Coupled material point and level set methods for simulating soils interacting with rigid objects with complex geometry

Article *in* Computers and Geotechnics · November 2023

DOI: 10.1016/j.compgeo.2023.105708

CITATION
1

READS
241

4 authors:

Yidong Zhao
Korea Advanced Institute of Science and Technology
10 PUBLICATIONS **86** CITATIONS
SEE PROFILE

Jinhyun Choo
Korea Advanced Institute of Science and Technology
69 PUBLICATIONS **1,529** CITATIONS
SEE PROFILE

Yupeng Jiang
Leibniz Universität Hannover
25 PUBLICATIONS **151** CITATIONS
SEE PROFILE

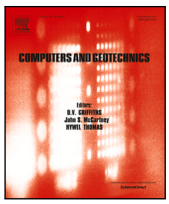
Liuchi Li
Johns Hopkins University
9 PUBLICATIONS **200** CITATIONS
SEE PROFILE

Computers and Geotechnics 163 (2023) 105708

Contents lists available at ScienceDirect

Computers and Geotechnics

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



Research paper

Coupled material point and level set methods for simulating soils interacting with rigid objects with complex geometry

Yidong Zhao [a], Jinhyun Choo [a,*], Yupeng Jiang [b], Liuchi Li [c]

[a] Department of Civil and Environmental Engineering, KAIST, South Korea
[b] Institute of Mechanics and Computational Mechanics, Leibniz Universität Hannover, Germany
[c] Hopkins Extreme Materials Institute, Johns Hopkins University, United States

ARTICLE INFO

Keywords:
Level set method
Material point method
Large deformation
Soil–structure interactions
Terramechanics
Numerical simulation

ABSTRACT

The material point method (MPM) is often used to simulate soils that interact with (nearly) rigid objects, such as structures, machines, or rocks. Yet MPM simulations of such problems are quite challenging when the objects have complex shapes. In this paper, we propose an efficient approach for incorporating geometrically complex rigid objects into MPM modeling. The proposed approach leverages the level set method, which can efficiently delineate arbitrary surface geometry, to represent the boundary of a discrete object. For coupling the level set object with the MPM domain, a robust algorithm is developed on the basis of contact mechanics. Through numerical examples of varied complexity, we verify the proposed approach and demonstrate its ability to efficiently simulate challenging problems wherein soils interact with complex rigid objects such as debris-resisting baffles, a vehicle wheel, and basal terrain.

1. Introduction

The material point method (MPM) has become a popular technique to simulate soils interacting with (nearly) rigid objects such as structures, machines, or rocks. These interactions are central to a variety of applications including infrastructure construction, geohazards mitigation, and terramechanics.

To represent stiff objects in MPM modeling, existing works have relied mainly on two types of approaches: (i) particle-based representations (*e.g.* Dunatunga and Kamrin, 2017; Xu et al., 2019; Agarwal et al., 2021; Zhao et al., 2021; Lei et al., 2022) and (ii) mesh-based representations (*e.g.* Lian et al., 2011; Chen et al., 2015; Song et al., 2020; Li et al., 2022). The former approach models an object as an assembly of material points that are far stiffer than soils. The latter approach represents an object with a separate mesh-based discretization scheme such as finite elements. It is noted that both types of approaches commonly require an algorithm for treating the contact between the soil domain (material points) and the object domain (an assembly of material points or a mesh). These approaches can well handle objects with relatively simple shapes.

When the object has complex geometry, however, both particle-based and mesh-based representations are sub-optimal. As an example, Fig. 1 illustrates how the two types of approaches represent a wheel of a real-world vehicle (NASA, 2020a). As can be seen, particle-based representations are inherently unable to describe complex shapes precisely. Mesh-based representations may be sufficiently accurate, but their coupling with the MPM involves high computational cost for implementation and execution. For these reasons, neither a particle-based nor a mesh-based approach is ideal for handling stiff objects that interact with an MPM domain.

In this work, we propose an efficient approach for simulating soils interacting with rigid objects with complex geometry. The proposed approach is built on two kinds of methods. The first one is the coupled material-point and discrete-element method (MP-DEM) (Jiang et al., 2020, 2022), which is a hybrid continuum–discrete approach for simulating interactions between soft materials and rigid objects. The upshot of the MP-DEM is that it only requires us to represent the *boundary* of an object, because it treats the object as rigid. This treatment has been verified and validated against analytical solutions and experimental data involving interactions between materials with high stiffness contrast. However, it should be noted that the existing MP-DEM framework is limited to objects with quite simple geometry (*e.g.* sphere and square) that can be represented in an analytical way.

The second method on which this work draws is the level set method (Osher and Fedkiw, 2003) – a powerful technique that can efficiently delineate arbitrary surface geometry – which has been utilized in a wide range of problems involving complex surfaces. Notably, the level set method has recently enabled the DEM to seamlessly

* Corresponding author.
E-mail address: jinhyun.choo@kaist.ac.kr (J. Choo).

https://doi.org/10.1016/j.compgeo.2023.105708
Received 29 March 2023; Received in revised form 6 July 2023; Accepted 29 July 2023
Available online 14 August 2023

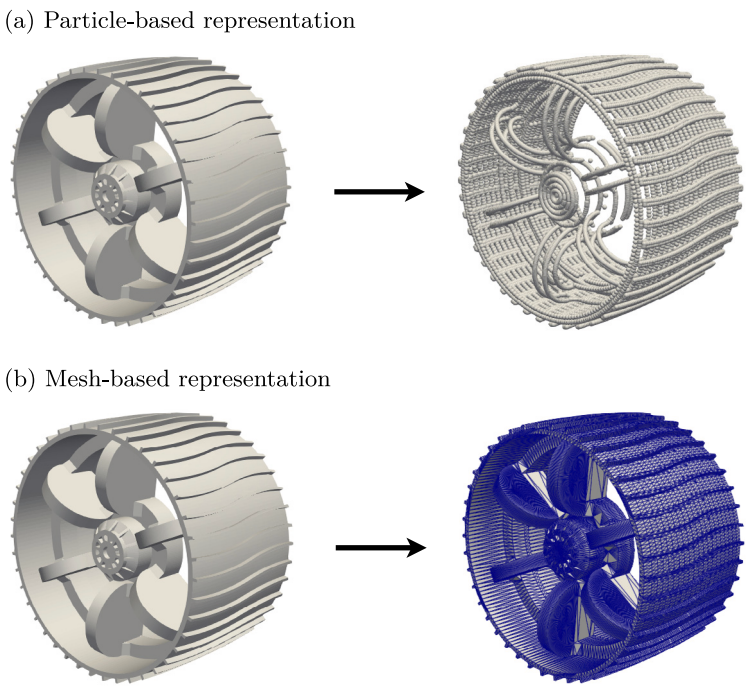


Fig. 1. Particle-based and mesh-based representations of a wheel (NASA, 2020a).

accommodate grains with complex shapes (Kawamoto et al., 2016, 2018; Li et al., 2019). However, it has not been utilized to represent rigid objects that interact with soft materials discretized by the MPM or any other continuum particle methods. To our knowledge, this work is the first to propose the use of the level set method to deal with geometrically complex rigid objects interacting with a continuum particle method such as the MPM.

The remainder of the paper proceeds as follows. In Section 2, we recapitulate the standard MPM formulation for soils undergoing large deformations. In Section 3, we present a level set approach for representing rigid objects with arbitrary boundary geometry. In Section 4, we develop an algorithm that couples the MPM domain and the level set objects on the basis of contact mechanics. In Section 5, we describe various numerical examples to verify and demonstrate the performance of the proposed approach. In Section 6, we provide concluding remarks.

2. Material point method for soils under large deformation

In this section, we summarize the standard MPM formulation for a continuum body undergoing large deformation. For more details of the MPM formulation and related backgrounds, we refer to Nguyen (2014), Jiang et al. (2016), Zhang et al. (2016) and de Vaucorbeil et al. (2020).

2.1. Problem statement

Consider a continuum body that fills in $\Omega \in \mathbb{R}^d$ in the current configuration, where d denotes the spatial dimension. The boundary of the domain is denoted by $\partial\Omega$ and decomposed into the displacement (Dirichlet) boundary $\partial_u\Omega$, and the traction (Neumann) boundary $\partial_t\Omega$, satisfying $\overline{\partial_u\Omega \cup \partial_t\Omega} = \partial\Omega$ and $\partial_u\Omega \cap \partial_t\Omega = \emptyset$. The time domain is denoted by $\mathcal{T} := (0, T]$ with $T > 0$.

To describe large-deformation kinematics accurately, one must use finite deformation theory that distinguishes between the reference and current configurations. Let us denote the position vectors in the reference and current configurations by $\boldsymbol{X}$ and $\boldsymbol{x}$, respectively. The displacement, velocity, and acceleration vectors are then defined as $\boldsymbol{u} := \boldsymbol{x} - \boldsymbol{X}$, $\boldsymbol{v} := \dot{\boldsymbol{u}}$ and $\boldsymbol{a} := \ddot{\boldsymbol{u}}$, respectively, where the dot denotes the material time derivative. The deformation gradient $\boldsymbol{F}$ is defined as

$$\boldsymbol{F} := \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \mathbf{1} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}}, \tag{1}$$

where $\mathbf{1}$ is the second-order identity tensor. The Jacobian (determinant) is defined as

$$J := \det(\boldsymbol{F}) = \mathrm{d}v/\mathrm{d}V > 0, \tag{2}$$

where $\mathrm{d}V$ and $\mathrm{d}v$ are the differential volumes in the reference and current configurations, respectively.

The standard MPM uses an updated Lagrangian method, whereby the balance of linear momentum is formulated in the current configuration as

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{F}) + \rho \boldsymbol{g} = \rho \boldsymbol{a} \quad \text{in } \Omega \times \mathcal{T}, \tag{3}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\nabla\cdot$ is the divergence operator calculated in the current configuration, ρ is the current mass density, and $\boldsymbol{g}$ is the gravitational acceleration vector. To close the formulation, one must introduce a constitutive law that relates the stress tensor and the deformation gradient tensor. A wide range of constitutive laws are available for geomaterials in various deformation regimes, see, *e.g.* Manzari and Dafalias (1997), Borja and Tamagnini (1998), Jop et al. (2006), Andrade et al. (2012), Borja and Choo (2016) and Choo (2018). To deal with elastoplastic constitutive equations in the finite deformation range, we employ the combination of hyperelasticity and multiplicative plasticity (Simo and Ortiz, 1985). Importantly, this approach allows one to bypass the issue of selecting a proper objective stress rate, which is an unresolved issue in the literature.

To furnish the initial–boundary-value problem of interest, let us introduce the initial condition as $\boldsymbol{u} = \boldsymbol{u}_0$ at $t = 0$, and the boundary conditions as

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \quad \text{on } \partial_u\Omega \times \mathcal{T}, \tag{4}$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} = \hat{\boldsymbol{t}} \quad \text{on } \partial_t\Omega \times \mathcal{T}, \tag{5}$$

where $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{t}}$ are the prescribed boundary displacement and boundary traction, respectively, and $\boldsymbol{n}$ is the unit outward normal vector. Finally, the strong form of the problem can be stated as follows: Find the

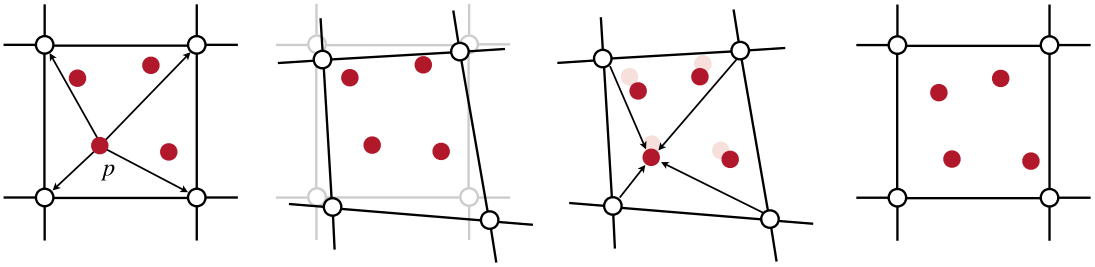


Fig. 2. MPM update procedure. After Zhao and Choo (2020) and Zhao et al. (2022b).

displacement field $\boldsymbol{u}$ that satisfies Eq. (3), subjected to the initial and boundary conditions above.

Following the standard procedure, the variational version of the governing equation (3) can be formulated as

$$\int_\Omega \boldsymbol{\eta}\cdot\rho\boldsymbol{a}\,\mathrm{d}v = -\int_\Omega \nabla^{\mathrm{s}}\boldsymbol{\eta}:\boldsymbol{\sigma}\,\mathrm{d}v + \int_\Omega \boldsymbol{\eta}\cdot\rho\boldsymbol{g}\,\mathrm{d}v + \int_{\partial_t\Omega}\boldsymbol{\eta}\cdot\hat{\boldsymbol{t}}\,\mathrm{d}a, \tag{6}$$

where $\boldsymbol{\eta}$ is the variation of $\boldsymbol{u}$, ∇^{s} is the symmetric gradient operator evaluated in the current configuration, and $\mathrm{d}a$ is the differential area.

2.2. MPM discretization

To begin MPM discretization, let us introduce a set of particles (material points) that fill in the initial domain and a background grid that accommodates the particles. In each time step, the MPM updates the states (*e.g.* positions, velocities, and stresses) of the particles through the four-stage procedure illustrated in Fig. 2. We provide a brief description of each step in the sequel. In what follows, we distinguish between quantities related to particles and background grid using subscripts $(\circ)_p$ and $(\circ)_i$, respectively. Also, we distinguish between quantities at time t^n and t^{n+1} using superscripts $(\circ)^n$ and $(\circ)^{n+1}$, respectively.

Particle-to-grid transfer. The first stage in the MPM procedure, called the particle-to-grid (P2G) transfer, is to project the quantities of the particles to the background grid. Specifically, the P2G transfer is conducted as

$$m_i = \sum_p w_i(\boldsymbol{x}_p^n)m_p, \tag{7}$$

$$m_i\boldsymbol{v}_i^n = \sum_p w_i(\boldsymbol{x}_p^n)m_p\boldsymbol{v}_p^n, \tag{8}$$

where m_i is the nodal mass, m_p is the particle mass, $\boldsymbol{v}_i$ is the nodal velocity, and $\boldsymbol{v}_p$ is the particle velocity. Also, $w_i(\boldsymbol{x}_p^n)$ is the interpolation weight associated with node i and particle p at position $\boldsymbol{x}_p$, and $\sum_p$ denotes the summation over particles supported by $w_i(\boldsymbol{x}_p^n)$. It is noted that a variety of choices are available for the interpolation functions, $w_i(\boldsymbol{x}_p^n)$. Examples range from the linear shape functions in the standard finite elements (used in the original MPM) to more advanced functions that are designed to avoid cell-crossing errors in MPM solutions (*e.g.* Bardenhagen and Kober, 2004; Gan et al., 2018; Steffen et al., 2008). In this work, we adopt quadratic B-splines which are free of cell-crossing errors (Steffen et al., 2008) and gaining popularity in the MPM community (*e.g.* Stomakhin et al., 2013; Gaume et al., 2018; Moutsanidis et al., 2020; Yamaguchi et al., 2021). Note, however, that the proposed approach should work equally well with other types of MPM interpolation functions.

Grid update. The second stage is to update the nodal quantities. As standard, we use the explicit Euler method to integrate the governing equation and update the nodal velocity as

$$m_i\boldsymbol{v}_i^{n+1} = m_i\boldsymbol{v}_i^n + \Delta t\boldsymbol{f}_i^n, \tag{9}$$

where $\Delta t := t^{n+1} - t^n$ is the time interval, and $\boldsymbol{f}_i^n$ is the sum of internal and external force vectors.

Grid-to-particle transfer. The third stage, called the grid-to-particle (G2P) transfer, is to map back the nodal velocities to the particles. The same interpolation functions used in the P2G transfer are used for this purpose. It is noted that there are two schemes for the velocity transfer, namely, the fluid-implicit-particle (FLIP) method (Brackbill and Ruppel, 1986) and the particle-in-cell method (Harlow, 1964). The FLIP scheme involves much less numerical dissipation than the PIC scheme, but it is also less stable numerically. As such, it is common to blend the two schemes as

$$\boldsymbol{v}_p^{n+1} = \eta\underbrace{\left(\boldsymbol{v}_p^n + \sum_i w_i(\boldsymbol{x}_p^n)(\boldsymbol{v}_i^{n+1} - \boldsymbol{v}_i^n)\right)}_{\text{FLIP}} + (1-\eta)\underbrace{\left(\sum_i w_i(\boldsymbol{x}_p^n)\boldsymbol{v}_i^{n+1}\right)}_{\text{PIC}}, \tag{10}$$

where $\eta \in [0,1]$ is the FLIP/PIC blending ratio. Unless specified otherwise, we set $\eta = 1$. We then update other quantities, such as the deformation gradient, stress, volume, and the position, of the individual particles.

Grid reset and repeat. The fourth and last stage is to reset the background grid and proceed to the next time step. The above-described four stages are repeated in the next time step.

3. Level set approach for rigid objects

In this section, we first describe how we make use of the level set method to delineate the boundary of a rigid object. Then we explain how to update the kinematics of the object when it is movable.

3.1. Level set representation of rigid object geometry

The key idea of the level set method is to represent a surface implicitly using a *signed distance function*. As depicted in Fig. 3a, the signed distance function gives rise to contour lines (or level curves) that are negative inside the surface and positive outside the surface. Then, based on the level set value and its gradient, we can calculate the distance from the surface and the direction normal to the surface. It is noted that for a rigid object whose shape is fixed, the level set values do not need to be updated after they are initialized.

To calculate the level set value at any point of interest, we have to discretize the level set values on a grid. For this purpose, we use an equi-spaced Cartesian grid, as illustrated in Fig. 3b. Note that here an equi-spaced grid is chosen for the sake of easy implementation, and other types of grids such as a hierarchical grid can also be used if desired. After the discretization, as shown in Fig. 3c, we can interpolate the discretized level set values to calculate the level set value at every point in the domain. Technically speaking, the interpolation can be done by any kind of interpolation functions. For consistency with the MPM discretization, here we adopt the same type of interpolation function used in the MPM (specifically, quadratic B-splines in this work) to interpolate the level set values as well. Also, the discretization of the level set values requires us to discretize the surface of the rigid

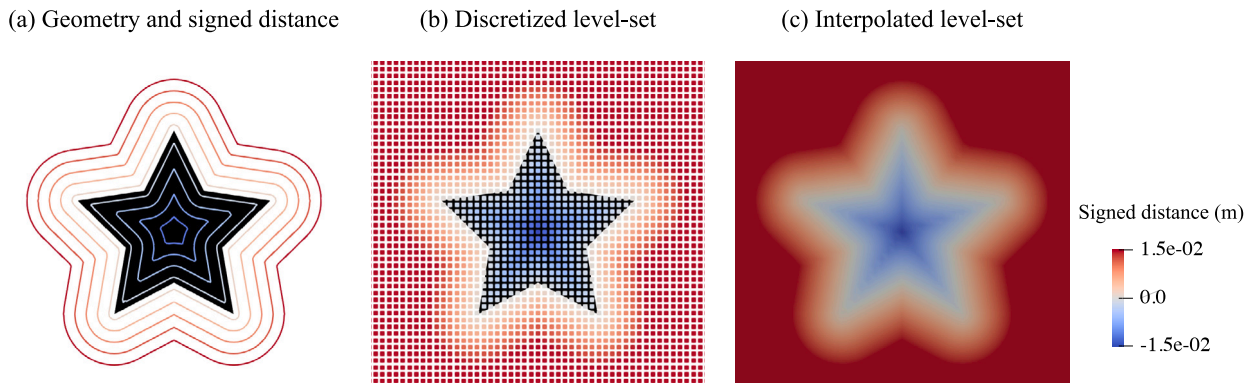


Fig. 3. Illustration of the signed distance and the level set.

object as well. So we discretize the object's surface with a lower-dimensional mesh composed of triangular elements, and calculate the level set values on the surface mesh using a standard point-to-triangle algorithm (Jones, 1995).

Importantly, the grid spacing of the level set discretization should be selected properly to balance accuracy and efficiency in computation. For accuracy, it is recommended that the spacing of the level set grid, h_{LS}, is not greater than the spacing of the MPM grid or that of the surface mesh of the rigid object. (This point will be demonstrated later through a numerical example.) Meanwhile, h_{LS} should not be too small for computational efficiency. As such, by default, we set h_{LS} to be equal to the smaller value between the spacing of the MPM grid and that of the surface mesh.

Having discretized the level set values, we can calculate the level set value at point $\boldsymbol{x}$, $\phi(\boldsymbol{x})$, and its gradient, $\nabla\phi(\boldsymbol{x})$, as

$$\phi(\boldsymbol{x}) = \sum_i w_i(\boldsymbol{x})\phi_i, \tag{11}$$

$$\nabla\phi(\boldsymbol{x}) = \sum_i \nabla w_i(\boldsymbol{x})\phi_i, \tag{12}$$

where ϕ_i is the discretized level set value at node i, and $\sum_i$ is the summation over the discretized level set grids supported by $w_i(\boldsymbol{x})$. The level set value corresponds to the nearest distance from point $\boldsymbol{x}$ to the surface of the object. Also, the unit vector outward normal to the surface can be calculated as

$$\boldsymbol{n}_{LS} = \frac{\nabla\phi(\boldsymbol{x})}{\|\nabla\phi(\boldsymbol{x})\|} \tag{13}$$

These two quantities, the distance and the surface normal vector, will be used in the coupling with MPM discretization, which will be described in the next section.

3.2. Kinematics of rigid object

When the rigid object of interest is movable, its kinematics should be updated based on standard rigid body dynamics, like the DEM (*e.g.* Lim and Andrade, 2014). In what follows, we describe how to integrate the equations of motion of a rigid object. We shall use subscript $(\circ)_r$ to denote quantities related to the rigid object, and superscript $(\circ)^{n-1}$, $(\circ)^{n-1/2}$, $(\circ)^n$, $(\circ)^{n+1/2}$, and $(\circ)^{n+1}$ to denote quantities at times t^{n-1}, $t^{n-1/2}$, t^n, $t^{n+1/2}$, and t^{n+1}, respectively.

The motion of a rigid object is comprised of two components: translation and rotation. The translational motion is governed by Newton's second law, expressed as

$$m_r\boldsymbol{a}_r = \boldsymbol{f}_r, \tag{14}$$

where m_r is the mass of the rigid object, $\boldsymbol{a}_r$ is the linear acceleration, and $\boldsymbol{f}_r$ is the reaction force on the rigid object. In the present work, the reaction force will be calculated based on the interaction of the object with the MPM domain, which will be explained in the next section. The rotation of the rigid object is governed by Euler's rotation equations, given by

$$\boldsymbol{I}_r\cdot\boldsymbol{\alpha}_r + \boldsymbol{\omega}_r\times(\boldsymbol{I}_r\cdot\boldsymbol{\omega}_r) = \boldsymbol{M}_r, \tag{15}$$

where $\boldsymbol{I}_r$ is the inertia matrix, $\boldsymbol{\alpha}_r$ is the angular acceleration, $\boldsymbol{\omega}_r$ is the angular velocity, and $\boldsymbol{M}_r$ is the moment. For convenience, Eq. (15) is usually calculated in the principal inertia axes as

$$I_{r,1}\alpha_{r,1} = M_{r,1} + \omega_{r,2}\omega_{r,3}(I_{r,2} - I_{r,3}), \tag{16}$$

$$I_{r,2}\alpha_{r,2} = M_{r,2} + \omega_{r,3}\omega_{r,1}(I_{r,3} - I_{r,1}), \tag{17}$$

$$I_{r,3}\alpha_{r,3} = M_{r,3} + \omega_{r,1}\omega_{r,2}(I_{r,1} - I_{r,2}), \tag{18}$$

where $I_{r,i}$ represents the ith principal moments of inertia, $\alpha_{r,i}$ and $\omega_{r,i}$ are the ith component of $\boldsymbol{\alpha}_r$ and $\boldsymbol{\omega}_r$, respectively, and $M_{r,i}$ is the ith component of $\boldsymbol{M}_r$.

To update the translational components of motion – the linear velocity and position – we use the central difference time integration scheme as in the standard DEM (Cundall and Strack, 1979). The linear velocity and position at time step n are updated as

$$\boldsymbol{v}_r^{n+1/2} = \boldsymbol{v}_r^{n-1/2} + \frac{\Delta t}{m_r}\boldsymbol{f}_r^n, \tag{19}$$

$$\boldsymbol{x}_r^{n+1} = \boldsymbol{x}_r^n + \Delta t\boldsymbol{v}_r^{n+1/2}. \tag{20}$$

The update of the rotational motion is more challenging because Euler's rotation equations (16)–(18) have nonlinear terms. Here we use the predictor–corrector algorithm suggested by Walton and Braun (1993), which can be summarized as follows:

1. Estimate the angular velocities at time step n based on the previous values:

$$\omega_{r,i}^{n,ES} = \omega_{r,i}^{n-1/2} + \frac{1}{2}\alpha_{r,i}^{n-1}\Delta t. \tag{21}$$

2. Predict the angular accelerations at time step n based on the estimated angular velocities:

$$\alpha_{r,1}^{n,PR} = \left[M_{r,1}^n + \omega_{r,1}^{n,ES}\omega_{r,3}^{n,ES}(I_{r,2} - I_{r,3})\right]/I_{r,1}, \tag{22}$$

$$\alpha_{r,2}^{n,PR} = \left[M_{r,2}^n + \omega_{r,3}^{n,ES}\omega_{r,1}^{n,ES}(I_{r,3} - I_{r,1})\right]/I_{r,2}, \tag{23}$$

$$\alpha_{r,3}^{n,PR} = \left[M_{r,3}^n + \omega_{r,1}^{n,ES}\omega_{r,2}^{n,ES}(I_{r,1} - I_{r,2})\right]/I_{r,3}. \tag{24}$$

3. Predict the angular velocities at time step n using the predicted angular accelerations:

$$\omega_{r,i}^{n,PR} = \omega_{r,i}^{n-1/2} + \frac{1}{2}\alpha_{r,i}^{n,PR}\Delta t. \tag{25}$$

4. Correct the angular accelerations based on the predicted angular velocities:

$$\alpha_{r,1}^{n,CR} = \left[M_{r,1}^n + \omega_{r,2}^{n,PR}\omega_{r,3}^{n,PR}(I_{r,2} - I_{r,3})\right]/I_{r,1}, \tag{26}$$

$$\alpha_{r,2}^{n,CR} = \left[M_{r,2}^{n} + \omega_{r,3}^{n,PR}\omega_{r,1}^{n,PR}(I_{r,3} - I_{r,1})\right]/I_{r,2}, \tag{27}$$

$$\alpha_{r,3}^{n,CR} = \left[M_{r,3}^{n} + \omega_{r,1}^{n,PR}\omega_{r,2}^{n,PR}(I_{r,1} - I_{r,2})\right]/I_{r,3}. \tag{28}$$

5. Update the angular velocities at time step $n + 1/2$ using the corrected angular accelerations:

$$\omega_{r,i}^{n+1/2} = \omega_{r,i}^{n-1/2} + \alpha_{r,i}^{n,CR}\Delta t. \tag{29}$$

After updating the angular velocities, we can update the rotation of the rigid object using the quaternion approach that allows one to avoid singularities in the Euler angle method (Evans and Murad, 1977). The update procedure in Walton and Braun (1993) is described in the following:

1. Calculate the coefficient matrix $\boldsymbol{B}_r$, which is required for updating the quaternions:

$$\boldsymbol{B}_r = \begin{pmatrix} 1 & -\beta_3 & \beta_1 & \beta_2 \\ \beta_3 & 1 & \beta_2 & -\beta_1 \\ -\beta_1 & -\beta_2 & 1 & -\beta_3 \\ -\beta_2 & \beta_1 & \beta_3 & 1 \end{pmatrix}, \tag{30}$$

where $\beta_i = \frac{\Delta t}{4}\omega_{r,i}^{n+1/2}$.

2. Update the quaternions:

$$\boldsymbol{q}_r^{n+1} = \boldsymbol{B}_r^{-1} \cdot \boldsymbol{B}_r^{T} \cdot \boldsymbol{q}_r^{n}, \tag{31}$$

where $\boldsymbol{q}_r^{n+1}$ and $\boldsymbol{q}_r^{n}$ represent the quaternions of the rigid object at times t^{n+1} and t^{n}, respectively.

3. Calculate the updated rotation matrix $\boldsymbol{A}_r^{n+1}$, which represents the rotation from the initial state to the current state at time step $n+1$, as

$$\boldsymbol{A}_r^{n+1} = \begin{pmatrix} -q_1^2 + q_2^2 - q_3^2 + q_4^2 & -2(q_1q_2 - q_3q_4) & 2(q_2q_3 + q_1q_4) \\ -2(q_1q_2 + q_3q_4) & q_1^2 - q_2^2 - q_3^2 + q_4^2 & -2(q_1q_3 - q_2q_4) \\ 2(q_2q_3 - q_1q_4) & -2(q_1q_3 + q_2q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix}, \tag{32}$$

where q_i represents the ith component of $\boldsymbol{q}_r^{n+1}$.

After updating the position and rotation matrix with Eqs. (20) and (32), we can determine the new configuration of the rigid object and proceed to the next time step.

4. Coupling material point method with level set object

In this section, we develop an algorithm to couple material points in the MPM domain with discrete objects represented by the level set method.

4.1. Coupling approach

To couple a material point and a rigid object, we build on the approach recently developed in Jiang et al. (2022). When the distance between a material point and a rigid object is lower than a certain value, the coupling force between the material point and the object is calculated based on a contact mechanics model. As for the specific contact model, here we adopt the barrier method (Li et al., 2020; Zhao et al., 2022a), which can strictly prevent inter-penetration in a relatively simple manner.

Specifically, we calculate the coupling force based on the nearest distance d_{LS}^{n} and the surface normal direction $\boldsymbol{n}_{LS}^{n}$ computed with the level set method in Eqs. (11) and (13), respectively. The normal coupling force, $\boldsymbol{f}_{rp,N}^{n}$, is calculated as

$$\boldsymbol{f}_{rp,N}^{n} = f_{rp,N}^{n}\boldsymbol{n}_{LS}^{n}, \tag{33}$$

$$f_{rp,N}^{n} = \begin{cases} \kappa(d_{LS}^{n} - r_p)\left[2\ln\left(\frac{d_{LS}^{n}}{r_p}\right) - \frac{r_p}{d_{LS}^{n}} + 1\right] & \text{if } 0 < d_{LS}^{n} < r_p, \\ 0 & \text{if } d_{LS}^{n} \geq r_p, \end{cases} \tag{34}$$

where $f_{rp,N}^{n}$ is the magnitude of the normal coupling force, κ is a parameter controlling the stiffness of the barrier model, and r_p is the maximum distance where the coupling force is nonzero. As in Jiang et al. (2022), we assume that each material point has a spherical volume and determine r_p as the radius of the volume. Alternatively, one may assume that the material point's volume is cubic and calculate r_p accordingly. From our experience, however, the result is not sensitive to the shape of the material point volume.

Next, we calculate the tangential coupling force, $\boldsymbol{f}_{rp,T}^{n}$, considering friction between the material point and the rigid object. To incorporate stick–slip transition behavior, we utilize a smoothed version of the Coulomb friction law (Li et al., 2020; Zhao et al., 2022a) and calculate $\boldsymbol{f}_{rp,T}^{n}$ as

$$\boldsymbol{f}_{rp,T}^{n} = m(u_T^n)\mu f_{rp,n}^{n}\boldsymbol{m}^{n}, \tag{35}$$

where $m(u_T^n) \in [0,1]$ is a friction smoothing function that continuously varies with the accumulative slip magnitude u_T^n, μ is the friction coefficient, and $\boldsymbol{m}^n$ is the unit vector pointing toward the slip direction. Specifically, $m(u_T^n)$ is given by

$$m(u_T^n) = \begin{cases} -\frac{{u_T^n}^2}{s_p^2} + \frac{2|u_T^n|}{s_p} & \text{if } |u_T^n| < s_p, \\ 1 & \text{if } |u_T^n| \geq s_p, \end{cases} \tag{36}$$

where s_p is a threshold parameter controlling the slip magnitude at which the slip condition is met. The accumulative slip magnitude, u_T^n, is calculated as

$$u_T^n = u_T^{n-1} + \Delta t(\boldsymbol{v}_p^n - \boldsymbol{v}_r^{n-1/2} - \boldsymbol{\omega}_r^{n-1/2} \times \boldsymbol{r}_{rp}^n) \cdot \boldsymbol{m}^n, \tag{37}$$

and $\boldsymbol{m}^n$ is given by

$$\boldsymbol{m}^n = \frac{\Delta t(\boldsymbol{v}_p^n - \boldsymbol{v}_r^{n-1/2} - \boldsymbol{\omega}_r^{n-1/2} \times \boldsymbol{r}_{rp}^n) - \Delta t\left[(\boldsymbol{v}_p^n - \boldsymbol{v}_r^{n-1/2} - \boldsymbol{\omega}_r^{n-1/2} \times \boldsymbol{r}_{rp}^n) \cdot \boldsymbol{n}_{LS}^n\right]\boldsymbol{n}_{LS}^n}{\left\|\Delta t(\boldsymbol{v}_p^n - \boldsymbol{v}_r^{n-1/2} - \boldsymbol{\omega}_r^{n-1/2} \times \boldsymbol{r}_{rp}^n) - \Delta t\left[(\boldsymbol{v}_p^n - \boldsymbol{v}_r^{n-1/2} - \boldsymbol{\omega}_r^{n-1/2} \times \boldsymbol{r}_{rp}^n) \cdot \boldsymbol{n}_{LS}^n\right]\boldsymbol{n}_{LS}^n\right\|}, \tag{38}$$

with $\boldsymbol{r}_{rp}^n$ denoting the vector from the center of mass of the rigid object to particle p. Finally, we get the total coupling force by adding the normal and tangential forces as

$$\boldsymbol{f}_{rp}^{n} = \boldsymbol{f}_{rp,N}^{n} + \boldsymbol{f}_{rp,T}^{n}. \tag{39}$$

The coupling force alters the MPM formulation (explained in Section 2) and the rigid body dynamics of the level set object (explained in Section 3) as follows. The nodal momentum update, Eq. (9), is modified as

$$m_i\boldsymbol{v}_i^{n+1} = m_i\boldsymbol{v}_i^{n} + \Delta t(\boldsymbol{f}_i^{n} + \boldsymbol{f}_{rp,i}^{n}), \tag{40}$$

where $\boldsymbol{f}_{rp,i}^{n} = \sum_p \boldsymbol{f}_{rp}^{n} w_i(\boldsymbol{x}_p^n)$ is the nodal coupling force. Next, let us denote the coupling force imposed on the rigid object by $\boldsymbol{f}_{pr}^{n}$, which should be equal to $-\boldsymbol{f}_{rp}^{n}$ according to Newton's third law. For the rigid object at time step n, the total reaction force $\boldsymbol{f}_r^n$ and the total moment $\boldsymbol{M}_r^n$, which appear on the right hand sides of Eqs. (14) and (15), respectively, are calculated as

$$\boldsymbol{f}_r^n = \sum_p \boldsymbol{f}_{pr}^{n} = \sum_p -\boldsymbol{f}_{rp}^{n}, \tag{41}$$

$$\boldsymbol{M}_r^n = \sum_p \boldsymbol{f}_{rp}^{n} \times \boldsymbol{r}_{rp}^{n}. \tag{42}$$

4.2. Implementation

The implementation of the above-described approach is straightforward, except for one thing: the calculation of the level set values when

the discrete object is moving. This is because the level set values of a moving object evolve in the global coordinate system. A general approach for updating level set values is to solve a Hamilton–Jacobi equation numerically, but its computational cost is non-trivial.

When the object of interest is rigid, one can update level set values during translation and rotation by introducing a local coordinate system attached to the rigid object (Kawamoto et al., 2016), which we shall refer to as the level set coordinate system. The origin of the level set coordinate system is set such that all discretized level set values are at the first octant of the coordinate system and thus the interpolation functions can be implemented straightforwardly. Also, the level set coordinate system has its axes aligned with the principal axes of the object. The use of the level set coordinate system greatly simplifies the level set calculation in two ways. First, as the level set coordinate system moves together with the rigid object, it eliminates the need to track translation and rotation separately. Second, because the object is rigid, its local surface distance remains constant. As such, the level set values generated in the level set coordinate system remain unchanged during the movement of the object. Thus the level set coordinate system allows us to calculate the level set values in a highly efficient manner. It is noted that the robustness of this calculation has been well demonstrated in level set DEM simulations, see, *e.g.* Kawamoto et al. (2016, 2018) and Li et al. (2019).

Algorithm 1 presents a modified MPM procedure for updating material points coupled with level set objects. This procedure extends the standard MPM procedure described in Section 2 to incorporate coupling with rigid objects represented by the level set method.

Algorithm 1 Procedure for updating material points coupled with level set objects.

Contact detection and coupling force calculation:

Calculate the positions of particles (material points), $\boldsymbol{x}^n_{p,LS}$, in the level set coordinate system.

Compute the distances from the particles to the rigid objects, d^n_{LS}, and the surface normal vectors, $\boldsymbol{n}^n_{LS}$:

$$d^n_{LS} = \sum_i w_i(\boldsymbol{x}^n_{p,LS})\phi_i,$$

$$\boldsymbol{n}^n_{LS} = \frac{\sum_i \nabla w_i(\boldsymbol{x}^n_{p,LS})\phi_i}{\| \sum_i \nabla w_i(\boldsymbol{x}^n_{p,LS})\phi_i \|}.$$

If $d^n_{LS} < r_p$, calculate the coupling forces $\boldsymbol{f}^n_{rp,LS}$ based on d^n_{LS} and $\boldsymbol{n}^n_{LS}$, as described in Eqs. (33)–(39)

Calculate the coupling forces in the global coordinate system through the rotational transformation:

$$\boldsymbol{f}^n_{rp} = \boldsymbol{A}^n \boldsymbol{f}^n_{rp,LS}.$$

Particle-to-grid transfer:

Transfer the masses and momenta of the particles to the background grid (standard particle-to-grid transfer).

Transfer the coupling forces of the particles to the background grid:

$$\boldsymbol{f}^n_{rp,i} = \sum_p \boldsymbol{f}^n_{rp} w_i(\boldsymbol{x}^n_p).$$

Grid update:

Update the nodal momenta:

$$m_i \boldsymbol{v}^{n+1}_i = m_i \boldsymbol{v}^n_i + \Delta t(\boldsymbol{f}^n_i + \boldsymbol{f}^n_{rp,i}).$$

Grid-to-particle transfer:

Transfer the updated nodal velocities to the particles (standard grid-to-particle transfer).

Update the kinematics of rigid objects:

Calculate the coupling forces and the moments imposed on the rigid objects:

$$\boldsymbol{f}^n_r = \sum_p -\boldsymbol{f}^n_{rp},$$

$$\boldsymbol{M}^n_r = \sum_p \boldsymbol{f}^n_{rp} \times \boldsymbol{r}^n_{rp}.$$

Update the kinematics of the rigid objects through standard rigid body dynamics, as described in Section 3.2.

5. Numerical examples

In this section, we evaluate the proposed approach through four examples of varied complexity. The first and second examples are intended to verify the simulation approach. For this reason, these two examples mainly consider objects with simple shapes that can also be described precisely in an analytical manner. The third and fourth examples are designed to demonstrate the performance of the approach for handling discrete objects with complex geometry. To this end, we use two geometrically complex objects in real-world applications, namely, a wheel of a vehicle and a basal terrain. Note that the rigid object (wheel) in the third example is not only geometrically complex but also subjected to dramatic translation and rotation.

For the numerical examples in this section, the level set values of the discrete objects are generated using the `pysdf` library (Yu, 2020). The overall MPM algorithm is implemented with the `Taichi` library (Hu et al., 2019).

5.1. Rolling of a sphere on an inclined plane

As our first example, we consider the problem of a sphere rolling down an inclined rigid plane, which has been used to verify contact algorithms in the MPM (*e.g.* Bardenhagen et al., 2001; Huang et al., 2011). Here we use the example to verify the contact algorithm that couples the MPM domain and the level set object in a simple setting. Fig. 4 depicts the geometry and boundary conditions of the problem.

When both the sphere and the plane are rigid, the displacement of the sphere permits an analytical solution (Huang et al., 2011). Thus, to emulate a rigid sphere, we treat the sphere as a purely elastic material with quite high stiffness, say, $E = 420$ MPa. The other elasticity parameter, Poisson's ratio, is set as $\nu = 0.4$. The density of the sphere is assigned as $\rho = 1.0$ t/m^3. We use 17,160 material points to discretize the sphere and employ a background grid comprised of 0.2-m-long cubic elements. We calculate the time increment as $\Delta t = 0.3(h/c) = 6.325 \times 10^{-5}$ s, where h is the element size of the MPM grid and c is the P-wave velocity. We discretize the level set values on a Cartesian grid with the spacing of $h_{LS} = 0.2$ m. To verify the approach under varied friction conditions, we consider three different friction coefficients, namely, $\mu = 0.0$, 0.2, and 0.4, for the interface between the sphere and the plane.

Fig. 5 shows the displacement of the center-of-mass of the sphere in the x direction, along with the analytical solutions. Note that the analytical solutions are different depending on the contact condition: Under the slip condition ($\mu = 0.0$ and 0.2), the solution is $u_x = 1/2\ gt^2(\sin\theta - \mu\cos\theta)$, and under the stick condition ($\mu = 0.4$), $u_x = 5/14\ gt^2 \sin\theta$. Irrespective of this difference, the numerical solutions always agree well with the analytical solutions. This agreement verifies the algorithm that couples the MPM domain and the rigid object represented by the level set method.

5.2. Impact of granular flow on columns

In our second example, we further verify the proposed approach under more complex conditions. To this end, we simulate the impact of granular flow on columns, which has relevance to debris flows and

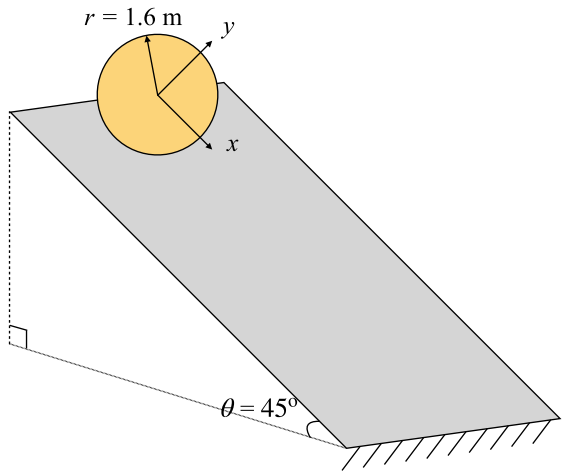


Fig. 4. Rolling of a sphere on an inclined plane: problem geometry and boundary conditions.

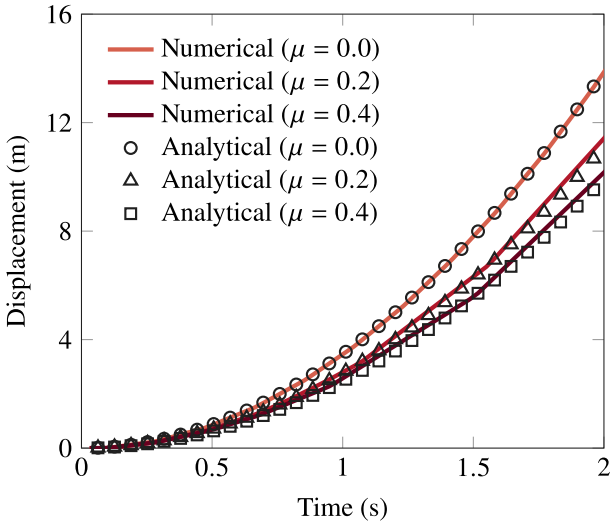


Fig. 5. Rolling of a sphere on an inclined plane: the x-displacement of the center-of-mass of the sphere.

their mitigation (*e.g.* Choi et al., 2014; Yang et al., 2021; Liang et al., 2023; Ng et al., 2023). The geometry and boundary conditions of this problem are illustrated in Fig. 6. It is noted that the bottom boundary is a frictional interface where the frictional coefficient is 0.1. As shown in the figure, we consider two types of baffle shapes, namely, pillar-shaped columns and star-shaped columns. For simplicity, the columns are considered made of quite smooth materials (*e.g.* steel) such that its friction coefficient with the granular flow can be set as zero.

The constitutive behavior of the granular material is considered elastoplastic. The elastic behavior is modeled by Hencky elasticity, with a Young's modulus of $E = 10$ MPa and a Poisson's ratio of $\nu = 0.3$. For the plastic behavior, we use the $\mu(I)$ rheology (Jop et al., 2006), which is a rate-dependent rheology model widely used for dry granular flows. The model parameters are adopted from Jiang et al. (2022) where the same constitutive model is used to simulate granular impact experiments with the MPM. The $\mu(I)$ rheology parameters adopted are as follows: the lowest friction angle $\phi_{\min} = 30^{\circ}$, the highest friction angle $\phi_{\max} = 34^{\circ}$, the reference inertia parameter $I_0 = 0.278$, and the mean particle size $d = 0.25$ mm. The initial density of the granular material is set as $\rho = 2.0$ t/m^3.

For MPM discretization, we introduce a background grid comprised of 0.01-m-long cubic elements and initialize each element in the granular domain with 8 material points. This discretization results in a total of 64,000 material points. The time increment is again calculated as $\Delta t = 0.3(h/c)$, which gives $\Delta t = 3.657 \times 10^{-5}$ s. To avoid volumetric locking that may emanate from the isochoric deformation of the $\mu(I)$ rheology, we apply the $\bar{F}$ approach proposed by Zhao et al. (2022b).

The main focus of this example is on how the grid spacing of the level set discretization, h_{LS}, affects the numerical solutions. For this purpose, we repeat each case with three choices of h_{LS}: $h_{LS} = 0.5\min(h_{\text{surface}}, h)$, $h_{LS} = \min(h_{\text{surface}}, h)$, and $h_{LS} = \max(h_{\text{surface}}, h)$, where h_{surface} refers to the spacing of the grid for discretizing the surfaces of the rigid objects (the columns herein). It would be worthwhile to note that the choice $h_{LS} = 2\max(h_{\text{surface}}, h)$, which is larger than the three choices above, has been found to be numerically unstable.

Fig. 7 presents the final configurations of the granular flows obtained with the three different values of level set grid spacings, for pillar-shaped columns. It can be seen that the three numerical solutions are more or less the same, and this is not surprising because the columns' shape is very simple. Also, while not presented for brevity, all these results are virtually identical to the numerical solution obtained by representing the columns analytically as in the existing MP-DEM of Jiang et al. (2022). So it can be confirmed that when the object of interest is geometrically simple, the proposed approach is not very sensitive to h_{LS} as long as it is small enough to provide numerical stability.

Fig. 8 shows how the results become different when the columns become star-shaped. As can be seen, the results obtained with $h_{LS} = 0.5\min(h_{\text{surface}}, h)$ and $h_{LS} = \min(h_{\text{surface}}, h)$ are nearly identical. Such close similarity indicates that the choice of $h_{LS} = \min(h_{\text{surface}}, h)$ can provide quite accurate results even when the object has complex geometry with sharp corners. However, the result obtained with $h_{LS} = \max(h_{\text{surface}}, h)$ exhibits non-trivial difference from the other two results, in terms of the flow morphology behind the baffles. This difference suggests that the choice of $h_{LS} = \max(h_{\text{surface}}, h)$ may be too large for objects with complex shapes.

In Fig. 9 we also plot how the reaction forces on the center columns are affected by the level set grid spacing, for the cases of pillar-shaped columns and star-shaped columns. Being consistent with the conclusion drawn from the flow morphology, the reaction force results indicate that (i) when the object geometry is simple (pillar), the results are not so sensitive to the value of h_{LS}, and that (ii) when the object geometry is rather complex (star), h_{LS} should be small enough and $h_{LS} = \min(h_{\text{surface}}, h)$ appears to give sufficiently accurate results that are similar to those obtained by a smaller value of h_{LS}. The latter point can be quantitatively confirmed by the maximum reaction forces in the figure (circles therein): those obtained with $h_{LS} = 0.5\min(h_{\text{surface}}, h)$ and $h_{LS} = \min(h_{\text{surface}}, h)$ are very close to each other. In light of these observations, we shall use $h_{LS} = \min(h_{\text{surface}}, h)$ in the following examples.

5.3. Soil–wheel interaction

Our third example simulates the interaction between a wheel of a real-world vehicle and a soil ground, which is the main topic in the field of terramechanics. The problem setup is depicted in Fig. 10. As shown, the wheel features highly complex geometry which cannot be handled easily by particle-based or mesh-based approaches.

We adopt the wheel geometry from a small-scale wheel model provided by NASA (2020a), scaling it to be the same as that of the real wheel described in NASA (2020b). The wheel moves with an angular velocity, ω, prescribed in the rolling direction. To demonstrate how the ground behaves differently by the wheel velocity, we consider three different angular velocities: $\omega = 1, 2$, and 4 rad/s. To emphasize the importance of the wheel geometry in soil–wheel interaction, we set the friction coefficient on the wheel surface as zero ($\mu = 0$), such that all

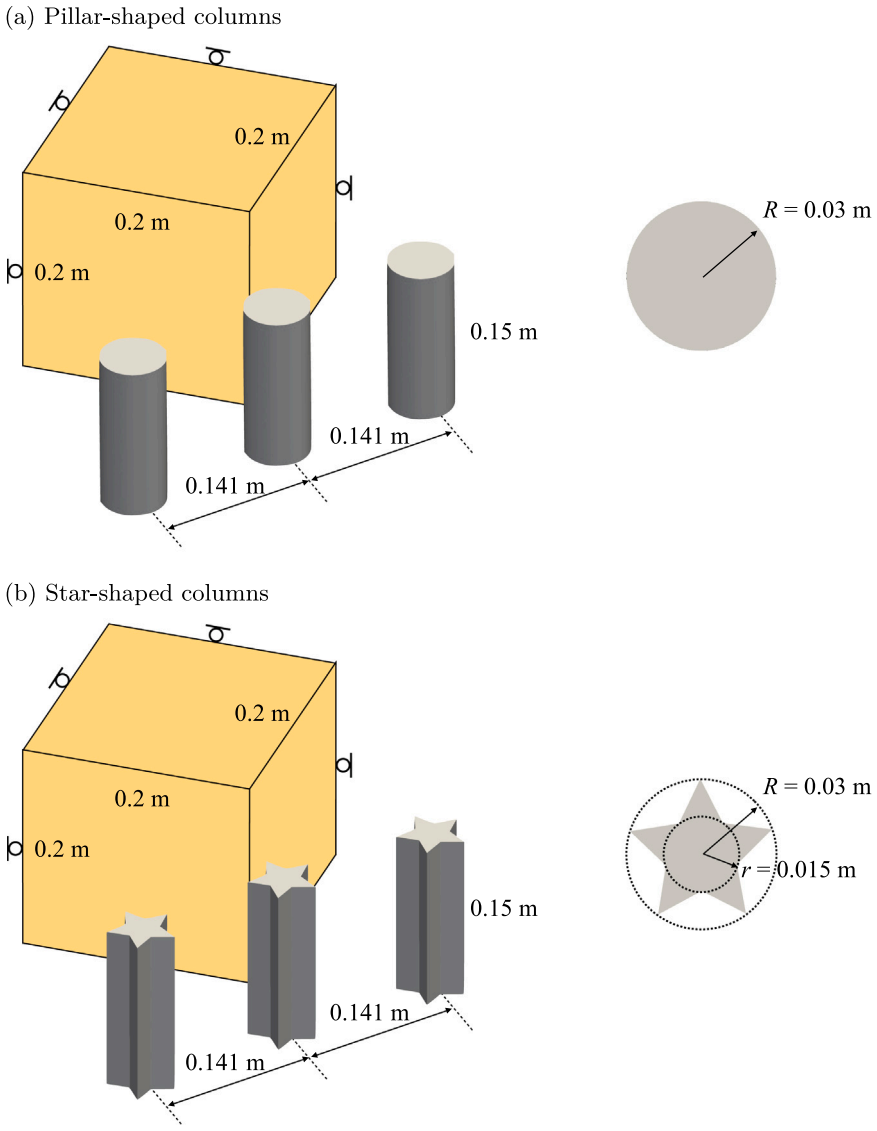


Fig. 6. Impact of granular flow on columns: problem geometry and boundary conditions.

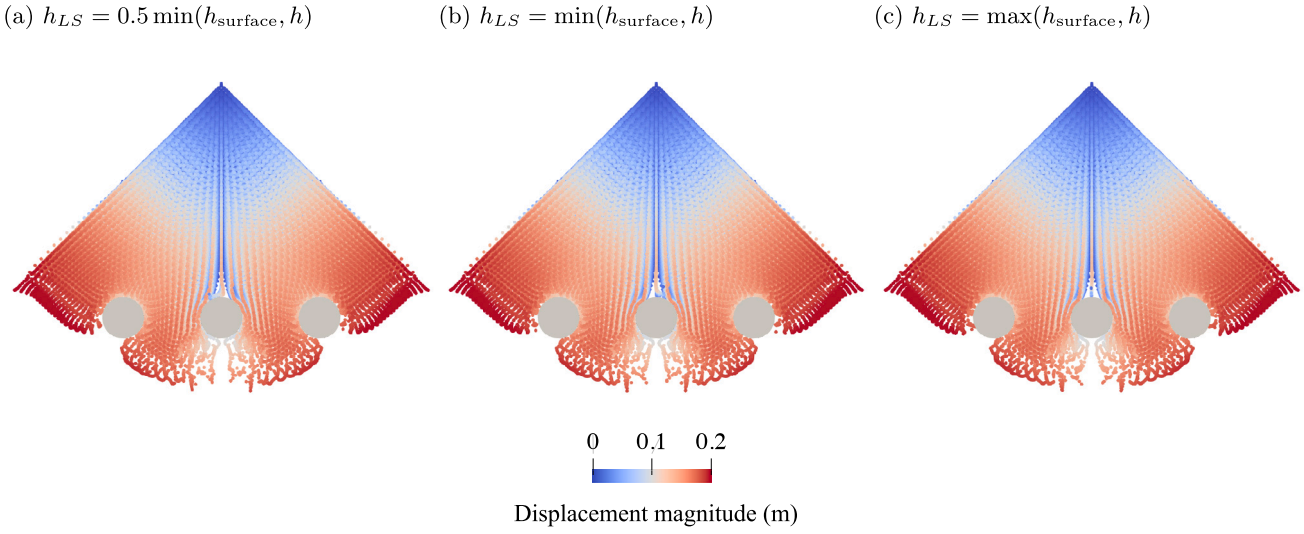


Fig. 7. Impact of granular flow on pillar-shaped columns: final configurations of the granular flows obtained with three different values of level set grid spacings. The particles are colored by the displacement magnitudes.

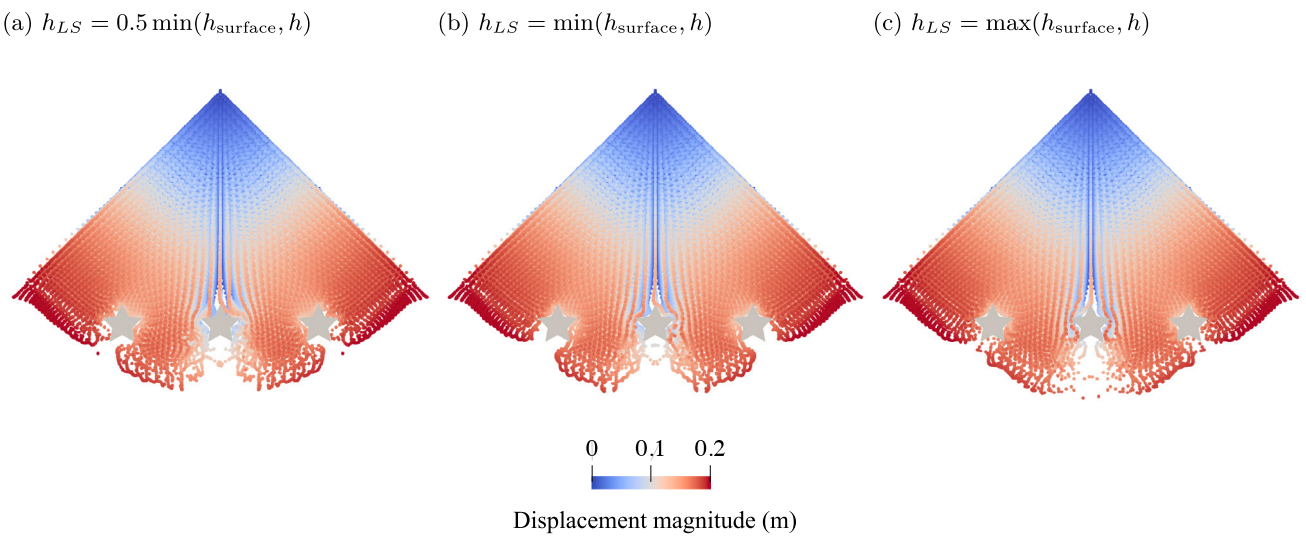


Fig. 8. Impact of granular flow on star-shaped columns: final configurations of the granular flows obtained with three different values of level set grid spacings. The particles are colored by the displacement magnitudes.

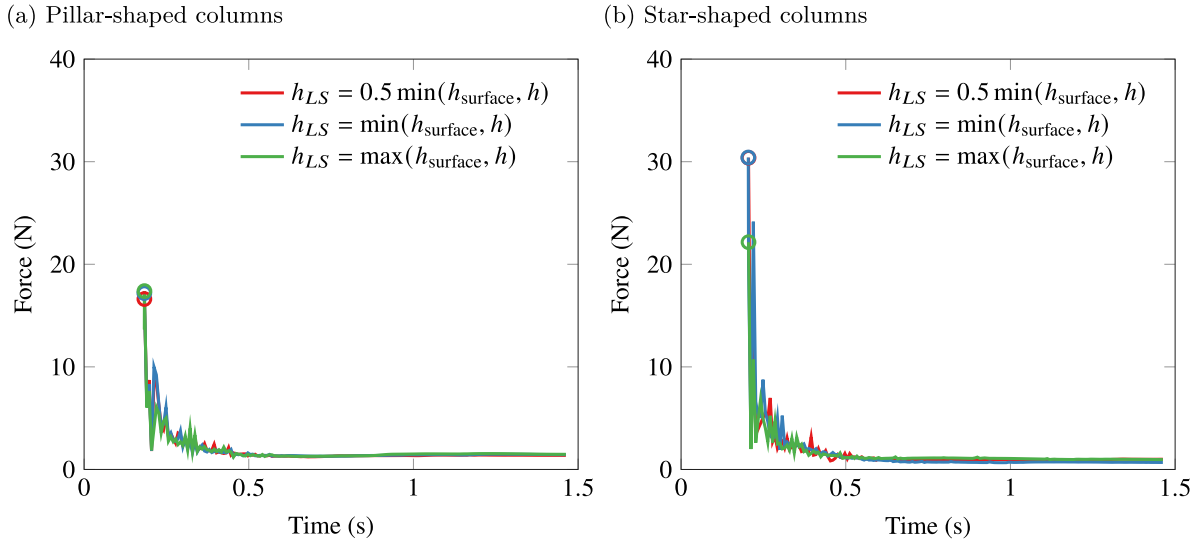


Fig. 9. Impact of granular flow on columns: reaction forces on the center columns obtained with three different values of level set grid spacings. Circles denote the maximum reaction forces.

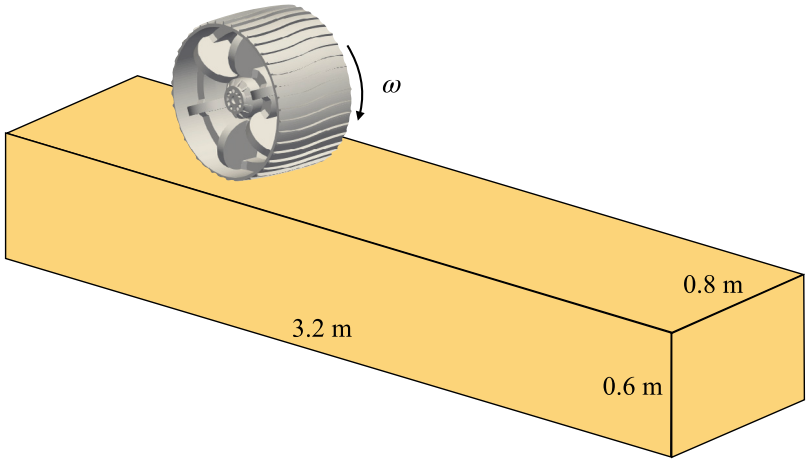


Fig. 10. Soil-wheel interaction: problem geometry.

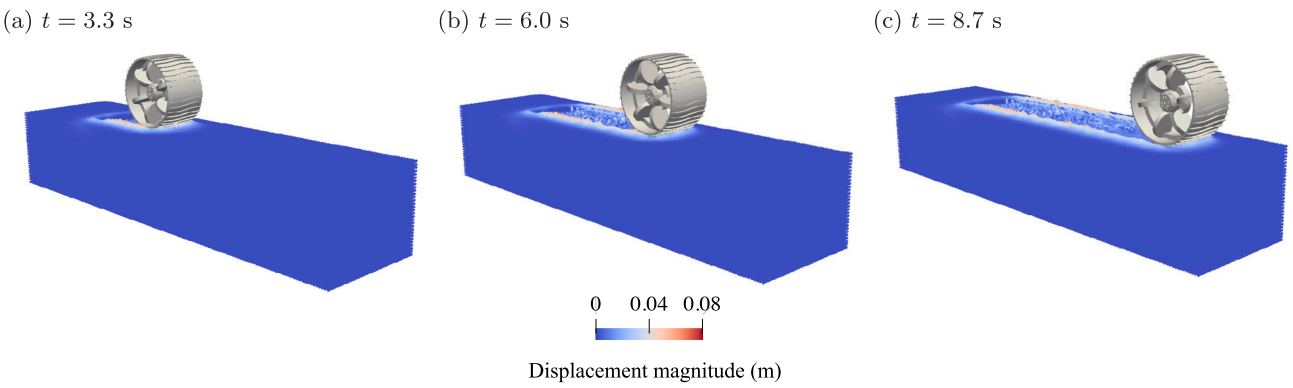


Fig. 11. Soil-wheel interaction: simulation snapshots with $\omega = 1$ rad/s. The particles are colored by their vertical displacements.

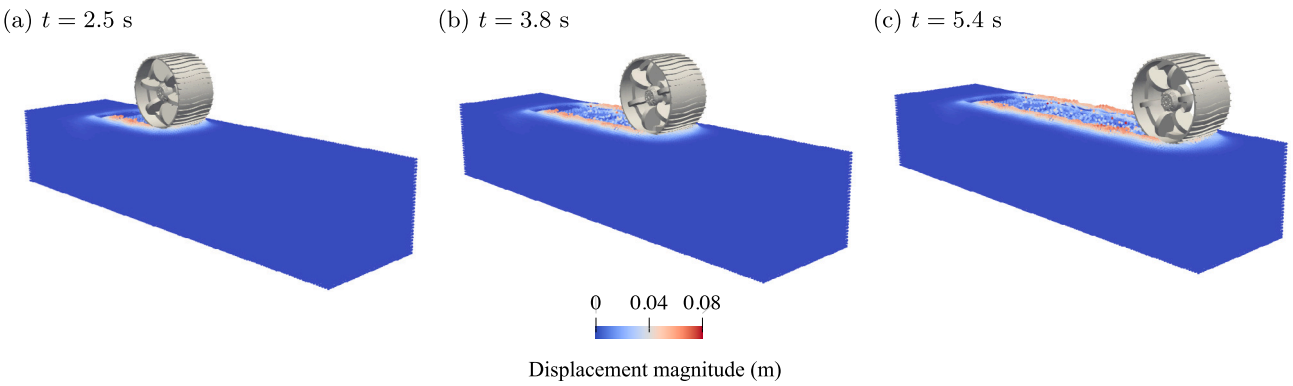


Fig. 12. Soil-wheel interaction: simulation snapshots with $\omega = 2$ rad/s. The particles are colored by their vertical displacements.

the frictional behavior at the soil–wheel interface is attributed to the non-smooth topology of the wheel surface. In other words, if the wheel were a smooth cylinder, it would not move forward despite the given angular velocity. The weight of the wheel is set as 0.171 t.

The soil ground is modeled by a combination of Hencky elasticity and Drucker–Prager plasticity. The elasticity parameter assigned are a Young's modulus of $E = 10$ MPa and a Poisson's ratio of $\nu = 0.3$. As for the plastic parameters, a friction angle of $\phi = 30°$ and a dilatancy angle of $\psi = 5°$ are assigned. The initial density of the soil is set as $\rho = 2.0$ t/m^3.

We discretize the problem as follows. Introducing a background grid comprised of 0.04-m-long cubes, we initialize each element in the ground with 8 material points, which results in a total of 192,000 material points. We again calculate the time increment as $\Delta t = 0.3(h/c) = 1.462 \times 10^{-4}$ s. We discretize the level set on a grid with $h_{LS} = \min(h, h_{\text{surface}}) = 0.008$ m, which equals to h_{surface} in the mesh scaled from NASA (2020a).

Figs. 11–13 illustrate the simulation results in the three cases of angular velocities. It can be seen that the wheel moves forward through reactions from the ground. As explained earlier, the wheel movement is possible due to the non-smooth geometry of the wheel surface, because the frictional coefficient is zero in this problem. Therefore the results highlight the critical role of the wheel surface geometry in soil–wheel interactions. Furthermore, the ground responds very differently to the wheel velocity, and it undergoes larger deformations as the wheel velocity becomes higher.

Fig. 14 presents how the wheel displacements evolve over time in the three cases. Also shown in the figure are the translational velocities of circular wheels rolling on a flat surface under a non-slip condition, $v = \omega r$, where r is calculated as the average radius of the wheel. It can be seen that the velocities (slopes) of the simulated wheels are initially lower than those of non-slipping circular wheels, because the friction coefficient is set as zero. (It is again noted that a circular wheel in this setting does not move at all.) As the wheels are further mobilized, however, the velocities approach those of circular wheels under non-slip conditions, indicating that the degree of "friction" due to the uneven surface geometry is physically plausible. The fact that the proposed method can capture such complex effects of surface geometry would be highly attractive for many problems in terramechanics and other types of soil–machine interactions.

5.4. Debris flow over a complex basal terrain

As our fourth and final example, we apply the proposed approach to the simulation of a granular flow over a complex basal terrain. This example is motivated by two facts: (i) an increasing number of numerical simulations are incorporating the geometry of real-world terrains characterized by various sensing techniques (*e.g.* Li et al., 2021; Cicoira et al., 2022; Xu et al., 2022; Zhao et al., 2023), and (ii) the existing simulations have spent substantial cost to represent the complex geometry of terrains with particles or other types of means (*e.g.* Xu et al., 2019; Lei et al., 2022; Zhao et al., 2021). The terrain geometry of this problem is adopted from that of a real-world debris flow occurred in South Korea (Yune et al., 2013). Fig. 15 shows the problem geometry and the locations of the source materials. The source materials are modeled as dry granular materials with the same constitutive model used in the second example. The model parameters are also adopted from the second example, except the mean particle diameter which needs to be scaled. To consider the difference between laboratory and field scales, we scale the parameter to 10 mm following Rauter et al. (2022).

For numerical simulation, we discretize the source materials with 6216 material points and introduce a background grid comprised of 2-m-long cubes. To represent the basal terrain, we discretize the terrain

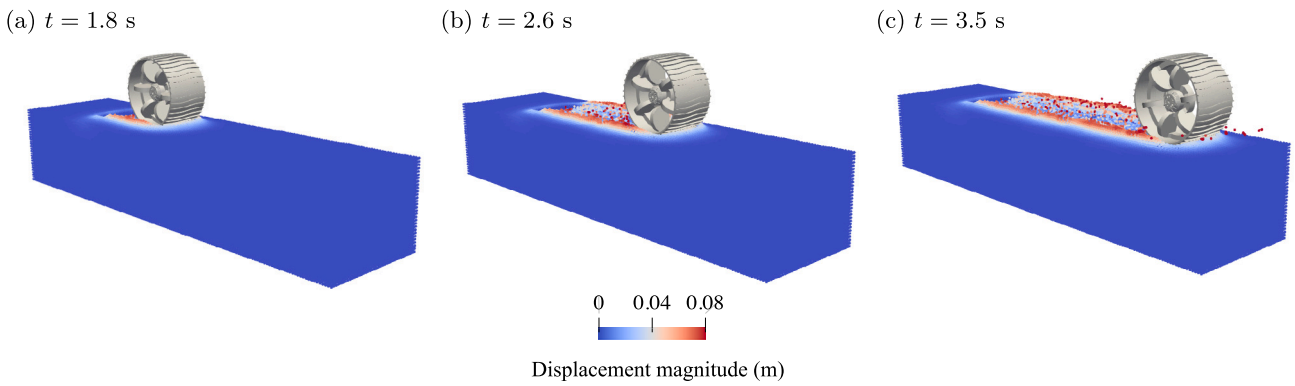


Fig. 13. Soil-wheel interaction: simulation snapshots with $\omega = 4$ rad/s. The particles are colored by their vertical displacements.

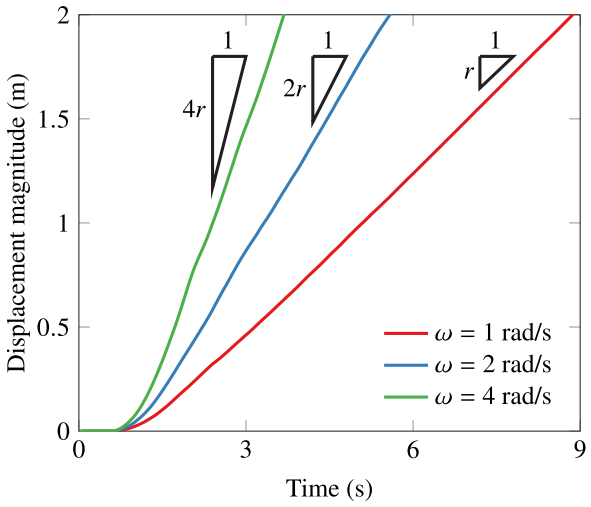


Fig. 14. Soil-wheel interaction: evolutions of the displacement magnitude, along with analytical solutions to translation velocities of cylindrical wheels rolling on a flat surface under a non-slip condition.

surface using triangular elements of $h_{\text{surface}} = 2$ m and discretize the level set with $h_{LS} = \min(h, h_{\text{surface}}) = 2$ m. We set the friction coefficient of the basal terrain as $\mu = 0.2$. For numerical stability, we set the FLIP/PIC blending ratio as $\eta = 0.95$. We simulate the problem until $t = 35$ s with a time increment of $\Delta t = 7 \times 10^{-3}$ s.

Fig. 16 presents snapshots of the granular-flow simulation, coloring the material points by their velocity magnitudes. The results demonstrate that the proposed approach can well simulate the sliding of the granular materials over the complex terrain. It is reminded that the existing MPM simulations have treated such complex terrains with particles (*e.g.* Xu et al., 2019; Lei et al., 2022; Zhao et al., 2021). Compared with the particle-based representation, the level set representation has two advantages: (i) it can delineate the terrain geometry more accurately, and (ii) it requires much less computational cost for simulation. These two advantages would be highly desirable for integrating MPM simulations with digitized terrains.

6. Closure

This paper has proposed an approach that can efficiently simulate interactions between soft materials and rigid objects with complex geometry. Built on the MPM, the proposed approach leverages the level set method to delineate arbitrary surface geometry of discrete objects that may interact with the MPM domain. A robust contact algorithm has been developed to couple the level set objects with the MPM domain. The proposed approach has been well verified for problems in which the object geometry can also be represented analytically. Various examples have further demonstrated that the proposed approach can efficiently simulate interactions between soft materials (*e.g.* soils) and topologically complex objects (*e.g.* debris-resisting baffles, vehicle wheel, and basal terrain), which would be far more difficult to simulate for other methods. Thus the proposed approach would be highly attractive for addressing an increasing number of geometrically complex objects in digital-twin simulations.

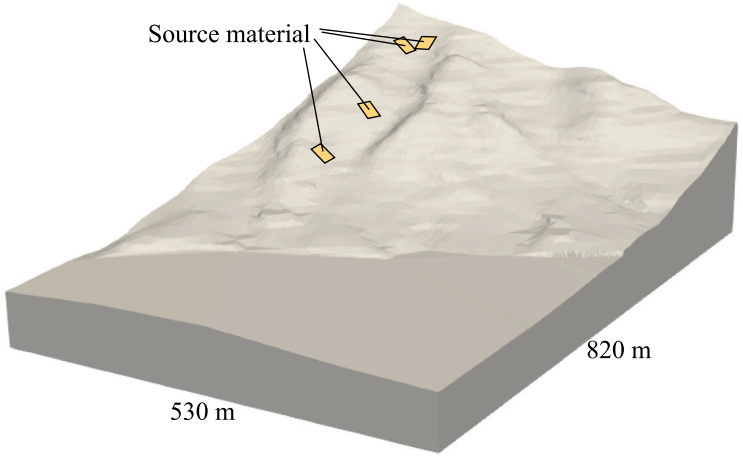


Fig. 15. Debris flow over a complex basal terrain: problem geometry.

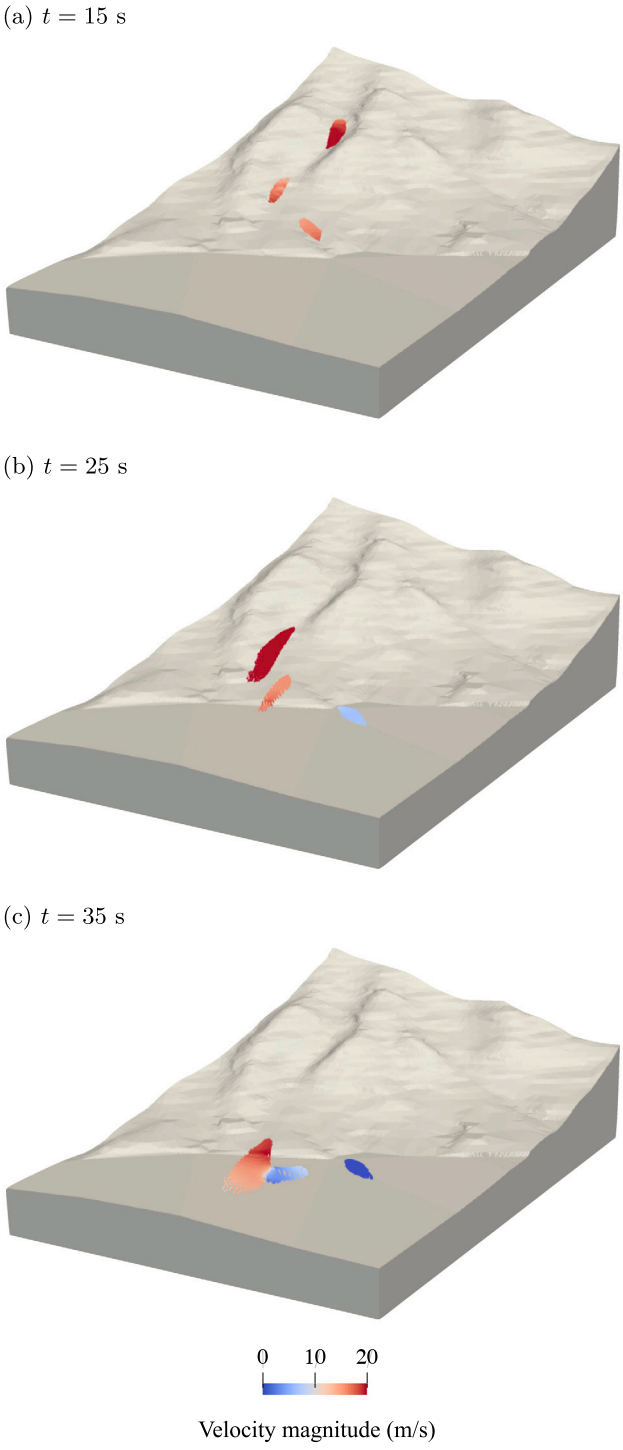


Fig. 16. Debris flow over a complex basal terrain: simulation snapshots. The particles are colored by their velocity magnitudes.

CRediT authorship contribution statement

Yidong Zhao: Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Visualization. **Jinhyun Choo:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Yupeng Jiang:** Methodology, Software, Validation. **Liuchi Li:** Methodology, Software, 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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (Nos. 2022R1F1A1065418 and RS-2023-00209799).

References

Agarwal, S., Karsai, A., Goldman, D.I., Kamrin, K., 2021. Surprising simplicity in the modeling of dynamic granular intrusion. Sci. Adv. 7 (17), eabe0631.
Andrade, J.E., Chen, Q., Le, P.H., Avila, C.F., Evans, T.M., 2012. On the rheology of dilative granular media: bridging solid-and fluid-like behavior. J. Mech. Phys. Solids 60 (6), 1122–1136.
Bardenhagen, S., Guilkey, J., Roessig, K., Brackbill, J., Witzel, W., Foster, J., 2001. Improved contact algorithm for the material point method and application to stress propagation in granular material. CMES Comput. Model. Eng. Sci. 2 (4), 509–522.
Bardenhagen, S.G., Kober, E.M., 2004. The generalized interpolation material point method. CMES Comput. Model. Eng. Sci. 5 (6), 477–496.
Borja, R.I., Choo, J., 2016. Cam-Clay plasticity, Part VIII: A constitutive framework for porous materials with evolving internal structure. Comput. Methods Appl. Mech. Engrg. 309, 653–679.
Borja, R.I., Tamagnini, C., 1998. Cam-Clay plasticity Part III: Extension of the infinitesimal model to include finite strains. Comput. Methods Appl. Mech. Engrg. 155 (1–2), 73–95.
Brackbill, J.U., Ruppel, H.M., 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. J. Comput. Phys. 65 (2), 314–343.
Chen, Z., Qiu, X., Zhang, X., Lian, Y., 2015. Improved coupling of finite element method with material point method based on a particle-to-surface contact algorithm. Comput. Methods Appl. Mech. Engrg. 293, 1–19.
Choi, C.E., Ng, C.W.W., Song, D., Kwan, J., Shiu, H., Ho, K.K.S., Koo, R.C., 2014. Flume investigation of landslide debris–resisting baffles. Can. Geotech. J. 51 (5), 540–553.
Choo, J., 2018. Mohr–Coulomb plasticity for sands incorporating density effects without parameter calibration. Int. J. Numer. Anal. Methods Geomech. 42 (18), 2193–2206.
Cicoira, A., Blatny, L., Li, X., Trottet, B., Gaume, J., 2022. Towards a predictive multi-phase model for alpine mass movements and process cascades. Eng. Geol. 310, 106866.
Cundall, P.A., Strack, O.D., 1979. A discrete numerical model for granular assemblies. Géotechnique 29 (1), 47–65.
Dunatunga, S., Kamrin, K., 2017. Continuum modeling of projectile impact and penetration in dry granular media. J. Mech. Phys. Solids 100, 45–60.
Evans, D.J., Murad, S., 1977. Singularity free algorithm for molecular dynamics simulation of rigid polyatomics. Mol. Phys. 34 (2), 327–331.
Gan, Y., Sun, Z., Chen, Z., Zhang, X., Liu, Y., 2018. Enhancement of the material point method using B-spline basis functions. Internat. J. Numer. Methods Engrg. 113 (3), 411–431.
Gaume, J., Gast, T., Teran, J., Van Herwijnen, A., Jiang, C., 2018. Dynamic anticrack propagation in snow. Nature Commun. 9 (1), 3047.
Harlow, F.H., 1964. The particle-in-cell computing method for fluid dynamics. Methods Comput. Phys. 3, 319–343.
Hu, Y., Li, T.-M., Anderson, L., Ragan-Kelley, J., Durand, F., 2019. Taichi: a language for high-performance computation on spatially sparse data structures. ACM Trans. Graph. 38 (6), 1–16.
Huang, P., Zhang, X., Ma, S., Huang, X., 2011. Contact algorithms for the material point method in impact and penetration simulation. Internat. J. Numer. Methods Engrg. 85 (4), 498–517.
Jiang, Y., Li, M., Jiang, C., Alonso-Marroquin, F., 2020. A hybrid material-point spheropolygon-element method for solid and granular material interaction. Internat. J. Numer. Methods Engrg. 121 (14), 3021–3047.
Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A., 2016. The material point method for simulating continuum materials. In: ACM SIGGRAPH 2016 Courses. pp. 1–52.
Jiang, Y., Zhao, Y., Choi, C.E., Choo, J., 2022. Hybrid continuum–discrete simulation of granular impact dynamics. Acta Geotech. 1–16.
Jones, M.W., 1995. 3D distance from a point to a triangle. University of Wales Swansea Technical Report CSR-5, Department of Computer Science, p. 5.

Jop, P., Forterre, Y., Pouliquen, O., 2006. A constitutive law for dense granular flows. Nature 441 (7094), 727–730.
Kawamoto, R., Andò, E., Viggiani, G., Andrade, J.E., 2016. Level set discrete element method for three-dimensional computations with triaxial case study. J. Mech. Phys. Solids 91, 1–13.
Kawamoto, R., Andò, E., Viggiani, G., Andrade, J.E., 2018. All you need is shape: Predicting shear banding in sand with LS-DEM. J. Mech. Phys. Solids 111, 375–392.
Lei, X., Chen, X., Yang, Z., He, S., Zhu, L., Liang, H., 2022. A simple and robust MPM framework for modelling granular flows over complex terrains. Comput. Geotech. 149, 104867.
Li, X., Fang, Y., Li, M., Jiang, C., 2022. BFEMP: Interpenetration-free MPM–FEM coupling with barrier contact. Comput. Methods Appl. Mech. Engrg. 390, 114350.
Li, M., Ferguson, Z., Schneider, T., Langlois, T.R., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M., 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Trans. Graph. 39 (4), 49.
Li, L., Marteau, E., Andrade, J.E., 2019. Capturing the inter-particle force distribution in granular material using LS-DEM. Granul. Matter 21, 1–16.
Li, X., Sovilla, B., Jiang, C., Gaume, J., 2021. Three-dimensional and real-scale modeling of flow regimes in dense snow avalanches. Landslides 1–14.
Lian, Y., Zhang, X., Zhou, X., Ma, Z., 2011. A FEMP method and its application in modeling dynamic response of reinforced concrete subjected to impact loading. Comput. Methods Appl. Mech. Engrg. 200 (17–20), 1659–1670.
Liang, Z., Choi, C.E., Zhao, Y., Jiang, Y., Choo, J., 2023. Revealing the role of forests in the mobility of geophysical flows. Comput. Geotech. 155, 105194.
Lim, K.-W., Andrade, J.E., 2014. Granular element method for three-dimensional discrete element calculations. Int. J. Numer. Anal. Methods Geomech. 38 (2), 167–188.
Manzari, M.T., Dafalias, Y.F., 1997. A critical state two-surface plasticity model for sands. Géotechnique 47 (2), 255–272.
Moutsanidis, G., Long, C.C., Bazilevs, Y., 2020. IGA-MPM: The isogeometric material point method. Comput. Methods Appl. Mech. Engrg. 372, 113346.
NASA, 2020a. M2020 wheel desktop model 5inch. https://github.com/nasa/NASA-3D-Resources/tree/master/3D%20Models/M2020_WHEEL_DESKTOP_MODEL-5inch/.
NASA, 2020b. The Mars 2020 Perseverance Rover Wheels and Legs. https://mars.nasa.gov/mars2020/spacecraft/rover/wheels/.
Ng, C.W.W., Jia, Z., Poudyal, S., Bhatta, A., Liu, H., 2023. Two-phase MPM modelling of debris flow impact against dual rigid barriers. Géotechnique 1–54.
Nguyen, V.P., 2014. Material point method: basics and applications. Cardiff University, Department of Civil Engineering.
Osher, S., Fedkiw, R., 2003. Level Set Methods and Dynamic Implicit Surfaces. Springer.
Rauter, M., Viroulet, S., Gylfadóttir, S.S., Fellin, W., Løvholt, F., 2022. Granular porous landslide tsunami modelling–the 2014 Lake Askja flank collapse. Nature Commun. 13 (1), 678.
Simo, J., Ortiz, M., 1985. A unified approach to finite deformation elastoplastic analysis based on the use of hyperelastic constitutive equations. Comput. Methods Appl. Mech. Engrg. 49 (2), 221–245.
Song, Y., Liu, Y., Zhang, X., 2020. A non-penetration FEM-MPM contact algorithm for complex fluid-structure interaction problems. Comput. & Fluids 213, 104749.
Steffen, M., Kirby, R.M., Berzins, M., 2008. Analysis and reduction of quadrature errors in the material point method (MPM). Internat. J. Numer. Methods Engrg. 76 (6), 922–948.
Stomakhin, A., Schroeder, C., Chai, L., Teran, J., Selle, A., 2013. A material point method for snow simulation. ACM Trans. Graph. 32 (4), 1–10.
de Vaucorbeil, A., Nguyen, V.P., Sinaie, S., Wu, J.Y., 2020. Material point method after 25 years: Theory, implementation, and applications. Adv. Appl. Mech. 53, 185–398.
Walton, O., Braun, R., 1993. Simulation of rotary-drum and repose tests for frictional spheres and rigid sphere clusters. Technical Report, Lawrence Livermore National Lab., CA (United States).
Xu, X., Jin, F., Sun, Q., Soga, K., Zhou, G.G., 2019. Three-dimensional material point method modeling of runout behavior of the Hongshiyan landslide. Can. Geotech. J. 56 (9), 1318–1337.
Xu, W.-J., Zhou, Q., Dong, X.-Y., 2022. SPH–DEM coupling method based on GPU and its application to the landslide tsunami. Part II: reproduction of the Vajont landslide tsunami. Acta Geotech. 17 (6), 2121–2137.
Yamaguchi, Y., Moriguchi, S., Terada, K., 2021. Extended B-spline-based implicit material point method. Internat. J. Numer. Methods Engrg. 122 (7), 1746–1769.
Yang, E., Bui, H.H., Nguyen, G.D., Choi, C.E., Ng, C.W.W., De Sterck, H., Bouazza, A., 2021. Numerical investigation of the mechanism of granular flow impact on rigid control structures. Acta Geotech. 16, 2505–2527.
Yu, A., 2020. pysdf. https://github.com/sxyu/sdf/.
Yune, C.-Y., Chae, Y.-K., Paik, J., Kim, G., Lee, S.-W., Seo, H.-S., 2013. Debris flow in metropolitan area—2011 Seoul debris flow. J. Mount. Sci. 10, 199–206.
Zhang, X., Chen, Z., Liu, Y., 2016. the Material Point Method: A Continuum-Based Particle Method for Extreme Loading Cases. Academic Press.
Zhao, Y., Choo, J., 2020. Stabilized material point methods for coupled large deformation and fluid flow in porous materials. Comput. Methods Appl. Mech. Engrg. 362, 112742.
Zhao, Y., Choo, J., Jiang, Y., Li, M., Jiang, C., Soga, K., 2022a. A barrier method for frictional contact on embedded interfaces. Comput. Methods Appl. Mech. Engrg. 393, 114820.
Zhao, S., He, S., Li, X., Deng, Y., Liu, Y., Yan, S., Bai, X., Xie, Y., 2021. The Xinmo rockslide-debris avalanche: An analysis based on the three-dimensional material point method. Eng. Geol. 287, 106109.
Zhao, Y., Jiang, C., Choo, J., 2022b. Circumventing volumetric locking in explicit material point methods: A simple, efficient, and general approach. arXiv preprint arXiv:2209.02466.
Zhao, S., Lai, Z., Zhao, J., 2023. Leveraging ray tracing cores for particle-based simulations on GPUs. Internat. J. Numer. Methods Engrg. 124 (3), 696–713.