proximity list. For example, if the particle is located at the red cross in Fig. 9, then only the host element ( $\Omega_e$ ) makes up the element proximity list. If the particle is instead located at the blue circle, then the list consists of the host element and Element 4. Finally, if the particle is located at the green plus sign, then the host element and Elements 1, 2, and 8 comprise the list. The addition of elements to the list is done at the beginning and end of the time step, i.e. based on  $\boldsymbol{\xi}_d^{n-1}$  and  $\boldsymbol{\xi}_d^n$ . With this, element proximity lists typically contain much fewer elements than element neighbor lists, resulting in a smaller number of collision inspections and, thus, lower computational cost. For the flows simulated in this work, we find  $S = 0.2$  to perform sufficiently well. Although not pursued in this work,  $S$  can also be adaptively set for each particle as a function of, for example, the particle size, particle trajectory, and element size.

We emphasize that obtaining  $\boldsymbol{\xi}_d^{n-1}$  and  $\boldsymbol{\xi}_d^n$  does not incur additional overhead since they are already computed in the particle search-locate procedure discussed in Section 3.3. Since the element-proximity-list approach directly relies on the geometric mapping, it can be easily embedded in the search-locate procedure as well. The only additional operation compared to a standard element-neighbor-list approach is evaluating whether the particle is inside or outside  $\hat{\Omega}_S$ , which is straightforward and cheap. Furthermore, the proposed algorithm is compatible with arbitrary, curved, unstructured elements. Note that for simple structured meshes, it is easy to determine the proximity of a particle to the faces of the host element by



**Fig. 9.** Procedure for adding elements to the element proximity list of a given particle based on  $\Omega_S$  and the reference position,  $\xi = [\xi, \eta]^T$ .



**Fig. 10.** Schematic for creating ghost particles in parallel simulations. The black solid lines denote element boundaries, while the dashed red lines indicate processor boundaries. The arrows indicate that a copy of the particle is to be sent to the corresponding processor.

simply computing distances in physical space, i.e. the geometric mapping and the subelement are not needed. However, the computational complexity increases for unstructured meshes and significantly more so for high-order curved elements. Specifically, in general for curved elements, an iterative procedure is required to compute the distance between a particle and a given face. Conversely, in the proposed approach, the particle-face proximity is naturally obtained by leveraging the search-locate procedure, irrespective of whether the mesh is unstructured and/or curved.

### 3.5.4. Parallelization

In parallel simulations, particles are stored in memory on the processors whose subdomains contain the respective host elements. Interparticle collisions near boundaries of different processors are handled using the standard ghost-particle approach [12,15]. For completeness, we describe it here. Specifically, if a particle is located in a *ghost element* associated with a neighboring processor, then a copy of the particle is created on the neighboring processor. In this work, an element is considered to be part of the ghost region of a neighboring processor if it shares nodes with any elements belonging to said processor. This is illustrated in Fig. 10, where the particles residing in the subdomain of Processor 1 are shown. The ghost region associated with Processor 2 is indicated by the vertical gray lines; the particles to be copied over to Processor 2 are filled with vertical blue lines. The ghost region and ghost particles associated with Processor 4 are denoted similarly, except with horizontal lines. The particle indicated by the empty circle located in the solid gray element is to be copied over to Processors 2, 3, and 4.

We note that a constraint set by the particle-pair selection algorithm discussed in Section 3.5.3 and the above parallelization strategy is that particles should not travel outside the set of node-sharing elements within a time step. However, this is typically not violated since particles are generally much smaller than the elements of the mesh and the time step of the disperse phase is often further limited by accuracy requirements, stability constraints of the temporal discretization, and/or

the time step of the carrier phase. This is also aided by the fact that high-order simulations allow for larger elements than lower-order simulations. Nevertheless, we can relax the aforementioned constraint by expanding the ghost region and by populating the element proximity list of a given particle while elements are searched as part of the search-locate procedure (see Section 3.3).

### 3.6. Numerical implementation

We conclude this section by outlining the entire solution procedure for the particle phase. This is given in Algorithm 1, where we assume that the time step of the disperse phase is equal to that of the carrier phase. If it is otherwise required to be smaller than the carrier phase time step, then substepping can be employed.

---

**Algorithm 1:** Solution procedure for particle phase at each time step.

---

```

Initialize collision list to zero collisions
for each particle do
    Interpolate state of carrier fluid to particle centroid
    Compute RHS of Eqs. (6)
    Compute reverse coupling source term (Eqs. (10a))
    Populate element proximity list based on  $\xi_d^{n-1}$  for host element from previous time step according to procedure
    in Section 3.5.3
    Advance particle in time from  $t^{n-1}$  to  $t^n$ 
    Identify host element and compute  $\xi_d^n$  according to search-locate procedure in Section 3.3
    if particle-wall collision is predicted then
        | Add collision to collision list (do not enact collision)
    end
    Add elements to element proximity list based on  $\xi_d^n$  for current host element according to procedure in
    Section 3.5.3
end
Create ghost particles
for all pairs of particles that share at least one element in their respective element proximity lists do
    Inspect particle pair for collision by solving Eq. (40)
    if collision is predicted to occur during current time step ( $0 < \lambda_1 \leq 1$ ) then
        | Add collision to collision list
    end
end
while at least one collision in collision list do
    Sort collision list in chronological order
    Advance all particles to time of first collision
    Enact first collision by solving Eqs. (28)
    Re-localize the recently collided particle(s) according to search-locate procedure in Section 3.3
    Remove from collision list all collisions involving the recently collided particle(s)
    Repopulate element proximity lists of the recently collided particle(s)
    Check for new collisions involving the recently collided particle(s) (making use of element proximity lists and
    Eq. (40))
    if new collisions predicted to occur then
        | Add collisions to collision list
    end
end
Delete ghost particles

```

---

## 4. Numerical tests

In this section, we discuss three test cases. The first one entails a kinetic theory analogy, which is used to not only verify the interparticle collision algorithm but also evaluate its robustness across different mesh topologies and its relative cost. The second test case involves a stream of particles impinging on a flat plate specimen, with erosion as the target quantity. In the third configuration, we compute hypersonic dusty flow over an entry capsule. Heat flux augmentation due to energy transfer by surface collisions is examined. In these final two cases, we investigate the effect of particle-particle collisions on the results.

**Table 1**

Input conditions for the motion of particles with a Maxwellian velocity distribution inside a 2D square domain.

$\rho_d$ (kg/m <sup>3</sup> )	$D$ (m)	$n_d$ (m <sup>-3</sup> )	$T_{\text{sys}}$ (K)
$10^{-9}$	$10^{-3}$	2000	$10^6$

#### 4.1. Case 1: Kinetic theory analogy

To test the efficiency and robustness of the proposed algorithm, we simulate monodisperse particles moving in an inviscid, quiescent fluid inside a regular box-shaped domain. Since  $\mathbf{F}$ ,  $\mathbf{T}$ , and  $\mathbf{Q}$  in Eqs. (6) are all equal to zero, particle trajectories change only as a result of particle-particle and particle-wall collisions. Particle positions are initialized at random non-overlapping locations, and particle velocities are initialized using a Maxwellian velocity distribution. The state of the carrier fluid remains fixed. Assuming that all collisions are perfectly elastic, the normal and tangential components of the impulse become

$$J_n = -2m_{ij}v_{d,n}^{(0)}, \quad (42a)$$

$$J_t = 0. \quad (42b)$$

The above setup is analogous to the motion of molecules in an ideal gas in thermodynamic equilibrium under the principles of kinetic theory. The equilibrium properties of the system can be completely determined from the following inputs: particle mass,  $m_d$ ; number density of the particles,  $n_d$ ; and “temperature” of the system, given by

$$T_{\text{sys}} = \frac{m_d \overline{|\mathbf{u}_d|^2}}{3k_b}, \quad (43)$$

where  $\overline{(\cdot)}$  denotes the mean quantity and  $k_b$  is the Boltzmann constant. The equilibrium properties can be used as target quantities to assess performance of the collision module. This was done, for example, by Sundaram and Collins [45] to test their collision routines. First, we focus on the 2D case, in which we ensure consistency of the results across various mesh topologies. Next, we move on to the 3D case, where we more thoroughly investigate prediction of equilibrium properties and compare computational cost of the element-proximity-list approach with that of the element-neighbor-list approach.

##### 4.1.1. 2D simulations

The inputs for this case are provided in Table 1. The 2D Maxwellian speed distribution is given by

$$f_M(|\mathbf{u}_d|) = 2\pi |\mathbf{u}_d| \left( \frac{m_d}{2\pi k_b T_{\text{sys}}} \right) \exp \left( -\frac{m_d |\mathbf{u}_d|^2}{2k_b T_{\text{sys}}} \right). \quad (44)$$

We employ three different meshes: a structured quadrilateral mesh, an unstructured quadrilateral mesh, and an unstructured triangular mesh. These are displayed in Fig. 11. The interparticle collision frequency according to kinetic theory is given by

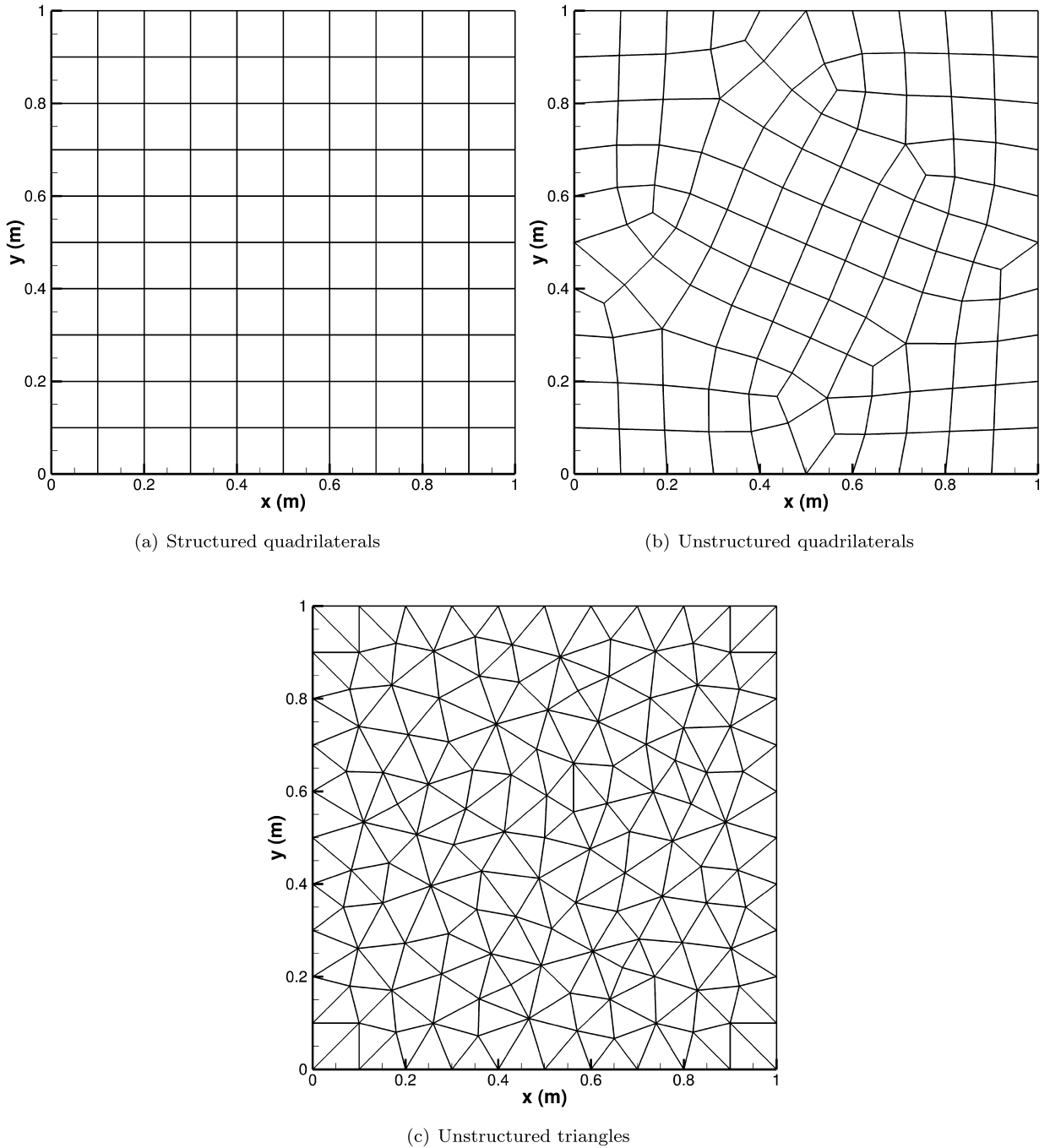
$$Z_d = Dn_d^2 \sqrt{\frac{\pi k_b T_{\text{sys}}}{m_d}}, \quad (45)$$

which, under the conditions in Table 1, gives a collision frequency of  $940.01 \text{ s}^{-1}$ . The temporal variations of  $f_d$  obtained from the simulations are shown in Fig. 12. The horizontal black line represents the expected collision frequency from kinetic theory. The exact same collision frequencies are observed for all three meshes, demonstrating the applicability of the proposed algorithm to different mesh topologies. The results also remain consistent when the element-neighbor-list approach is used and when the number of processors used for the simulation is changed, verifying the ghost-particle strategy discussed in Section 3.5.4. The temporally averaged collision frequency from the DG simulations is  $938.33 \text{ s}^{-1}$ . Running for longer times would yield closer agreement with the theoretical value, as will be observed in the next section. However, the results among the three meshes will eventually begin to slightly deviate from each other as a consequence of round-off error, which is magnified due to the iterative nature of the search-locate procedure (Section 3.3).

##### 4.1.2. 3D simulations

Table 2 lists the input conditions for this problem. The domain is a cube with a side length of one meter. We employ a mesh with two hexahedral elements in each direction, for a total of eight elements. The 3D Maxwellian speed distribution is written as



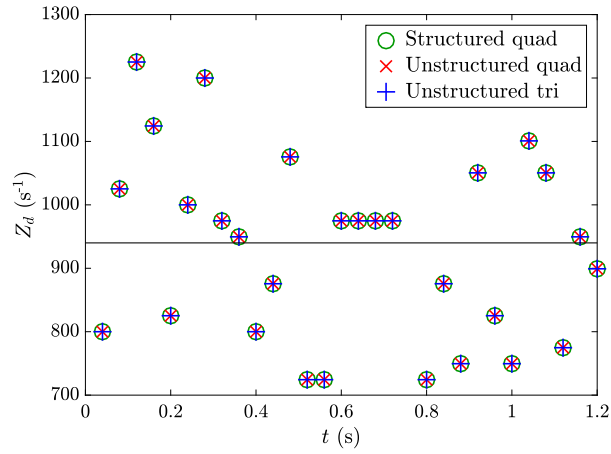


**Fig. 11.** Meshes employed for simulating the motion of particles with a Maxwellian velocity distribution inside a 2D square domain.

$$f_M(|\mathbf{u}_d|) = 4\pi |\mathbf{u}_d|^2 \left( \frac{m_d}{2\pi k_b T_{\text{sys}}} \right)^{3/2} \exp \left( -\frac{m_d |\mathbf{u}_d|^2}{2k_b T_{\text{sys}}} \right), \quad (46)$$

which is displayed in Fig. 13 for the conditions in Table 2. A histogram of the particle speeds from the end of the simulation is also shown, demonstrating consistency between the numerical and theoretical speed distributions. According to kinetic theory, the particle-particle collision frequency is given by

$$Z_d = \frac{\pi}{2} D^2 n_d^2 \sqrt{\frac{16k_b T_{\text{sys}}}{\pi m_d}}, \quad (47)$$

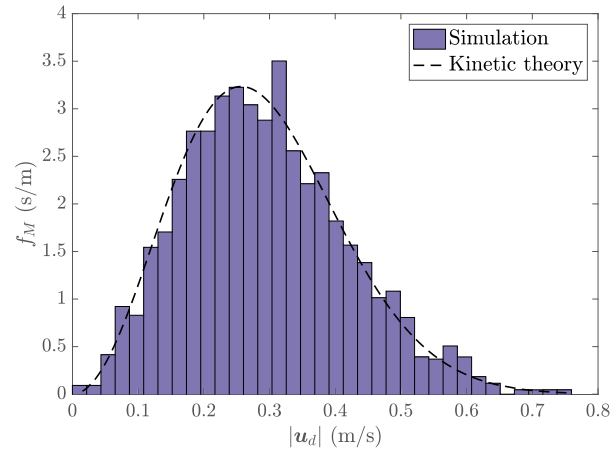


**Fig. 12.** Temporal variations of interparticle collision frequency obtained with the three meshes in Fig. 11. The horizontal black line represents the expected collision frequency from kinetic theory.

**Table 2**

Input conditions for the motion of particles with a Maxwellian velocity distribution inside a 3D cube domain.

$\rho_d$ (kg/m <sup>3</sup> )	$D$ (m)	$n_d$ (m <sup>-3</sup> )	$T_{\text{sys}}$ (K)
$10^{-9}$	0.02	2000	$10^7$



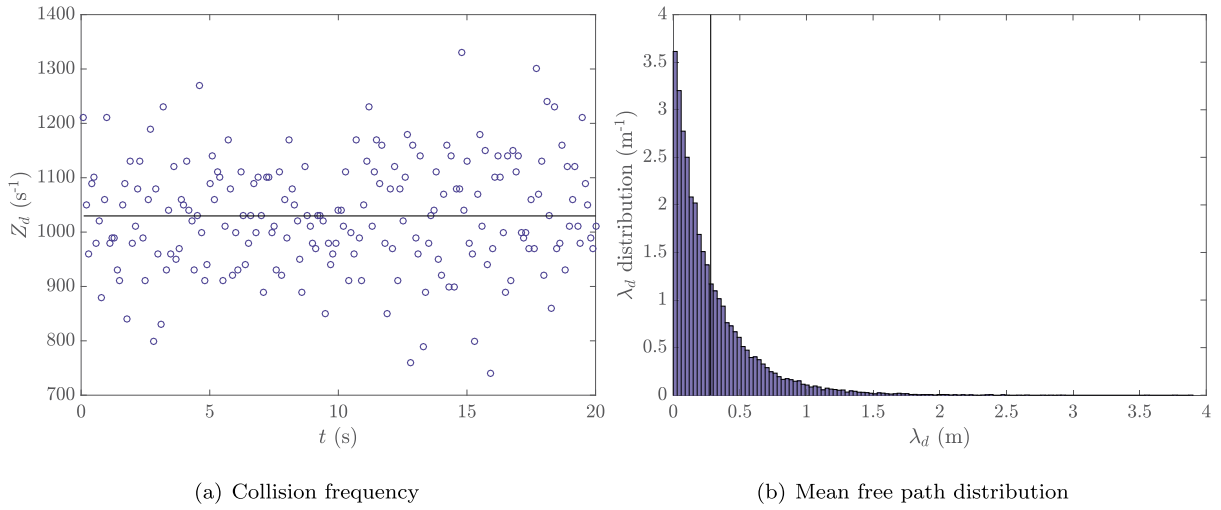
**Fig. 13.** Distribution of particle speeds from both the simulation and the kinetic theory of ideal gases.

and the mean free path, which is the average distance traveled by a particle before a collision occurs, is given by

$$\lambda_d = \frac{1}{\sqrt{2}\pi D^2 n_d}. \quad (48)$$

Fig. 14 shows the temporal evolution of collision frequency and the mean free path distribution throughout the simulation. The horizontal solid line in Fig. 14(a) represents the theoretical collision frequency, 1029.6 collisions per second. The temporally averaged collision frequency from the simulation is 1029.9 collision per second. Similarly, the vertical solid line in Fig. 14(b) represents the theoretical mean free path, 0.281 m, with the simulation predicting  $\lambda_d = 0.287$  m. Overall, there is very good agreement between the simulation and kinetic theory.

Next, we compare the computational cost between the proposed element-proximity-list approach and the element-neighbor-list approach. On the aforementioned  $2 \times 2 \times 2$  mesh, the former is approximately eleven times faster than the latter. We note that a major reason for the considerable speedup is that within a given time step, many of the particles remain in one element and only traverse a relatively small portion of the element. Specifically, according to kinetic theory, the most probable speed is



**Fig. 14.** Temporal evolution of collision frequency and mean free path distribution for the motion of particles with a Maxwellian velocity distribution inside a 3D cube domain. The black solid lines represent the expected values from kinetic theory.

**Table 3**

Disperse phase conditions for simulating a particle stream impacting a flat plate specimen. The particle material is SiO<sub>2</sub>.

$\rho_d$ (kg/m <sup>3</sup> )	$D$ (μ m)	$e_n$	$e_t$	$\mu_f$
2600	49	0.1	−0.1	0.4

$$u_{d,mp} = \sqrt{\frac{2k_b T_{sys}}{m_d}}, \quad (49)$$

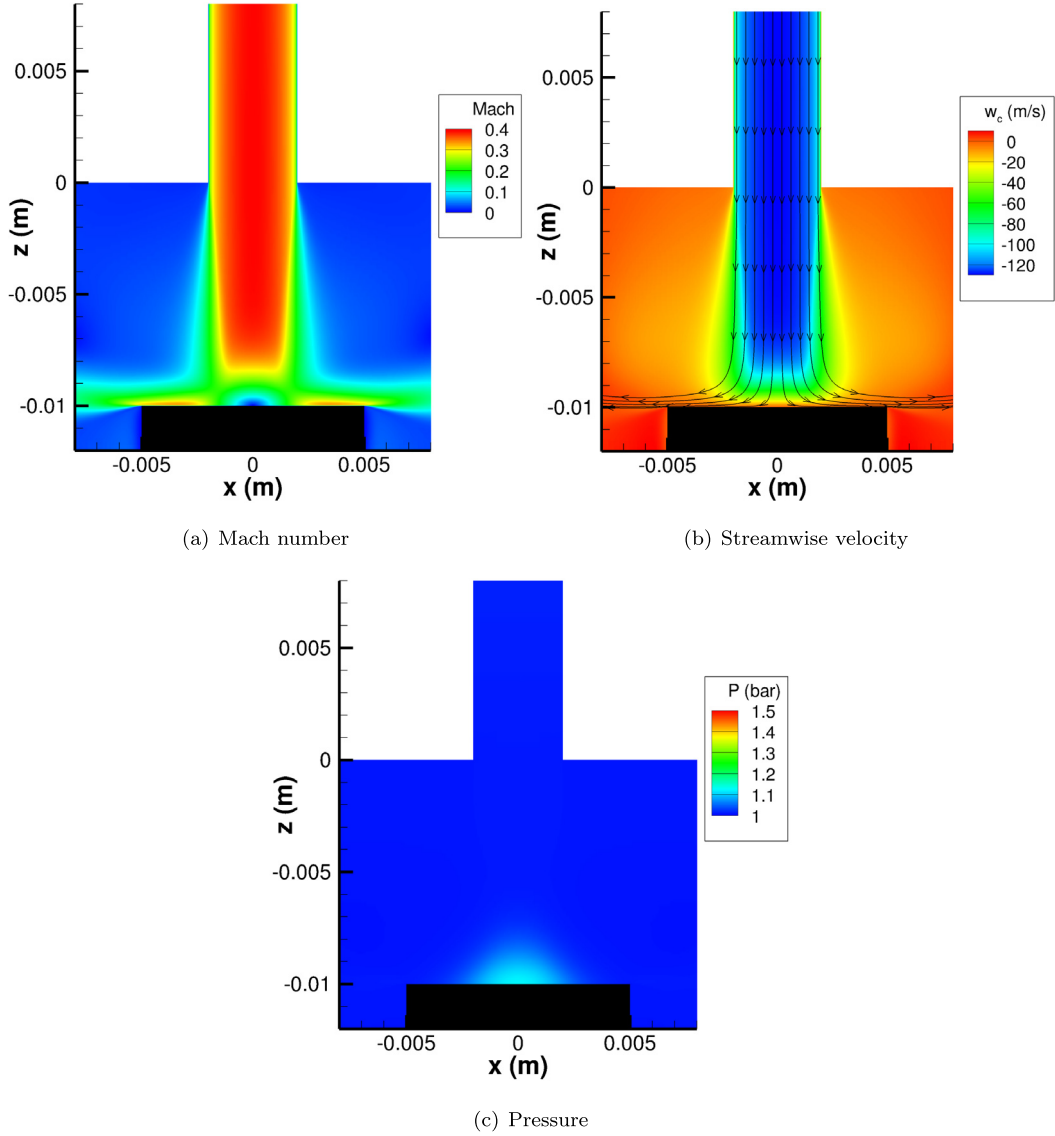
which is 0.257 m/s for the conditions in Table 2. With a time step of  $\Delta t = 0.1$  s, the distance traveled by a particle with speed  $u_{d,mp}$  is evidently less than the element side length of 0.5 m. Furthermore, the particle size is significantly smaller than the element size. Despite this, the element-neighbor-list approach adds all eight elements to the element list of every particle, yielding many superfluous collision inspections. Conversely, the element-proximity-list approach is much more conservative in adding elements to a given particle's element list, resulting in notably fewer collision inspections. In light of this, we repeat the above calculation with a mesh with  $4 \times 4 \times 4$  hexahedral elements. The volume of influence swept by a particle during a time step is larger relative to the element size than for the previous mesh. In this case, the proposed strategy is approximately five times faster than the element-neighbor-list approach. Nevertheless, the speedup is still significant. In large-scale simulations of compressible multiphase flows, there will be many cases in which the volume of influence swept by a particle during a time step is much smaller than the host element. We note that handling interparticle collisions can often be the most expensive step in such simulations, especially for high particle concentrations.

#### 4.2. Case 2: Impingement of particle stream on a flat plate

In this problem, we simulate a subsonic stream of particles accelerating through a nozzle and impacting a flat plate specimen. This is relevant to sandblasting and shot peening applications. Specifically, we simulate one of the experiments performed by Oka et al. [46]. The target quantity is erosion of the specimen. The nozzle diameter is 4 mm, and its length is 1.5 m. The specimen is a square prism with a cross-sectional area of 100 mm<sup>2</sup> and a depth of 2 mm, positioned 10 mm downstream of the nozzle and oriented normal to the flow.

The disperse phase is comprised of SiO<sub>2</sub> particles with  $D = 49$  μ m. Key parameters of the disperse phase are listed in Table 3. The coefficients of restitution for interparticle collisions are chosen based on the high collisional velocities observed between the nozzle exit and the specimen [47,48], as well as the angular shapes of the particles [49], as reported by Oka et al. [46]. The friction coefficient is obtained from Ref. [50]. Four cases with varying particle mass fluxes are simulated:  $\dot{m}_d = 27.9, 38.5, 47.8, \text{ and } 81.1$  kg/m<sup>2</sup>/s. The aim here is to evaluate the effect of particle mass flux on the average erosion rate of the specimen. The erosion damage (mm<sup>3</sup>/kg) is computed as [51,52]

$$D_e = g(\alpha) D_{e,90}, \quad (50)$$



**Fig. 15.**  $y = 0$  slice of pure-gas solution of subsonic nozzle flow impinging on a flat plate specimen zoomed-in on the nozzle exit and specimen (colored in black). Streamlines are included in (b).

which denotes the volume of material removed per unit mass of particles.  $D_{e,90}$  is the erosion damage for a normal impact, and  $g(\alpha)$  accounts for the dependence on the impact angle,  $\alpha$ , calculated as

$$D_{e,90} = K(Hv)^{k_1} \left( \frac{|u_d^{(0)}|}{u_{\text{ref}}} \right)^{k_2} \left( \frac{D}{D_{\text{ref}}} \right)^{k_3}, \quad (51a)$$

$$g(\alpha) = (\sin \alpha)^{n_1} [1 + Hv(1 - \sin \alpha)^{n_2}], \quad (51b)$$

where  $Hv$  is the Vickers hardness of the material and the remaining parameters depend on other properties of the specimen and the particles. The average erosion rate is computed as

$$\bar{D}_e = \sum_{i=1}^{N_{\text{coll}}} \frac{V_{e,i}}{m_{d,\text{tot}}} = \sum_{i=1}^{N_{\text{coll}}} \frac{D_{e,i} m_{d,i}}{m_{d,\text{tot}}}, \quad (52)$$

where  $V_e$  is the volume removed from an impact,  $m_{d,\text{tot}}$  is the total particle mass consumed, and the sum is over all particle-specimen collisions. The erosion rates presented here are normalized by the erosion rate for the lowest mass flux,

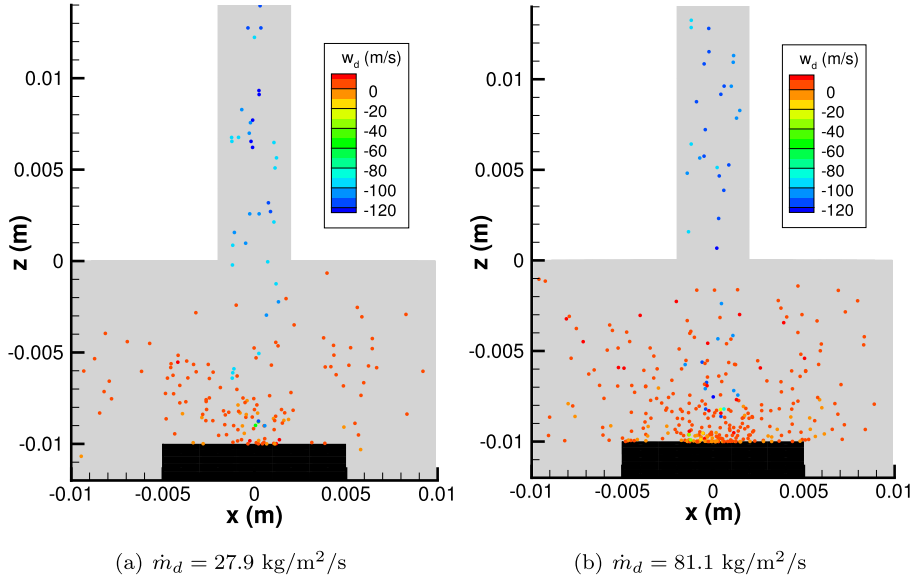


Fig. 16. Locations of particles, colored by streamwise velocity, for the lowest and highest particle fluxes, 27.9 and 81.1 kg/m<sup>2</sup>/s.

both for consistency with the results presented by Oka et al. [46] and to account for the fact that certain parameters in the erosion model are not calibrated specifically for the conditions considered. These are denoted  $\bar{D}_{e,rel}$ .

The computational domain consists of the entire nozzle and a region outside the nozzle exit containing the specimen. We perform  $p = 2$  and  $p = 3$  simulations on a mesh with 13,000 hexahedral elements. The Spalart-Allmaras model is employed to account for turbulence [53,54]. Boundary conditions at the nozzle inlet are based on Riemann invariants. The prescribed total temperature and total pressure are 330 K and 1.85 bar, respectively. These conditions yield an average impact velocity of approximately 110 m/s for a one-way coupled flow, as noted by Oka et al. [46]. The nozzle and specimen surfaces are no-slip adiabatic walls, and atmospheric pressure is imposed at the outflow. A pure-gas (particle-free) steady-state solution is first computed. Particles are then continuously injected at random locations along the nozzle inlet, and the simulation is computed in a time-accurate manner until quasi-steady-state conditions are attained. Second-order backward differencing and the third-order Adams-Bashforth are used to evolve the carrier phase and disperse phase in time, respectively.

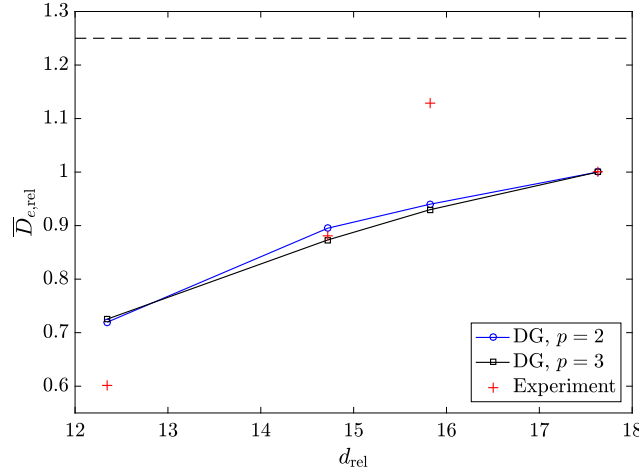
Fig. 15 displays the  $y = 0$  slices of the pure-gas distributions of Mach number, streamwise velocity ( $w_c$ ), and pressure in the region surrounding the nozzle exit and specimen. The air accelerates in the nozzle to a moderate Mach number and a high streamwise velocity. The pressure in the vicinity of the stagnation point is lower than the total pressure prescribed at the nozzle inlet due to frictional losses inside the nozzle. The flow is deflected parallel to the specimen prior to exiting the domain, as indicated by the streamlines in Fig. 15(b).

Instantaneous particle locations for the lowest and highest particle mass fluxes are given in Fig. 16. The particles are colored by streamwise velocity. For the lowest mass flux, there is slight accumulation of particles in the region near the surface of the specimen directly under the nozzle. For the highest mass flux, this accumulation near the stagnation point is significantly increased, which has important implications for the average erosion rate (discussed next). Particles exiting the nozzle collide with other particles or the specimen wall, in which case they rebound with positive streamwise velocities.

The predicted erosion rates, defined as the ratio of the volume removed to the mass of particles consumed, for all cases are given in Fig. 17. These erosion rates are displayed as a function of the relative interparticle distance, defined by

$$d_{rel} = \left( \frac{U_{imp} \pi R_n^2 \dot{m}_d}{\dot{m}_d D^3} \right)^{1/3}, \quad (53)$$

where  $U_{imp}$  is the average impact velocity as discussed above (110 m/s) and  $R_n$  is the nozzle radius. Note that a higher mass flux yields a lower relative interparticle distance. The erosion rate decreases with mass flux. This is largely because of the accumulation of particles near the surface of the specimen, as shown in Fig. 16. These particles are likely to be struck by high-velocity particles exiting the nozzle. Due to the inelasticity of the particle-particle collisions, the average impact velocities of particles hitting the specimen are reduced, resulting in less erosion. This attenuation of the erosion rate is magnified for higher particle fluxes because of greater interparticle collision frequencies. The consistency between the  $p = 2$  and  $p = 3$  results indicates grid convergence. Good agreement between the DG predictions and the experimental results is observed. The horizontal dashed line represents the erosion rate for a one-way coupled calculation, approximately 25% greater than that for  $\dot{m}_d = 27.9$  kg/m<sup>2</sup>/s. Note that with only one-way coupling, this erosion rate is independent of the particle loading since there are no interactions between particles or between the carrier and disperse phases. Physically,



**Fig. 17.** Variation of relative erosion rate with relative interparticle distance for a particle stream impinging on a flat plate specimen. A higher interparticle distance corresponds to a lower particle flux.

the one-way coupled case corresponds to a very low particle flux, such that both particle-particle collisions and the reverse coupling of particles to the carrier gas can be neglected.

Figs. 18(a) and 18(b) show the locations of all collisions between particles and the specimen surface for  $\dot{m}_d = 27.9$  and  $81.1 \text{ kg/m}^2/\text{s}$ , respectively, over a time period of 0.002 seconds. Each impact location is colored by its corresponding value of  $V_{e,rel}$ , the volume removed normalized by the maximum volume removed. The surface mesh of the specimen is superimposed. In both cases, the majority of impacts occurs in the circular region with radius 0.002 meters directly under the nozzle exit. However, for  $\dot{m}_d = 27.9 \text{ kg/m}^2/\text{s}$ , a significant portion of these impacts is associated with high  $V_e$ . On the other hand, for  $\dot{m}_d = 81.1 \text{ kg/m}^2/\text{s}$ ,  $V_e$  is noticeably attenuated as a result of the increase in inelastic particle-particle collisions above the specimen. Nevertheless, both results are consistent with the observation by Oka et al. [46] that over 90% of the total damage is concentrated directly under the nozzle exit. Specifically, 98% and 95% of the total damage is constrained to said region for  $\dot{m}_d = 27.9 \text{ kg/m}^2/\text{s}$  and  $\dot{m}_d = 81.1 \text{ kg/m}^2/\text{s}$ , respectively.

For comparison, the impact locations for a one-way coupled case are given in Fig. 18(c). The particle flux here is  $81.1 \text{ kg/m}^2/\text{s}$ ; however, as previously discussed, without four-way coupling, the particle loading affects only the quantity of impacts, not the  $V_e$  distribution or the average erosion rate. Here, the highly erosive impacts are heavily concentrated within a radius of approximately 0.0015 meters. Over 99% of the total damage is constrained to the circular region directly under the nozzle exit. This is in stark contrast with the four-way coupled cases, in which the proportion of high-erosion impacts is lower and the  $V_w$  distribution is more isotropic. This is further illustrated in Fig. 19, which gives the variation of  $\bar{V}_{e,rel}$  with  $r$ , where  $r$  is the distance from the center of the specimen. To compute  $\bar{V}_{e,rel}$ , the circumscribed circle of the specimen surface is uniformly divided into annuli. For the  $j$ th annulus, with outer radius  $r_j$ ,  $\bar{V}_{e,rel}$  computed as

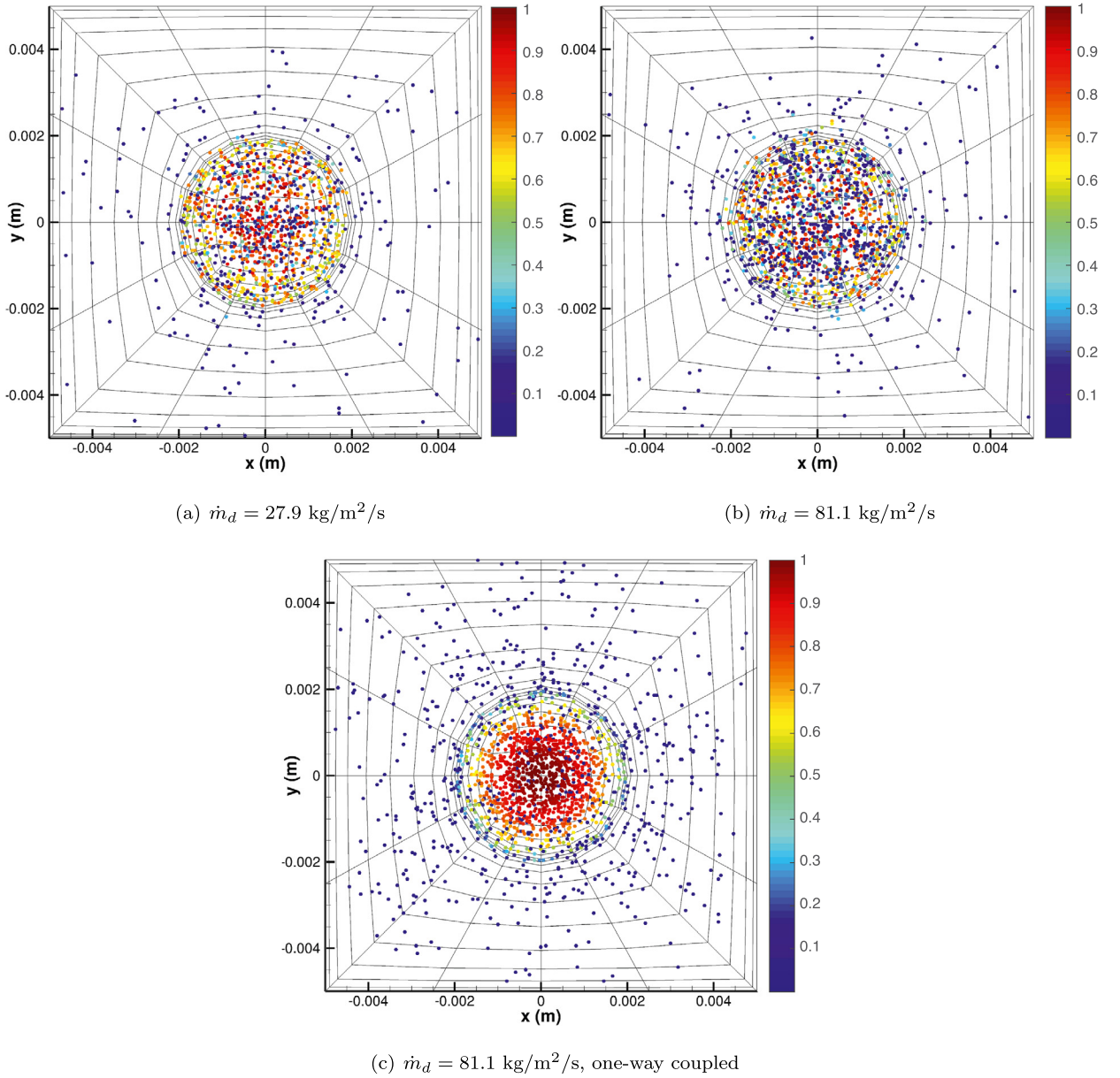
$$\bar{V}_{e,rel}^j = \sum_{i=1}^{N_{coll}^j} \frac{V_{e,rel}^j}{N_{coll}^j}, \quad (54)$$

where the sum is over the collisions inside the annulus. For the one-way coupled case,  $\bar{V}_{e,rel}$  decreases smoothly and monotonically up to  $r \approx 2.3 \text{ mm}$ . It then remains nearly zero for greater  $r$ . With consideration of interparticle collisions,  $\bar{V}_{e,rel}$  is more evenly distributed over the specimen. A higher mass flux further attenuates  $\bar{V}_{e,rel}$ . Running for longer times would likely yield a smoother  $\bar{V}_{e,rel}$  profile for  $\dot{m}_d = 81.1 \text{ kg/m}^2/\text{s}$ . Note that since the arithmetic mean of  $V_{e,rel}$  is calculated, the relative impact frequencies inside each annulus are not taken into account.

Overall, the results in this section demonstrate the ability of the proposed particle collision algorithm, combined with the DG method, to accurately predict a complex flow with relevant engineering applications.

#### 4.3. Case 3: High-speed dusty flow over an entry capsule

In this section, we investigate supersonic/hypersonic dusty flows over the forebody of an entry capsule at Mars atmospheric conditions. The main goal here is to assess the importance of accounting for particle-particle collisions for these types of flows. We target four trajectory points from the ExoMars Schiaparelli mission [55], detailed in Table 4. The freestream Mach numbers range from 3.55 to 11.7. For simplicity, the angle of attack is set to zero. The capsule geometry is slightly different from the Schiaparelli geometry. As roughly approximated from Ref. [56], the wall temperatures for the S2, S3, S5, and S7 points are set to 600, 600, 500, and 400 K, respectively.  $\text{CO}_2$  is used as the working gas.  $\text{SiO}_2$  is selected to represent the dust material, which was also done in the studies by Ozawa et al. [57] and Majid et al. [58].



**Fig. 18.** All impact locations on the specimen, colored by  $V_{e,\text{rel}}$ , for the lowest and highest particle fluxes, 27.9 and 81.1  $\text{kg/m}^2/\text{s}$ , over a period of 0.002 seconds. One-way coupled results are also shown. The surface mesh is superimposed. In (a), 25% of total impacts are shown; in (b) and (c), 10% of total impacts are shown.

**Table 4**

Freestream gas conditions for high-speed dusty flow over an entry capsule, taken from the trajectory of the ExoMars Schiaparelli vehicle [55].

Trajectory point	Mach	$\rho \times 10^3 \text{ (kg/m}^3\text{)}$	$P \text{ (Pa)}$	$T \text{ (K)}$
S2	11.7	1.54	56.6	192
S3	8.97	1.98	74.1	195
S5	5.43	2.96	114	202
S7	3.55	4.08	164	210



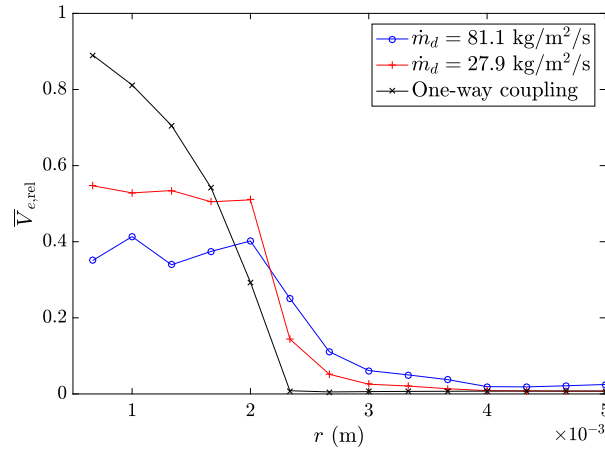


Fig. 19. Variation of  $\bar{V}_{e,rel}$  with  $r$  for low particle flux, high particle flux, and one-way coupling.

Dust particles suspended in the Mars atmosphere can influence the flow field as well as the thermal protection system of the capsule. Particles cross the shock at high velocities and low temperatures. This strong nonequilibrium between the two phases results in complex interphase momentum and energy transfer. For instance, the deceleration of particles in the shock layer causes an increase in the momentum and kinetic energy of the carrier gas. Additionally, upon entering the shock layer, the cold particles drain thermal energy from the flow. Large particles may not deviate appreciably from their initial trajectories or freestream temperatures; however, smaller particles can rise in temperature significantly and then deposit thermal energy to the colder boundary layer, augmenting surface heat fluxes. Extremely small particles equilibrate with the flow rapidly and can be advected by the flow past the capsule, missing the boundary layer completely. These complex interphase interactions were investigated in detail in Refs. [21] and [34], with a focus on relatively small particles whose velocities and temperatures change significantly in the shock layer. Using the physical model of the disperse phase described in Section 3.1 (excluding particle rotations and collisions with other particles), we obtained good agreement with the experiments performed by Vasilevskii et al. [59,60] of hypersonic dusty flow over a sphere. In this problem, we also consider large particles that strike the surface of the body at high speeds. These high-energy impacts can not only induce erosion of the heat shield [61,62] but also increase surface heat fluxes. In this case, the heating augmentation is expected to be primarily due to kinetic energy lost from inelastic particle-wall collisions, as opposed to momentum and energy exchange between the carrier and disperse phases in the shock layer. Interparticle collisions are then more likely to be important since particles striking the body at high energies rebound at very low velocities. This accumulation of particles in the boundary layer can then shield the surface from subsequent incoming particles, attenuating the impact energy of the particle-wall collisions. This is akin to the phenomenon discussed in Section 4.2. Volkov et al. [63] found particle-particle collisions to be important in the simulation of a supersonic blunt-body flow. Here, we use heating augmentation induced by surface collisions as a measure of the importance of accounting for particle-particle collisions as a function of freestream conditions and particle loading in the context of Mars atmospheric entry. The qualitative conclusions drawn from these results regarding the significance of interparticle collisions can be similarly extended to heat shield erosion, which is also directly related to particle-wall collisional dynamics.

The results in this section are obtained with  $p = 2$  polynomials on a  $q = 2$  mesh with approximately 55,000 3D hexahedral elements. The elements near the capsule wall are of high aspect ratios. No significant changes are observed with  $p = 3$ . Freestream conditions are imposed at the inflow, the capsule surface is an isothermal no-slip wall, and extrapolation is employed at the outflow. To deal with the bow shock, intra-element solution variations are used for shock detection and smooth artificial viscosity is employed for stabilization [29]. A steady-state pure-gas solution is first computed, followed by continuous injection of particles at random locations along a cut plane upstream of the shock. The particles are initialized in equilibrium with the freestream carrier flow. The back-coupling of particles to the carrier gas is ignored, not only because of our focus on particle-wall collisional energy transfer but also to circumvent numerical instabilities caused by the reverse coupling of larger particles. As previously discussed, smooth projection kernels [35], instead of the delta functions in Eqs. (10), can aid with such instabilities.

Fig. 20 displays the pure-gas temperature fields for all trajectory points in Table 4. In all cases, the bow shock is smoothly captured and free from apparent oscillations. As the Mach number increases, the temperature jump across the shock increases and the shock is closer to the capsule. The slight smearing of the shock is due to the use of artificial viscosity for stabilization. We note that for the S2 trajectory point, thermal nonequilibrium and dissociation are likely to become important given the high shock layer temperature, despite our assumption of a nonreacting, thermally perfect gas. Furthermore, the post-shock temperature exceeds the vaporization temperature of  $\text{SiO}_2$  (at the given post-shock pressure) [62], inducing phase change of small dust particles that quickly attain equilibrium with the carrier gas. However, for the purpose of evaluating the importance of accounting for interparticle collisions, we do not expect the above simplifications to significantly

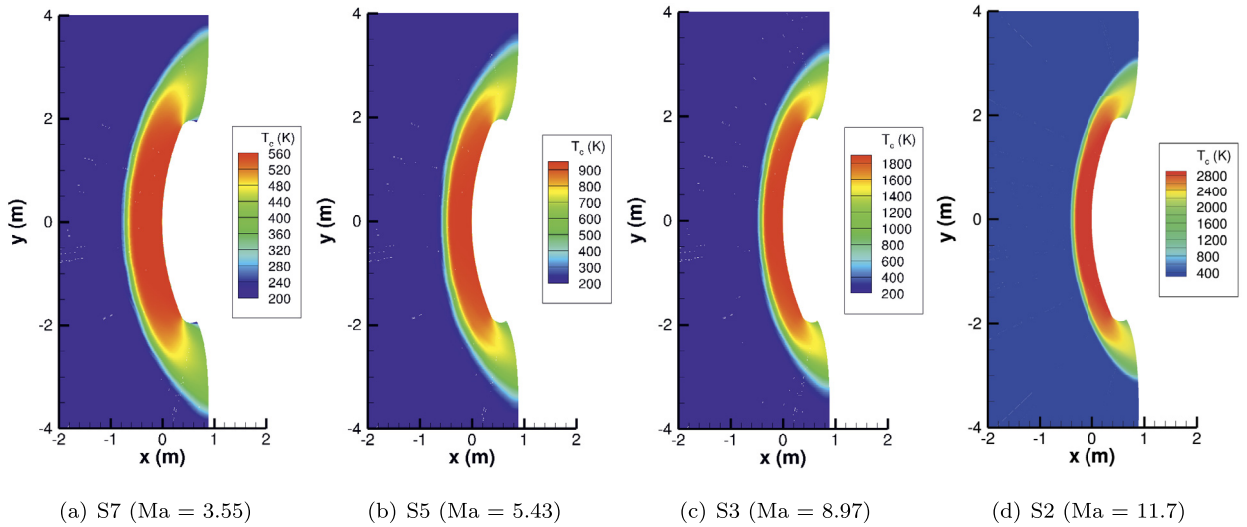


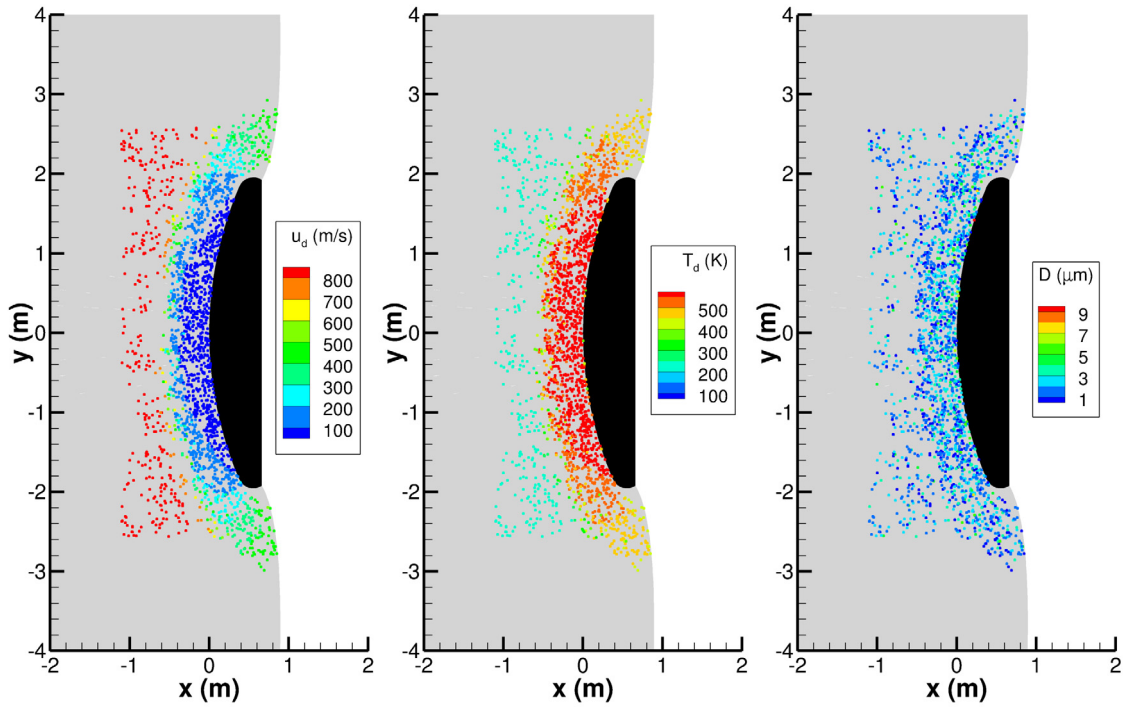
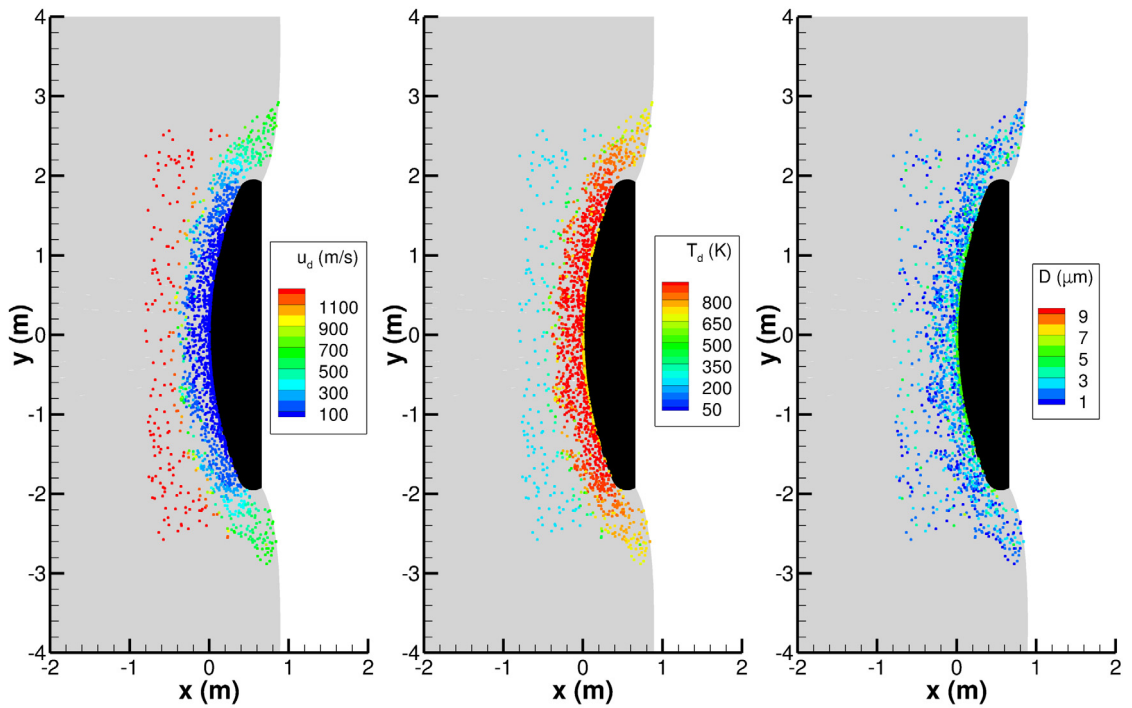
Fig. 20. Pure-gas temperature fields for supersonic/hypersonic flows over an entry capsule with flow conditions defined in Table 4.

change the results, especially since larger particles, whose high translational and thermal inertias reduce departure from their freestream trajectories and temperatures, are primarily responsible for surface collisional energy transfer.

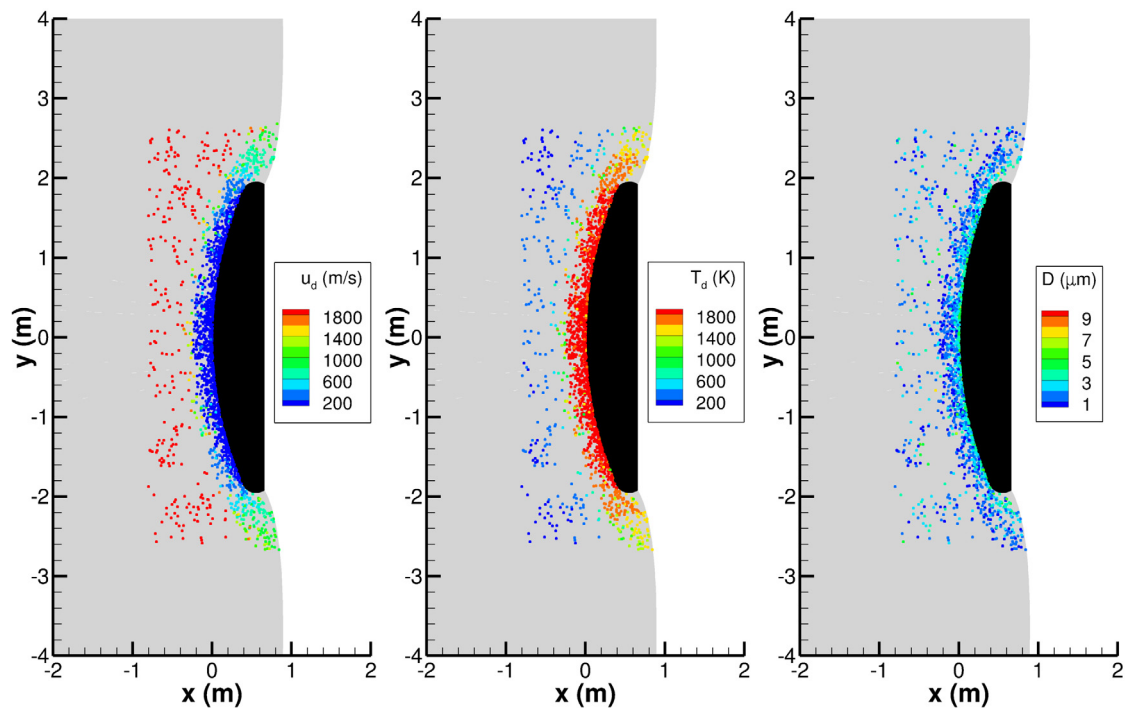
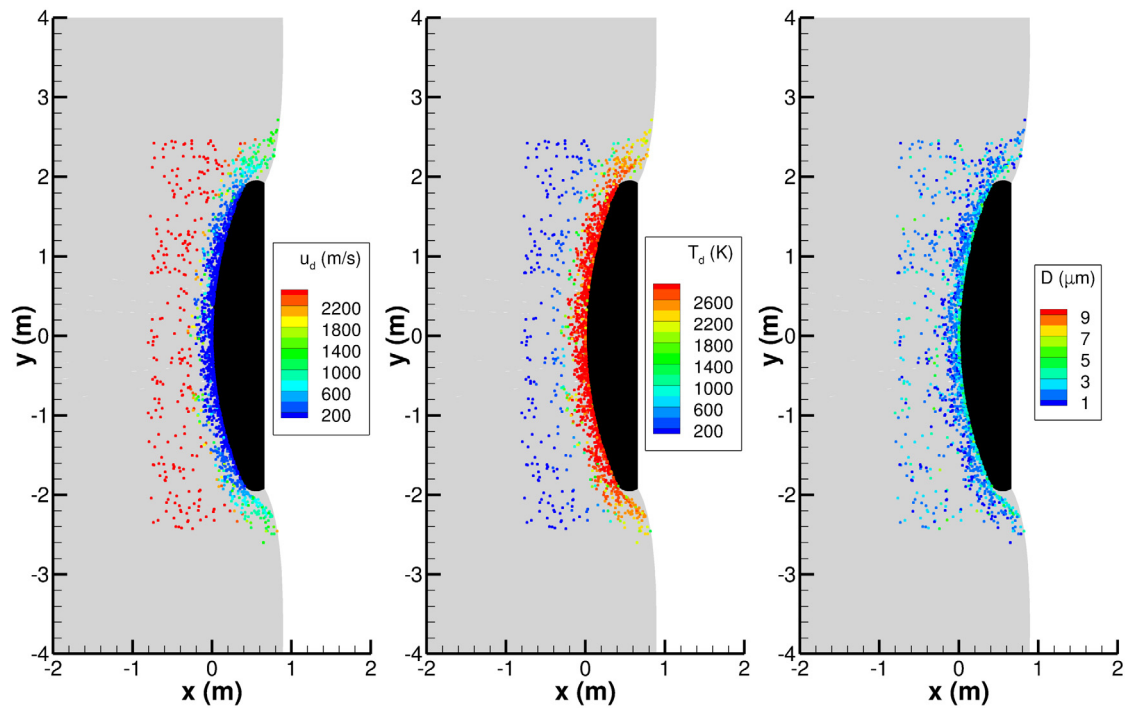
For particle size, we employ the modified gamma distribution by Tomasko et al. [64], which gives a geometric cross-section-weighted mean particle diameter of  $3.2 \mu\text{m}$ . We first consider a baseline mass loading ratio, denoted  $\beta$ , defined as the ratio of the mass flux of the disperse phase to that of the carrier phase, of 0.01%, as done by Papadopoulos [65]. Note that this is slightly higher than the mass loading ratio of 0.0069% corresponding to the July 2007 global dust storm, which can yield non-negligible erosion of the heat shield [62]. As will be discussed later in this section, at a 0.01% mass loading ratio, the surface collisional energy transfer is very small and particle-particle collisions are insignificant. Therefore, to determine approximately at which point particle-particle collisions become important, we also consider mass loading ratios of 1% and 5%.

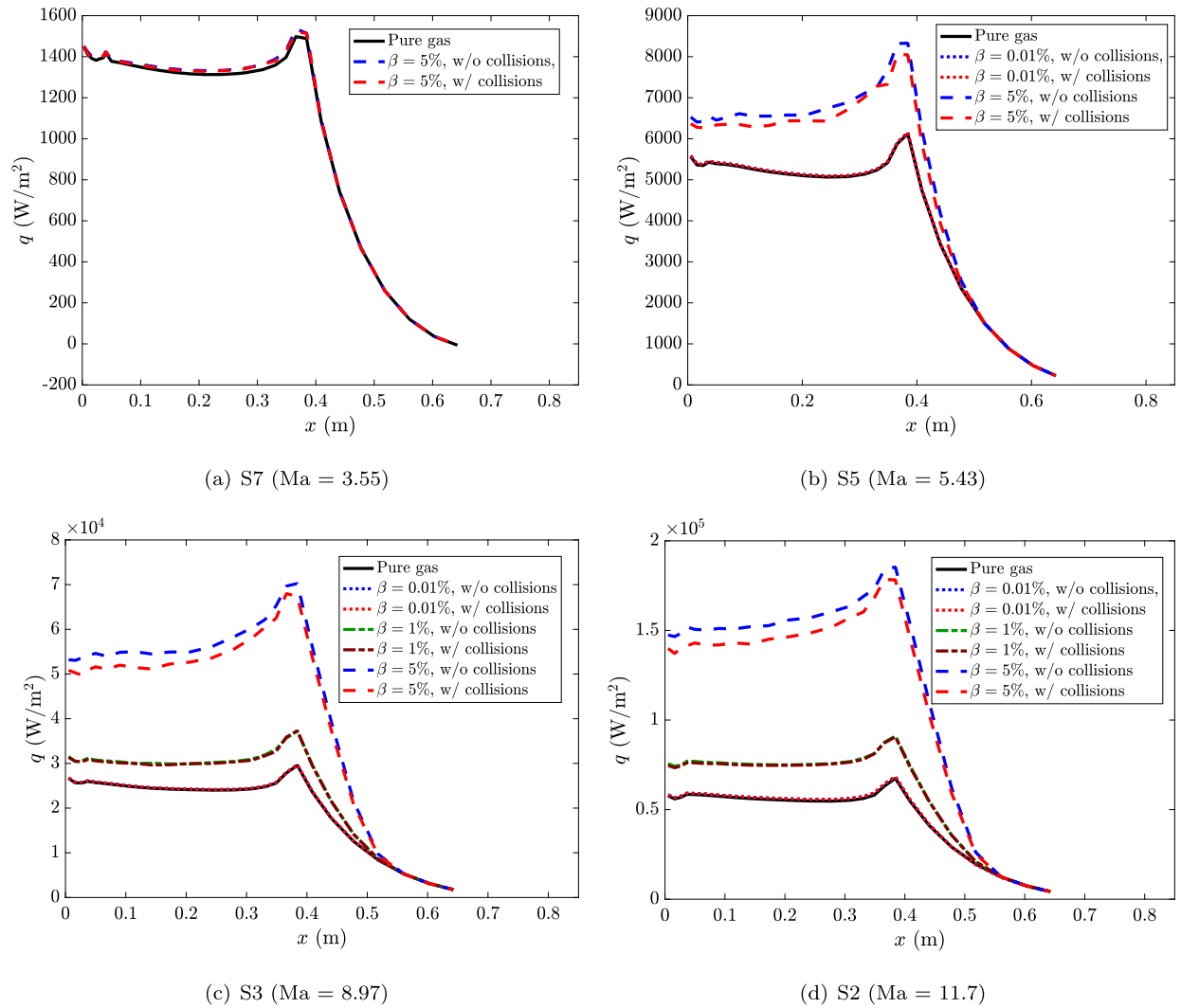
Fig. 21 shows instantaneous particle locations along the  $xy$ -plane after a quasi-steady-state is reached. The particles are colored by their streamwise velocities (left), temperatures (middle), and diameters (right). In all cases, the overall  $u_d$  and  $T_d$  distributions mirror the corresponding carrier gas distributions, although some delay in the particles equilibrating with the flow can be observed, particularly for larger particles. For the lower Mach numbers, nearly all particles close to the capsule surface have equilibrated with the carrier gas; however, for the higher Mach numbers, a greater proportion of particles remains close to the freestream conditions. One major reason is that at the lower Mach numbers, particles are initialized at lower speeds and therefore have less momentum to transfer to the shock layer. Furthermore, since the shock is located further away from the capsule, there is a longer residence time for particles to approach the velocity and temperature of the flow. Especially for the S7 trajectory point, there is very little accumulation of particles near the surface since they are largely advected away by the carrier gas before they can reach the capsule surface. This is also the reason for the relatively uniform spatial distribution of the particle diameters for the S7 and S5 trajectory points. Conversely, for the S3 and S2 trajectory points, larger particles are generally located closer to the capsule since they have sufficient inertia to resist being displaced by the flow. After colliding with the capsule wall, they are reflected at low velocities and accumulate in the boundary layer. This has significant implications for the importance of particle-particle collisions, discussed next.

Fig. 22 shows the surface heat flux profiles, averaged azimuthally and temporally, for the four trajectory points. Results of calculations at various mass loading ratios with and without particle-particle collisions are given. The pure-gas results represent the surface heating in the absence of dust particles. The dusty-gas results account for the addition of particle-wall collisional energy transfer. For the S7 trajectory point, only the  $\beta = 5\%$  prediction is shown. As previously explained, the heating augmentation due to particle-wall collisions is negligible since the majority of particles equilibrate with the carrier gas before reaching the boundary layer. For the S5 trajectory point, the heat flux amplification at  $\beta = 5\%$  is larger, and there is a modest influence of taking into account particle-particle collisions. At the S3 and S2 trajectory points, the dusty-gas heat flux at  $\beta = 5\%$  is significant, and noticeable attenuation of the heating augmentation is observed due to particle-particle collisions. This is a direct result of the accumulation of particles near the capsule surface, inducing the shielding effect described in the beginning of this section. At the intermediate mass loading ratio of 1%, the heat flux augmentation is reduced, with negligible influence by interparticle collisions. For  $\beta = 0.01\%$  at all trajectory points, the surface collisional energy transfer is very small. In general, as the Mach number and mass loading ratio increase, both collisional energy transfer and the effect of interparticle collisions become more significant. This is due to higher particle velocities, the thinner shock layer, and greater frequency of particle-wall and particle-particle collisions.

(a) S7 ( $Ma = 3.55$ )(b) S5 ( $Ma = 5.43$ )

**Fig. 21.** Instantaneous particle locations, indicated by filled circles, along the  $xy$ -plane for the four trajectory points described in Table 4. The particles are colored by their streamwise velocities (left), temperatures (middle), and diameters (right). The capsule forebody is colored black, and the computational domain is colored gray.

(c) S3 ( $Ma = 8.97$ )(d) S2 ( $Ma = 11.7$ )**Fig. 21.** (continued)



**Fig. 22.** Pure-gas and dusty-gas surface heat flux profiles, averaged azimuthally and temporally, for the four trajectory points described in Table 4. Results for computations at various mass loading ratios with and without particle-particle collisions are given. Particle-wall collisions are still accounted for in all cases (except for the pure-gas case).  $x$  is the streamwise direction.

## 5. Concluding remarks

In this work, we develop algorithms for treating hard-sphere particle-wall and particle-particle collisions under an Euler-Lagrange framework in which the carrier phase is solved with a high-order discontinuous Galerkin method. The proposed method for handling particle-wall collisions builds on that developed in Ref. [21] by better accounting for finite particle size. Two different approaches are presented: in the first, the geometric mappings of both the boundary face and the element are modified by shifting the boundary nodes by a distance equal to the particle diameter. This is directly combined with the particle search-locate procedure presented by Allievi and Bermejo [36]. This approach is efficient, but can neglect certain pathological scenarios, such as that in which a particle exits and reenters the computational domain at the same boundary face within a time step. The second approach can deal with this scenario, but may result in superfluous Newton searches. Regardless, both approaches are applicable to curved, high-aspect-ratio elements and can appreciably improve prediction of particle trajectories in flows involving larger particles colliding with walls.

The developed particle-particle collision algorithm focuses on the identification of particle pairs for collision inspection. For a very wide distribution of particle velocities, popular bin-based approaches can experience significant issues. Instead, the proposed strategy builds on the element-neighbor-list approach. Specifically, information on relative particle trajectories provided by the geometric mapping and the particle search-locate procedure is used to further reduce the number of collision inspections. The developed algorithm is simple, cheap, and compatible with arbitrary, curved, unstructured elements.

These algorithms are applied to three test cases. The first case involves particles colliding elastically within an quiescent inviscid fluid. With particle velocities sampled from a Maxwellian distribution, this problem is analogous to the motion of molecules comprising an ideal gas in thermodynamic equilibrium. We find that the proposed formulation is able to accurately recover equilibrium properties from kinetic theory such as collision frequency and mean free path. Results are robust across various mesh topologies. The developed interparticle collision algorithm is significantly faster than the element-neighbor-list approach.

The second case entails a subsonic stream of particles accelerating through a nozzle and impinging on a flat plate specimen. Increase in particle loading reduces the average erosion rate due to the accumulation of particles near the specimen that shield it from subsequent incoming particles via inelastic particle-particle collisions. Good agreement in the erosion rate with experiments by Oka et al. [46] is observed. Also consistent with experimental results is the localization of at least 90% of the total damage to the circular region directly under the nozzle exit.

In the last test case, we simulate high-speed dusty flow over an entry capsule. Four trajectory points from the ExoMars Schiaparelli mission [55] are selected. We find that particle-particle collisions can be important at high Mach numbers and particle loading ( $Ma \gtrsim 5$  and  $\beta \gtrsim 5\%$ ). Specifically, the impact energy of particle-wall collisions is reduced by way of the shielding effect observed in the previous configuration. At smaller Mach numbers and particle loading, particle-particle collisions can be neglected because of the lower particle-wall and particle-particle collision frequencies. Additionally, with slower particles, there is not only increased likelihood that particles are carried by the flow away from the capsule, but also less energy lost by inelastic interparticle collisions.

As a final remark, we note that the proposed formulation can be straightforwardly extended to arbitrary element shapes and geometric orders. It can also be applied to other finite-element- and spectral-element-based methods that typically employ the same type of geometric mapping.

### CRedit authorship contribution statement

**Eric J. Ching:** Conceptualization, Methodology, Writing – original draft. **Matthias Ihme:** Funding acquisition, Supervision, Writing – review & editing.

### 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 a Stanford Graduate Fellowship and an Early Career Faculty grant (NNX15AU58G) from the NASA Space Technology Research Grants Program. Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center.

### References

- [1] A.B. Morris, D.B. Goldstein, P.L. Varghese, L.M. Trafton, Lunar dust transport resulting from single- and four-engine plume impingement, *AIAA J.* 54 (4) (2016) 1339–1349.
- [2] K. Balakrishnan, J. Bellan, High-fidelity modeling and numerical simulation of cratering induced by the interaction of a supersonic jet with a granular bed of solid particles, *Int. J. Multiph. Flow* 99 (2018) 1–29.
- [3] M.A. Dehnavi, S. Shahhosseini, S.H. Hashemabadi, S.M. Ghafleebashi, CFD simulation of hydrodynamics and heat transfer in gas phase ethylene polymerization reactors, *Int. Commun. Heat Mass Transf.* 37 (4) (2010) 437–442.
- [4] G.A. Bokkers, M. van Sint Annaland, J.A.M. Kuipers, Mixing and segregation in a bidisperse gas–solid fluidised bed: a numerical and experimental study, *Powder Technol.* 140 (3) (2004) 176–186.
- [5] C. González-Montellano, A. Ramirez, E. Gallego, F. Ayuga, Validation and experimental calibration of 3D discrete element models for the simulation of the discharge flow in silos, *Chem. Eng. Sci.* 66 (21) (2011) 5116–5126.
- [6] P.J. Owen, P.W. Cleary, Prediction of screw conveyor performance using the discrete element method (DEM), *Powder Technol.* 193 (3) (2009) 274–288.
- [7] C.T. Crowe, J.D. Schwarzkopf, M. Sommerfeld, Y. Tsuji, *Multiphase Flows with Droplets and Particles*, CRC Press, 2011.
- [8] L. Verlet, Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.* 159 (1) (1967) 98–103.
- [9] B.P.B. Hoomans, J.A.M. Kuipers, W.J. Briels, W.P.M. van Swaaij, Discrete particle simulation of bubble and slug formation in a two-dimensional gas–fluidised bed: a hard-sphere approach, *Chem. Eng. Sci.* 51 (1) (1996) 99–118.
- [10] S. Sundaram, L.R. Collins, Numerical considerations in simulating a turbulent suspension of finite-volume particles, *J. Comput. Phys.* 124 (2) (1996) 337–350.
- [11] B.P.B. Hoomans, *Granular dynamics of gas–solid two-phase flows*, Ph.D. Thesis, Twente University, Netherlands, 2000.
- [12] J. Capecelatro, O. Desjardins, An Euler–Lagrange strategy for simulating particle-laden flows, *J. Comput. Phys.* 238 (2013) 1–31.
- [13] L. Huilin, W. Shuyan, Z. Yunhua, L. Yang, D. Gidaspow, J. Ding, Prediction of particle motion in a two-dimensional bubbling fluidized bed using discrete hard-sphere model, *Chem. Eng. Sci.* 60 (12) (2005) 3217–3231.
- [14] B. Quentrec, C. Brot, New method for searching for neighbors in molecular dynamics computations, *J. Comput. Phys.* 13 (3) (1973) 430–432.
- [15] D. Zwick, S. Balachandar, A scalable Euler–Lagrange approach for multiphase flow simulation on spectral elements, *Int. J. High Perform. Comput. Appl.* 34 (3) (2020) 316–339.
- [16] H. Sigurgeirsson, A. Stuart, W.-L. Wan, Algorithms for particle-field simulations with collisions, *J. Comput. Phys.* 172 (2) (2001) 766–807.



- [17] Z. Yao, J.-S. Wang, G.-R. Liu, M. Cheng, Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method, *Comput. Phys. Commun.* 161 (1–2) (2004) 27–35.
- [18] M. Breuer, M. Alletto, Efficient simulation of particle-laden turbulent flows with high mass loadings using LES, *Int. J. Heat Fluid Flow* 35 (2012) 2–12.
- [19] V. Ogarko, S. Luding, A fast multilevel algorithm for contact detection of arbitrarily polydisperse objects, *Comput. Phys. Commun.* 183 (4) (2012) 931–936.
- [20] D. Krijgsman, V. Ogarko, S. Luding, Optimal parameters for a hierarchical grid data structure for contact detection in arbitrarily polydisperse particle systems, *Comput. Particle Mech.* 1 (3) (2014) 357–372.
- [21] E.J. Ching, S.R. Brill, Y. Lv, M.D. Barnhardt, M. Ihme, An Euler-Lagrange method for simulating high-speed multiphase flows with two-way-coupling using discontinuous Galerkin schemes on arbitrary curved elements, *J. Comput. Phys.* 405 (2020).
- [22] B.J. McBride, M.J. Zehe, S. Gordon, NASA Glenn coefficients for calculating thermodynamic properties of individual species, 2002, NASA TP-2002-211556.
- [23] J.C. Slattery, *Momentum, Energy, and Mass Transfer in Continua*, McGraw-Hill Chemical Engineering Series, 1972.
- [24] T.B. Anderson, R. Jackson, Fluid mechanical description of fluidized beds. Equations of motion, *Ind. Eng. Chem. Fundam.* 6 (4) (1967) 527–539.
- [25] G.S. Shallcross, R.O. Fox, J. Capecelatro, A volume-filtered description of compressible particle-laden flows, *Int. J. Multiph. Flow* 122 (2020).
- [26] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* 43 (2) (1981) 357–372.
- [27] F. Bassi, S. Rebay, GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations, in: B. Cockburn, C.W. Shu (Eds.), *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Springer, Berlin, 2000, pp. 197–208.
- [28] H.C. Yee, M. Vinokur, M.J. Djomehri, Entropy splitting and numerical dissipation, *J. Comput. Phys.* 162 (1) (2000) 33–81.
- [29] E.J. Ching, Y. Lv, P. Gnoffo, M. Barnhardt, M. Ihme, Shock capturing for discontinuous Galerkin methods with application to predicting heat transfer in hypersonic flows, *J. Comput. Phys.* 376 (2019) 54–75.
- [30] C.B. Henderson, Drag coefficients of spheres in continuum and rarefied flows, *AIAA J.* 14 (6) (1976) 707–708.
- [31] T.W. Fox, C.W. Rackett, J.A. Nicholls, Shock wave ignition of magnesium powders, in: 11th International Shock Tubes and Waves Symposium, Seattle, Washington, 1978, pp. 262–268.
- [32] A.N. Volkov, Transitional flow of a rarefied gas over a spinning sphere, *J. Fluid Mech.* 683 (2011) 320–345.
- [33] E. Loth, Compressibility and rarefaction effects on drag of a spherical particle, *AIAA J.* 46 (9) (2008) 2219–2228.
- [34] E.J. Ching, M. Barnhardt, M. Ihme, Sensitivity study of high-speed dusty flows over blunt bodies simulated using a discontinuous Galerkin method, in: *AIAA Scitech 2019 Forum*, 2019, AIAA-2019-0895.
- [35] E.J. Ching, M. Ihme, Efficient projection kernels for discontinuous Galerkin simulations of disperse multiphase flows on arbitrary curved elements, *J. Comput. Phys.* 435 (2021).
- [36] A. Allievi, R. Bermejo, A generalized particle search-locate algorithm for arbitrary grids, *J. Comput. Phys.* 132 (1997) 157–166.
- [37] M. Ceze, K.J. Fidkowski, Constrained pseudo-transient continuation, *Int. J. Numer. Methods Eng.* 102 (11) (2015) 1683–1703.
- [38] A. Haselbacher, F.M. Najjar, J.P. Ferry, An efficient and robust particle-localization algorithm for unstructured grids, *J. Comput. Phys.* 225 (2) (2007) 2198–2213.
- [39] P. Ortwein, S.M. Copplestone, C.-D. Munz, T. Binder, W. Reschke, S. Fasoulas, A particle localization algorithm on unstructured curvilinear polynomial meshes, *Comput. Phys. Commun.* 235 (2019) 63–74.
- [40] Y.M. Tsirkunov, S.V. Panfilov, M.B. Klychnikov, Semiempirical model of impact interaction of a disperse impurity particle with a surface in a gas suspension flow, *J. Eng. Phys. Thermophys.* 67 (5–6) (1994) 1018–1025.
- [41] A.N. Volkov, Y.M. Tsirkunov, Kinetic model of a collisional admixture in dusty gas and its application to calculating flow past bodies, *Fluid Dyn.* 35 (3) (2000) 380–392.
- [42] A.L. Stasenko, Velocity recovery factors of a particle repelled from a solid surface, *J. Eng. Phys. Thermophys.* 80 (5) (2007) 885–891.
- [43] H.R. Norouzi, R. Zarghami, R. Sotudeh-Gharebagh, N. Mostoufi, *Coupled CFD-DEM Modeling: Formulation, Implementation and Application to Multiphase Flows*, John Wiley & Sons, 2016.
- [44] B. Vreman, B.J. Geurts, N.G. Deen, J.A.M. Kuipers, J.G.M. Kuerten, Two- and four-way coupled Euler-Lagrangian large-eddy simulation of turbulent particle-laden channel flow, *Flow Turbul. Combust.* 82 (1) (2009) 47–71.
- [45] S. Sundaram, L.R. Collins, Collision statistics in an isotropic particle-laden turbulent suspension. Part 1. Direct numerical simulations, *J. Fluid Mech.* 335 (1997) 75–109.
- [46] Y.I. Oka, M. Nishimura, K. Nagahashi, M. Matsumura, Control and evaluation of particle impact conditions in a sand erosion test facility, *Wear* 250 (1–12) (2001) 736–743.
- [47] N.V. Brilliantov, F. Spahn, J.-M. Hertzsch, T. Pöschel, Model for collisions in granular gases, *Phys. Rev. E* 53 (5) (1996) 5382.
- [48] L. Wang, B. Wu, Z. Wu, R. Li, X. Feng, Experimental determination of the coefficient of restitution of particle-particle collision for frozen maize grains, *Powder Technol.* 338 (2018) 263–273.
- [49] T. Poppe, J. Blum, T. Henning, Analogous experiments on the stickiness of micron-sized preplanetary dust, *Astrophys. J.* 533 (1) (2000) 454.
- [50] C.A.R. Duarte, F.J. de Souza, R. de Vasconcelos Salvo, V.F. dos Santos, The role of inter-particle collisions on elbow erosion, *Int. J. Multiph. Flow* 89 (2017) 1–22.
- [51] Y.I. Oka, K. Okamura, T. Yoshida, Practical estimation of erosion damage caused by solid particle impact. Part 1: effects of impact parameters on a predictive equation, *Wear* 259 (1–6) (2005) 95–101.
- [52] Y.I. Oka, T. Yoshida, Practical estimation of erosion damage caused by solid particle impact. Part 2: mechanical properties of materials directly associated with erosion damage, *Wear* 259 (1–6) (2005) 102–109.
- [53] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, 1992, AIAA-92-0439.
- [54] T.A. Oliver, A high-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-averaged Navier-Stokes equations, Massachusetts Institute of Technology, 2008, Ph.D. Thesis.
- [55] A. Gülhan, T. Thiele, F. Siebe, R. Kronen, T. Schleutker, Aerothermal measurements from the ExoMars Schiaparelli capsule entry, *J. Spacecr. Rockets* 56 (1) (2018) 68–81.
- [56] G. Pinaud, J. Bertrand, J. Soler, P. Tran, H. Ritter, ExoMars mission 2016: a preliminary post-flight performance analysis of the heat shield during entry on Mars atmosphere, in: *AIAA Scitech 2019 Forum*, 2019, AIAA-2019-0244.
- [57] T. Ozawa, T. Suzuki, H. Takayanagi, K. Fujita, Investigation of Martian-dust drag and heat transfer for Mars sample return mission, *J. Thermophys. Heat Transf.* 25 (3) (2011) 341–353.
- [58] A. Majid, M. Fertig, Two-phase flow solver for hypersonic entry flows in a dusty Martian atmosphere, *J. Thermophys. Heat Transf.* 30 (2) (2016).
- [59] E.B. Vasilevskii, A.N. Osipov, A.V. Chirikhin, L.V. Yakovleva, Heat exchange on the front surface of a blunt body in a high-speed flow containing low-inertia particles, *J. Eng. Phys. Thermophys.* 74 (6) (2001) 1399–1411.
- [60] E. Vasilevskii, A. Osipov, Experimental and numerical study of heat transfer on a blunt body in dusty hypersonic flow, in: 33rd AIAA Thermophysics Conference, Norfolk, Virginia, 1999, AIAA 1999-3563.
- [61] G. Palmer, Y.-K. Chen, P. Papadopoulos, M. Tauber, Reassessment of effect of dust erosion on heatshield of Mars entry vehicle, *J. Spacecr. Rockets* 37 (6) (2000) 747–752.
- [62] G.E. Palmer, E.J. Ching, M. Ihme, D. Allofs, A. Gülhan, Modeling heatshield erosion due to dust particle impacts for Martian entries, *J. Spacecr. Rockets* 57 (5) (2020) 857–875.



- [63] A.N. Volkov, Y.M. Tsirkunov, Effect of a dispersed admixture on the flow pattern and heat transfer in a supersonic dusty-gas flow around a cylinder, *Fluid Dyn.* 40 (4) (2005) 561–574.
- [64] M.G. Tomasko, L.R. Dose, M. Lemmon, P.H. Smith, E. Wegryn, Properties of dust in the Martian atmosphere from the Imager on Mars Pathfinder, *J. Geophys. Res., Planets* 104 (E4) (1999) 8987–9007.
- [65] P.E. Papadopoulos, Computation of Heat Shield Erosion in a Dusty Martian Atmosphere, Stanford University Ph.D. Thesis, 1994.