Computational Particle Mechanics

ISSN 2196-4378
Volume 5
Number 3

Comp. Part. Mech. (2018) 5:387-409
DOI 10.1007/s40571-017-0176-1

Computational Particle Mechanics (2018) 5:387–409
https://doi.org/10.1007/s40571-017-0176-1

A new approach to fluid–structure interaction within graphics hardware accelerated smooth particle hydrodynamics considering heterogeneous particle size distribution

Adnan Eghtesad[1] · Marko Knezevic[1]

Received: 8 July 2017 / Accepted: 21 November 2017 / Published online: 5 December 2017

Abstract
A corrective smooth particle method (CSPM) within smooth particle hydrodynamics (SPH) is used to study the deformation of an aircraft structure under high-velocity water-ditching impact load. The CSPM-SPH method features a new approach for the prediction of two-way fluid–structure interaction coupling. Results indicate that the implementation is well suited for modeling the deformation of structures under high-velocity impact into water as evident from the predicted stress and strain localizations in the aircraft structure as well as the integrity of the impacted interfaces, which show no artificial particle penetrations. To reduce the simulation time, a heterogeneous particle size distribution over a complex three-dimensional geometry is used. The variable particle size is achieved from a finite element mesh with variable element size and, as a result, variable nodal (i.e., SPH particle) spacing. To further accelerate the simulations, the SPH code is ported to a graphics processing unit using the OpenACC standard. The implementation and simulation results are described and discussed in this paper.

Keywords Smooth particle hydrodynamics · Heterogeneous particle size distribution · Fluid–structure interaction · Graphics processing unit · OpenACC

1 Introduction

Fluid–structure interactions (FSI) are encountered in a wide range of applications including water ditching of aircrafts, where the aircraft structure experiences severe distortions [1]. A number of FSI problems have been studied using finite element method (FEM), finite volume method (FVM), boundary element method (BEM), and their combinations [2–8]. Case studies such as bird strike simulations using these methods have been reported in [9–12].

In order to circumvent the issues of large element distortion in FEM and inaccurate predictions of mechanical fields, as a consequence of remeshing followed by interpolation, particle-based numerical algorithms such as smooth particle hydrodynamics (SPH) have been conceived. SPH was first proposed to solve cosmological problems in three-dimensional open space such as the simulations of binary stars and stellar collisions [13,14]. Since then, the SPH method has been applied in computational fluid dynamics for studies that include multi-phase flow [15–19], incompressible flow [20] and free surface flow analyses [21–25]. Benz and Asphaug applied SPH to simulate fracture in brittle solids [26–28]. Studies have also been conducted using SPH to better understand ductile damage [29–33]. Johnson and Libersky have made outstanding contributions in the application of SPH to impact problems [34,35]. Chen et al. and other researchers [36–39] have implemented an improvement for tensile instability in SPH using renormalization schemes, which was found to improve the accuracy of predictions at free surface boundaries. Eghtesad et al. investigated the dynamic mixed ductile–brittle behavior of metal–ceramic functionally graded materials under high-velocity FSI impact loads [40]. Rabczuk et al. implemented an immersed particle method as a modification of SPH to model FSI phenomena through the simulation of high-velocity impact and explo-

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s40571-017-0176-1) contains supplementary material, which is available to authorized users.

✉ Marko Knezevic
marko.knezevic@unh.edu

[1] Department of Mechanical Engineering, University of New Hampshire, 33 Academic Way, Kingsbury Hall, W119, Durham, NH 03824, USA

sion [41]. Recently, a wide range of FSI simulations have been investigated using SPH [42–46]. Most of the studies on FSI problems using SPH algorithm focus primarily on the elastic deformations involving small strains or the fluid–rigid body interactions rather than large deformations under high-velocity impact load. The applied load used in such studies are relatively low-speed impact, not sufficient to cause the structure undergo large inelastic deformations. The SPH method has also been implemented in surgery and used in biomedical applications [47]. Very recently, SPH has been used for studying microstructural evolution during severe plastic deformation processing [48,49].

In the presented paper, a corrective smooth particle method (CSPM) within SPH is used to study the strain rate-dependent inelastic deformation of a real-scale aircraft structure in 3D under high-velocity water ditching. Specifically, the CSPM treatment is applied to both fluid and solid particles to have a more realistic free surface behavior, which required the development of an algorithm for detecting the free surface boundary particles during the solution process. The latter is important, because, if the algorithms used to detect the particles that lay on the boundary of water free surface with air are not accurate, spurious numerical results due to unphysical oscillations and interpenetration of particles occur. The solid structure follows the Johanson–Cook plasticity constitutive model interacting with a weakly compressible fluid, which is defined using a liner equation of state (EOS) expressing a relation between sound speed, pressure, and density. The fluid–solid interface and underlying interaction loads arising from a strong two-way interaction, where the fluid domain interacts with solid and vice versa, are all considered. Instead of using a penalty repulsive force, a new scheme is proposed to improve the interface contact behavior between the fluid and solid structure. The new scheme relies on definition of a third particle-type neighboring, which is not considered as a fluid or a solid but both. This particle-type neighboring only exists in the interface region and a set of specific equations are solved to model the boundary. This treatment not only prevents the interpenetration of fluid and solid particles, but also maintains the necessary gap between fluid and solid boundary particles representing the interaction interface, which remains consistent during the simulation process. Furthermore, a heterogeneous particle size distribution over a complex three-dimensional geometry is used to reduce the simulation time relative to a uniform particle size SPH simulation. The variable particle size is achieved from a finite element mesh with variable resolution using the ANSYS WORKNBECN meshing module in conjunction with the ICEM CFD meshing software. The procedure consists of exporting nodal coordinates from the ICEM CFD software and then importing them as centers of SPH particles in the CSPM-SPH code while ensuring their interconnectivity. Finally, the SPH code is ported to a graphics processing unit (GPU) using the OpenACC standard to accelerate the calculations, which is of significant importance for SPH simulations in 3D space. Simulation results show that using the novel interface algorithm together with CSPM treatment within SPH is well suited for modeling the deformation of structure under water-ditching impact loads. The locations of mechanical field localizations in the aircraft structure upon water ditching are successfully predicted. The contours showed no sudden discontinuities intrinsic to the heterogeneity nature of SPH algorithm. Moreover, the impacted interfaces showed no artificial particle penetrations.

The paper is organized as follows: Sect. 2 summarizes the main SPH governing equations; Sect. 3 defines the solid structure constitutive relations; Sect. 4 elaborates the SPH heterogeneous particles size generation; Sect. 5 presents the novel FSI algorithm; Sect. 6 describes the GPU implementation of CSPM-SPH model using the OpenACC standard; Sect. 7 presents a benchmark validation case study; Sect. 8 presents the application case study of an aircraft structure under high-velocity water-ditching impact load; Sect. 9 presents the main conclusions.

2 Governing equations in SPH

In SPH formulation, a structure is represented using a number of particles. These particles interact with each other through a domain named kernel support carrying field variables. The field variables of a given particle are computed using related values of the neighboring particles in a summation form through a smooth function called the kernel function [50]. The kernel functions are smooth splines, which are easily differentiable. Figure 1 illustrates a schematic view of kernel support covering a smooth kernel on neighboring particles. The kernel support's radius is defined as a product between a constant k and a smoothing length h. Next, we summarize the main equations pertaining to SPH.

In our notation, tensors are denoted by bold not italic letters, while tensor components and scalars are not bold but are italic. The dot product is denoted by "·" and the tensor product is denoted by "⊗". α, β, γ denote the component indices and i, j represent two neighboring particles. Also, the relation $\square_{ij} = \square_i - \square_j$ holds true for all variables.

2.1 Continuity

The continuity equation ensures the conservation of mass and is expressed using the following material time derivative involving the divergence of velocity vector:

$$\frac{D\rho}{Dt} = -\rho\left(\nabla \cdot \mathbf{v}\right), \tag{1}$$

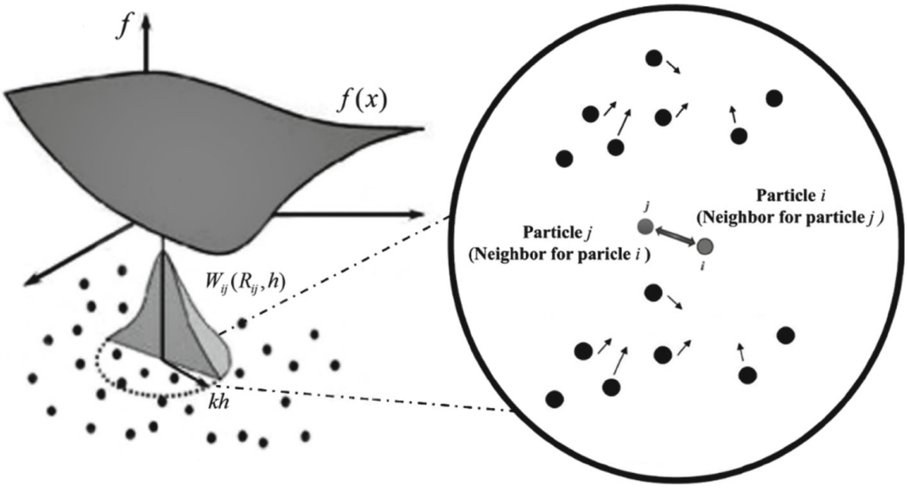


Fig. 1 A schematic view of kernel support covering a smooth kernel on SPH neighboring particles

where ρ and $\mathbf{v}$ are the density and velocity vectors, respectively. The continuity equation in the SPH discretized form can be represented as:

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_j}\left(\mathbf{v}_i - \mathbf{v}_j\right) \cdot \frac{\partial W_{ij}\left(R_{ij}, h\right)}{\partial \mathbf{x}_i}, \tag{2}$$

where h is the smoothing length used to determine the supporting domain size. The symbol $R_{ij} = \frac{r_{ij}}{h} = \frac{\left|\mathbf{x}_i - \mathbf{x}_j\right|}{h}$ is the normalized relative distance between particles i and j. The symbol $r_{ij} = \left|\mathbf{x}_i - \mathbf{x}_j\right|$ denotes the magnitude (modulus) of the relative distance between the two particles, the vector $\mathbf{r}_{ij} = \mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ is joining the position vector of particles i (i.e., $\mathbf{x}_i$) and j (i.e., $\mathbf{x}_j$), and m_i denotes mass of the particle i. W_{ij} defines the smoothing kernel function of particle i evaluated at particle j.

2.2 Linear momentum balance

The balance of linear momentum in terms of the stress tensor upon the localization equation has the following form:

$$\frac{D\mathbf{v}}{Dt} = \frac{1}{\rho}\nabla \cdot \boldsymbol{\sigma}. \tag{3}$$

In above equation, $\boldsymbol{\sigma}$ denotes the total Cauchy stress tensor, which can be split as:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{s}, \tag{4}$$

where p is the hydrodynamic pressure, $\mathbf{I}$ is the identity tensor, and $\mathbf{s}$ is the deviatoric part of the stress tensor.

Using Jaumann rate to enforce objectivity, the following constitutive relation is assumed for the deviatoric stress tensor,

$$\frac{D\mathbf{s}}{Dt} = 2G\left(\dot{\boldsymbol{\varepsilon}} - \frac{1}{3}\mathbf{I}\dot{\varepsilon}^{\gamma\gamma}\right) + \mathbf{s}\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{\omega}}\mathbf{s}, \tag{5}$$

where G is shear modulus for a solid phase made of Al 6061-T6, which is 26 GPa and $\dot{\boldsymbol{\varepsilon}}$ is the strain rate tensor given by:

$$\dot{\boldsymbol{\varepsilon}} = sym\left\{\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right\}. \tag{6}$$

The SPH form of Eq. (6) is expressed as:

$$\dot{\boldsymbol{\varepsilon}}_i = \sum_j \frac{m_j}{\rho_j} sym\left\{\left(\mathbf{v}_j - \mathbf{v}_i\right) \otimes \frac{\partial W_{ij}\left(R_{ij}, h\right)}{\partial \mathbf{x}_i}\right\}, \tag{7}$$

in which "*sym*" means the symmetric part of included equation.

The variable $\dot{\boldsymbol{\omega}}$ is the rotation rate tensor given by

$$\dot{\boldsymbol{\omega}} = asym\left\{\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right\}. \tag{8}$$

The SPH form of Eq. (8) can be written as:

$$\dot{\boldsymbol{\omega}}_i = \sum_j \frac{m_j}{\rho_j} asym\left\{\left(\mathbf{v}_j - \mathbf{v}_i\right) \otimes \frac{\partial W_{ij}\left(R_{ij}, h\right)}{\partial \mathbf{x}_i}\right\}, \tag{9}$$

in which "*asym*" means the asymmetric part of included formula.

The SPH discretized form of the momentum equation can be written as follows:

$$\frac{D\mathbf{v}_i}{Dt} = \sum_j m_j \left(\frac{\boldsymbol{\sigma}_i}{\left(\rho_i\right)^2} + \frac{\boldsymbol{\sigma}_j}{\left(\rho_j\right)^2} + \Pi_{ij}\right) \frac{\partial W_{ij}\left(R_{ij}, h\right)}{\partial \mathbf{x}_i}. \tag{10}$$

The artificial viscosity Π_{ij} in SPH methodology and other numerical methods has been included in the momentum equation to prevent unphysical particle interpenetration and to improve the numerical stability on capturing the shock waves generated by impact. The viscosity can be considered as a shear and bulk component affecting the material behavior [51–54],

$$\Pi_{ij} = \begin{cases} \frac{-a\bar{C}_{s_{ij}}\left(\phi_{ij}-\frac{b}{a}\phi_{ij}^2\right)}{\rho_{ij}} & \mathbf{v}_{ij}\cdot\mathbf{r}_{ij}<0 \\ 0 & \mathbf{v}_{ij}\cdot\mathbf{r}_{ij}\geq 0 \end{cases}, \tag{11a}$$

where

$$\phi_{ij} = \frac{h_{ij}\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}}{r_{ij}^2+\eta^2}, \bar{C}_{s_{ij}} = \frac{1}{2}(C_{s_i}+C_{s_j}), \rho_{ij} = \frac{1}{2}(\rho_i+\rho_j), h_{ij} = \frac{1}{2}(h_i+h_j). \tag{11b}$$

In above equations, C_{s_i} and C_{s_j} are the sound speed of particle i and particle j respectively, a and b are the non-dimensional coefficients with recommended values $a = 1, b = 2$ or $a = b = 2.5$. In order to choose the right value for our work, a comparison with experimental results was done in which the crater and debris dimensions for a hyper velocity impact of a projectile into a target were validated and proved to be accurate. This validation will be elaborated in Sect. 7 later in the text. In this study, we used $a = 1, b = 2$. The value of parameter η is chosen to be $0.1h_{ij}$ to achieve the numerical stability especially in the regions where particles are approaching to each other.

2.2.1 Energy

In order to account for the conservation of energy while solving the Navier–Stokes equations for the fluid domain, the energy equation needs to be solved together with continuity and momentum equations. The energy equation can be written in the following form:

$$\frac{De}{Dt} = -\frac{p}{\rho}\nabla\cdot\mathbf{v}+\frac{\mu}{\rho}\dot{\boldsymbol{\varepsilon}}\cdot\dot{\boldsymbol{\varepsilon}}, \tag{12}$$

where e and $\dot{\boldsymbol{\varepsilon}}$ denote the energy magnitude and shear strain rate, respectively. The energy balance equation in the SPH discretized form can be defined as:

$$\frac{De_i}{Dt} = \frac{1}{2}\sum_j m_j\left(\frac{p_i}{(\rho_i)^2}+\frac{p_j}{\left(\rho_j\right)^2}\right)\left(\mathbf{v}_i-\mathbf{v}_j\right)\cdot\frac{\partial W_{ij}\left(R_{ij},h\right)}{\partial\mathbf{x}_i} + \frac{\mu}{2\rho_i}\dot{\boldsymbol{\varepsilon}}\cdot\dot{\boldsymbol{\varepsilon}}. \tag{13}$$

For Newtonian fluids, the viscous shear stress can be written as:

$$\boldsymbol{\tau}_i = \mu\dot{\boldsymbol{\varepsilon}}_i. \tag{14}$$

where μ is the dynamic viscosity of the fluid. The value for dynamic viscosity of water used here is 8.9×10^{-4}(Pa). The strain rate used in Eq. (14) for fluid particles can be written as:

$$\dot{\boldsymbol{\varepsilon}}_i = \sum_j\frac{2m_j}{\rho_j}sym\left\{\left(\mathbf{v}_j-\mathbf{v}_i\right)\otimes\frac{\partial W_{ij}\left(R_{ij},h\right)}{\partial\mathbf{x}_i}\right\} - \frac{2}{3}\mathbf{I}\sum_j\frac{m_j}{\rho_j}(\mathbf{v}_j-\mathbf{v}_i)\cdot\frac{\partial W_{ij}\left(R_{ij},h\right)}{\partial\mathbf{x}_i}. \tag{15}$$

2.3 Free surface tracking

In order to better describe the fluid behavior at free surfaces, a search algorithm is applied to all particles to detect the free surface particles. To implement this process, the following equality is checked every time step,

$$\sum_j\frac{m_j}{\rho_j}W_{ij}\left(R_{ij},h\right) = 1. \tag{16}$$

Since particles located at free surface boundaries do not have enough neighbor particles in their support domain, the left hand side of the above equation is much less than unity for such particles. When free surface particles have been identified, the value of pressure is set to zero for all those particles.

2.4 XSPH correction

SPH particles of a medium, especially those within a supporting domain, are expected to move together in a uniform way. (i.e., particles of water falling from a source at a higher level to a lower height follow the same path and stream lines). It is unphysical to see whether a portion of this control volume of the water medium is scattered without any changes in the fluid path or source. A common issue in some SPH algorithms for fluid particles is that a set of particles do not have enough time to update themselves (their velocities) in a high-velocity regime, which results in an unphysical disarrangement of a number of particles with respect to the other particles. XSPH is a modification applied to SPH to prevent spurious oscillations in fluid particles [55–58]. The XSPH velocity correction is introduced according to the following relation:

$$\frac{D\mathbf{x}_i}{Dt} = \mathbf{v}_i - \propto\sum_j\frac{m_j}{\rho_j}(\mathbf{v}_i-\mathbf{v}_j)W_{ij}\left(R_{ij},h\right), \tag{17}$$

where $\propto$ is a constant parameter between zero and unity. Using the XSPH modification, the particles (fluid elements) move with an average velocity in a similar manner in a same regime of flow. The exact value of $\propto$ is problem dependent, but usually a value between 0.1 and 0.5 gives good results.

In our simulations, we used a value of 0.1 to assure that no overestimation of kernel approximation is attained over the particles. XSPH correction should be applied carefully to prevent any out of range results [55,59–61]. While there are other filters and schemes as alternatives to XSPH such as Shepard filter [62] primarily used in fluid dynamics, XSPH is proven to work well for the FSI studies [63].

2.4.1 CSPM correction for improving the boundary treatment

Another common numerical issue at high pressures is the so called tensile instability problem. The tensile instability shows itself in different types of numerical inconsistencies. For example, in a simple tensile test simulation using a common SPH formulation without any modifications, the particles will detach from each other (i.e., form a spurious fracture) leading to an unrealistic physical behavior. On the other hand, at ultrahigh pressures created at the impact zone under a high-velocity collision, the particles penetrate into each other, causing the solution to diverge. To relax these issues, several approaches have been proposed in the literature [36,64,65]. One of the most efficient and easy to implement approaches is the CSPM correction suggested in [36–40,66]. Essentially, this algorithm resolves the problem of particle deficiency (i.e., lacking enough particles) at the boundaries. The CSPM correction extends the kernel estimate function using its higher order Taylor series expansion involving higher order derivatives of the desired quantity (i.e., density, velocity) over the neighboring particles. Thus, the kernel support domains for boundary particles improves reaching the size of the domains for interior particles [67]. CSPM not only improves the tensile instability problem, but also increases the accuracy in boundary surfaces with the applied normalization. The CSPM correction has the following form:

$$\frac{D\rho_i}{Dt} = \frac{\rho_i \sum_j \frac{m_j}{\rho_j} \left(\mathbf{v}_i - \mathbf{v}_j\right) \cdot \frac{\partial W_{ij}(R_{ij},h)}{\partial \mathbf{x}_i}}{\sum_j \frac{m_j}{\rho_j} \left(\mathbf{x}_i - \mathbf{x}_j\right) \cdot \frac{\partial W_{ij}(R_{ij},h)}{\partial \mathbf{x}_i}}. \tag{18}$$

A case study was conducted in order to demonstrate the effect of CSPM on the behavior of boundaries and interfaces. To this end, a simulation was run to study the cube on cube impact of Al 6061-T6 into steel 4340 at the impact velocity of 500 m/s. Velocity magnitude is chosen to be high enough to ensure a fair judgment for large strain scenarios. Figure 2 represents such comparison for the SPH solver with and without CSPM modification. Results confirm that the CSPM modification corrects the unphysical particle clustering, disorder, and spurious interpenetration of particles of two materials into each other, by ensuring the gap maintained at the interface.

In addition to the CSPM correction, we used an initial smoothing length and particle spacing suggested in [68] for improving the tensile instability. The initial smoothing length is set to be $h_0 = 1.558 r_{ij_0}$, where h_0, r_{ij_0} are the initial smoothing length and the initial particle spacing, respectively.

2.5 Choice of a kernel

Common kernels used in SPH approximation are Gaussian, cubic (B-spline), quadratic, quartic, quintic, and the new quartic kernel. The most commonly used kernel is the cubic kernel because of its accuracy and robustness. In our work, we use the new quartic kernel [50], which has the same form as the cubic kernel but with a smoother derivative. This kernel has the following form:

$$W_{ij}\left(R_{ij}, h\right) = \alpha_d \times \begin{cases} \left(\frac{2}{3} - \frac{9}{8} R_{ij}^2 + \frac{19}{24} R_{ij}^3 - \frac{5}{32} R_{ij}^4\right), & 0 \leq R_{ij} < 2 \\ 0 & R_{ij} > 2 \end{cases}, \tag{19}$$

where the parameter α_d is a constant that normalizes the kernel function on the support domain The values for α_d are $1/h$, $15/7\pi h^2$, $315/207\pi h^3$ in one-, two-, and three-dimensional spaces, respectively.

2.6 Time step control and convergence behavior

In order to obtain convergence in the explicit time integration scheme used in SPH, a variable time step was selected as a function of velocity and sound speed as following:

$$\Delta t = \min\left(\alpha \frac{h}{S_s \left(\max\left(abs(v)\right)\right) + C_s}\right), \tag{20}$$

where S_s is the slope in shock velocity of Hugoniot curve [69–72], which is defined later in Sect. 3.2 of the text. Having a variable time step also helps to improve the computational efficiency by varying it according to the case study instead of using a very small constant value during the whole simulation, where larger time steps are allowed later in the solution. The constant α varies between 0 and 1 depending on the problem size and boundary conditions applied. Usually, for studying merely the fluid behavior, a time step in the range of 0.0001 s ensures the convergence. We recognize that there are other versions of SPH developed specifically to study the flow of water (i.e., free surface flows, dam break), in which an incompressible SPH (ISPH) algorithm [73–75] is implemented allowing a larger time increment in orders of 0.001–0.01 s.

On the other hand, in high-velocity impact simulations, due to the large velocity magnitude in the denominator

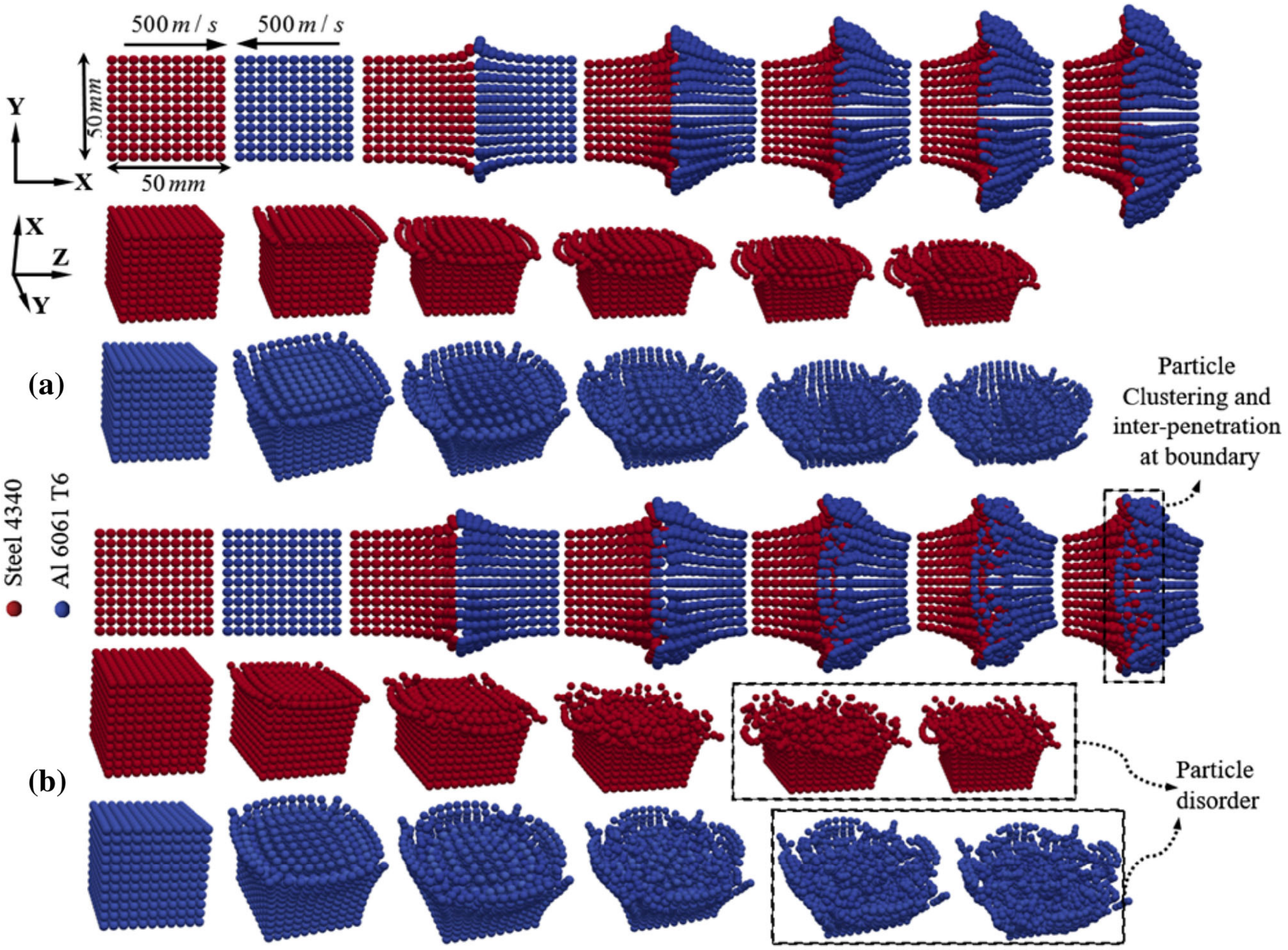


Fig. 2 A comparison between SPH simulation results of cube (Al) on cube (steel) normal impact carried out **a** with CSPM and **b** without CSPM. Evidently, the CSPM corrects unphysical particle clustering, disorder, and spurious penetration of two materials into each other by ensuring the gap at the interface

of Eq. (20), a smaller time step is needed to obtain the solution convergence. Furthermore, the fluid–structure interaction problems introduce one more level of complexity at the interface which demand smaller time steps comparing to fluid-only simulations. This time resolution is necessary to ensure capturing the motion and deformation of materials at the FSI interface, accurately.

3 Strain rate-dependent plastic deformation and shock wave response

3.1 Equation of state for water

In order to simulate water as a weakly compressible fluid, the following linear equation of state (EOS) is used which defines the relation between pressure, density, and sound speed:

$$P = C_s^2(\rho - \rho_0). \tag{21}$$

The subscript zero defines the reference value of the related property. The sound speed in water can be obtained using the following equation:

$$C_s = \sqrt{\frac{\zeta}{\rho}}, \tag{22}$$

where ζ is the compressibility modulus of water with a constant value of $2.2 \times 10^9 (Pa)$ for a large range of pressures.

3.2 Equation of state for high-pressure shock response of solids

To define the pressure–density relation for solids including metals, the Mie-Grüneisen form of EOS is used [50]. The Mie Gruneisen EOS has the following form:

$$P(\rho, e) = \left(1 - \frac{1}{2}\Gamma\eta\right) P_H(\rho) + \Gamma\rho e, \tag{23}$$

Table 1 Mie Gruneisen EOS model constants for Al 6061-T6

Material constant	Value
ρ_0	2710 kg/m^3
C_s	6320 m/s
S_s	1.337
Γ	2.1

where Γ is the Gruneisen parameter, which depends on a material. The variable η can be represented as a function of initial and current density using:

$$\eta = \frac{\rho}{\rho_0} - 1. \tag{24}$$

The subscript '0' indicates the initial quantity. P_H is referred to the pressure on Hugonoit reference curve [69–72], and can be written as:

$$P_H = \begin{cases} a_0\eta + b_0\eta^2 + c_0\eta^3 & \eta > 0 \\ a_0\eta & \eta < 0 \end{cases}, \tag{25}$$

where $a_0, b_0, and c_0$ are defined as:

$$\begin{aligned} a_0 &= \rho_0 C_s^2, \\ b_0 &= a_0 \left[1 + 2\left(S_s - 1\right)\right], \\ c_0 &= a_0 \left[2\left(S_s - 1\right) + 3\left(S_s - 1\right)^2\right]. \end{aligned} \tag{26}$$

The Hugoniot slope of the linear relation between the shock and particle velocity is defined as:

$$S_s = \frac{U_s - C_s}{U_p}. \tag{27}$$

where U_p and U_s denote the solid particle velocity and shock wave velocity, respectively. Table 1 provides the values for the Mie Gruneisen EOS used in this study.

3.3 Johnson–Cook (JC) material model

Strain rate-dependent JC strength model is used to introduce a dynamic flow stress criteria including strain hardening and thermal softening properties [76]. The JC model for the flow stress has the following form:

$$\sigma = \left[A + B\varepsilon^n\right]\left[1 + C\, ln\left(\dot{\varepsilon}^*\right)\right]\left[1 - \left(T^*\right)^m\right], \tag{28a}$$

where T^* is the homologous temperature expressed as:

$$T^* = \frac{T - T_{\text{ambient}}}{T_{\text{melt}} - T_{\text{ambient}}}. \tag{28b}$$

Table 2 Johnson–Cook flow stress model constants for Al6061-T6

Material constant	Value
A	265 MPa
B	426 MPa
C	0.015
m	1.0
n	0.34

The dimensionless strain rate can be written as:

$$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}, \tag{28c}$$

where ε, $\dot{\varepsilon}^*$, $\dot{\varepsilon}$, $\dot{\varepsilon}_0$, T, T_{ambient}, T_{melt} define the equivalent plastic strain, dimensionless equivalent strain rate, equivalent plastic strain rate, reference equivalent plastic strain rate, current temperature, ambient temperature, and melting temperature, respectively. *A, B, C, m, n* are material-dependent parameters, which are obtained for different materials by calibration using mechanical testing data. The values of aforementioned parameters for Al6061-T6 [76] are included in Table 2.

4 A methodology for achieving heterogeneous particle size distribution

Intrinsic to SPH algorithm is a searching method to identify the neighboring particles based on the interparticle distances (i.e., kh), which usually makes the code computationally intensive. In order to improve the efficiency of the search algorithm, a heterogeneous particle sizing distribution has been considered by our code. Figure 3 illustrates a solid discretized into finite elements with variable element size and SPH particles with heterogeneous resolution. Nodes of finite element mesh have been converted into centers of SPH particles, which resulted in variable particle size. As can be seen that particles of smaller size (higher resolution) have been concentrated in the corner to better represent the geometry. A case study was conducted in order to validate the implementation of a variable particle size distribution, in our SPH code and to justify our claim of having more efficient but still accurate solution while using a heterogeneous particles resolution comparing to a uniform distribution. To this end, high velocity impact of a sphere of Al 6061-T6 into an armor plate of steel 4043 was simulated at the impact velocity of 1000 m/s for 100 ms after initiating the impact. Figure 4 represents the initial configuration and the conversion of FEM mesh into SPH particles with uniform and variable size distribution. Projectile was modeled as a sphere with diameter of 50 mm, and the target was modeled as a thin square plate with the length of 200 mm and thickness of 10 mm . In both uniform

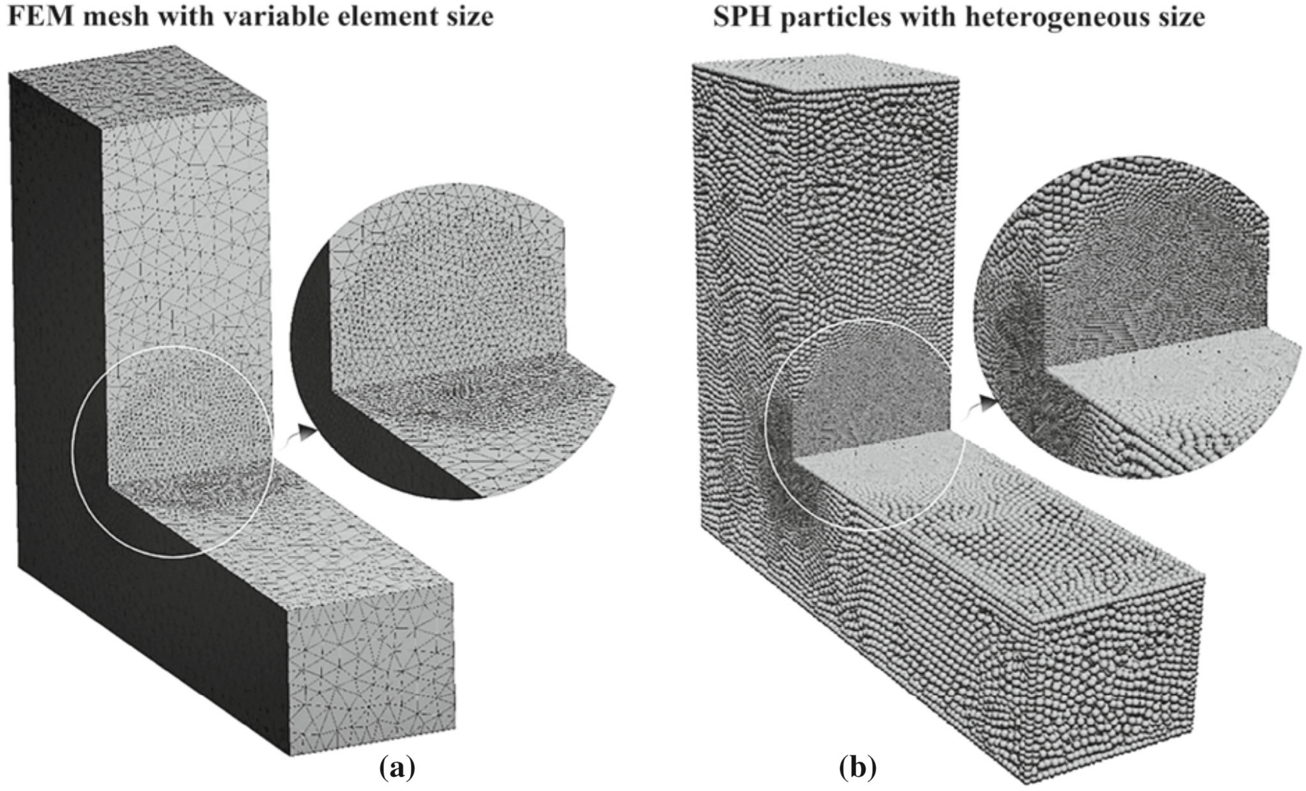

FEM mesh with variable element size
SPH particles with heterogeneous size
(a)
(b)

Fig. 3 Finite element mesh with variable element size converted into SPH particles with heterogeneous size distribution

and variable particle distributions, the sphere was discretized using total number of 343 SPH particles with a uniform sizing of 5 mm. The target plate was represented using 5043 and 1713 SPH particles for the uniform and variable size distribution, respectively. Note that, in order to apply the particle concentration in the vicinity of impact region, a particle size of 5 mm was used while far from this zone, a 10 mm particle size was selected. Table 3 shows the material parameters used for Johnson–Cook (JC) strength model to predict the flow stress in the steel target. The parameters for aluminum are already presented in Table 2. Figure 5 illustrates the deformation response and Johnson–Cook flow stress distribution in the projectile and target plate after 100 ms of impact initiation. It can be observed that using the variable particle sizing with an appropriate refinement in the vicinity of points of interest (i.e., cracks, voids, contacts, interface, impact area), the SPH code gives almost identical results while much more efficient. Table 4 shows the CPU time for running the simulations with uniform and variable particle distribution. It is observable that the configuration with variable particle sizing runs almost four times faster than the uniform size and this is due to less number of particles implemented in the simulation which in turn results in less computational effort in the neighbor search algorithm.

The methodology for achieving heterogeneous particle resolution for a complicated geometry is obtained using a process chain, in which FEM explicit mesh nodes are converted to SPH particles. To this end, a computer aided design (CAD) geometry is meshed using the explicit dynamics (i.e., in order to extract the right nodal information for an explicit type simulation, including the impact simulation we want to perform using the SPH explicit solver) mesh type in ANSYS WORKBENCH meshing module using tetrahedrons for a better control over the mesh size, then the mesh data is exported to ICEM CFD solver and then exported to solver input for LSDYNA software. The file is directly used as the input file for the SPH code, where the initial particle spacing and the variable smoothing length are calculated and assigned to each particle. The initial smoothing length scaling factor k was set to 2.

The aircraft geometry for the application case study has been designed using SOLIDWORKS [77]. It should be noted that many geometry details have been ignored and only the main structure was maintained for the simulation purpose. The real-scale CAD model is illustrated in Fig. 6. The length, height, and width of body dimensions are 19.5, 3.7, and 2 m, respectively. The main wing has a length and width of 28 and 4.6 m, respectively. Figure 7 illustrates steps involved in achieving heterogeneous particle distribution for the aircraft. The ANSYS meshing tool together with the ICEM CFD management of mesh size and quality facilitates a consider-

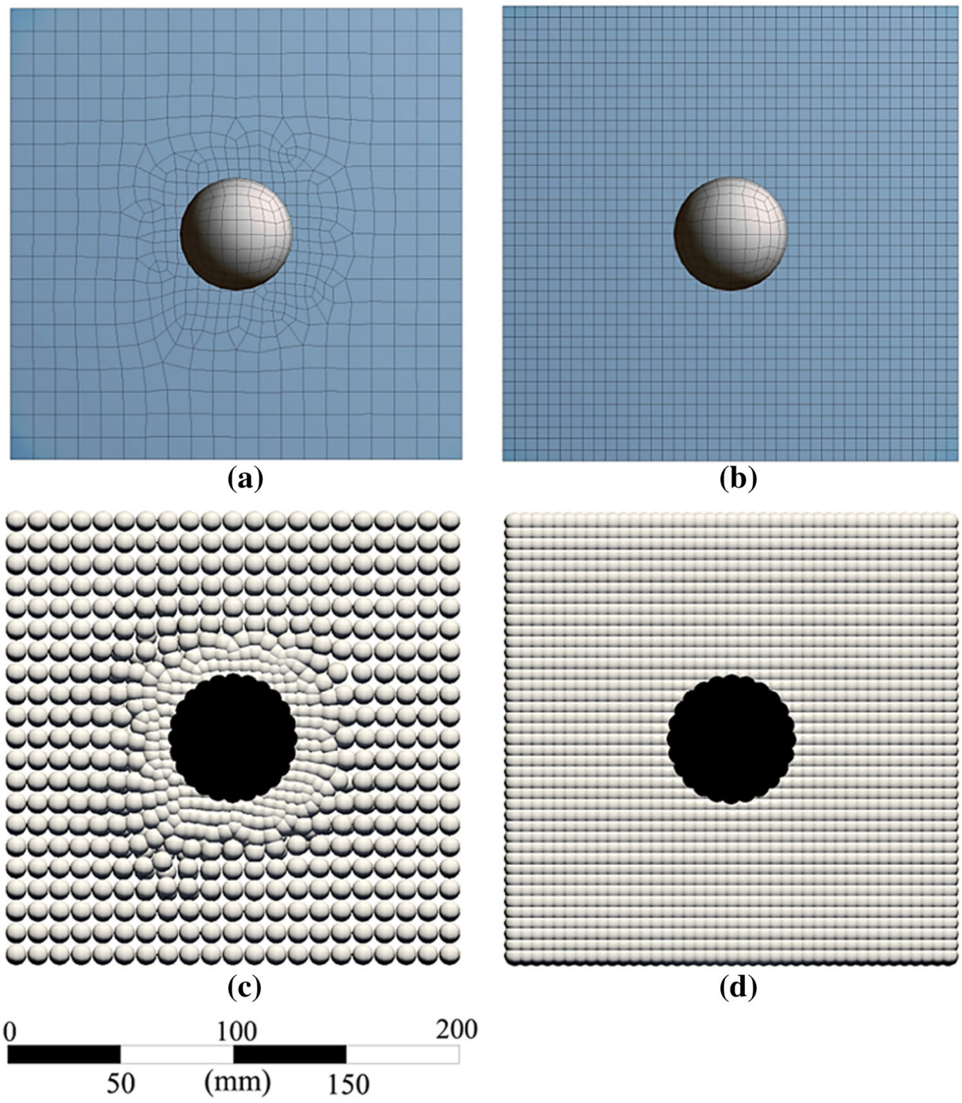


Fig. 4 Conversion of FEM mesh into SPH particles with uniform and variable size distribution for the high-velocity impact simulation of aluminum (i.e., Al 6061-T6) projectile into steel (i.e., Steel 4340) target: **a** variable mesh resolution, **b** uniform mesh size, **c** variable particle size distribution converted from **a**, and **d** uniform particle size distribution converted from **b**

able flexibility to produce a desired particle resolution for the complex geometries.

As illustrated in Fig. 7, variable particle resolution has been used to discretize the aircraft structure. The locations in which structure has sharper edges and corners, include particles with higher resolution (smaller size) resulting in a non-uniform particle distribution or a heterogeneous SPH particle distribution. Each particle has its own volume, mass, and smoothing length differing from others. To ensure more stability and accuracy, all particles are forced to have an initial smoothing length with the maximum value in the whole domain, which in turn is followed by the advantage of having no particles suffer from the lack of enough neighboring particles. Furthermore, the smoothing length is adaptively changed during the simulation using a time- and space-dependent formulation based on the local density rate for each particle according to the following relation:

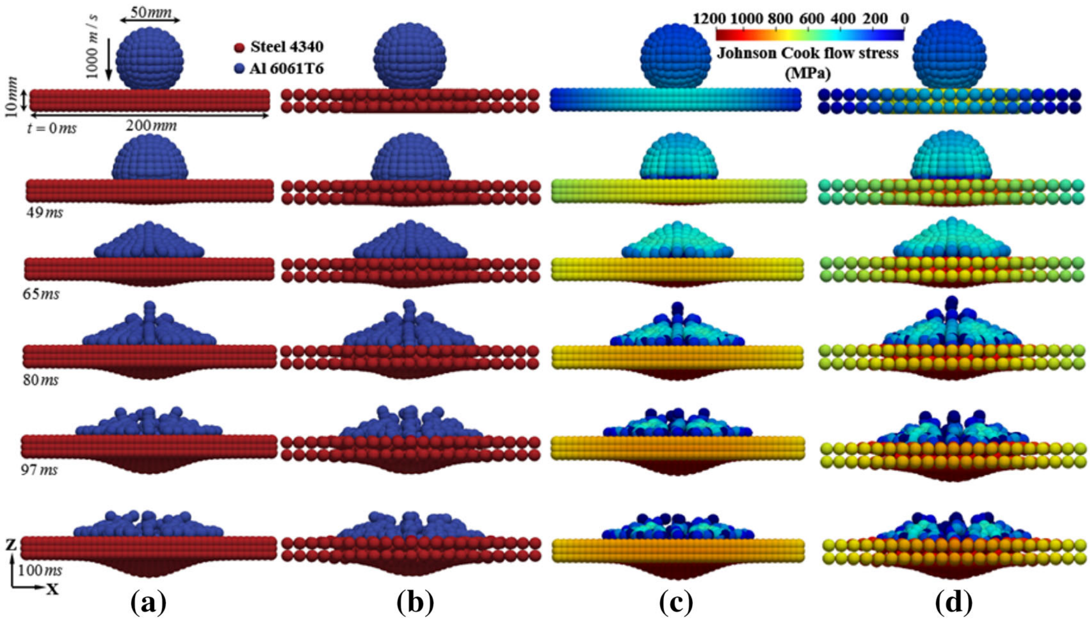
Table 3 Johnson–Cook flow stress model constants for Steel 4340

Material constant	Value
A	792 MPa
B	510 MPa
C	0.014
m	1.03
n	0.26

$$\frac{Dh}{Dt} = -\frac{1}{d}\frac{h}{\rho}\frac{D\rho}{Dt}, \tag{29}$$

where d is the dimension parameter (i.e., $d = 1, 2, 3$ for one-, two-, and three-dimensional domains, respectively). As our simulation is 3D in nature, we used $d = 3$ for Eq. (28).

The former equation can be rewritten in the SPH formulation using the following equation:



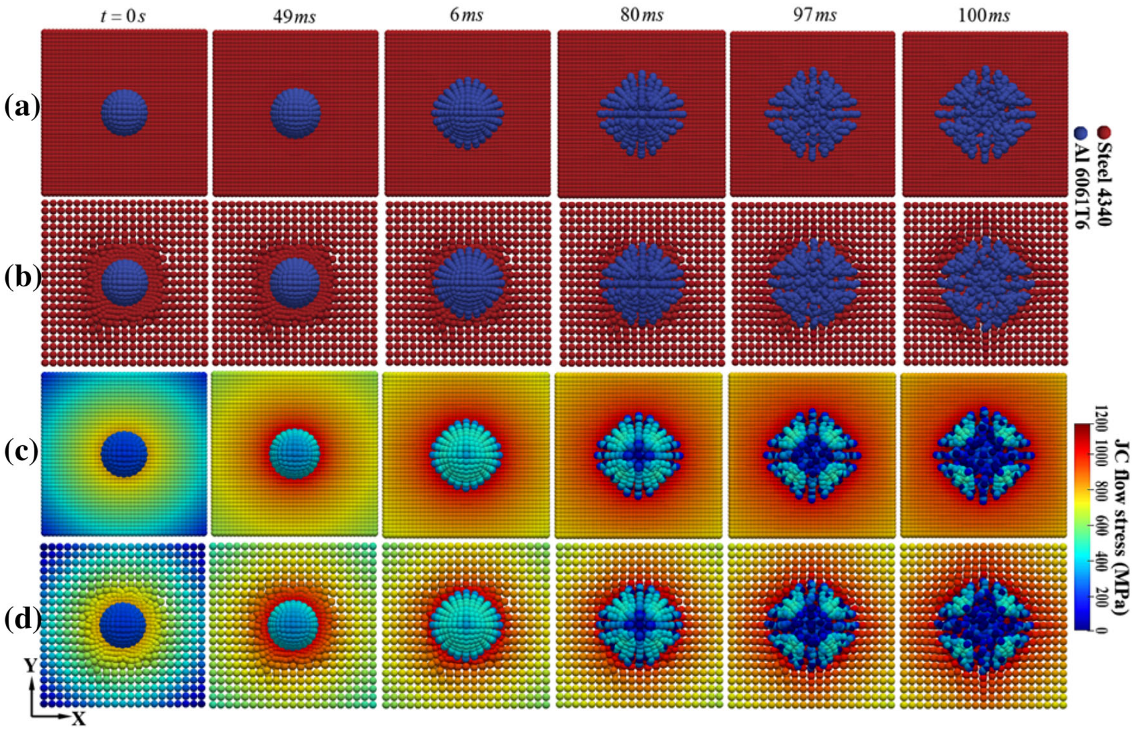


Fig. 5 Deformation response and Johnson–Cook flow stress distribution for the high-velocity impact simulation of aluminum (i.e., Al 6061-T6) projectile into steel (i.e., Steel 4340) target during 100 ms after impact initiation with the impact velocity of 1000 m/s: **a** response for the uniform particle size distribution, **b** response for the variable particle size distribution, **c** Johnson–Cook flow stress for the uniform particle size distribution, and **d** Johnson–Cook flow stress for the variable particle size distribution

$$\prec \frac{Dh}{Dt} \succ_i = -\frac{1}{d}\frac{h}{\rho} \prec \frac{D\rho}{Dt} \succ_i , \qquad (30)$$

where $\prec \ \succ$ indicates the average value over the kernel support domain for each particle.

5 Fluid–solid interaction treatment

Analysis of contact between fluid and solid in the interaction zone is an important step for successful FSI numerical simulations. In our two phase SPH problem (i.e., solid and fluid),

Table 4 Uniform and variable particle sizing configurations along with the simulation CPU time for the case study performed in Fig. 5

Particle distribution	# of projectile particles	# of target particles	Total # of particles	CPU time (s)	Speedup
Uniform particle size	343	5043	5386	968.53	3.812
Variable particle size	343	1713	2056	254.07	

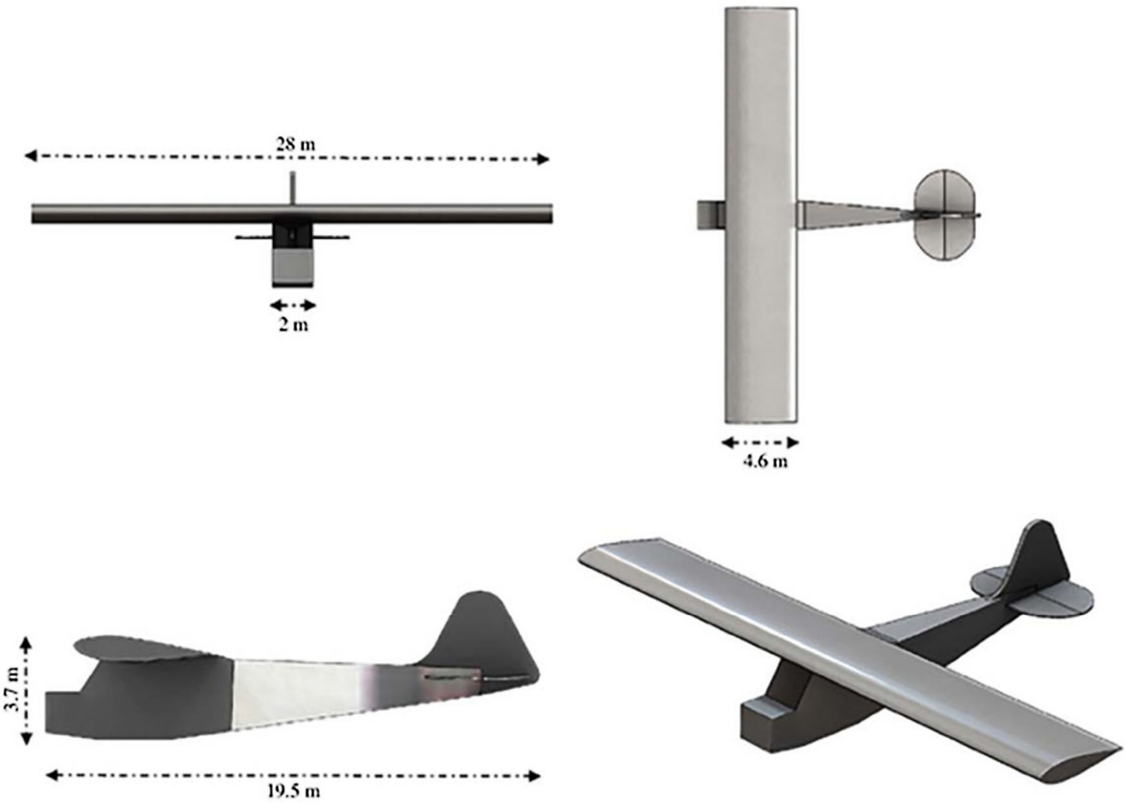


Fig. 6 Aircraft geometry in different views with the overall dimensions

we encounter three particle-type neighborhoods: (i) particles surrounded only by the fluid neighbor particles, (ii) particles surrounded only by the solid neighbor particles, and (iii) those with both fluid and solid particles within a given supporting domain. The first two types do not need any special treatment because the fluid domain equations are solved for fluid particles while the solid domain equations are solved for solid particles. As explained in Sect. 2.4.1, the CSPM correction is applied to improve free surfaces in both fluid and solid domains. The third type of particle neighbors are addressed in a different way. To this end, a new interface scheme is proposed to identify the fluid and solid particles in the domain of a third type particle and then the coupled field equations are solved. The FSI underlying interface loads result from a strong two-way interaction, where the fluid domain interacts with solid and vice versa. The momentum equation for a particle in the third type group is expressed as:

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j m_j \left(\frac{p_{i_{\text{solid}}}}{\left(\rho_{i_{\text{solid}}}\right)^2} + \frac{p_{i_{\text{fluid}}}}{\left(\rho_{j_{\text{fluid}}}\right)^2} + \Pi_{ij} \right) \frac{\partial W_{ij}\left(R_{ij}, h\right)}{\partial \mathbf{x}_i} + \sum_j m_j \left(\frac{\mathbf{s}_{i_{\text{solid}}}}{\left(\rho_{i_{\text{solid}}}\right)^2} + \frac{\boldsymbol{\tau}_{j_{\text{fluid}}}}{\left(\rho_{j_{\text{fluid}}}\right)^2} \right) \frac{\partial W_{ij}\left(R_{ij}, h\right)}{\partial \mathbf{x}_i}, \quad (31)$$

where $\mathbf{s}_{i_{\text{solid}}}$ and $\boldsymbol{\tau}_{j_{\text{fluid}}}$ are given by Eqs. (5) and (14), respectively. One of the common approaches for treating FSI, is a repulsive force (a penalty force) approach [78–80]. Magnitude of the repulsive force is controlled by an arbitrary parameter determining the interface gap. Since the exact value of this parameter is difficult to know, the penalty force approach is usually accompanied by a non-physically large gap (which is a result of a large repulsive force between two particles on both sides of interface). Moreover, the penalty approach is even more difficult to use with a variable smoothing length. Finally, since the interface evolves

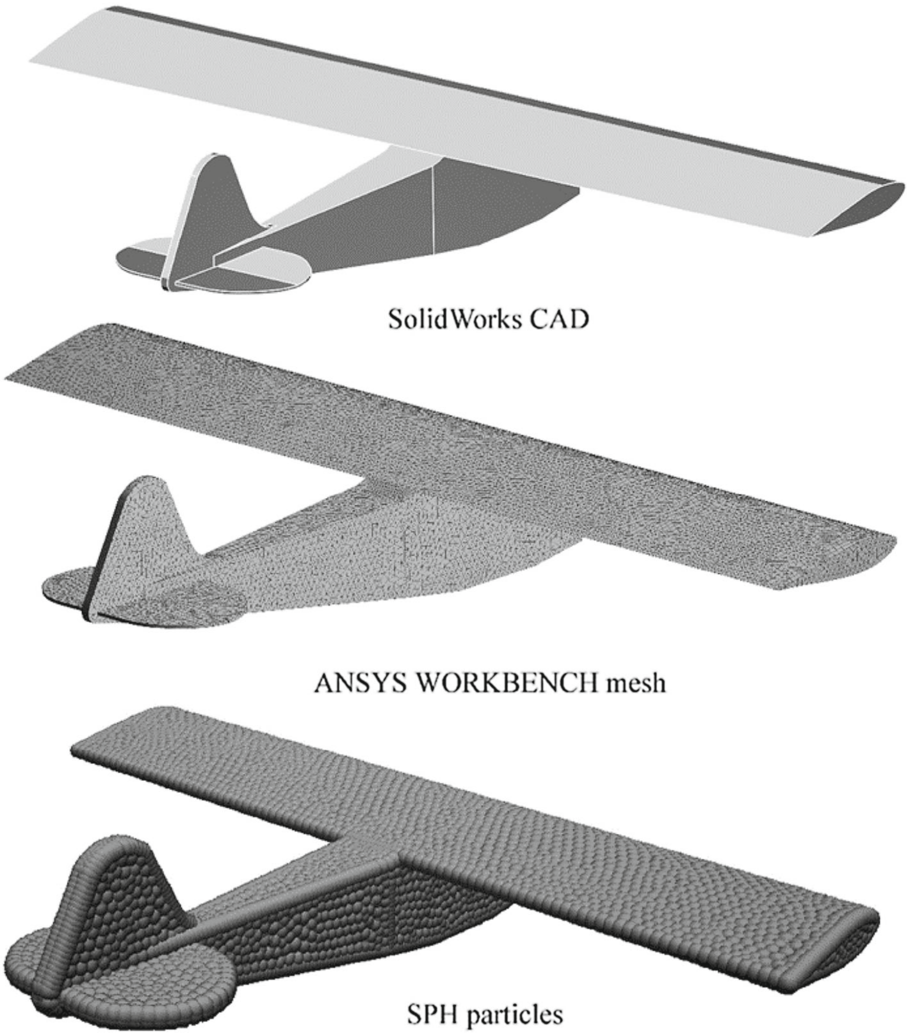


Fig. 7 A schematic of steps involved in achieving variable particle size distribution from a CAD model over meshing using ANSYS and ICEM CFD to SPH particles

during simulation, the penalty force may result in shooting particles (in the case of being very large) or may result in particle interpenetration (not large enough).

In contrast to the penalty repulsive force between interface particles, which is numerically unstable under very high pressures, the new interface scheme described above not only prevents penetration of fluid and solid particles, but also improves the gap distance between the fluid and solid boundaries in the contact regions. Figure 8 represents this improved gap for a high speed water jet impact into a thin plate of aluminum at 500 m/s for 0.5 s of the simulation. The gap is maintained at a reasonable value during the simulation without facing the problem of interparticle penetration (i.e., particle clustering) or spurious particle oscillations. Moreover, due to the consistent gap during the whole simulation, no particle shooting is experienced in the solver. The new approach takes the advantage of evolving smoothing length (i.e., Eq. 29). The smoothing length value plays a significant role in the accuracy of simulations. It has also a direct effect on the computational efficiency (i.e., with increase in smoothing length, more particles take part as the neighboring particles and in turn, the maximum number of pairs and interactions will increase, followed by more computation expense). If the smoothing length is too small, then no information will be provided from the neighbors in order to perform the kernel estimation; on the other hand, if the smoothing length is too large, then not only results are inaccurate due to overestimation but the computational efficiency is sacrificed due to more time spent in the search algorithm for finding the neighbors as well as more equations are being solved. Because the problem we are simulating is not isotropic/uniform in nature (i.e., FSI interface, large deformations resulted from high-velocity impact), choosing the smoothing length to be a variable in time and space, results in considerable increase of computational efficiency while maintaining the accuracy. As it was shown from the variable particle size study, having a non-uniform particle distribution results in less total number of interactions for and in turn less computations. The same analogy exists for the variable smoothing length, because having a smaller smoothing

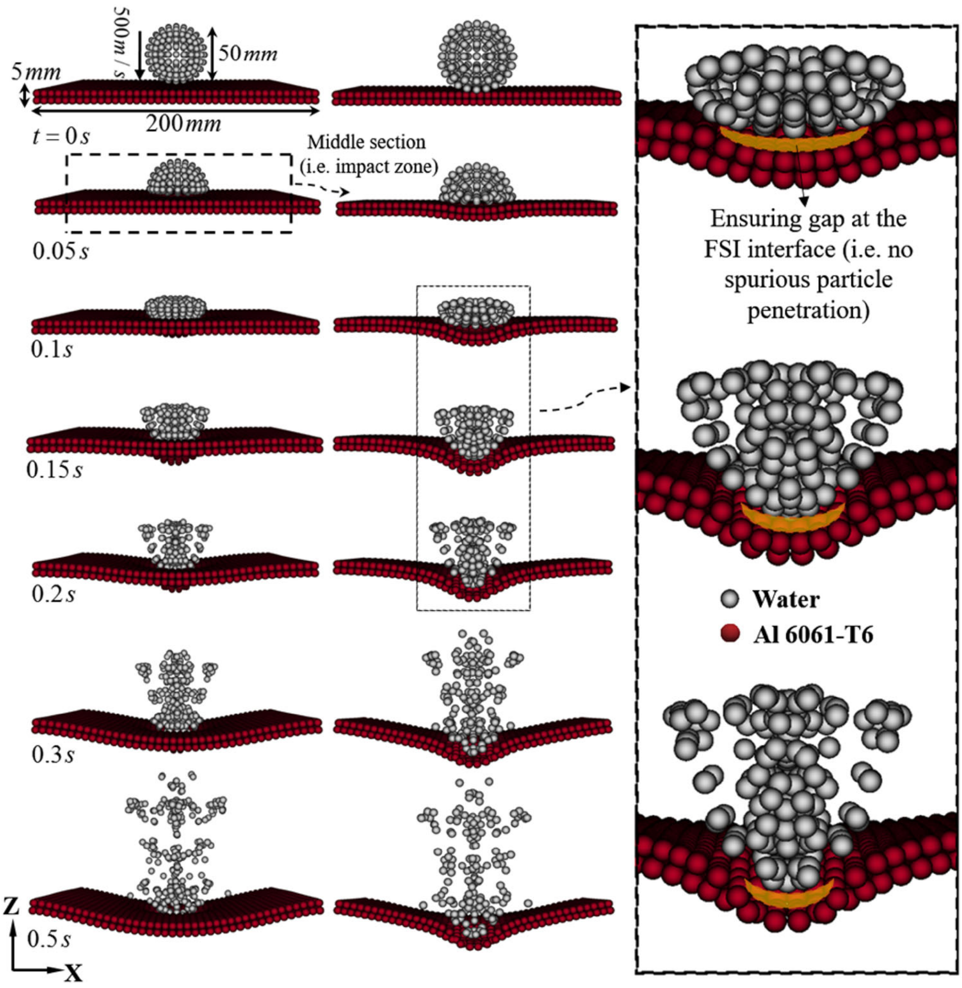


Fig. 8 High-velocity impact of water jet simulation into aluminum (i.e., Al 6061-T6). The introduced novel FSI interface scheme ensures a reasonable gap during the simulation at the water-metal FSI boundaries (i.e., no particle clustering nor inter penetration is observed)

length for a smaller particle size results in having less number of pairs and, consequently, more efficient calculations.

6 GPU-based implementation of CSPM-SPH using OpenACC

Real-life large-scale simulations in 3D SPH, necessitate a high level of computational efficiency to perform the case studies in practically acceptable times. In large-scale SPH simulations, the CPU time increases proportionally with the number of particles in the solution domain.

Power of modern GPUs to include many built-in CUDA (Compute Unified Device Architecture) cores in hardware architecture designed efficiently by NVIDIA corporation, introduces an advantage of GPUs over CPUs acceleration.

The Portland Group, Inc. (PGI) and CRAY have introduced compilers to implement the GPU hardware for massive computations. Programming in CUDA is a common approach for GPU implementation. There exist well-known SPH codes in the literature that can run on GPU using CUDA. DualSPHysics [81], GPUSPH [82], and AQUAgpusph [83] are the open-source SPH codes taking the advantage of CUDA language. Despite the fact that CUDA is very efficient and powerful, a resulting GPU code written by CUDA language is not portable for different environments and hardware architectures. The code essentially needs to be restructured and rewritten in another language (e.g., CUDA FORTRAN and CUDA C++). A CUDA code programmed and optimized for NVIDIA GPU might not have the same efficiency running on AMD Radeon graphics card and in order to optimize it for the new hardware, the code must be modified significantly.

In contrast, OpenACC application program interface is one of the efficient GPU implementation techniques to make scientific applications parallel on GPU without making significant changes to the code structure itself. OpenACC, originally developed by three major vendors CAPS [84], CRAY [85], and PGI [86], is a high level programming model based on directives that are added to annotate the code. Once the code is ported to GPU using OpenACC, user is able to either run the serial version, multicore version (i.e., OpenACC code can be treated as OPENMP at the same time addressing multi-CPUs) or a GPU version by switching the compile-time flags (i.e., "–ta=tesla" for GPU, "–ta=multicore" for multi-CPUs, and "=ta=host" for serial run). Moreover, OPENACC is capable of addressing all GPU architectures efficiently at the same time (i.e., both ATI and NVIDIA architecture can be addressed using the same developed code). Another advantages include the capability of OpenACC with OpenMP or MPI to implement multiple GPUs using a prewritten OPENMP or MPI version of the code. Instead of putting in a great deal effort to write in CUDA, OpenACC helps us to run our code on GPU as efficient as CUDA, and in turn allowing us to spend more time on tuning the code algorithm itself rather than concerning about the mere implementation of GPU.

In the present paper, in order to reduce the computation time of the 3D SPH simulation on a single machine, a loop level parallelism using OpenACC directive pragmas was implemented to run the code on GPU. The hardware we used to run our simulations was a low-end laptop graphics card with the specification shown in Table 6. Implementation of the code on high-end advanced Nvidia Tesla GPUs will be furthered investigated in our feature work.

In order to efficiently port the code to GPUs, it is mandatory to identify the hot spots (i.e., most time-consuming routines and functions) first. The PGI Performance Profiler (PGPROF) v15.10 was used to this end. Appendix A shows the results of this profiling. Using PGPROF, we were able to identify the computational intensities (i.e., ratio of computation to data transfer) for different regions, helping us to avoid huge data transfers. Generally, loops with intensities higher than 4 are suitable to run on GPU [87] while intensities lower than 1 are not good candidates for GPU parallelism, as they may result in making the code even running slower on GPU because of unnecessary amount of data transfer between the host (i.e., CPU) and the device (i.e., GPU) memories. Appendix B represents a pseudocode and shows how we used OPENACC to port our code to the GPU.

The "!$acc" pragma, indicates the abbreviation of "accelerator" operating on GPU. The "!$acc kernels" pragma defines the region that should be parallelized using GPU cores. The "!$acc data" pragma performs the data transform and management between the main system memory and the GPU dedicated memory, playing a significant role in the efficiency of the ported code (i.e., less data movement means more efficiency). It is always better to keep the data resident on the GPU as much as possible to avoid excessive data transfer. Figure 9 illustrates the block diagram and detailed hardware architecture for NVIDIA GPUs and demonstrates how communication and data transfer is performed between CPU and GPU through the PCI express channel. For more information about OPENACC features, readers are referred to [88–91].

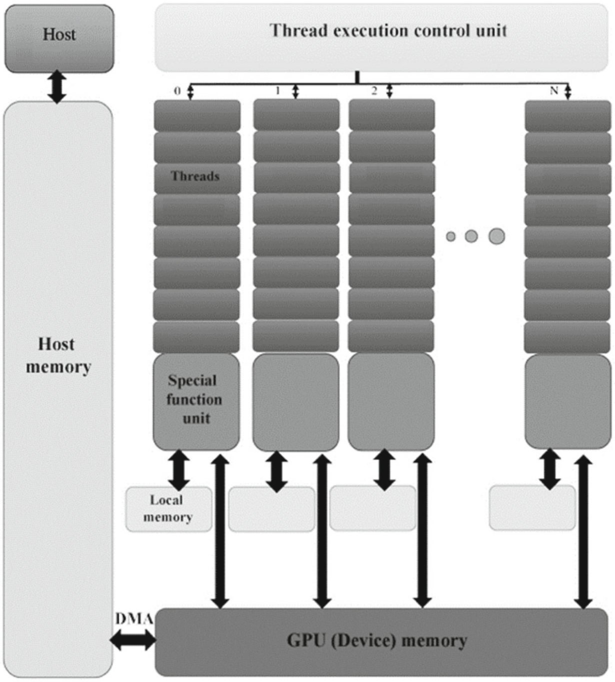


Fig. 9 Block diagram and architecture for an NVIDIA GPU including the data communication details. DMA stands for direct memory access between CPU and GPU. Numbers 0, 1, 2 . . . N represent blocks of threads. Local memory is a part of global device memory specific to each thread block. Control unit performs the operations and calculations as instructed by the computer program

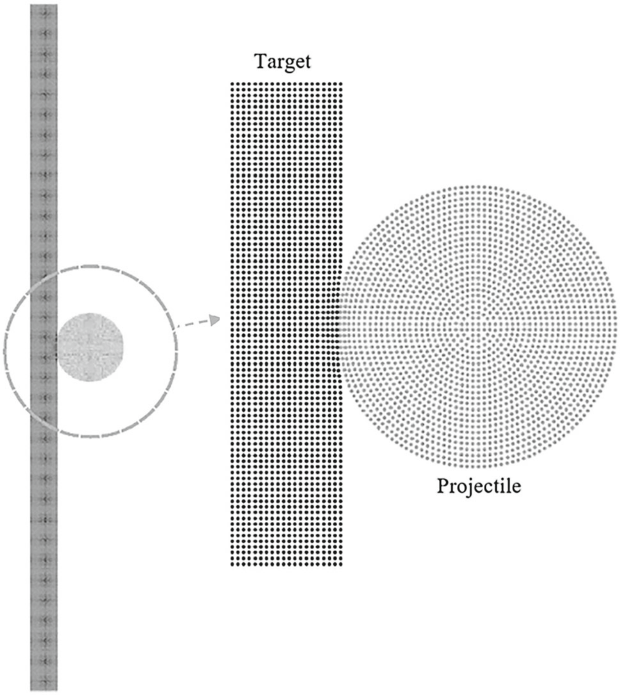


Fig. 10 Initial impact configuration for the benchmark test

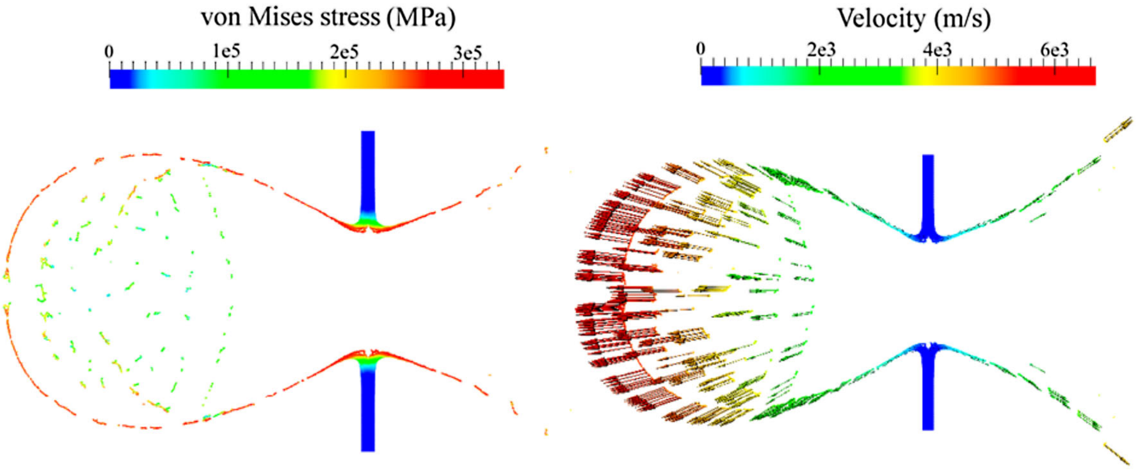


Fig. 11 Distribution of von Miss stress and velocity in resulted debris 20 μs after impact

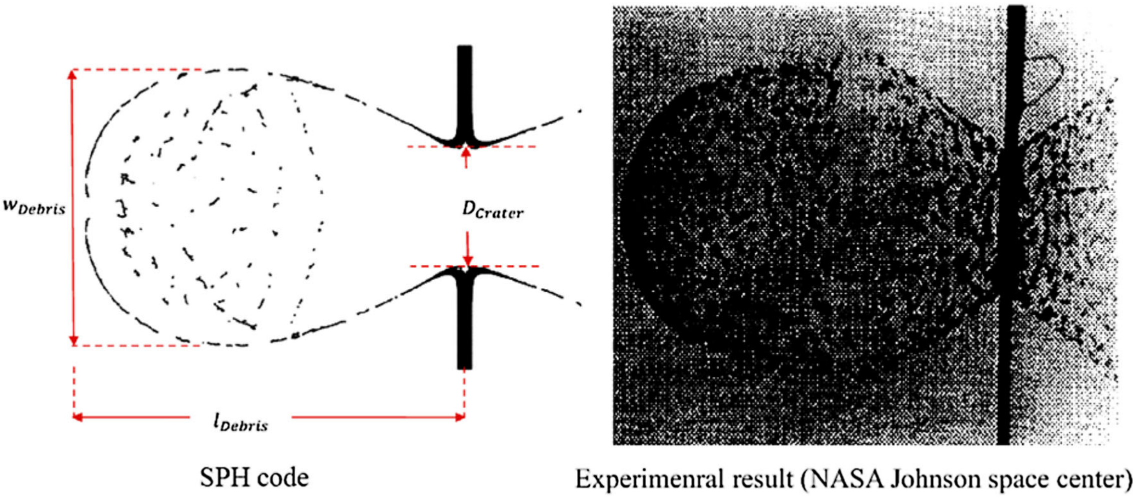


Fig. 12 Experimental data used to benchmark the new CSPM-SCP implementation [92]

7 Model validation with experimental measurements

The developed SPH code was validated using the experimental data taken from [92]. Specifically, the numerical simulation was performed to study the crater and debris resulted from a hyper velocity impact of a projectile into a thin armor plate. The initial configuration of the SPH simulation is illustrated in Fig. 10. Both projectile and target are made of Al 6061-T6. Projectile collides into the plate with the initial velocity of 6180 m/s. The projectile is a sphere with a diameter of 1 cm, while the plate target has a length of 10 cm and thickness of 0.4 cm. Initial SPH particle spacing for domain discretization is considered to be 0.2 mm. Therefore, projectile consists of 1956 particles, while the target plate was represented using 10,000 particles. The impact simulation was run for 20 μs after the impact initiation. Figure 11 shows the velocity vector field and von Mises stress distribution 20 μs after impact initiation. Figure 12 represents the comparison of crater and debris dimensions for the SPH simulation using our code versus experiments carried out using high-pressure light-gas guns at NASA Johnson space center [93]. Hyper velocity impact-induced crater diameter, length, and width of the produced debris cloud for both SPH and experimental setup are presented in Table 5. We regard the SPH simulation results to be in good agreement with the experimental data.

Table 5 Comparison of crater diameter and length/width ratio of debris in impact of Al 6061-T6 sphere into Al 6061-T6 plate. The values are in *cm*

	SPH results	Experimental data
Crate diameter	3.5	3.45
Debris length to width ratio	1.4	1.39

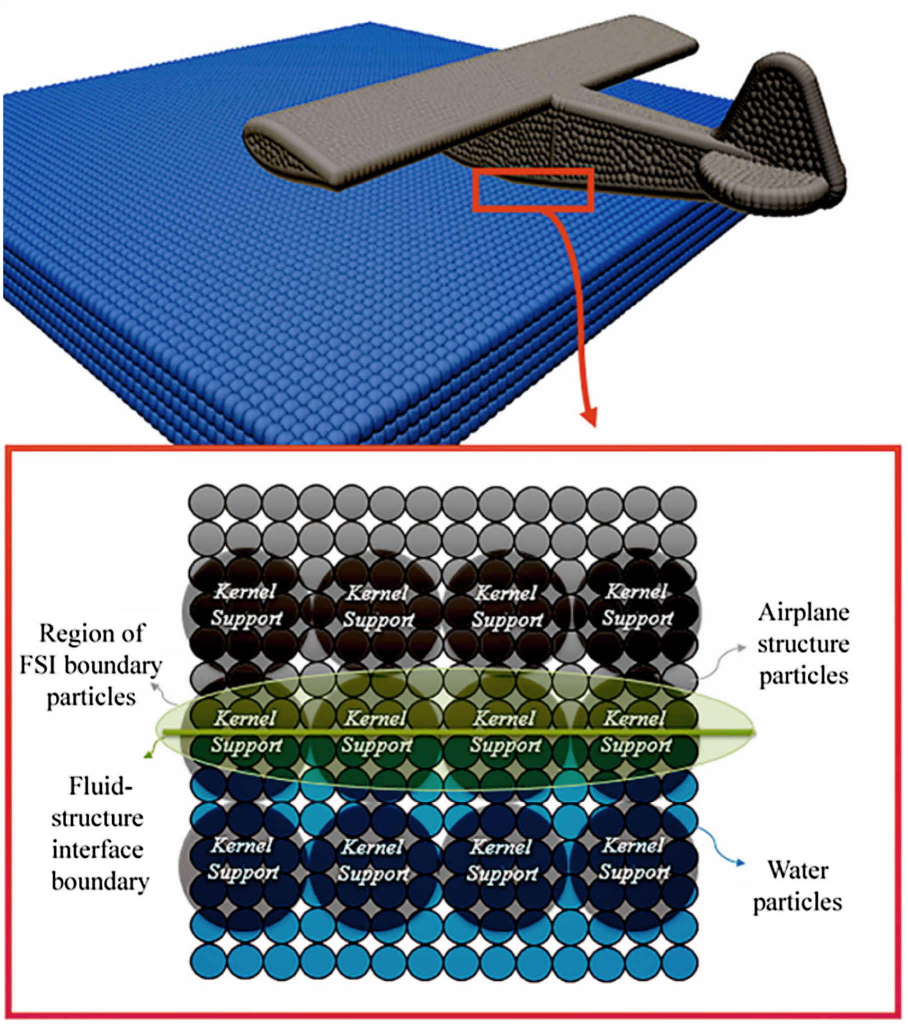

Kernel Support
Region of FSI boundary particles
Airplane structure particles
Fluid-structure interface boundary
Water particles

Fig. 13 Particle representation within kernel support domains at the aircraft structure/water interface

Table 6 Platform configuration for GPU parallelization

CPU	Corei7 ivy-bridge @ 3GHZ
GPU	GTX660m
OS	Windows 7
RAM	8GB
GPU driver version	NVidia-latest (WHQL)
CUDA toolkit	7.5
Host compiler	PGI 15.10

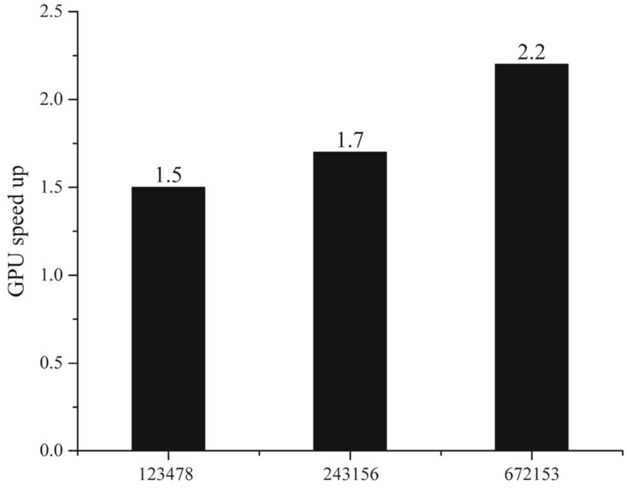


Fig. 14 OpenACC-GPU speedup on GTX 660m laptop GPU over a serial execution code for a given number of particles

8 Aircraft dynamic structural response under high-velocity impact load of water-ditching case study

We begin the presentation of the application case study by showing in Fig. 13 the kernel support for FSI of an aircraft while crashing into the water surface. The particle neighbors type used for the simulation of FSI have been discussed earlier in the text and are also illustrated in the figure.

The collision of falling aircraft into the water occurs at different impact angles. In order to make the case study more comprehensive, the simulations were performed for 30°, 45°

Fig. 15 Distribution of velocity as a function of impact angle: **a** 30°, **b** 45°, and **c** 60°

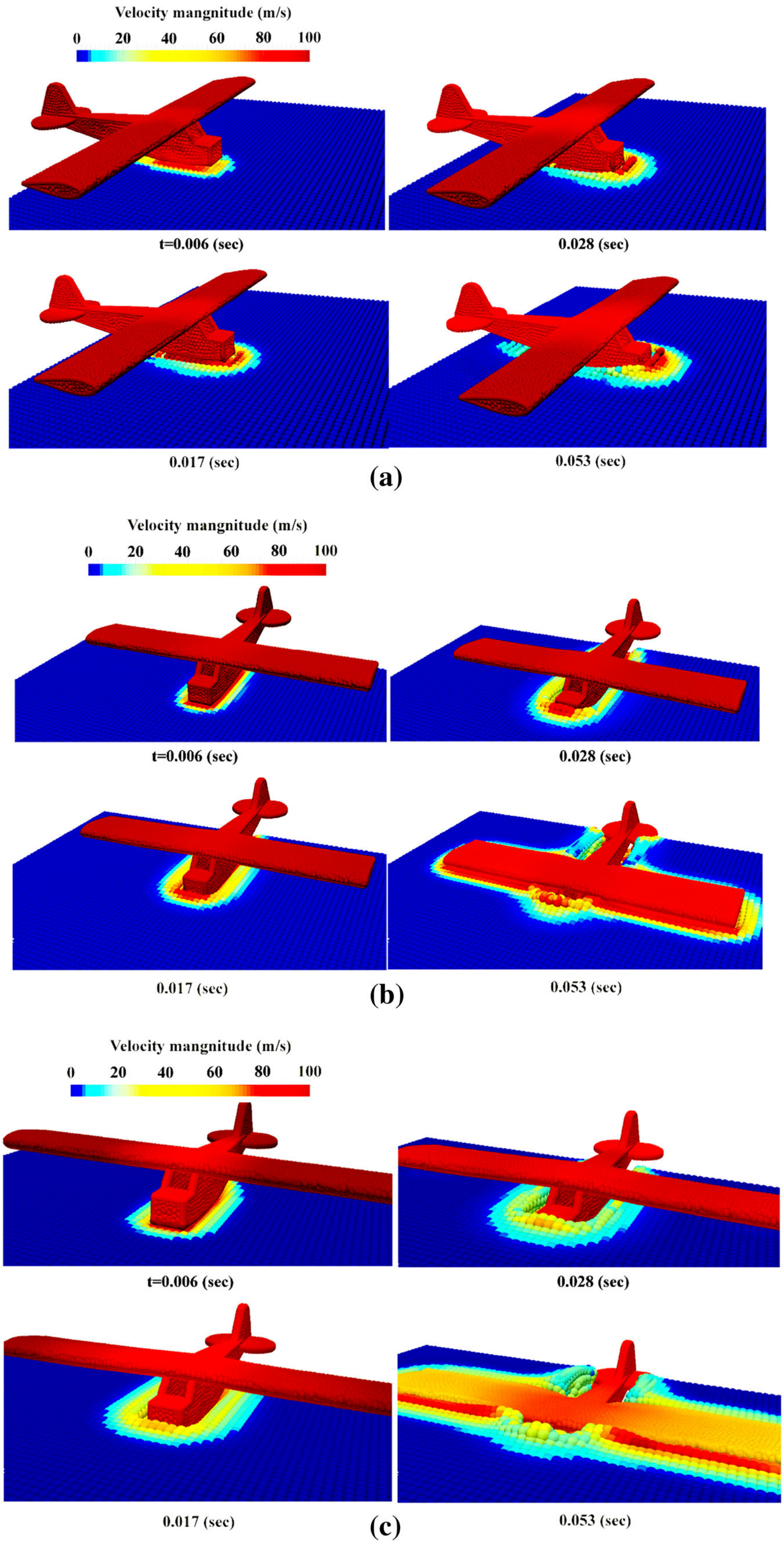


Fig. 16 Distribution of pressure as a function of impact angle: **a** 30°, **b** 45°, and **c** 60°

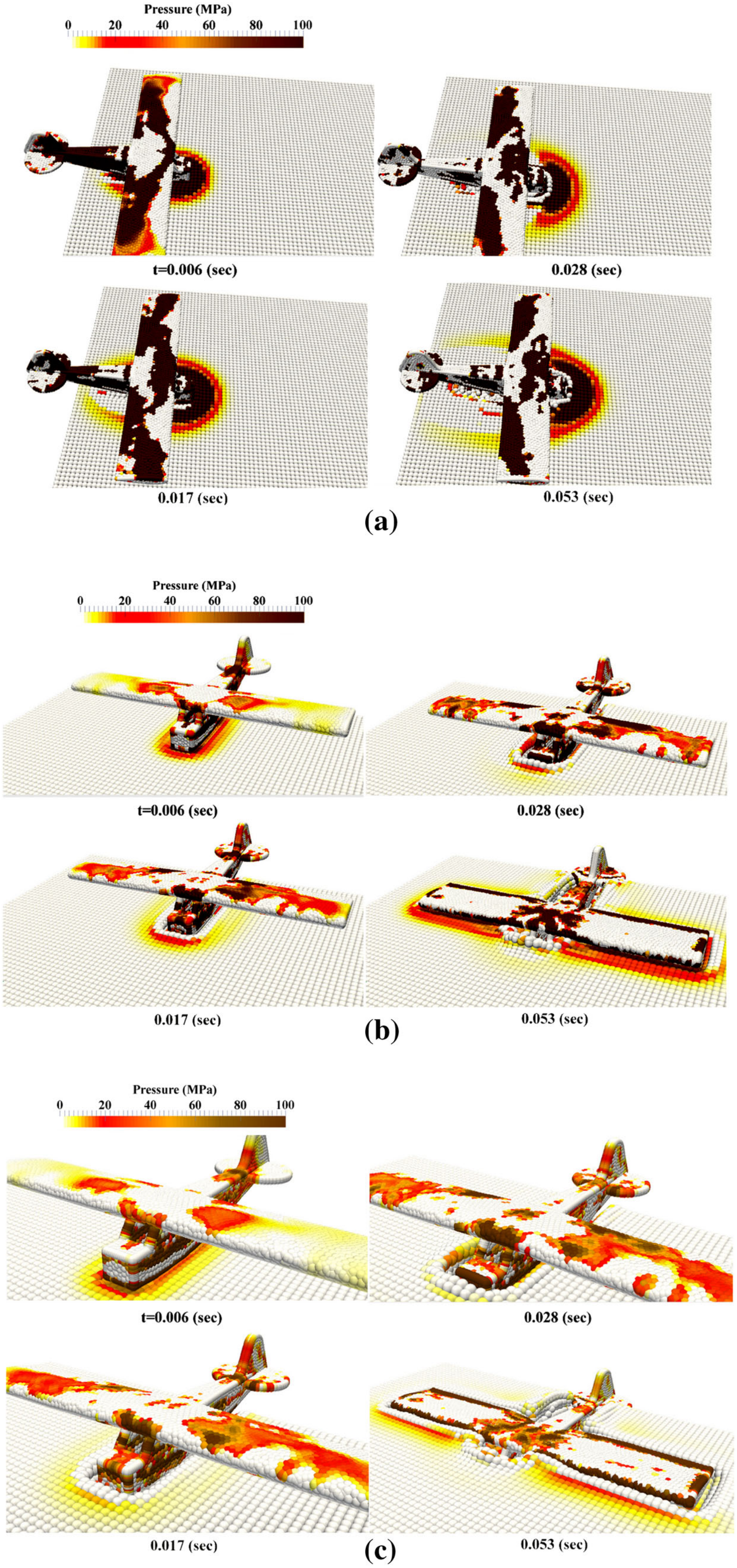
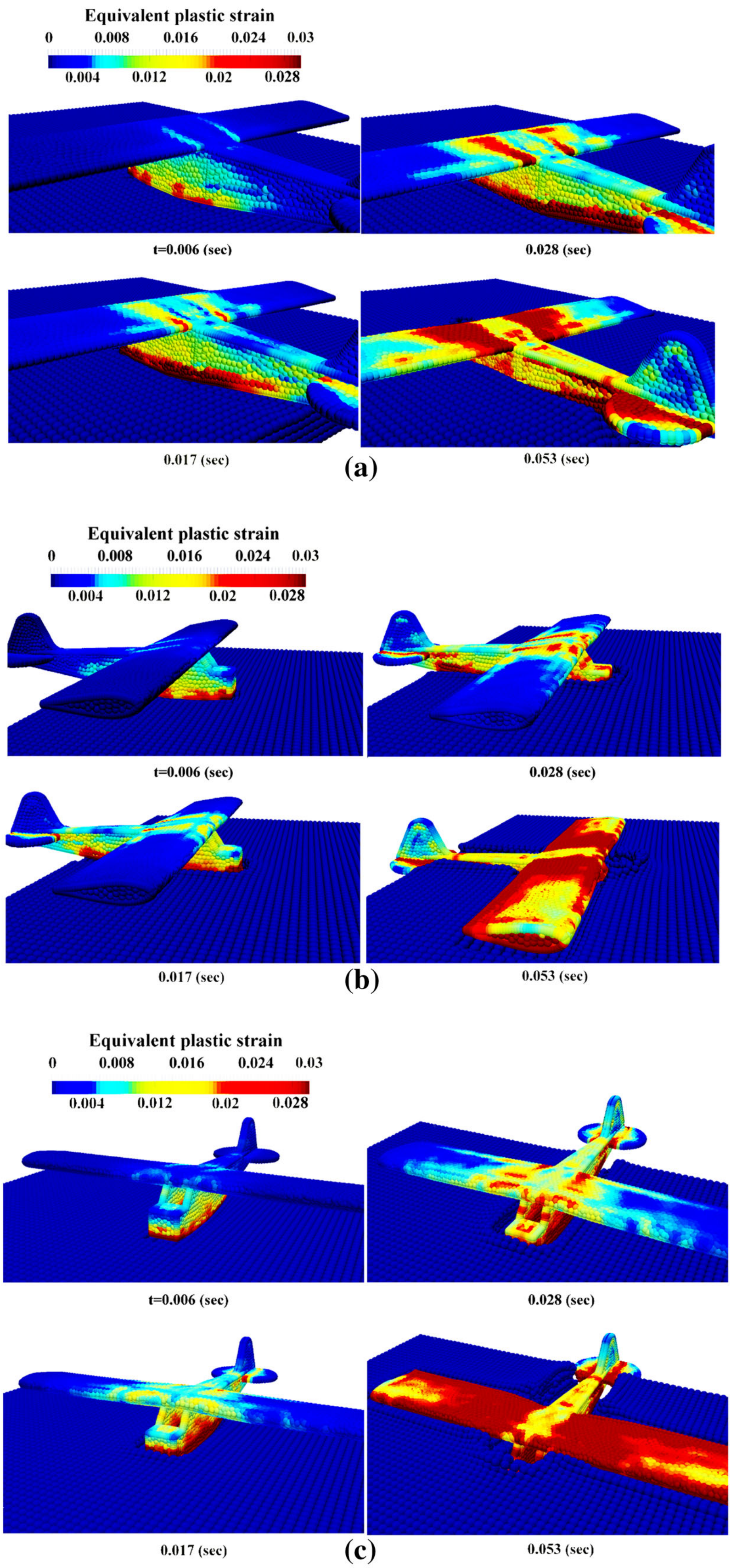


Fig. 17 Distribution of equivalent plastic strain as a function of impact angle: **a** 30°, **b** 45°, and **c** 60°



and 60° angles of crash relative to the water surface. Table 6 represents the platform configuration for our simulations. With the use of heterogeneous particle resolution and total number of 123,478, 243,156, and 672,153 particles, respectively, the SPH was run on the GPU, and 1.5×, 1.7×, and 2.2× speedups, respectively, were obtained comparing the serial execution. The speed up information is provided in Fig. 14 for better illustration. We emphasize that the GPU runs performed in this work are aimed at proving the concept rather than obtaining large speed ups. The GTX 660m on which the simulations were run is a low end laptop graphics card, which is not designed for high performance computing. For obtaining large speed-ups, modern GPU platforms (i.e., TESLA K80 and P100) are necessary.

Figures 15, 16 and 17 represent the distribution of velocity, pressure, and equivalent plastic strain contours in the aircraft structure for the collision angles of 30°, 45°, and 60°, respectively. For a better illustration, three simulations movies are also provided in the supplementary material. The reason behind choosing such angles is to cover the most probable crashing angles. Crashes at 0° or 90° relative to the water surface are not expected to happen. Visualization of the results was carried out in PARAVIEW [94]. It can be observed that the magnitude of the fields increases with an increase in the angle from 30° to 60°. From the detailed analysis of carried out simulations, it can be understood that SPH with the novel FSI method can capture the fields in the anticipated zones of localizations (i.e., evident from Figs. 15, 16 and 17). The non-uniform particle distribution of the SPH algorithm implemented in this research, captures the different length scales of simulation followed by smooth predicted contours (i.e., no discontinuities is observed in the pressure, velocity, and strain fields). It is worth pointing out that the FEM simulations would not be able to predict such smooth interface characteristics due to the extreme mesh distortions at the impact region. Using FEM solvers require frequent remeshing leading to inaccurate predictions due to the interpolations of variables from old to new mesh.

9 Conclusions

This work presented a CSPM within SPH considering a heterogeneous particle resolution applied to an aircraft structure under high-velocity impact load. A new FSI algorithm ensuring no interparticle penetration is implemented to facilitate simulations of high-velocity impact. Furthermore, a procedure for generating variable particle size over any complex input geometry is developed. The heterogeneous particle resolution is achieved for an arbitrary geometry from a non-uniform finite element mesh generated using the ANSYS WORKBENCH meshing module in conjunction with the ICEM CFD software. Finally, the SPH code is ported to run on GPU using the OpenACC standard to accelerate the calculations. Results show that the novel FSI algorithm together with the non-uniform SPH particle distribution is well suited for modeling the deformation of structures under high-velocity crash and impact into the water as evident from the predicted stress and strain localizations in the aircraft structure critical regions as well as the integrity of the impacted interfaces, which showed no artificial particle penetrations. Feature efforts will implement the present SPH package on multiple NVIDIA Tesla GPUs to facilitate more detailed large-scale simulations. Another point of interest for the future research is to enable the SPH code to carry out the simulations using advanced crystal plasticity constitutive laws taking into account microstructure evolution of polycrystalline materials under high-velocity impact loading conditions [95–99].

Acknowledgements This work is based upon a project supported by the US National Science Foundation under grant no. CMMI-1650641. The authors gratefully acknowledge this support.

Compliance with ethical standards

Conflicts of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix A

The % time spent in subroutines ordered from top to bottom as identified using PGPROF:

1. Subroutine Direct_find (neighbor search subroutine, 38% of total execution time)
2. Subroutine Int_Force (force calculation subroutine, 23% of total execution time)
3. Subroutine Time_Intg (time integration subroutine, 12% of total execution time)
4. Main program SPH (main program, 9% of total execution time)
5. Subroutine Cont_Density (continuity subroutine including the CSPM modification, 8% of total execution time)
6. Subroutine H_Upgrade (update smoothing length subroutine, 5% of total execution time)
7. Other subroutines (5% of total execution time)

In this work, subroutines 1–5 were ported to GPU.

Appendix B

(i) Neighbor particles search within computational domain:

```
!$acc data copyin (Input Variables [particle position, particle density, particle velocity,
…]) copyout & !$acc  (Output Variables [particle acceleration, particle force, … ]) copy
(Input/output Variables)
kernels loop Private (Private Variables) Reduction ([+,−,× ]:Reduction Variable)
Do i=1, Maximum interactions
[Perform Neighbor Search]
EndDo
!$acc end kernels
!$acc end data
```

(ii) Force calculation:

```
!$acc data copyin (Input Variables [particle position, particle density, particle velocity,
…]) copyout & !$acc (Output Variables [particle acceleration, particle force, … ]) copy
(Input/output Variables)
!$acc kernels loop Private (Private Variables) Reduction ([+,−,× ]:Reduction Variable)
Do i=1, Total Number of Domain Particles
[                                        ]
 EndDo
!$acc end kernels
!$acc end data
```

(iii) Time integration:

```
!$acc data copyin (Input Variables [particle position, particle density, particle velocity,
…]) copyout  &         !$acc (Output Variables [particle acceleration, particle force, … ])
copy (Input/output Variables)
!$acc kernels loop Private (Private Variables) Reduction ([+,−,× ]:Reduction Variable)
Do i=1, Total Number of Domain Particles
[Perform Time Integration]
EndDo
!$acc end kernels
!$acc end data
```

(iv) Continuity:

```
!$acc data copyin (Input Variables [particle position, particle density, particle velocity,
…]) copyout  &         !$acc (Output Variables [particle acceleration, particle force, … ])
copy (Input/output Variables)
!$acc kernels loop Private (Private Variables) Reduction ([+,−,× ]:Reduction Variable)
Do i=1, Total Number of Domain Particles
[Perform continuity equations]
EndDo
!$acc end kernels
!$acc end data
```

(v) SPH main program:

```
!$acc data copyin (Input Variables [particle position, particle density, particle velocity,
…]) copyout  &         !$acc (Output Variables [particle acceleration, particle force, … ])
copy (Input/output Variables)
!$acc kernels loop Private (Private Variables) Reduction ([+,−,× ]:Reduction Variable)
Do i=1, Total Number of Domain Particles
[Perform calls to the routines]
EndDo
!$acc end kernels
!$acc end data
```

A loop from the continuity subroutine is presented below to better illustrate how the OpenACC data and kernels directives can be used to run the loop in parallel on GPU. The "reduction" and "private" clauses ensure that there are no race conditions while accessing the summation over the scalar "vcc" using the GPU threads. Additionally, "copyin" clauses show the arrays data input from CPU (host) to the device (GPU).

```
!$acc data copyin (vx,rho,mass,pair_i,pair_j,dwdx) copy (drhodt)
!$acc kernels loop Private (k,d) Reduction (+:vcc)
Do k=1,niac
i = pair_i(k)
j = pair_j(k)
Do d=1,dim
dvx(d) = vx(d,i) - vx(d,j)
EndDo
vcc = dvx(1)*dwdx(1,k)
Do d=2,dim
vcc = vcc + dvx(d)*dwdx(d,k)
EndDo
drhodt(i) = drhodt(i) + mass(j)*vcc/rho(j)
drhodt(j) = drhodt(j) + mass(i)*vcc/rho(i)
EndDo
!$acc end kernels
!$acc end data
```

In above loop, "x", "vx", "rho", "drhodt", "pair_i", "pair_j", "niac", "dwdx", and "dim" represent particle position, velocity, density, time rate of density, neighbor particle "i" interacting with particle "j", neighbor particle "j" interacting with particle "i", gradient of kernel function, total number of interacting pairs, and domain dimension, respectively.

Supplementary material

Movies showing the evolution of pressure, equivalent plastic strain, and von Mises stress during aircraft water ditching at an angle of 60°.

References

1. Malan A, Oxtoby O (2013) An accelerated, fully-coupled, parallel 3d hybrid finite-volume fluid–structure interaction scheme. Comput Methods Appl Mech Eng 253:426–438
2. Hesch C, Gil A, Carreno AA, Bonet J (2012) On continuum immersed strategies for fluid–structure interaction. Comput Methods Appl Mech Eng 247:51–64
3. Koh HM, Kim JK, Park JH (1998) Fluid–structure interaction analysis of 3-D rectangular tanks by a variationally coupled BEM–FEM and comparison with test results. Earthq Eng Struct Dyn 27:109–124
4. Tong Z, Zhang Y, Zhang Z, Hua H (2007) Dynamic behavior and sound transmission analysis of a fluid–structure coupled system using the direct-BEM/FEM. J Sound Vib 299:645–655
5. Czygan O, Von Estorff O (2002) Fluid–structure interaction by coupling BEM and nonlinear FEM. Eng Anal Bound Elem 26:773–779
6. Soares D Jr, Von Estorff O, Mansur W (2005) Efficient non-linear solid–fluid interaction analysis by an iterative BEM/FEM coupling. Int J Numer Methods Eng 64:1416–1431
7. He Z, Liu G, Zhong Z, Zhang G, Cheng A (2011) A coupled ES-FEM/BEM method for fluid–structure interaction problems. Eng Anal Bound Elem 35:140–147
8. He T, Zhou D, Bao Y (2012) Combined interface boundary condition method for fluid–rigid body interaction. Comput Methods Appl Mech Eng 223:81–102
9. Heimbs S (2011) Computational methods for bird strike simulations: a review. Comput Struct 89:2093–2112
10. Georgiadis S, Gunnion AJ, Thomson RS, Cartwright BK (2008) Bird-strike simulation for certification of the Boeing 787 composite moveable trailing edge. Compos Struct 86:258–268

11. Smojver I, Ivančević D (2011) Bird strike damage analysis in aircraft structures using Abaqus/Explicit and coupled Eulerian Lagrangian approach. Compos Sci Technol 71:489–498
12. Smojver I, Ivančević D (2010) Numerical simulation of bird strike damage prediction in airplane flap structure. Compos Struct 92:2016–2026
13. Lucy LB (1977) A numerical approach to the testing of the fission hypothesis. Astron J 82:1013–1024
14. Gingold RA, Monaghan JJ (1977) Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon Not R Astron Soc 181:375–389
15. Monaghan J, Kocharyan A (1995) SPH simulation of multi-phase flow. Comput Phys Commun 87:225–235
16. Chen Z, Zong Z, Liu M, Zou L, Li H, Shu C (2015) An SPH model for multiphase flows with complex interfaces and large density differences. J Comput Phys 283:169–188
17. Tartakovsky AM, Panchenko A (2016) Pairwise force smoothed particle hydrodynamics model for multiphase flow: surface tension and contact line dynamics. J Comput Phys 305:1119–1146
18. Ming FR, Sun PN, Zhang AM (2017) Numerical investigation of rising bubbles bursting at a free surface through a multiphase SPH model. Meccanica 52:2665–2684
19. Tong M, Browne DJ (2014) An incompressible multi-phase smoothed particle hydrodynamics (SPH) method for modelling thermocapillary flow. Int J Heat Mass Transf 73:284–292
20. Morris JP, Fox PJ, Zhu Y (1997) Modeling low Reynolds number incompressible flows using SPH. J Comput Phys 136:214–226
21. Monaghan JJ (1994) Simulating free surface flows with SPH. J Comput Phys 110:399–406
22. Peng C, Xu G, Wu W, H-s Yu, Wang C (2017) Multiphase SPH modeling of free surface flow in porous media with variable porosity. Comput Geotech 81:239–248
23. Violeau D, Rogers BD (2016) Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future. J Hydraul Res 54:1–26
24. Bankole AO, Dumbser M, Iske A, Rung T (2017) A meshfree semi-implicit smoothed particle hydrodynamics method for free surface flow. In: Griebel M, Schweitzer MA (eds) Meshfree methods for partial differential equations VIII. Springer, Berlin, pp 35–52
25. Abdelrazek AM, Kimura I, Shimizu Y (2016) Simulation of three-dimensional rapid free-surface granular flow past different types of obstructions using the SPH method. J Glaciol 62:335–347
26. Benz W, Asphaug E (1993) Explicit 3D continuum fracture modeling with smooth particle hydrodynamics. Lunar and planetary science conference
27. Benz W, Asphaug E (1995) Simulations of brittle solids using smooth particle hydrodynamics. Comput Phys Commun 87:253–265
28. Das R, Cleary P (2013) A mesh-free approach for fracture modelling of gravity dams under earthquake. Int J Fract 179:9–33
29. Monaghan J (2012) Smoothed particle hydrodynamics and its diverse applications. Annu Rev Fluid Mech 44:323–346
30. Caleyron F, Combescure A, Faucher V, Potapov S (2012) Dynamic simulation of damage–fracture transition in smoothed particles hydrodynamics shells. Int J Numer Methods Eng 90:707–738
31. Eghtesad A, Shafiei AR, Mahzoon M (2012) Study of dynamic behavior of ceramic-metal FGM under high velocity impact conditions using CSPM method. Appl Math Model 36:2724–2738
32. Eghtesad A, Shafiei AR, Mahzoon M (2011) Predicting fracture and fragmentation in ceramic using a thermo-mechanical basis. Theor Appl Fract Mech 56:68–78
33. Mehra V, Sijoy C, Mishra V, Chaturvedi S (2012) Tensile instability and artificial stresses in impact problems in SPH. In: Journal of physics: conference series. IOP Publishing, pp 012102
34. Johnson GR, Petersen EH, Stryk RA (1993) Incorporation of an SPH option into the EPIC code for a wide range of high velocity impact computations. Int J Impact Eng 14:385–394
35. Randles P, Libersky L (2000) Normalized SPH with stress points. Int J Numer Methods Eng 48:1445–1462
36. Chen J, Beraun J, Jih C (1999) An improvement for tensile instability in smoothed particle hydrodynamics. Comput Mech 23:279–287
37. Gao R, Ren B, Wang G, Wang Y (2012) Numerical modelling of regular wave slamming on subface of open-piled structures with the corrected SPH method. Appl Ocean Res 34:173–186
38. Xu F, Zhao Y, Yan R, Furukawa T (2013) Multidimensional discontinuous SPH method and its application to metal penetration analysis. Int J Numer Methods Eng 93:1125–1146
39. Korzilius S, Schilders W, Anthonissen M (2016) An improved CSPM approach for accurate second-derivative approximations with SPH. J Appl Math Phys 5:168
40. Eghtesad A, Shafiei A, Mahzoon M (2012) Study of dynamic behavior of ceramic–metal FGM under high velocity impact conditions using CSPM method. Appl Math Model 36:2724–2738
41. Rabczuk T, Gracie R, Song J-H, Belytschko T (2010) Immersed particle method for fluid–structure interaction. Int J Numer Methods Eng 22:48
42. Gong K, Shao S, Liu H, Wang B, Tan S-K (2016) Two-phase SPH simulation of fluid–structure interactions. J Fluids Struct 65:155–179
43. Dai Z, Huang Y, Cheng H, Xu Q (2017) SPH model for fluid–structure interaction and its application to debris flow impact estimation. Landslides 14:917–928
44. Stasch J, Avci B, Wriggers P (2016) Numerical simulation of fluid–structure interaction problems by a coupled SPH–FEM approach. PAMM 16:491–492
45. Liu X, Shao S, Lin P, Tan S (2016) 2D numerical ISPH wave tank for complex fluid–structure coupling problems. Int J Offshore Polar Eng 26:26–32
46. Groenenboom P, Siemann M (2016) Fluid-structure interaction by the mixed SPH-FE method with application to aircraft ditching. Int J Multiphys 9:249–265
47. Zhu B, Gu L, Peng X, Zhou Z (2010) A point-based simulation framework for minimally invasive surgery. In: International symposium on biomedical simulation. Springer, Berlin, pp 130–138
48. Pan W, Li D, Tartakovsky AM, Ahzi S, Khraisheh M, Khaleel M (2013) A new smoothed particle hydrodynamics non-Newtonian model for friction stir welding: process modeling and simulation of microstructure evolution in a magnesium alloy. Int J Plast 48:189–204
49. Ma A, Hartmaier A (2016) A crystal plasticity smooth-particle hydrodynamics approach and its application to equal-channel angular pressing simulation. Model Simul Mater Sci Eng 24:085011
50. Liu G-R, Liu MB (2003) Smoothed particle hydrodynamics: a meshfree particle method. World Scientific, Singapore
51. Monaghan J, Gingold R (1983) Shock simulation by the particle method SPH. J Comput Phys 52:374–389
52. Rice W, Paardekooper S-J, Forgan D, Armitage P (2014) Convergence of simulations of self-gravitating accretion discs-II. Sensitivity to the implementation of radiative cooling and artificial viscosity. Mon Not R Astron Soc 438:1593–1602
53. Li X, Zhang T, Zhang Y, Liu G (2014) Artificial viscosity in smoothed particle hydrodynamics simulation of sound interference. In: Proceedings of meetings on acoustics 168ASA ASA, p 040005
54. Li X, Zhang T, Zhang YO (2015) Time domain simulation of sound waves using smoothed particle hydrodynamics algorithm with artificial viscosity. Algorithms 8:321–335

55. Monaghan J (1989) On the problem of penetration in particle methods. J Comput Phys 82:1–15
56. Takahashi T, Fujishiro I, Nishita T (2014) A velocity correcting method for volume preserving viscoelastic fluids. In: Proceedings of the computer graphics international
57. Maindl TI (2013) SPH for simulating impacts and collisions. University of Vienna, Vienna
58. Deb D, Pramanik R (2013) Failure process of brittle rock using smoothed particle hydrodynamics. J Eng Mech 139:1551–1565
59. Monaghan JJ (1992) Smoothed particle hydrodynamics. Annu Rev Astron Astrophys 30:543–574
60. Imaeda Y, S-i Inutsuka (2002) Shear flows in smoothed particle hydrodynamics. Astrophys J 569:501
61. Paiva A, Petronetto F, Lewiner T, Tavares G (2009) Particle-based viscoplastic fluid/solid simulation. Comput Aided Des 41:306–314
62. Crespo AJC, Domínguez JM, Rogers BD, Gómez-Gesteira M, Longshaw S, Canelas R, Vacondio R, Barreiro A, García-Feal O (2015) DualSPHysics: open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH). Comput Phys Commun 187:204–216
63. Paredes RJ, Imas L (2014) Application of multiphase SPH to fluid structure interaction problems. In: 9th international SPHERIC workshop. CNAM Paris, France
64. Dyka C, Ingel R (1995) An approach for tension instability in smoothed particle hydrodynamics (SPH). Comput Struct 57:573–580
65. Monaghan JJ (2000) SPH without a tensile instability. J Comput Phys 159:290–311
66. Korzilius S, Schilders W, Anthonissen M (2015) An improved CSPM approximation for multi-dimensional second-order derivatives. In: Proceedings of the 10th international SPHERIC workshop
67. Chen J, Beraun J, Carney T (1999) A corrective smoothed particle method for boundary value problems in heat conduction. Int J Numer Methods Eng 46:231–252
68. Crespo AJC (2008) Application of the smoothed particle hydrodynamics model SPHysics to free surface hydrodynamics. Universidade de Vigo, Vigo
69. Davison L (2008) Fundamentals of shock wave propagation in solids. Springer, Berlin
70. Ben-Dor G, Igra O, Elperin T (2000) Handbook of shock waves, three volume set. Academic Press, New York
71. Zharkov VN, Kalinin VA, Tybulewicz A (1971) Equations of state for solids at high pressures and temperatures. Springer, Berlin
72. Raizer YP (2002) Physics of shock waves and high-temperature hydrodynamic phenomena. Courier Corporation, North Chelmsford
73. Goda T (2017) Numerical study on seepage-induced failure of Caisson type breakwaters using a stabilized ISPH. TVVR 17/5007
74. Zheng X, Shao S, Khayyer A, Duan W, Ma Q, Liao K (2017) Corrected first-order derivative ISPH in water wave simulations. Coast Eng J 59:1750010
75. Pahar G, Dhar A (2017) Coupled incompressible smoothed particle hydrodynamics model for continuum-based modelling sediment transport. Adv Water Resour 102:84–98
76. Johnson GR, Cook WH (1983) A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. In: Proceedings of the 7th international symposium on ballistics. The Hague, The Netherlands, pp 541–547
77. SolidWorks I (2002) Solidworks Corporation, Concord
78. Amini Y, Emdad H, Farid M (2011) A new model to solve fluid–hypo-elastic solid interaction using the smoothed particle hydrodynamics (SPH) method. Eur J Mech B Fluids 30:184–194
79. Yang Q, Jones V, McCue L (2012) Free-surface flow interactions with deformable structures using an SPH-FEM model. Ocean Eng 55:136–147
80. Rafiee A, Thiagarajan KP (2009) An SPH projection method for simulating fluid–hypoelastic structure interaction. Comput Methods Appl Mech Eng 198:2785–2795
81. DualSPHysics. www.dual.sphysics.org
82. GPUSPH. http://www.gpusph.org
83. AQUAgpusph. http://canal.etsin.upm.es/aquagpusph
84. CAPS OpenACC Compiler (2012) The fastest way to many-core programming. http://www.caps-entreprise.com
85. OpenACC (2012) OpenACC accelerator directives. http://www.training.praceri.eu/uploads/tx_pracetmo/OpenACC.pdf
86. Lebacki B, Wolfe M, Miles D (2012) The pgi fortran and c99 openacc compilers. Cray User Group, Redmond
87. Using OpenACC with MPI Tutorial—PGI Compilers. www.pgroup.com/doc/openaccmpi17tut.pdf
88. Wienke S, Springer P, Terboven C, an Mey D (2012) OpenACC—first experiences with real-world applications. In: European conference on parallel processing. Springer, Berlin, pp 859–870
89. Herdman J, Gaudin W, McIntosh-Smith S, Boulton M, Beckingsale DA, Mallinson A, Jarvis SA (2012) Accelerating hydrocodes with OpenACC, OpenCL and CUDA. In: High performance computing, networking, storage and analysis (SCC), 2012 SC companion. IEEE, pp 465–471
90. Hart A, Ansaloni R, Gray A (2012) Porting and scaling OpenACC applications on massively-parallel, GPU-accelerated supercomputers. Eur Phys J Spec Top 210:5–16
91. Farber R (2016) Parallel programming with OpenACC. Newnes, London
92. Hiermaier S, Könke D, Stilp A, Thoma K (1997) Computational simulation of the hypervelocity impact of Al-spheres on thin plates of different materials. Int J Impact Eng 20:363–374
93. https://www.nasa.gov/centers/johnson/capabilities/orbital_debris/hvit/index.html
94. Henderson A, Ahrens J, Law C (2004) The ParaView guide. Kitware, Clifton Park
95. Knezevic M, Crapps J, Beyerlein IJ, Coughlin DR, Clarke KD, McCabe RJ (2016) Anisotropic modeling of structural components using embedded crystal plasticity constructive laws within finite elements. Int J Mech Sci 105:227–238
96. Knezevic M, Kalidindi SR, Fullwood D (2008) Computationally efficient database and spectral interpolation for fully plastic Taylor-type crystal plasticity calculations of face-centered cubic polycrystals. Int J Plast 24:1264–1276
97. Knezevic M, McCabe RJ, Tomé CN, Lebensohn RA, Chen SR, Cady CM, Gray Iii GT, Mihaila B (2013) Modeling mechanical response and texture evolution of α-uranium as a function of strain rate and temperature using polycrystal plasticity. Int J Plast 43:70–84
98. Knezevic M, Kalidindi SR (2017) Crystal plasticity modeling of microstructure evolution and mechanical fields during processing of metals using spectral databases. JOM 69:830–838
99. Zecevic M, Beyerlein IJ, McCabe RJ, McWilliams BA, Knezevic M (2016) Transitioning rate sensitivities across multiple length scales: microstructure–property relationships in the Taylor cylinder impact test on zirconium. Int J Plast 84:138–159