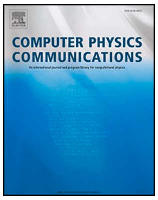
Contents lists available at ScienceDirect

Computer Physics Communications

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



Computer Programs in Physics

GeoDualSPHysics: a high-performance SPH solver for large deformation modelling of geomaterials with two-way coupling to multi-body systems

Ruofeng Feng [a,*], Jidong Zhao [a,*], Georgios Fourtakas [b], Benedict D Rogers [b]

[a] Department of Civil and Environmental Engineering, Hong Kong University of Science and Technology, China
[b] School of Engineering, The University of Manchester, UK

ARTICLE INFO

Edited by Prof. Andrew Hazel

Keywords:
Smoothed particle hydrodynamics
GPU
Large deformation
Granular flow
DualSPHysics

ABSTRACT

This paper presents GeoDualSPHysics, an open-source, graphics processing unit (GPU)-accelerated smoothed particle hydrodynamics (SPH) solver designed for simulating large-deformation geomaterial and their interactions with multi-body systems. Built upon the popular open-source SPH solver DualSPHysics, the solver leverages its highly parallelised SPH scheme empowered by the CUDA parallelisation while extending its capabilities to large-deformation geomechanics problems with particles up to the order of 10^8 on a single GPU. The SPH geomechanics model is enhanced by a noise-free stress treatment technique that stabilizes and accurately resolves stress fields, as well as an extended modified Dynamic Boundary Condition (mDBC) ensuring first-order consistency in solid boundary modelling. Additionally, the coupling interface between DualSPHysics and the multi-body dynamics solver Project Chrono is adapted for simulating interactions between geomaterials and multiple interacting rigid bodies. Benchmark validations confirm the solver's accuracy in resolving geotechnical failures, impact forces on solid boundaries, and geomaterial-multibody system interactions. GPU profiling of the newly implemented CUDA kernels demonstrates their performance metrics are similar to those of the original DualSPHysics solver. Performance evaluations demonstrate its saving in memory usage of 30-50% and improvements in computational efficiency over existing SPH geomechanics solvers, achieving practical simulation speeds for systems with tens of millions of particles and showing a speedup of up to 180x compared to the optimised multi-core CPU implementation. These advances position GeoDualSPHysics as a versatile, efficient tool for high-fidelity simulations of complex geotechnical systems.

Program summary: Program title: GeoDualSPHysics

CPC Library link to program files: https://doi.org/10.17632/z4sh62y97g.1

Licensing provisions: GNU Lesser General Public License

Programming language: C++ and CUDA

Nature of problem: Simulating large deformations in geomaterials and their interactions with movable or fixed solid bodies is critical for addressing engineering challenges such as landslides, soil-machine interactions, and off-road vehicle mobility. While the Smoothed Particle Hydrodynamics (SPH) method is well-suited for modelling continuum-based geomaterial behaviour in these scenarios, critical computational barriers persist, including: (1) numerical instabilities and unphysical noise in large-deformation regimes, (2) inefficiency in scaling simulations to millions of particles for real-world systems, and (3) inadequate frameworks for robust, two-way coupling between deformable geomaterials and multi-body systems. Overcoming these limitations demands stabilized SPH formulations, high-performance computing architectures, and two-way coupling with multibody dynamics solvers.

Solution method: The GeoDualSPHysics solver addresses the above challenges by combining (1) a stabilised SPH formulation for geomaterials, featuring a noise-free stress treatment to eliminate spurious oscillations in large deformations and an extended modified Dynamic Boundary Condition (mDBC) for first-order consistent solid boundary modelling; (2) high-performance CUDA-based GPU parallelization inherited from DualSPHysics, enabling efficient simulations of tens of millions of particles; and (3) two-way coupling with Project Chrono via

* Corresponding authors.
E-mail addresses: ruofengfeng@ust.hk (R. Feng), jzhao@ust.hk (J. Zhao).

https://doi.org/10.1016/j.cpc.2025.109965
Received 18 June 2025; Received in revised form 11 November 2025; Accepted 20 November 2025
Available online 2 December 2025

the DSPHChronoLib library, which integrates collision detection, frictional contact models, and joint constraints to resolve interactions between deformable geomaterials and multi-body systems.

1. Introduction

The simulation of geomaterials and their flow behaviour has broad applications across various scientific and engineering disciplines, including snow and rock avalanches, debris flows, soil cutting and tillage, and terramechanics. In the numerical modelling of such problems, the main challenges are the presence of free surfaces and extremely large deformations with complex topological changes. Discrete methods, such as the discrete element method (DEM) [1–3], and continuum-based meshfree methods, including material point method (MPM) [4–6], smoothed particle hydrodynamics (SPH) [7–10], particle finite element method (PFEM) [11,12], and peridynamics (PD) [13–15] are generally two types of numerical approaches that have demonstrated the capacity to address these challenges.

DEM excels at modelling the behaviour of granulate material and discontinuous interactions but fully resolved DEM simulations of practical granular flows in a large scale typically generate large degrees of freedom, presenting demanding computational cost and data storage challenges. For such large-scale applications, continuum-based meshfree methods are particularly advantageous. Among the aforementioned meshfree methods, SPH is a truly meshfree, Lagrangian method. No grid is required for the integration of the governing partial differential equations in SPH, thereby avoiding grid-related constraints and numerical issues. In the recent years, SPH has gained increasing popularity for efficiently modelling geomechanics problems (e.g., [16–23]).

Although SPH has demonstrated many advantages in modelling highly nonlinear free-surface flow, its relatively high computational cost is the main challenge for large-scale simulations, as the SPH interpolation for a single particle requires computing interactions with a large number of neighbours and the explicit scheme necessitates a small timestep [24]. However, the local interpolation nature and explicit formulation of SPH make it particularly suitable for parallel computation. Leveraging advances in parallel computing architecture, research attempts have been made to accelerate the SPH simulation in geomechanics problems. Peng et al. [25] initiated the use of NVIDIA Compute Unified Device Architecture (CUDA) in SPH modelling of geotechnical problems. A maximum 15 million number of particle simulation has been achieved on GTX 1080Ti 11 GB with 0.41 timesteps per second for the ratio of the smoothing length (h) to initial particle distance (dp) of $h/\mathrm{dp} = 2.8$. Yang et al. [26] developed a parallelized SPH scheme for simulating granular flow using message passing interface (MPI) communication which is scalable on large CPU clusters. Their implementation features a simulation with 11.7 million particles running on 512 CPU cores to simulate 140,000 time-steps in 12–24 hours (1.62-3.24 steps/s). Huang et al. [27] simulated granular flow with up to 50 million particles using DualSPHysics (v.4.0), requiring one week of computation time for a 1 s physical time simulation on TITAN RTX 24 GB. The surging needs for larger-scale, higher-resolution simulations is driving the development of a more efficient solver in terms of both computational speed and memory consumption.

In the past decade, several open-source codes have been developed for SPH applications in geomechanics, such as the C++ code on the Linux platform, PersianSPH [28], the GPU-accelerated C++/CUDA code LOQUAT [25], and the Fortran code SPHERA [29]. While existing open-source solvers have enabled effective simulation of geomaterials, few of them have been extended to simulate geophysical flows interacting with structures. The recently released software GeoXPM (https://www.geoxpm.com), developed by Bui et al. [30], offers the possibility to simulate complex large-deformation geomechanics problems involving soil-structure interactions. However, the software is not yet open-source and adopts a CPU-based parallel scheme with certain particle number limits applied to the public release version. In addition, the existing solver mainly employs the boundary treatment with up to zeroth order consistency along with Shepard filtering to reduce numerical noise in the stress field during large deformations. These techniques are known to introduce numerical fluctuations and empiricism into the analysis. The recent development in boundary conditions and smoothing techniques by Feng et al. [31] has emerged as an alternative framework for the simulation of geomaterials with improved accuracy and stability.

DualSPHysics is an open-source SPH solver based on C++/CUDA/OpenMP for free-surface flow simulation, compatible with both Windows and Linux operating systems [32]. In recent years, DualSPHysics has gained growing popularity as a powerful tool for solving fluid dynamics problems and as a basis for SPH development. Several variants and enhancements of the DualSPHysics solver have been developed to incorporate complex physics, advanced algorithms, and methodology improvements, such as DualSPHysics with SPH-based structural modelling for fluid-structure interaction [33], VecDualSPHysics for improved computational efficiency [34], DualSPHysics-Chrono for the coupling of fluid dynamics with multibody dynamics [35], DualSPHysics with surface tension models for drop dynamics [36], DualSPHysics+ for enhanced accuracy in fluid simulation [37], and DualSPHysics with variable resolution for multi-scale modelling [38]. The extension of DualSPHysics to address geomechanics problems stands to benefit from both its growing community and newly incorporated physics, algorithms, and parallelization optimizations. New users interested in geomechanics simulation can also take advantage of the solver's powerful pre-/post-processing tools, extensive documentation/tutorials, and active community support.

This work presents GeoDualSPHysics, a new branch of the DualSPHysics code for large-scale large-deformation problems in geomechanics, geophysics and geotechnical engineering. GeoDualSPHysics inherits the highly parallelised SPH framework from the original DualSPHysics code and further enables the simulation of geomaterial with a stabilised, noise-free treatment and an accurate resolution of solid boundaries. The two-way coupling between the geophysical flow and multiple interactive structures is also achieved through the inclusion of DSPHChronoLib by Martínez-Estévez et al. [35]. Special attention is given to the performance of the newly implemented functionality, in which optimisations are made with the NVIDIA Nsight Compute to maximise the GPU occupancy while minimising the memory requirements.

The remaining part of this paper is organised as follows: Section 2 introduces the governing equations and SPH formulations for geomaterials modelling. In Section 3, two-way coupling between GeoDualSPHysics and Project Chrono for the interaction between geomaterials and multibody system is presented. This is followed by the code documentation for the new solver as described in Section 4. Sections 5 presents the adopted benchmark tests for validations. The performance analysis of the new solver is shown in Section 6. Finally, conclusions are summarized in Section 7.

2. Numerical model for geomaterials

2.1. SPH fundamentals

In SPH, the computational domain is discretised as a set of particles (i.e., computational points), where each particle carries physical quantities and moves according to governing equations. As no computational mesh is adopted, SPH relies on point-based interpolation. The value at a point $\mathbf{x}$ is computed using a differentiable, compactly supported kernel function W, whose value decreases with the distance between particles $|\mathbf{x} - \mathbf{x}'|$ and depends on the smoothing length h. The SPH interpolation is achieved in two steps: integral approximation and then discrete approximation. The integral approximation of the value of a field function $f(\mathbf{x})$ at the point $\mathbf{x}$ is defined as,

$$\langle f(\mathbf{x})\rangle = \int_{\Omega} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{1}$$

where Ω is the local interpolation region, and $\langle\ldots\rangle$ denotes the SPH interpolation.

The discrete form of Eq. (1) can be written as,

$$\langle f(\mathbf{x})\rangle_i = \sum_{j=1}^{N} f_j W\left(\mathbf{x}_i - \mathbf{x}_j, h\right)V_j \tag{2}$$

where the subscript i and j denote the interpolating particle and its neighbours, respectively, N is the number of particles distributed in the interpolation domain, $f_j = f(\mathbf{x}_j)$, and V_j stands for the associated volume of particle j.

For the gradient of a function $\nabla f(\mathbf{x})$, its integral approximation in SPH reads,

$$\langle \nabla f(\mathbf{x})\rangle = \int_{\Omega} \nabla f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{3}$$

The discrete approximation of Eq. (3) is given by,

$$\langle \nabla f(\mathbf{x})\rangle_i = -\sum_{j=1}^{N} f_j \nabla W\left(\mathbf{x}_i - \mathbf{x}_j, h\right)V_j \tag{4}$$

For further details on the SPH approximations of differential operators (including gradient, divergence, and Laplacian operators), the reader is referred to Violeau and Rogers [39].

In this work, the 5th-order Wendland kernel C^2, which is stable against clumping instability, is adopted. The kernel function takes the following form,

$$W(q,h) = \begin{cases} \alpha_d \left(1 - \dfrac{q}{2}\right)^4 (2q+1) & 0 \le q \le 2 \\ 0 & q > 2 \end{cases} \tag{5}$$

where α_d is the normalization constant. In one-, two- and three-dimensional space, α_d is equal to $3/4h$, $7/(4\pi h^2)$, and $21/(16\pi h^3)$ respectively and q is the dimensionless distance between points at $\mathbf{x}$ and $\mathbf{x}'$, defined as $q = |\mathbf{x} - \mathbf{x}'|/h$.

2.2. Governing equations

The Lagrangian form of mass and momentum balance equations for the continuum representation of geomaterials is written as,

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{v} \tag{6}$$

$$\frac{d\mathbf{v}}{dt} = \frac{1}{\rho}\nabla\cdot\boldsymbol{\sigma} + \mathbf{b} \tag{7}$$

where $\mathbf{v}$ represents the velocity, $\boldsymbol{\sigma}$ denotes the Cauchy stress tensor, ρ indicates the density, and $\mathbf{b}$ is the body force per unit volume.

2.3. Constitutive models

To close the governing equations Eqs (6) and (7), an additional constitutive equation is needed. In GeoDualSPHysics, the elastoplastic Drucker–Prager model is implemented and tested. Other advanced constitutive models can be readily implemented in the new code by modifying the functionalities in the time integration. The Jaumann stress rate is adopted to ensure the objectivity of stress under rigid body rotation. The semi-implicit stress updating algorithm by Bui and Nguyen [9] is adopted and its algorithm is presented in Table 1. The equivalent deviatoric plastic strain κ is also calculated in the stress update.

Table 1
Algorithm for semi-implicit stress updating.

```
Algorithm1 Stress update algorithm
//-Interaction_Forces
for each particle $i$ do
  Compute the strain rate and spin rate tensor
  Compute the stress rate based on elastic stiffness $\dot{\boldsymbol\sigma} = \mathbf{D}_e : \dot{\boldsymbol\varepsilon} - \dot{\boldsymbol\omega}\cdot\boldsymbol\sigma + \boldsymbol\sigma\cdot\dot{\boldsymbol\omega}$
end for
//-Time_integration
for each particle $i$ do
  Elastic trial solution $\boldsymbol\sigma^{trial}_{t+dt} = \boldsymbol\sigma_t + \dot{\boldsymbol\sigma}dt;\ \kappa_{t+dt} = \kappa_t$
  Check yielding condition $f(\boldsymbol\sigma^{trial}_{t+dt}, \kappa_{t+dt}) < 0$
    If $f(\boldsymbol\sigma^{trial}_{t+dt}, \kappa_{t+dt}) < 0$ then
      Update stress as elastic trial solution $\boldsymbol\sigma_{t+dt} = \boldsymbol\sigma^{trial}_{t+dt};\ \kappa_{t+dt} = \kappa_t$
    else
      Perform return mapping $\boldsymbol\sigma_{t+dt} = \boldsymbol\sigma^{trial}_{t+dt} - d\boldsymbol\sigma_p,\ \kappa_{t+dt} = \kappa_t + d\kappa$
    end if
end for
```

In this approach, the stress rate is firstly calculated according to the elastic stiffness matrix according to,

$$\dot{\boldsymbol\sigma} = \mathbf{D}_e : \dot{\boldsymbol\varepsilon} - \dot{\boldsymbol\omega}\cdot\boldsymbol\sigma + \boldsymbol\sigma\cdot\dot{\boldsymbol\omega} \tag{8}$$

where $\mathbf{D}_e$ is the elastic stiffness tensor, $\boldsymbol\varepsilon$ is the strain tensor, and $\boldsymbol\omega$ is the spin tensor.

The strain rate tensor $\dot{\boldsymbol\varepsilon}$ and the spin rate tensor $\dot{\boldsymbol\omega}$ can be related to velocity gradient from kinematic relations according to,

$$\dot{\boldsymbol\varepsilon} = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right) \tag{9a}$$

$$\dot{\boldsymbol\omega} = \frac{1}{2}\left(\nabla\mathbf{v} - (\nabla\mathbf{v})^T\right) \tag{9b}$$

The calculated stress rate using the elastic assumption results in a trial solution of the stress increment $d\boldsymbol\sigma^{trial}$ [40]. If the resulting stress condition reaches the yield condition f, the plastic return mapping is applied according to,

$$d\boldsymbol\sigma = d\boldsymbol\sigma^{trial} - \mathbf{D}_e : \left(d\lambda\frac{\partial g(\boldsymbol\sigma)}{\partial\boldsymbol\sigma}\right) \tag{10}$$

where g is the plastic potential function, $d\lambda$ is the plastic multiplier.

With the Drucker-Prager yield criterion, the yield function and plastic potential function are given by,

$$f = \alpha_\phi I_1 + \sqrt{J_2} - k_c \tag{11a}$$

$$g = \alpha_\psi I_1 + \sqrt{J_2} \tag{11b}$$

where I_1 and J_2 denote the first principal stress invariant and the second deviatoric stress, respectively, α_ϕ and k_c are Drucker-Prager constants that can be related to cohesion c and internal friction ϕ according to, for plane strain conditions,

$$\alpha_\phi = \frac{\tan\phi}{\sqrt{9+12\tan^2\phi}}, k_c = \frac{3c}{\sqrt{9+12\tan^2\phi}} \tag{12a}$$

for 3D conditions that the Drucker-Prager yield surface circumscribes the Mohr–Coulomb yield surface,

$$\alpha_\phi = \frac{2\sin\phi}{\sqrt{3}(3-\sin\phi)}, k_c = \frac{6c\cos\phi}{\sqrt{3}(3-\sin\phi)} \tag{12b}$$

for 3D conditions that the Drucker-Prager yield surface middle circumscribes the Mohr–Coulomb yield surface,

$$\alpha_{\phi} = \frac{2\sin\phi}{\sqrt{3}(3+\sin\phi)}, k_c = \frac{6c\cos\phi}{\sqrt{3}(3+\sin\phi)} \tag{12c}$$

α_{ψ} is a dilatancy factor, whose value is related to the dilation angle ψ in a similar manner to that between α_{ϕ} and friction angle φ.

The implementation of the constitutive model has been validated by the simple shear test suggested by Bui and Nguyen [9]. Details of the validation can be found in Feng et al. [31].

2.4. SPH discretisation

The SPH discretized forms of the governing equations (Eqs. 6 and 7), along with the strain rate, spin rate equations (Eqs. 9a and 9b) and the position equation, for particle i representing the geomaterials, are expressed as,

$$\left\langle \frac{d\rho}{dt} \right\rangle_i = \rho_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(v_i^{\alpha} - v_j^{\alpha} \right) \frac{\partial W_{ij}}{\partial x_i^{\alpha}} \tag{13a}$$

$$\left\langle \frac{dv^{\alpha}}{dt} \right\rangle_i = \frac{1}{\rho_i} \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(\sigma_i^{\alpha\beta} + \sigma_j^{\alpha\beta} \right) \frac{\partial W_{ij}}{\partial x_i^{\beta}} + g^{\alpha} \tag{13b}$$

$$\langle \dot{\varepsilon}^{\alpha\beta} \rangle_i = \frac{1}{2} \left(\sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(v_j^{\alpha} - v_i^{\alpha} \right) \frac{\partial W_{ij}}{\partial x_i^{\beta}} + \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(v_j^{\beta} - v_i^{\beta} \right) \frac{\partial W_{ij}}{\partial x_i^{\alpha}} \right) \tag{13c}$$

$$\langle \dot{\omega}^{\alpha\beta} \rangle_i = \frac{1}{2} \left(\sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(v_j^{\alpha} - v_i^{\alpha} \right) \frac{\partial W_{ij}}{\partial x_i^{\beta}} - \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(v_j^{\beta} - v_i^{\beta} \right) \frac{\partial W_{ij}}{\partial x_i^{\alpha}} \right) \tag{13d}$$

$$\frac{dx_i^{\alpha}}{dt} = v_i^{\alpha} \tag{13e}$$

where the subscripts ij represents the difference in value in the function f_{ij}=f_i-f_j and g^{α} is the gravitational force. For clarity, the equations are presented in indicial notation and Einstein convention.

2.5. Stabilisation techniques

2.5.1. Artificial viscosity

In the absence of a numerical dissipative term in the governing equations, SPH may exhibit unphysical oscillations and numerical instability due to the centered and collocated scheme. The classical artificial viscosity term [41] is often adopted to dissipate these oscillations and stabilise the numerical algorithm. The artificial viscous term is introduced into the pressure term of the momentum equation (13b) in the following way,

$$\left\langle \frac{dv^{\alpha}}{dt} \right\rangle_i = \sum_{j=1}^{N} m_j \left(\frac{\sigma_i^{\alpha\beta} + \sigma_j^{\alpha\beta}}{\rho_i \rho_j} - \Pi_{ij} \delta^{\alpha\beta} \right) \frac{\partial W_{ij}}{\partial x_i^{\beta}} + g^{\alpha} \tag{14}$$

where

$$\prod_{ij} = \begin{cases} \dfrac{-\alpha c_s \mu_{ij}}{\overline{\rho}_{ij}} & v_{ij}^{\beta} x_{ij}^{\beta} \le 0 \\ 0 & v_{ij}^{\beta} x_{ij}^{\beta} > 0 \end{cases} \tag{15a}$$

$$\mu_{ij} = \frac{h v_{ij}^{\beta} x_{ij}^{\beta}}{x_{ij}^{\gamma} x_{ij}^{\gamma} + \eta^2} \tag{15b}$$

$$\overline{\rho}_{ij} = 0.5\left(\rho_i + \rho_j\right) \tag{15c}$$

α is the empirical constant used to control the magnitude of numerical dissipation and normally takes the value of $\alpha = 0.1$. The factor η=0.1h is added to the denominator to avoid numerical singularity. c_s is the speed of sound of the material, which is computed according to,

$$c_s = \sqrt{(4G/3 + K)/\rho} \tag{16}$$

where K and G are bulk modulus and shear modulus of the material, respectively.

2.5.2. Numerical diffusion term

In the SPH simulation of large deformations in geomaterials, high-frequency noise may arise in the stress field due to the collated nature of the method. To smooth the stress field, the numerical diffusion term is introduced to reduce numerical noise. This involves the inclusion of a stress diffusion term into the stress rate equation according to [31],

$$\frac{d\sigma_i^{\alpha\beta}}{dt} = D_e^{\alpha\beta\gamma l} \dot{\varepsilon}_i^{\gamma l} - \dot{\omega}^{\alpha\gamma} \sigma^{\gamma\beta} + \sigma^{\alpha\gamma} \dot{\omega}^{\gamma\beta} + D_i^{\alpha\beta} \tag{17}$$

where $D_i^{\alpha\beta}$ is the diffusion term that has a general form as follows,

$$D_i^{\alpha\beta} = 2\zeta h c_s \sum_j \psi_{ij}^{\alpha\beta} \frac{x_{ij}^{\gamma}}{x_{ij}^{l} x_{ij}^{l} + \eta^2} \frac{\partial W_{ij}}{\partial x_i^{\gamma}} \frac{m_j}{\rho_j} \tag{18}$$

where ζ is the diffusion coefficient that controls the magnitude of diffusion. Its value normally takes as 0.1 for most application [42]. The parameter $\psi_{ij}^{\alpha\beta}$ is the diffusion operator, varying depending on the formulation. Its dimensions correspond to those of the diffused physical quantities. To maintain consistency between the stress rate and strain rate, the strain rate may need to be corrected based on the magnitude of the diffusion term. Since the strain rate serves as an intermediate variable in this work, such a correction is not necessary. However, this adjustment could become essential when employing a strain-history dependent constitutive model.

Two types of diffusion operator are implemented in the new solver. The first formula is originated from Molteni and Colagrossi [43], given by,

$$\psi_{ij}^{\alpha\beta} = \sigma_i^{\alpha\beta} - \sigma_j^{\alpha\beta} \tag{19}$$

The second form of stress diffusion operator is proposed by Feng et al. [31] that can be expressed as,

$$\begin{cases} \psi_{ij}^{\alpha\beta} = \sigma_{ij}^{\alpha\beta} & \alpha \neq \beta \\ \psi_{ij}^{xx} = \sigma_{ij}^{xx} - K_0 \rho_0 g z_{ij} & \\ \psi_{ij}^{yy} = \sigma_{ij}^{yy} - K_0 \rho_0 g z_{ij} & \\ \psi_{ij}^{zz} = \sigma_{ij}^{zz} - \rho_0 g z_{ij} & \end{cases} \tag{20}$$

where K_0 is Jaky's earth pressure coefficient at rest, defined as $K_0 = 1$-sinϕ, ρ_0 is the initial density, and z_{ij} is the distance between particles i and j in z-axis.

The first form, given in Eq. (19), approximates the Laplacian operator by Morris et al. [44], providing spatial smoothing of the given quantities. While it performs well in non-static cases, it has free-surface inconsistencies due to the kernel truncation, which become particularly noticeable in static cases involving free surfaces. The second form addresses this issue by considering only the dynamic components of the diffusion term and its formulation (Eq. 20) reduces to the first form (Eq. 19) in the absence of gravitational forces. The implemented diffusion term in Eq. (20) is expected to work mostly for gravity-dominated flows. For general applications to any type of flow and consistent formulations, the extension to the diffusion term in Feng et al. [45] is required. More details regarding the comparisons of different formulations and the in-depth analysis of noise reduction by the diffusion term can be found in Feng et al. [31]. Other methods, such as the use of density diffusion for improving the isotropic part of the stress as proposed by Khayyer et al. [46], which demonstrates favourable energy conservation, can also be considered a potential option for stress smoothing.

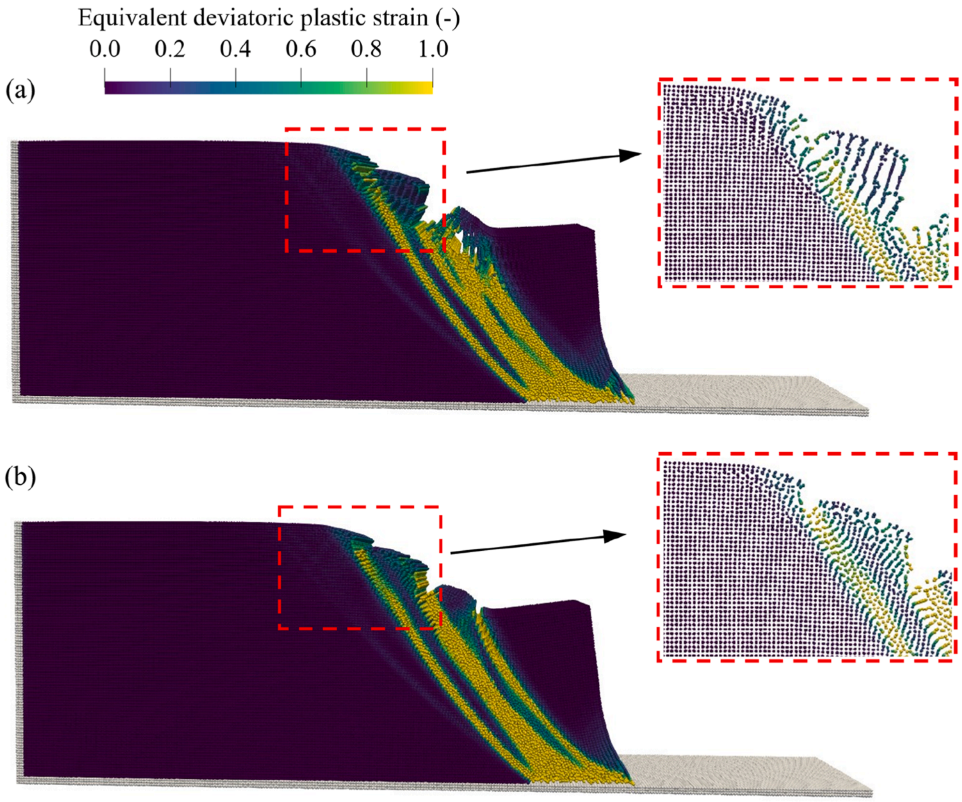


Fig. 1. Three-dimensional modelling of cohesive granular column collapse (a) without shifting (b) with shifting.

2.5.3. *Particle shifting*

In geomechanics SPH applications, numerical issues such as particle clustering and voids can arise due to tensile instability. These effects are especially pronounced in cohesive material simulations [9]. To mitigate these issues, a particle shifting technique is often applied to regularize particle distributions (e.g., [21,47,48]). The shifting algorithm in the original solver DualSPHysics is retained for improving the particle distributions. With the shifting technique, each particle is shifted a certain distance according to the concentration of particles C_i at the end of each time step, allowing particles in denser areas to move towards areas with lower particle concentration [47]. The shifting distance with free-surface corrections is calculated as,

$$\delta x_i^\alpha = \begin{cases} -A_{FSC.i} A h v_{mag.i} dt \dfrac{\partial C_i}{\partial x^\alpha} & if \partial_\alpha x_i^\alpha > A_{FST} \\ 0 & else \end{cases} \tag{21}$$

where A is the dimensionless shifting coefficient, whose value is independent of the case scenario and is set to 2 by default [48]. $v_{mag.i}$ is the velocity magnitude calculated as $v_{mag.i} = \sqrt{v_i^\alpha v_i^\alpha}$. $\partial_\alpha x_i^\alpha$ is the particle divergence equivalent to $(\nabla\cdot\mathbf{x})_i$. A_{FST} is the threshold to identify particles located at the free surface, and its recommended values are 1.5 for 2D and 2.75 for 3D. A_{FSC} is the shifting coefficient considering free-surface correction, given by,

$$A_{FSC.i} = \frac{\partial_\alpha x_i^\alpha - A_{FST}}{A_{FSM} - A_{FST}} \tag{22}$$

in which, A_{FSM} is the maximum value of the particle divergence given by,

$$A_{FSM} = \begin{cases} 2 & for2D \\ 3 & for3D \end{cases} \tag{23}$$

The performance of the employed shifting algorithm in modelling cohesive granular materials is demonstrated in Fig. 1, which shows the results of three-dimensional simulations of a cohesive granular column collapse with and without the shifting algorithm. The particle shifting algorithm effectively mitigates the unphysical clumping and voids that arise from tensile instability. However, it is also noted that the present shifting method adopts simplified free-surface corrections, for example preventing shifting at the free surface, may not perform as well in the vicinity of the free surface compared to those with consistent free-surface correction technique, such as optimized particle shifting (OPS) by Khayyer et al. [49]. With some extra computational cost, a consistent free-surface correction can lead to a more regular particle distribution near free surface. An example of modifying OPS for geomechanics problems can be found in Feng et al. [21]. In addition, the unified transport velocity formulation by Zhang et al. [37] can also be considered as an option to improve particle distributions near the free surface.

A further improvement to the shifting algorithm is to introduce conservative considerations, such as the SPH-ALE formulation [50], and volume conservation shifting [51], to avoid unphysical, numerically-induced volume expansion or compression in explicit scheme.

It is worth mentioning that, in addition to the implemented stabilization techniques, other advanced methods are also available in SPH to address the aforementioned stability issues in fluid and solid dynamics applications, such as Riemann-based stabilization term [52], density diffusion in elastic structural dynamics [46], and the hourglass control method [53,54]. The application of these techniques to geomechanics problems will be considered in future investigations.

2.6. *Boundary treatment*

2.6.1. *Extension of modified dynamic boundary condition*

The boundary treatment is a challenging problem in SPH due to the kernel truncation issue for particles near the boundary. In GeoDualSPHysics, the boundary treatment by Feng et al. [31] for granular material modelling, which builds upon the modified dynamic boundary

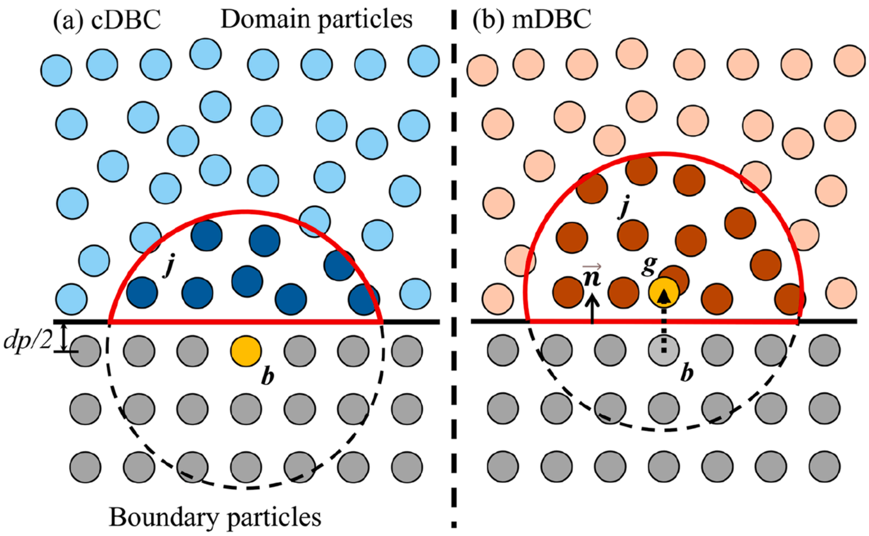


Fig. 2. Schematic diagram of the solid boundary treatment (a) corrected dummy boundary condition (cDBC); (b) modified dynamic boundary condition (mDBC).

condition (mDBC) by English et al. [55] , is used. As shown in Fig. 2, several layers of dummy boundary particles are created for completing the kernel support of particles truncated by boundary. For each boundary particle, b, there are ghost nodes, g, projected into the computational domain to interpolate field variables from surrounding material particles, j.

The stress and stress gradients of ghost particles are computed using the first-order consistent SPH interpolant introduced by Liu and Liu [56], given by,

$$\mathbf{A}_g \cdot \begin{bmatrix} \sigma_g^{\alpha\beta} \\ \partial_x \sigma_g^{\alpha\beta} \\ \partial_y \sigma_g^{\alpha\beta} \\ \partial_z \sigma_g^{\alpha\beta} \end{bmatrix} = \begin{bmatrix} \sum_j \sigma_j^{\alpha\beta} W_{gj} V_j \\ \sum_j \sigma_j^{\alpha\beta} \partial_x W_{gj} V_j \\ \sum_j \sigma_j^{\alpha\beta} \partial_y W_{gj} V_j \\ \sum_j \sigma_j^{\alpha\beta} \partial_z W_{gj} V_j \end{bmatrix} \tag{24}$$

where the renormalisation matrix $\mathbf{A}_g$ is given by,

$$\mathbf{A}_g = \begin{bmatrix} \sum_j W_{gj} V_j & \sum_j (x_j - x_g) W_{gj} V_j & \sum_j \left(y_j - y_g\right) W_{gj} V_j & \sum_j (z_j - z_g) W_{gj} V_j \\ \sum_j \partial_x W_{gj} V_j & \sum_j (x_j - x_g) \partial_x W_{gj} V_j & \sum_j \left(y_j - y_g\right) \partial_x W_{gj} V_j & \sum_j (z_j - z_g) \partial_x W_{gj} V_j \\ \sum_j \partial_y W_{gj} V_j & \sum_j (x_j - x_g) \partial_y W_{gj} V_j & \sum_j \left(y_j - y_g\right) \partial_y W_{gj} V_j & \sum_j (z_j - z_g) \partial_y W_{gj} V_j \\ \sum_j \partial_z W_{gj} V_j & \sum_j (x_j - x_g) \partial_z W_{gj} V_j & \sum_j \left(y_j - y_g\right) \partial_z W_{gj} V_j & \sum_j (z_j - z_g) \partial_z W_{gj} V_j \end{bmatrix} \tag{25}$$

The final stress value of the boundary particle $\sigma_b^{\alpha\beta}$ is extrapolated from the position of ghost nodes, using the first-order Taylor series expansion according to,

$$\sigma_b^{\alpha\beta} = \sigma_g^{\alpha\beta} + (x_b - x_g)\partial_x \sigma_g^{\alpha\beta} + \left(y_b - y_g\right)\partial_y \sigma_g^{\alpha\beta} + (z_b - z_g)\partial_z \sigma_g^{\alpha\beta} \tag{26}$$

The invertibility of the renormalisation matrix $\mathbf{A}_g$ is checked by calculating its determinant. In the situation where the determinant is very small (by default, lower than 0.001) the matrix becomes ill-conditioned, and a zeroth order consistent SPH interpolant (Shepard function) is used, given by,

$$\sigma_b^{\alpha\beta} = \sigma_g^{\alpha\beta} = \frac{\sum_j \sigma_j^{\alpha\beta} W_{gj} V_j}{\sum_j W_{gj} V_j} \tag{27}$$

In addition to the zero-velocity boundary option, the no-slip boundary condition is also enabled. For the no-slip case, the velocity at the ghost nodes is calculated using the Shepard corrected interpolation, defined as,

$$v_g^\alpha = \frac{\sum_j v_j^\alpha W_{gj} V_j}{\sum_j W_{gj} V_j} \tag{28}$$

To enforce no-slip and no-penetration conditions, the velocity of boundary particles is set as,

$$v_b^\alpha = 2v_M^\alpha - v_g^\alpha \tag{29}$$

where v_M^α is the motion velocity of the solid boundary, for example, in cases of rigid body dynamics or solid boundaries with prescribed motions. It is noted that only v_M^α is the velocity to update the positions of boundary particles.

2.6.2. *Corrected dummy boundary condition*

The extended mDBC provides an option for accurately resolving stress and velocity condition at the solid boundaries. However, the requirement for boundary normals presents complexities when dealing with complex geometries, such as real terrains. For this, GeoDualSPHysics also implements a new alternative boundary treatment method, named corrected dummy boundary condition (cDBC). Similar to mDBC, dummy boundary particles are used in cDBC, as presented in Fig. 2(a), but the boundary properties are evaluated by the local first-order interpolant at boundary particles b over neighbouring material particles j without the use of ghost node. The stress value at the dummy boundary particles is calculated according to,

$$\sigma_b^{\alpha\beta} = \begin{vmatrix} \sum_j \sigma_j^{\alpha\beta} W_{bj} V_j & \sum_j (x_j - x_b) W_{bj} V_j & \sum_j \left(y_j - y_b\right) W_{bj} V_j & \sum_j (z_j - z_b) W_{bj} V_j \\ \sum_j \sigma_j^{\alpha\beta} \partial_x W_{bj} V_j & \sum_j (x_j - x_b) \partial_x W_{bj} V_j & \sum_j \left(y_j - y_b\right) \partial_x W_{bj} V_j & \sum_j (z_j - z_b) \partial_x W_{bj} V_j \\ \sum_j \sigma_j^{\alpha\beta} \partial_y W_{bj} V_j & \sum_j (x_j - x_b) \partial_y W_{bj} V_j & \sum_j \left(y_j - y_b\right) \partial_y W_{bj} V_j & \sum_j (z_j - z_b) \partial_y W_{bj} V_j \\ \sum_j \sigma_j^{\alpha\beta} \partial_z W_{bj} V_j & \sum_j (x_j - x_b) \partial_z W_{bj} V_j & \sum_j \left(y_j - y_b\right) \partial_z W_{bj} V_j & \sum_j (z_j - z_b) \partial_z W_{bj} V_j \end{vmatrix} \Bigg/ \begin{vmatrix} \sum_j W_{bj} V_j & \sum_j (x_j - x_b) W_{bj} V_j & \sum_j \left(y_j - y_b\right) W_{bj} V_j & \sum_j (z_j - z_b) W_{bj} V_j \\ \sum_j \partial_x W_{bj} V_j & \sum_j (x_j - x_b) \partial_x W_{bj} V_j & \sum_j \left(y_j - y_b\right) \partial_x W_{bj} V_j & \sum_j (z_j - z_b) \partial_x W_{bj} V_j \\ \sum_j \partial_y W_{bj} V_j & \sum_j (x_j - x_b) \partial_y W_{bj} V_j & \sum_j \left(y_j - y_b\right) \partial_y W_{bj} V_j & \sum_j (z_j - z_b) \partial_y W_{bj} V_j \\ \sum_j \partial_z W_{bj} V_j & \sum_j (x_j - x_b) \partial_z W_{bj} V_j & \sum_j \left(y_j - y_b\right) \partial_z W_{bj} V_j & \sum_j (z_j - z_b) \partial_z W_{bj} V_j \end{vmatrix} \tag{30}$$

Similar to mDBC, if the determinant of the renormalisation matrix in the denominator of Eq. (30) is very small (by default, less than 0.001), the interpolant reduces to a zeroth-order Shepard interpolation. The enforcement of velocity conditions also follows the same manner of mDBC in Eqs (28) and (29), but the interpolation is taken at the boundary particles instead of the ghost nodes.

2.7. *Time stepping*

The Velocity Verlet scheme and Symplectic Position Verlet scheme in the original DualSPHysics solver are adopted for the time integration of governing equations Eqs (13a) to (13e). The velocity Verlet scheme is a low computational cost scheme with second order accuracy in time. The position Verlet scheme is a symplectic integrator that operates in two substeps while maintaining second-order temporal accuracy. The details of these two schemes can be found in [32].

The time-stepping schemes are constrained by the Courant-Friedrichs-Lewy (CFL) condition, the maximum force term, and the numerical speed of sound. The variable time step is calculated according to,

$$\Delta t_f = \underset{i}{min}\left(\sqrt{h/\parallel \mathbf{f}_i \parallel}\right) \tag{31a}$$

$$\Delta t_{cv} = \underset{i}{min}\frac{h}{c_s + \underset{j}{max}\left|\frac{h\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}}{\mathbf{r}_{ij}^2+\eta^2}\right|} \tag{31b}$$

$$\Delta t = C_0 \min\left(\Delta t_{cv}, \Delta t_f\right) \tag{31c}$$

where Δt_f is the time step governed by force, Δt_{cv} is the time step constrained by the CFL condition, $\mathbf{f}$ is the force per unit mass (i.e., the acceleration computed in momentum equation), C_0 is the Courant number that is set to be 0.2 by default.

3. Modelling of interactions between geomaterials and multi-body systems

3.1. *Rigid body dynamics*

The motion of a rigid object driven by the movement of *geomaterials* is solved based on the fundamental equations of rigid body dynamics. In current framework, the rigid body is discretised as particles and is treated as solid boundary using method described in Section 2.6. The force on each particle k representing the rigid object is computed according to the sum of the contribution of all surrounding material particles j. Forces per unit mass exerted on the rigid body particles $\mathbf{f}_k$ by material particles is calculated as [57],

$$\mathbf{f}_k = \sum_{j \in geomaterials} \mathbf{f}_{kj} \tag{32a}$$

and its SPH formulation in current implementation is given by,

$$\mathbf{f}_k = \frac{1}{\rho_k} \sum_{j \in geomaterials}^{N} \frac{m_j}{\rho_j} (\boldsymbol{\sigma}_k + \boldsymbol{\sigma}_j) \cdot \nabla_k W_{kj} \tag{32b}$$

The motion of *rigid body* is calculated by solving the discretised Newton's equation as follow,

$$M\frac{d\mathbf{V}}{dt} = \sum_{k \in body} m_k \mathbf{f}_k \tag{33a}$$

$$\mathbf{I} \cdot \frac{d\boldsymbol{\Omega}}{dt} = \sum_{k \in body} m_k (\mathbf{r}_k - \mathbf{R}_0) \times \mathbf{f}_k \tag{33b}$$

where M is the mass of the object, $\mathbf{I}$ is the moment of inertia matrix, $\mathbf{V}$ is the linear velocity, $\boldsymbol{\Omega}$ is the angular velocity, $\boldsymbol{r}_k$ is the position of particle

Table 2
Algorithm for geomaterial-multibody interaction.

```
Algorithm2 Interaction between geomaterial and multibody system
//-Interaction_Forces
for each particle $k$ do
    Compute the forces exerted on particles representing rigid bodies using Eq. (32)
end for
//-RunFloating
for each rigid body $N$ do
    Compute the motion of rigid body $d\mathbf{V}/dt$, $d\boldsymbol{\Omega}/dt$ using Eqs. (33a) and (33b)
end for
//-DSPHChronoLib
Compute $\mathbf{F}$, $\mathbf{T}$ of rigid bodies according to $d\mathbf{V}/dt$, $d\boldsymbol{\Omega}/dt$
Transfer $\mathbf{F}$, $\mathbf{T}$, and $\Delta t_{SPH}$ to Project Chrono
//-Project Chrono
Detect collisions and apply mechanical constraints for $\mathbf{f}_c$
Compute the $\mathbf{V}$, $\boldsymbol{\Omega}$, and $\mathbf{R}_0$ of the rigid bodies according to Eqs. (35a) to (35c)
Transfer the $\mathbf{V}$, $\boldsymbol{\Omega}$, and $\mathbf{R}_0$ of the rigid bodies back to SPH solver
//-FtUpdate
for each rigid body $N$ do
    Update positions of the rigid bodies using Eq. (34)
end for
```

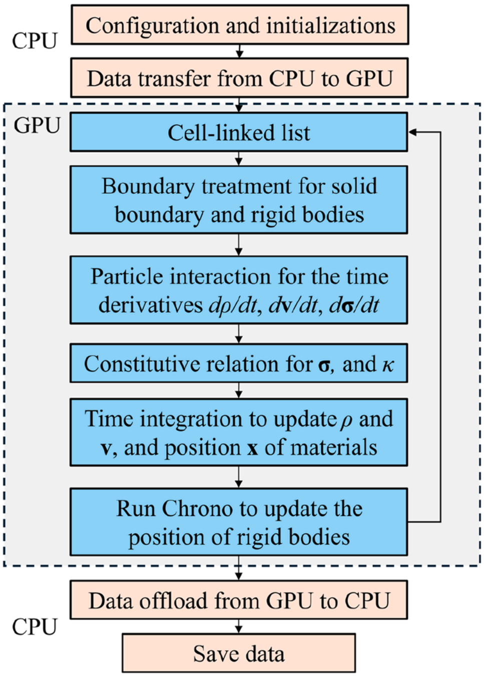

CPU
Configuration and initializations
Data transfer from CPU to GPU
GPU
Cell-linked list
Boundary treatment for solid boundary and rigid bodies
Particle interaction for the time derivatives $d\rho/dt$, $d\mathbf{v}/dt$, $d\boldsymbol{\sigma}/dt$
Constitutive relation for $\boldsymbol{\sigma}$, and κ
Time integration to update ρ and $\mathbf{v}$, and position $\mathbf{x}$ of materials
Run Chrono to update the position of rigid bodies
Data offload from GPU to CPU
CPU
Save data

Fig. 3. Computational workflow and GPU parallel framework of GeoDualSPHysics.

k, and $\mathbf{R}_0$ is the centre of the mass.

The computed linear ($d\mathbf{V}/dt$) and angular acceleration ($d\boldsymbol{\Omega}/dt$) are then transferred to Project Chrono to compute $\mathbf{V}$, $\boldsymbol{\Omega}$, and $\mathbf{R}_0$ in the global frame, taking into account mechanical constraints. Finally, the velocity of the particles that represents the rigid body is calculated by,

$$\mathbf{u}_k = \mathbf{V} + \boldsymbol{\Omega} \times (\mathbf{r}_k - \mathbf{R}_0) \tag{34}$$

3.2. Multibody dynamics

The interactions between rigid bodies are solved by the multibody dynamics library Project Chrono [58], which adopts the Discrete Element Method (DEM). The dynamics of the multibody system are solved by,

$$\frac{d\mathbf{q}}{dt} = \mathbf{L}(\mathbf{q})\mathbf{v} \tag{35a}$$

$$\mathbf{M}\frac{d\mathbf{v}}{dt} = \mathbf{f}_t(t, \mathbf{q}, \mathbf{v}) \tag{35b}$$

$$\mathbf{f}_t(t, \mathbf{q}, \mathbf{v}) = \mathbf{f}_e - \mathbf{f}_c \tag{35c}$$

where $\mathbf{q}$ and $\mathbf{v}$ are respectively the positions and velocities of the multibody system, and they are related by using a linear transformation matrix $\mathbf{L}(\mathbf{q})$. $d\mathbf{q}/dt$ includes the time derivatives of both the position and Euler parameters, describing the rigid body's motion in the global frame. $\mathbf{v}$ contains the linear velocity as well as the angular velocity in the body-fixed frame. Detailed formulations can be found in [59,60]. $\mathbf{f}_t$ is the sum of the total forces that consists of external forces $\mathbf{f}_e$ and constraint forces $\mathbf{f}_c$. $\mathbf{M}$ is the matrix of mass of the multibody system.

The external forces $\mathbf{f}_e$ are considered as the forces from the geomaterials, solved in the SPH solver, while the constraint forces $\mathbf{f}_c$ are the forces within multibody systems due to mechanical constraints, resolved in the DEM solver Project Chrono. In this work, the main functionality tested and adopted in Project Chrono is the collision module with frictional constraints for the interaction between multiple rigid bodies. However, other mechanical constraints such as spherical joints, hinges, or springs are also available in the code and can be applied to boarder case scenarios such as vehicle-terrain interaction.

Two frictional contact models are available in Project Chrono, including smooth contacts (SMC) and non-smooth contacts (NSC) [58]. In SMC, overlaps between bodies are allowed, and frictional forces are resolved based on the partial deformation of the bodies in contact using the penalty-based method. In NSC, complementarity conditions are introduced to impose non-penetrations constraints for the bodies in contact, along with a Coulomb friction model to solve the frictional forces.

3.3. Coupling with Project Chrono

The two-way coupling between the geomechanics SPH solver and multibody dynamics DEM solver is achieved through the communication interface named DSPHChronoLib developed by Martínez-Estévez et al. [35]. The coupling procedures are listed in Table 2 and are briefly described below. For more details of the implementation, readers can refer to Martínez-Estévez et al. [35].

In each time step, the coupling is completed in three parts: (1) the SPH solver computes forces exerted by geomaterials on the rigid bodies according to Eq. (32). Then, the linear ($d\mathbf{V}/dt$) and angular acceleration ($d\boldsymbol{\Omega}/dt$) of the rigid bodies are obtained by Eqs. (33a) and (33b), and they are transferred to the DSPHChronoLib to compute the linear forces $\mathbf{F}$ and torsional forces $\mathbf{T}$; (2) DEM solver receives $\mathbf{F}$ and $\mathbf{T}$ from DSPHChronoLib and applies them as external forces ($\mathbf{f}_e$), alongside constraints forces ($\mathbf{f}_c$) computed internally, to solve the dynamics of the rigid bodies. Project Chrono then outputs linear velocity $\mathbf{V}$, angular velocity $\boldsymbol{\Omega}$, and the centre-of-mass position $\mathbf{R}_0$ to GeoDualSPHysics; (3) After receiving $\mathbf{V}$, $\boldsymbol{\Omega}$, and $\mathbf{R}_0$, velocities of the particles representing rigid bodies are updated by solving Eq. (34). The entire system is then updated in GeoDualSPHysics for the next time step calculation.

4. Code documentation

4.1. Overview of implementations

The implementations of the aforementioned algorithms and numerical schemes are based on DualSPHysics v5.2, utilizing its built-in OpenMP-based and CUDA-based parallel computing architecture and established SPH computational workflow. These frameworks have been adapted for solving geomechanics problems in the GeoDualSPHysics. Fig. 3 presents the computational workflow and GPU parallel

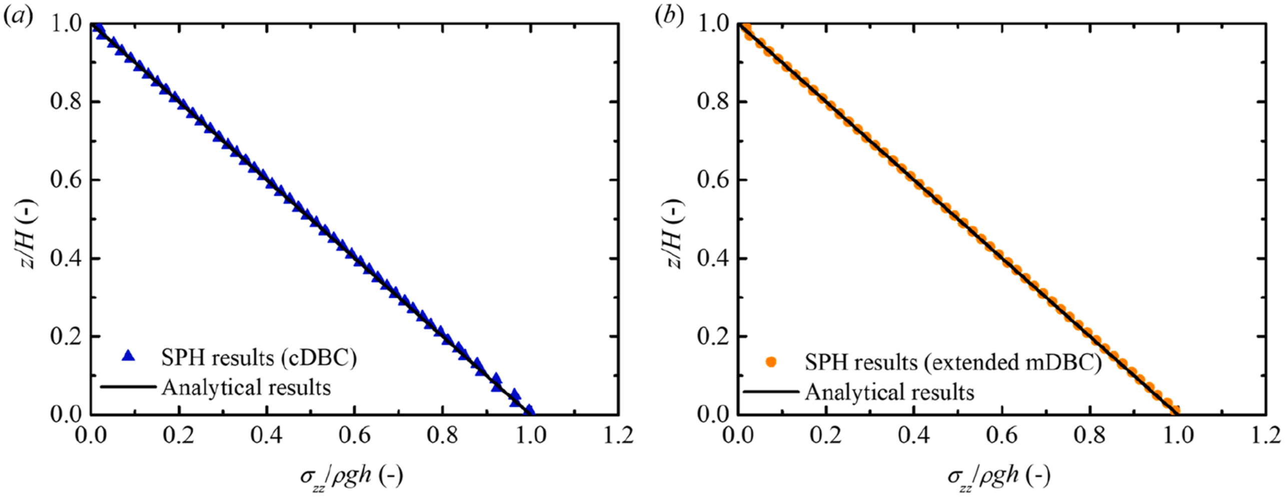


Fig. 4. Normalised vertical stress profile at final instant (t = 20 s) for the finest resolution using (a) cDBC and (b) extended mDBC.

framework.

The main added variables include the stress tensor $\boldsymbol{\sigma}$, the stress rate tensor $d\boldsymbol{\sigma}/dt$, and the equivalent deviatoric plastic strain κ. This necessitates the revision to the memory allocation for new variables on both CPU and GPU, data transfer of the newly allocated arrays between CPU and GPU, and cell-linked list for new variables. Furthermore, the configuration and initialisation are also modified to load the material properties and execution parameters from the XML file.

The newly incorporated physics and algorithms are included in the (1) boundary treatment, involving boundary-material interaction and rigid body-material interaction for the stress and velocity conditions on solid boundaries and rigid bodies, as presented in section 2.6, through functions *JSphCpu::Interaction_CdbcCorrection, JSphCpu::Interaction_MdbcCorrection, KerInteractionCdbcCorrection* and *KerInteractionMdbcCorrection*, (2) particle interaction, including material-material interaction, material-boundary interaction, and material-rigid body interaction for the evaluation of governing equations expressed in Eqs. (13a) to (13d), as well as rigid body-material interaction for forces exerted on rigid bodies from material in Eq. (32), shifting distance in Eq. (21), and the diffusion term in Eq. (17), all within functions *JSphCpu::InteractionForcesFluid* and *KerInteractionForcesFluid*, (3) time integration for updating stress and plastic strain using the semi-implicit stress update algorithm listed in Table 1, implemented in function *JSphCpu::ComputeVerlet, JSphCpu::ComputeSymplecticPre, JSphCpu::ComputeSymplecticCorr, KerComputeStepVerlet, KerComputeStepSymplecticPre*, and *KerComputeStepSymplecticCor*.

In GeoDualSPHysics, the CUDA configuration parameters remain the same as DualSPHysics. BlockSize (threads per block) has three calculation modes and can be changed using executions parameter -blocksize:< mode>. By default (-blocksize:0), BlockSize is fixed at 128 threads. Other options include: (-blocksize:1) where the optimal BlockSize is determined by the CUDA Occupancy Calculator, and (-blocksize:2) where the optimal BlockSize is calculated empirically based on the data used in the CUDA kernels. The number of blocks is determined by dividing the number of particles by the number of threads per block, using the formula (particleNumber + blockSize - 1) / blockSize to ensure that every particle is assigned to a thread. Shared memory is used selectively within CUDA kernels, primarily for caching particle information during neighbour searches and for reduction operations. For most particle data, global memory is used due to the large size of the datasets. Synchronization between threads is managed using the _syncthreads() function. This operation is applied only when coordination among threads is necessary to ensure maximum efficiency, for example, when calculating the maximum value in a data array, creating the list of new periodic particles, or generating the cell-linked list for neighbour searching.

GeoDualSPHysics employs a mixed precision strategy inherited from the original DualSPHysics code. Double precision is used where high precision is necessary for accuracy, such as for storing and calculating particle positions, while the rest adopts single precision. This strategy is a performance optimization for GPUs, as their architecture is designed to process single-precision operations significantly faster than double-precision operations. Previous studies of the DualSPHysics code have shown that this optimisation provides a significant performance improvement without loss of accuracy on GPU models [61].

In addition to the implementation, code optimisations are also conducted to achieve efficient instruction execution, reduced memory usage, and high GPU occupancy. Variables with multiple components are converted to built-in vector types during GPU execution, for instance, using float2 arrays for stress tensor computation on the GPU. CUDA built-in functions are employed for mathematical operations to enhance computational efficiency. Memory management in the original DualSPHysics code follows a static allocation approach: arrays are allocated when needed and deallocated immediately after use. Taking advantage of this feature, the memory allocation for the stress tensor is reduced since the array for the stress rate tensor is deallocated after the time integration, leaving the allocated memory for the extra stress tensor array required in neighbour list computation. Furthermore, the NVIDIA Nsight Compute for GPU profiling is adopted to enable the optimisation of performance metrics such as register usage per thread, improving occupancy through strategies such as reducing local variables and intermediate values. For example, in the Symplectic Position Verlet scheme, the density update is used in its standard form rather than the modified version in the original DualSPHysics solver, which includes additional local variables (i.e., *epsilon_rdot*) to improve density/pressure stability. This change does not affect the results because, in the current application, density is not directly linked to stress computation and is not prone to instability.

4.2. Compiling and running the code

The source code and files attached to this manuscript include precompiled GeoDualSPHysics executables along with dynamic libraries for DSPHChrono-Lib and Project Chrono. This allows users to immediately run the released version on CPU/GPU without need to compile the source code. In the situation that compilation is needed, the GNU C++ compiler is required for Linux-based systems, while Visual Studio IDE can be adopted for Windows systems with the use of provided Visual Studio 2022 solution in original DualSPHysics package or CMAKE. The CUDA toolkit that includes the NVCC compiler is required to compile

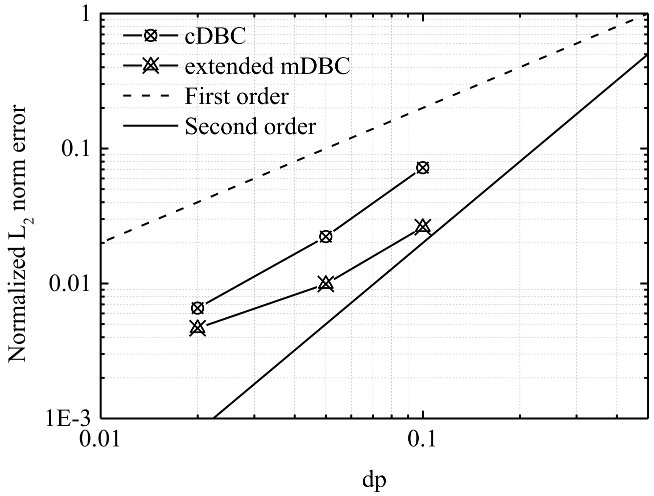


Fig. 5. Normalised L_2 error norms of vertical stress at $t = 20s$.

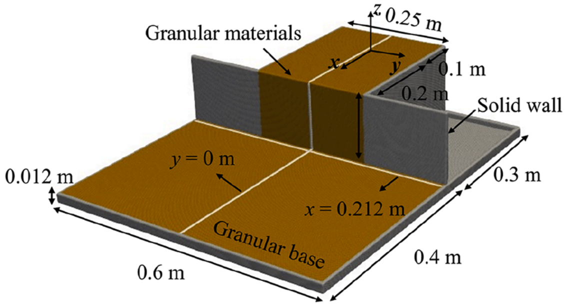


Fig. 6. Granular failure on deformable base: initial numerical configuration.

and run GeoDualSPHysics on an NVIDIA GPU card.

To compile GeoDualSPHysics with Project Chrono, the files *libChronoEngine.so* and *libdsphchrono.so* are included in the path of *src/lib/linux_gcc* and *bin/linux*. The GNU C++ compiler version 11, or later, should be used to compile the code. For details regarding the compilation with Project Chrono, readers can refer to Martínez-Estévez et al. [35].

Once the executable files have been built, the other files required to launch the simulation include the Case_Def.xml (eXtensible Markup Language) file and either a .sh (Shell) or .bat (batch) script. Identical to the hydrodynamics DualSPHysics package, users can modify Case_Def.xml to adjust geometries, material properties, or execution settings. For example, the “Boundary” option can be changed to value = “1” for cDBC, while value = “2” for mDBC. The script .sh/.bat specifies the location of the executables and contains command flags for these executable files.

The Case_Def.xml file is processed by GenCase, a precompiled preprocessing tool developed by DualSPHysics, which generates the particle discretisation and prepares input files for simulation. External geometries (e.g., complex terrain) can also be integrated into the numerical model. For guidance on generating geometries using XML or loading external geometries into the simulation, one can refer to the XML_GUIDE document in the DualSPHysics package for detailed instructions. The benchmark tests presented in the manuscript are provided in the folder *GeoDualSPHysics/examples*. For each benchmark, scripts are available to execute the code on Linux or Windows.

Upon launching the simulation, the script will call GenCase to generate the initial numerical configuration, and then GeoDualSPHysics will be called to run the simulation. The binary files output from GeoDualSPHysics will be finally processed by post-processing tools such as PartVtk. The output files include the format of VTK (Visualization Toolkit), which can be conveniently loaded into a visualization software such as Paraview for results analysis.

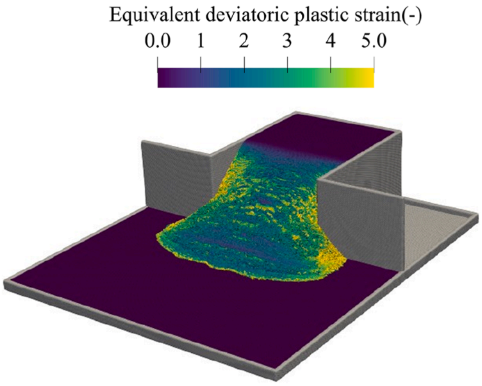


Fig. 7. The final deposit profile obtained from the simulation.

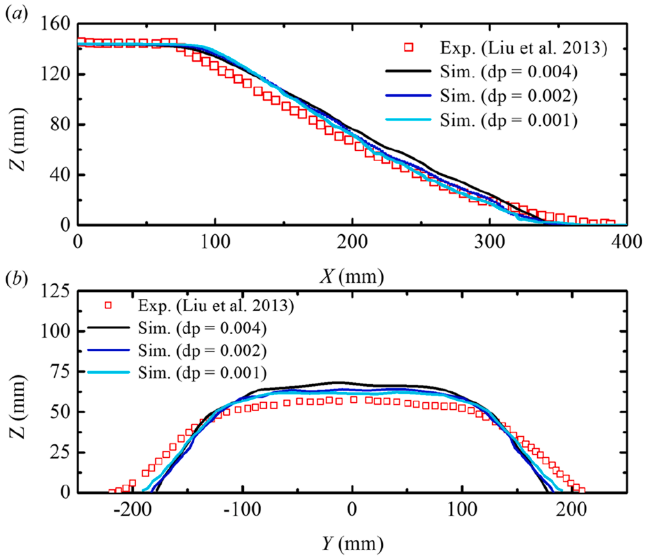


Fig. 8. Comparison with experiment measured free-surface profile at cross-sections of (a) $y = 0$ m and (b) $x = 0.212$ m.

5. Validation cases

The validity of GeoDualSPHysics in simulating large deformations of geomaterial and their interaction with multiple rigid bodies is assessed in this section through six benchmark examples with the comparison to available analytical solutions, numerical solutions and experimental data.

5.1. Static soil column

The simulation of a soil column at rest is conducted as an elemental benchmark test to study the convergence behaviour of the implemented SPH scheme and the performance of boundary treatments. The numerical configuration is taken the same as the previous work by Feng et al. [31]. The height of the soil column is set as 1 m, with periodic boundary treatment applied to the lateral boundaries and a solid boundary applied to the bottom. An increased value of artificial viscosity coefficient $\alpha = 1.0$ is used to settle down oscillations in the system due to the sudden application of gravitational forces [62]. The material properties are set as Young’s modulus $E = 10$ MPa, Poisson’s ratio $\upsilon = 0.3$, friction angle $\varphi = 33^\circ$, cohesion $c = 10$ kPa, dilation angle $\psi = 0^\circ$, and density $\rho = 2100$

Table 3
Statistical parameters A_r and P_d for different particle spacing at cross-sections of $y = 0$ m and $x = 0.212$ m.

	dp = 0.004 *m*	dp = 0.002 *m*	dp = 0.001 *m*
P_d ($x = 0.212$ *m*)	0.198	0.181	0.143
A_r ($x = 0.212$ m)	1.118	1.063	1.055
P_d ($y = 0$ *m*)	0.084	0.071	0.066
A_r ($y = 0$ m)	1.049	1.044	1.041

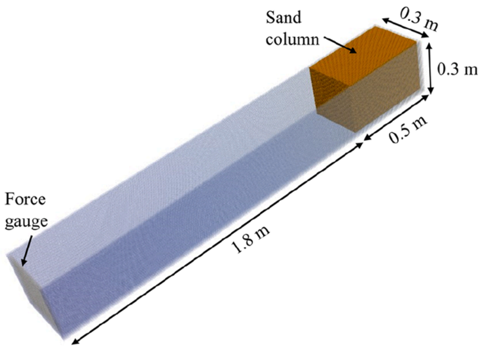


Fig. 9. Impact force of sand on a rigid wall: initial numerical configuration.

kg/m^3. The initial particle spacings are taken as dp = 0.1, 0.05, and 0.02 m. The simulations are conducted for 20 s of physical time.

Fig. 4 shows the normalised vertical stress from the simulation with dp = 0.02 m at the final time instant, compared against analytical solutions. Overall, both boundary options agree well with the analytical results. The cDBC exhibits slight fluctuations near the boundary, whereas the extended mDBC yields a smoother prediction with closer agreement with analytical solutions.

In addition, the normalized L_2 error norms of vertical stress at $t = 20s$ for three different initial particle spacings are presented in Fig. 5. The implemented SPH scheme shows a convergent behaviour with both the cDBC and the extended mDBC. The extended mDBC achieves overall higher accuracy than the cDBC, especially at coarse particle resolutions. However, for this case, the gap in accuracy between the two boundary methods narrows as the refinement of particle resolutions.

5.2. Granular failure on deformable base

This case simulates the collapse of granular material from a restricted box onto a deformable base to validate the solver's capacity in modelling large-deformation granular flow. This test has been experimentally investigated by Liu et al. [63] with measurements of the final deposit profile at $x = 0.212$ m and $y = 0$ m available. The initial numerical configuration is set to keep consistent with experiment setup as shown in Fig. 6. The material properties adopted in the simulation are referenced from Huang et al. [27], taking value of $E = 2$ MPa, $\upsilon = 0.3$, $\varphi = 33°$, $c = 0$ kPa, $\psi = 0°$, and density $\rho = 1530$ kg/m^3. For evaluating the convergence behaviour of the solver, three sets of initial particle spacings, i.e., dp = 0.001, 0.002, and 0.004 m, are adopted, resulting in a total particle number of 0.35 million, 2.29 million, and 16.45 million, respectively.

Fig. 7 displays the final deposit profile from the simulation using the finest particle resolution (dp = 0.001 m), coloured by deviatoric plastic strain. The results demonstrate that the simulation captures the mechanism of granular failure from the confined container, where the large deformation region and stationary region are clearly distinguished.

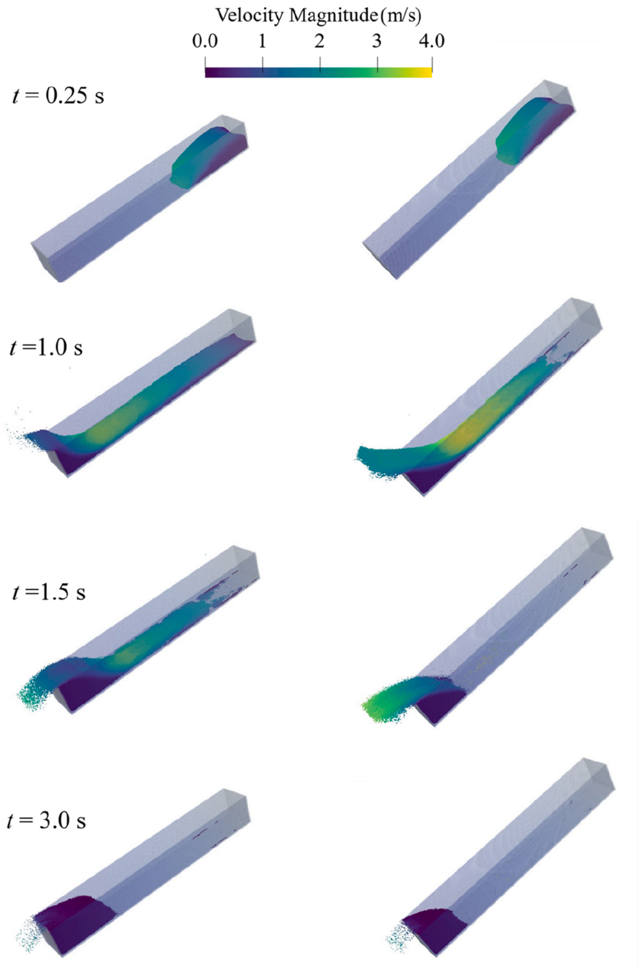


Fig. 10. The impact process of sand on the 45° inclined plane (left column) and 55° inclined plane (right column).

The post-processing tool IsoSurface is used to extract the location of the free-surface profile from the simulation output. However, since IsoSurface is less effective to capture non-planar surfaces, the profile at $x = 0.212$ m is generated by directly identifying the free-surface particles (e.g., based on the calculated particle divergence).

Fig. 8 compares the simulated final deposit profiles along two cross-sections of $x = 0.212$ m and $y = 0$ m with experiment results. The

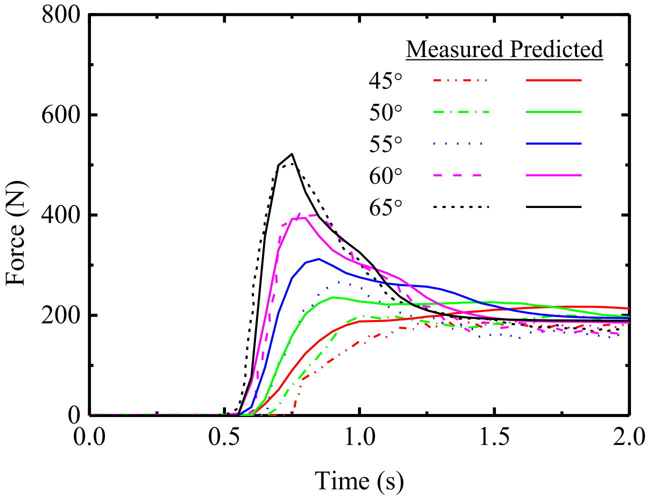


Fig. 11. Time history of the impact forces predicted in simulation with comparison against experiment measurements.

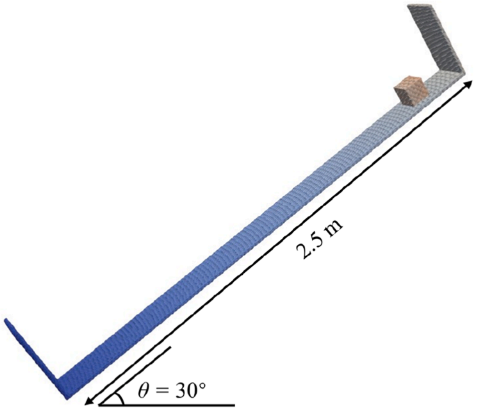


Fig. 12. Sliding motion of a cube on a slope: initial numerical configuration.

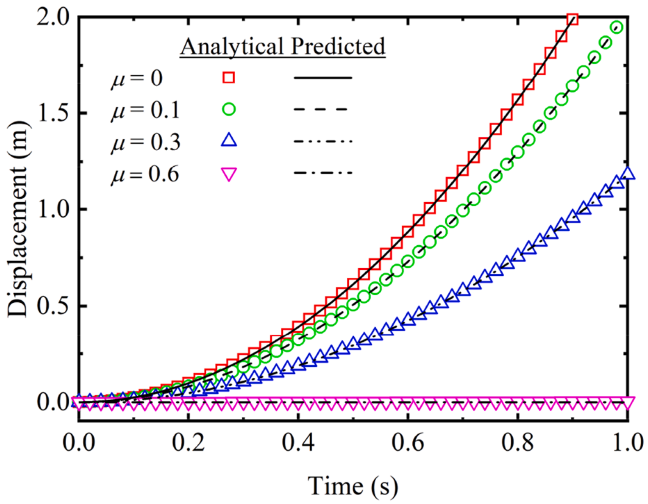


Fig. 13. The displacement of the sliding cube over time for different frictional coefficients.

differences between numerical and experimental results are quantified using statistical parameters defined as [64],

$$A_r = \left(\sum_{j}^{N} \left(f_j^{\mathrm{num}} \right)^2 \Big/ \sum_{j}^{N} \left(f_j^{\mathrm{exp}} \right)^2 \right)^{1/2} \tag{36a}$$

$$P_d = \left(\sum_{j}^{N} \left(f_j^{\mathrm{num}} - f_j^{\mathrm{exp}} \right)^2 \Big/ \sum_{j}^{N} \left(f_j^{\mathrm{exp}} \right)^2 \right)^{1/2} \tag{36b}$$

where A_r represents the relative amplitude between both datasets, in which perfect agreement yields $A_r \to 1$. P_d denotes the phase difference between both datasets, with perfect agreement resulting in $P_\mathrm{d} \to 0$. f represents the variable of interest (in this case, the position of free-surface particles), where subscripts denote experimental (exp) or numerical (num) values.

Table 3 lists the value of A_r and P_d obtained for different particle spacing at the cross-sections of $x = 0.212$ m and $y = 0$ m. Both statistical parameters imply a reasonable agreement between numerical results and experiment measurements. The value of P_d at $x = 0.212\ m$ appears relatively large but results become more accurate with decreasing particle spacing, indicating convergence of the predicted free-surface profile toward the experimental results. The discrepancies with experiment data may be attributed to greater granular material accumulation in the numerical simulation within the range of x =0.1 to 0.2 m compared to experiments, as shown in Fig. 8(a), resulting in less material flowing downslope. Potential reasons for this observation could vary, including soil heterogeneities, wall friction, and the accuracy of the numerical scheme. Future work may focus on these aspects to improve predictions.

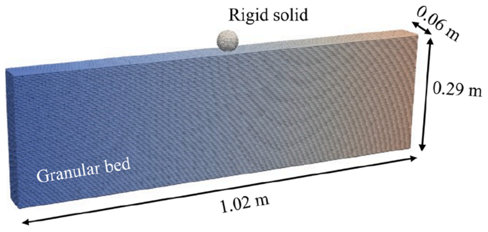


Fig. 14. Spherical solid impacting on a granular bed: initial numerical configuration.

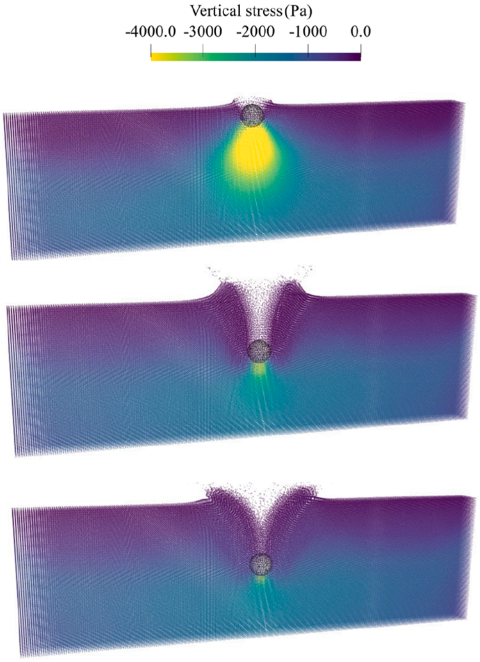


Fig. 15. Simulation results with the initial velocity of 3.03 m/s at different time instants (from top to bottom): (a) $t = 0.015$s, (b) $t = 0.08$s and (c) t =0.15s.

5.3. Impact force of sand on a rigid wall

The experiment by Moriguchi et al. [65] is simulated to validate impact forces of soil flow on a rigid wall. The configuration of the numerical model is illustrated in Fig. 9. A sand column is initially confined at the top of a flume and impacts the bottom rigid wall upon release. The impact forces are measured using a force sensor in the experiment.

The strength parameters are taken as $E = 21.6$ MPa, $\upsilon = 0.3$, $\varphi = 35^\circ$, $c = 0$ kPa, and $\psi = 0^\circ$, following Lei et al. [66]. In accordance with their study, the constants for the Drucker-Prager yield surface that middle circumscribes the Mohr-Coulomb yield surface are adopted. The density is back analysed from the volume to match the total mass of the sand column with the experimentally measured value of 50 kg, leading to $\rho =$ 1111 kg/m^3. The numerical model is discretised with the initial particle spacing of dp = 0.005 m, leading to a total number of 0.63 million particles. The simulated physical time is 3 s. All five inclination angles tested in the experiment, i.e., 45°, 50°, 55°, 60°, and 65°, are simulated in this work.

Fig. 10 shows particles coloured according to the velocity magnitude at different time instants for the inclination angles of 45° and 55°. The simulation with a higher inclination angle demonstrates greater

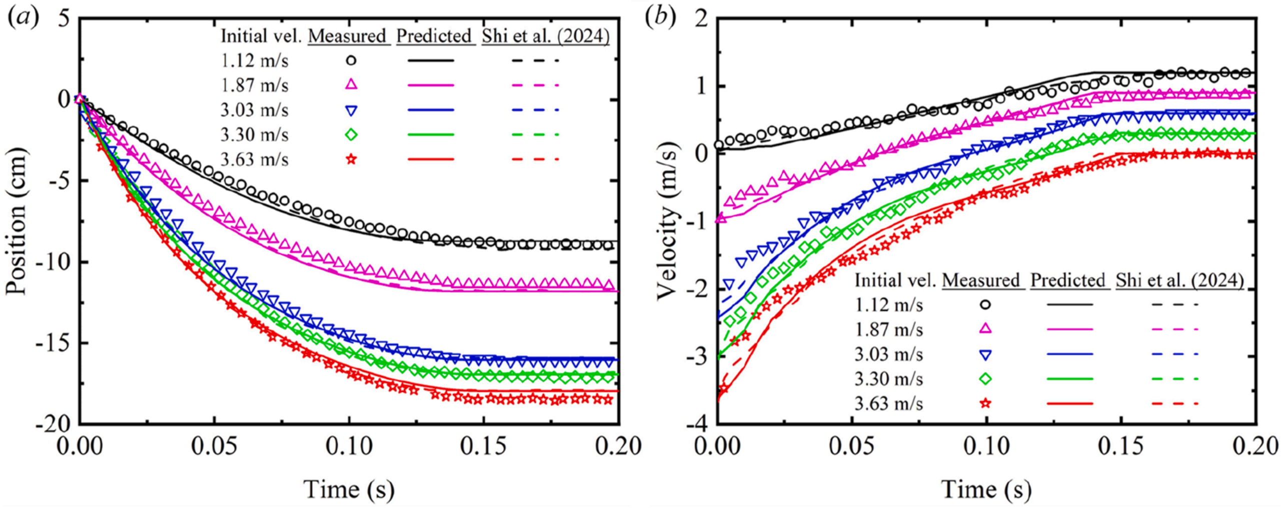


Fig. 16. Time evolution of (a) position and (b) velocity of the spherical solid (the y coordinate of each curve has been shifted by 0.3 sequentially for better clarity).

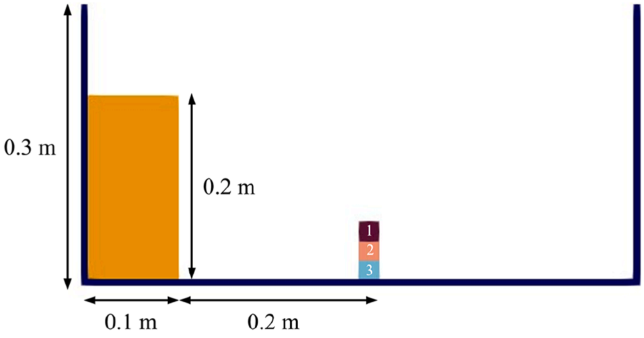


Fig. 17. Dynamic impact of granular flow on blocks: initial numerical configuration.

mobility of the sand flow. Upon the impact onto the rigid wall, the accumulation of sand at the bottom starts, forming a growing sand pile. The size of the final deposit deceases as the increase of inclination angles. The simulation results generally capture the experimental observation reported by Moriguchi et al. [65].

The forces per unit mass exerted on the rigid wall are evaluated by summing the forces on particles representing the wall using Eq. (32b). The force gauge option in the original solver is modified to achieve this purpose. It is noted that the forces obtained from simulation are the total force while the impact force measured in the experiment is the force normal to the rigid wall. Therefore, the normal vector $\boldsymbol{n}$ of the rigid wall is applied in the calculation of impact force in a similar manner to Chen et al. [7]. Fig. 11 displays the predicted impact force over time in simulation with comparison against experiment measurements. The prediction by the new solver shows a reasonable agreement with the experiment measurements by Moriguchi et al. [65].

5.4. Sliding motion of a cube on a slope

The sliding of a cube on a slope is simulated in this case to validate the developed solver in solving the rigid-rigid interaction with frictional constraints using Project Chrono. Fig. 12 shows the configuration of the numerical model. A solid cube is positioned on an inclined plane and slides down the slope due to the combined action of gravity force and frictional force. The cube has a size of 0.1 m × 0.1 m × 0.1 m, and the slope has a length of 2.5 m with the inclination of $\theta = 30^\circ$. The numerical model is discretised with initial particle spacing of dp = 0.005. The material properties of the cube and slope are both set as Young's modulus $E = 2$ GPa, Poisson ratio $\nu = 0.2$, and density $\rho = 2500$ kg/m^3. The friction coefficient of the slope is set to vary from $\mu = 0$, 0.1, 0.3, to 0.6.

Table 4
Material properties adopted in the simulation of granular flow impact on blocks.

Granular column	
Density (kg/m^3)	1300
Young's modulus (MPa)	0.5
Poisson ratio (-)	0.3
Cohesion (Pa)	0
Friction angle (°)	22
Dilation angle (°)	0
Wooden blocks	
Density (kg/m^3)	500
Young's modulus (MPa)	50
Poisson ratio (-)	0.5
Fictional coefficient (-)	0.6

The time-dependent analytical solution of the displacement of the sliding cube is $g(sin\theta - \mu cos\theta)t^2/2$. Fig. 13 presents the comparison between the analytical solution and numerical results for the displacement of the block for different values of μ. It can be seen that the simulation well captures the frictional contact of the rigid-rigid interaction.

5.5. Spherical solid impacting on a granular bed

In this case, the simulation of spherical solid impacting on a granular bed is performed for testing the effectiveness of GeoDualSPHysics in modelling the interaction of a single rigid solid with geomaterials. The experiment setup described in Pica Ciamarra et al. [67] is adopted for the numerical simulation. Fig. 14 illustrates the geometry of the numerical model. A spherical solid with a radius of 0.0223 m is placed just above the granular bed and released with varying initial velocities. The initial particle spacing is set as dp = 0.004 m, resulting in a total number of 0.44 million particles.

The material properties used in the simulation are taken from Shi et al. [2], taking values of $\rho = 600$ kg/m^3, $E = 1.3$ MPa, $\upsilon = 0.2$, $\varphi = 16^\circ$, $c = 0$ kPa, $\psi = 0^\circ$ for the granular bed and $\rho = 7850$ kg/m^3 for the rigid solid. Note that, similar to Shi et al. [2], the Young's modulus of granular beds is calibrated in this study by matching numerical results to the corresponding experiment measurement from the case with an initial velocity of 3.30 m/s.

Fig. 15 shows the numerical results for the initial velocity of 3.3 m/s at different time instants. It is noted that the predicted stress profile of

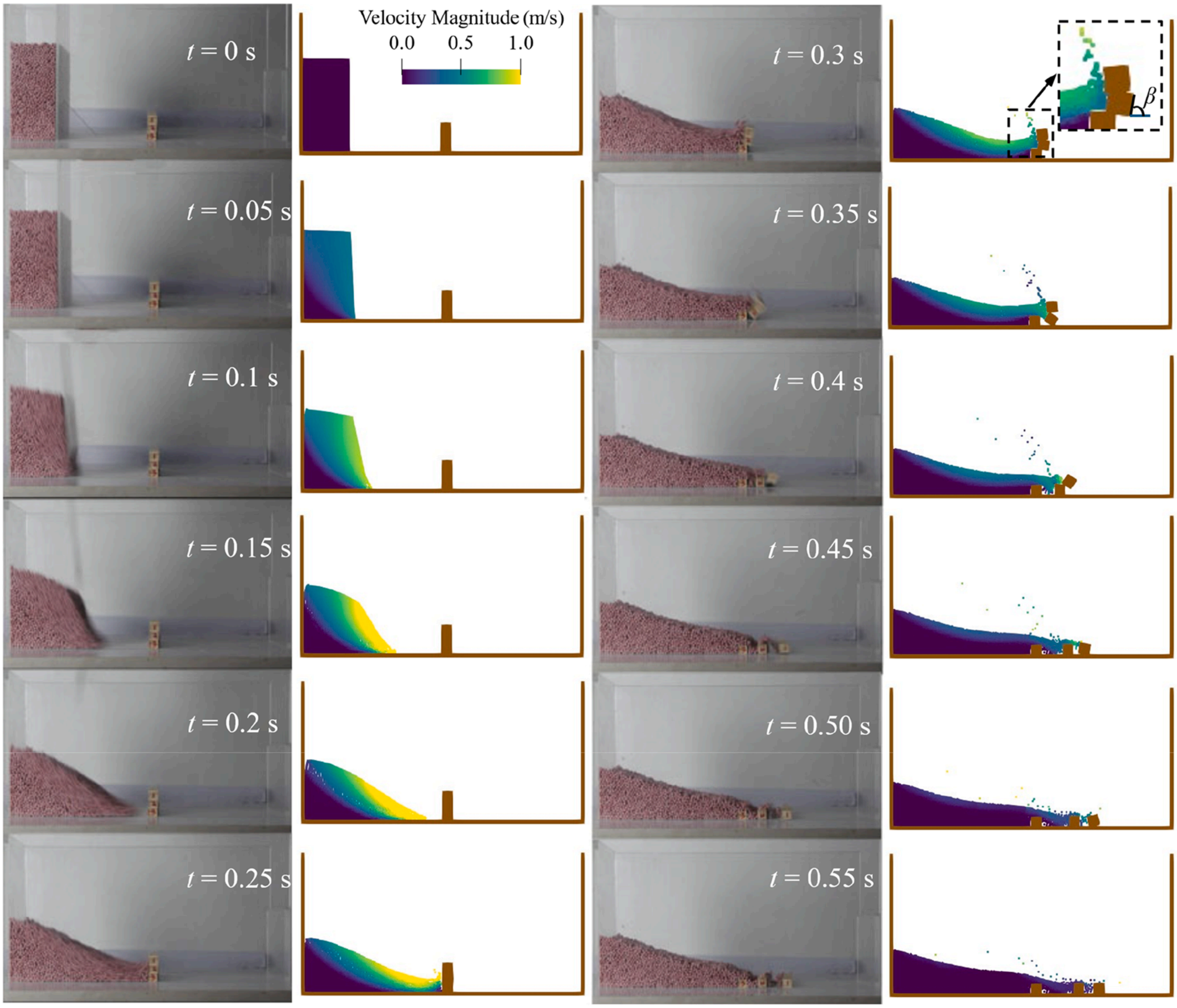


Fig. 18. Comparison between the experiment observation [68] and simulated impact process. (adapted from [68], with permission from © John Wiley & Sons, Ltd.).

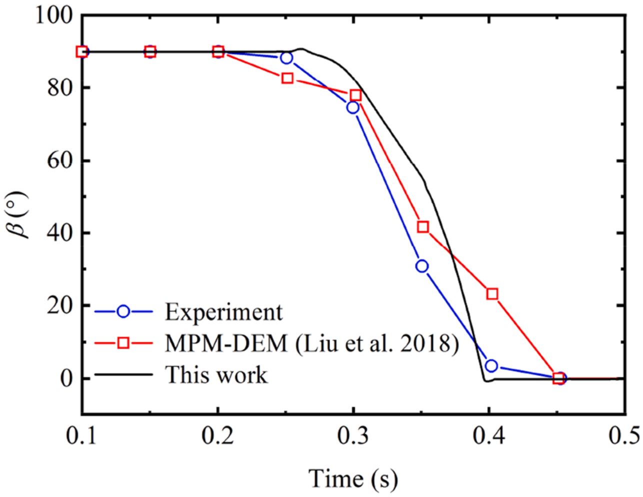


Fig. 19. Temporal evolution of the rotation angle β of the block no.2.

Table 5
Information of adopted devices.

GPU	NVIDIA GeForce RTX 4090	NVIDIA Tesla V100	CPU	Intel Xeon Gold 6248R
Compute capacity	8.9	7.0	Clock Speed	3.00 GHz
Global memory	24 GB	16 GB	Cores	24
CUDA cores	16384	5120	Threads	48
Clock rate	2.52 GHz	1.53 GHz	Cache	35.75 MB
Memory bandwidth	1008 GB/s	900 GB/s	RAM	256 GB
TFLOPS (Single precision)	82.6	15.7	Memory bandwidth	140.8 GB/s

the granular material remains smooth, even in this high-velocity impact scenario. Additionally, Fig. 16 presents a comparison with the position and velocity data reported by Pica Ciamarra et al. [67], as well as the MPM-DEM simulation results by Shi et al. [2]. The results show that the predictions from the developed solver match well with the experimental measurements as well as MPM-DEM results, indicating that the new solver can effectively capture the interaction between the discrete solid body and the granular continuum in large deformations.

Table 6
The resolutions for the performance analysis.

Particle spacing (m)	Material particles (million)	Boundary particles (million)	Total particles (million)
0.0008	24.7	3.7	28.4
0.001	2.36	12.64	15.00
0.00125	6.48	1.51	7.99
0.0015	3.79	1.00	4.79
0.002	1.59	0.59	2.18
0.0025	0.816	0.379	1.195
0.004	0.201	0.149	0.350
0.008	0.0267	0.0357	0.0624

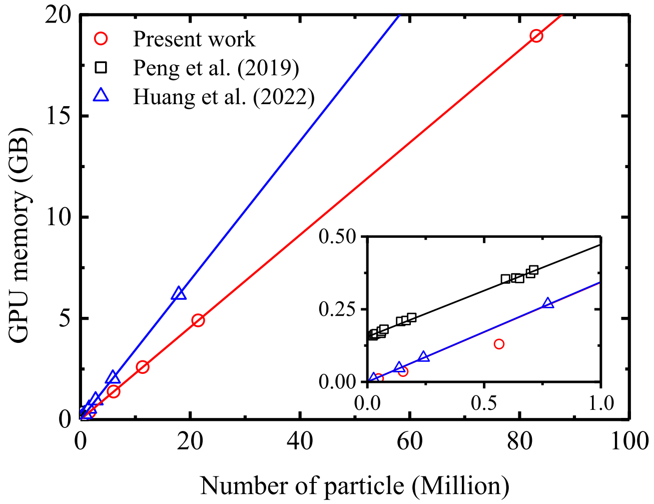


Fig. 20. Relationship between GPU memory usage and number of particles.

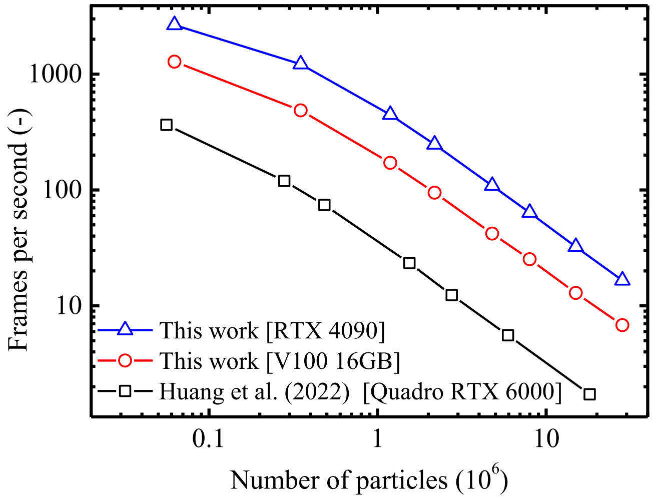


Fig. 21. Relationship between computational speed (FPS) and number of particles.

5.6. *Dynamic impact of granular flow on blocks*

This case refers to the simulation of experiment by Liu et al [68] on the impact of granular flow on blocks, to validate the developed solver in modelling the interaction of geomaterials with multiple rigid bodies. This benchmark example has been widely used to validate the interaction between granular flow and rigid multibody systems (e.g., [5,18, 69]). The 2-D setup is adopted to align with the numerical configuration and material properties reported in the literature.

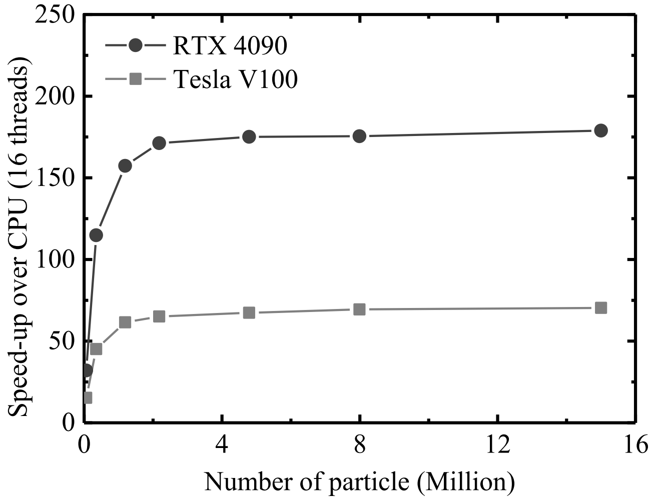


Fig. 22. Speed-up over CPU (16 threads) for 3-D granular column collapse test.

The setup of numerical model is shown in Fig. 17. The granular column has a width of 0.1 m and a height of 0.2 m. The block dimensions are 0.02 m × 0.018 m. The block no. 3 is fixed in space to keep consistent with the experiment setup while the blocks no.1 and no.2 are movable. Table 4 lists the material parameters adopted in the simulation, which are taken from the experiment data reported by Liu et al. [68]. The initial particle spacing is set to be 0.0025 m.

Fig. 18 presents the flow process of the granular column and its impact on the blocks. The snapshots from the experiment are also included for comparison. It can be seen that the impact occurs at $t = 0.25$ s and causes the translation and rotation of both block no.1 and no. 2. Block no.2 touches the ground and stops rotation at around $t = 0.4$ s, while the block no.3 fully contacts the ground at about $t = 0.45$ s. The physical process observed in the experiment is effectively reproduced in the simulation. Moreover, no unphysical gap or numerical instability arises between the granular flow and the multibody system, indicating a stable and consistent coupling.

In addition, the rotation angle β between the left boundary of block No. 2 and the x-axis is measured in simulation and compared with experiment data and MPM-DEM results by Liu et al. [68] in Fig. 19. The SPH results generally agree with the experiment data and MPM-DEM results, indicating the new solver can accurately capture the dynamic interaction between granular flow and multibody systems.

6. Performance analysis

The performance of the new solver on GPU is tested and reported in this section, with a focus on the memory usage, computational speed, speedup over a multi-core CPU, and profiling analysis of key CUDA kernels. The three-dimensional granular column collapse test, widely adopted in the literature for performance test, is selected as the benchmark case. The NVIDIA GeForce RTX 4090 is adopted to run GPU simulations, along with simulations on the NVIDIA Tesla V100 for comparison. The Intel Xeon Gold 6248R is adopted for multi-core CPU simulations. The details of adopted devices are listed in Table 5.

The granular circular column collapse test simulates the gravity-driven free fall of a cylindrical granular column. The initial dimensions of the column are set to radius $r_0 = 0.2$ m and height $h_0 = 0.1$ m, corresponding to an initial aspect ratio $\alpha = 0.5$. The material parameters used in the simulation are identical to those in Huang et al. [27], taking value of $\rho = 2600$ kg/m^3, $E = 15$ MPa, $\upsilon = 0.3$, $\varphi = 30^\circ$, $c = 0$ kPa. The initial particle spacings are adopted ranging from dp $= 0.008$ m to dp $= 0.0008$ m. The adopted resolutions and resulting particle numbers in the simulation are presented in Table 6. All the simulations are performed for the physical time of $t = 0.1$s.

Table 7
Profiling data for the main kernels in GeoDualSPHysics.

Performance metrics	KerMDBC	KerCDBC	KerInteract	KerSymPre	KerSymCorr	KerVerlet
Duration (ms)	0.251	0.104	3.24	0.458	0.429	0.471
Compute Throughput (%)	34.33	31.58	78.79	54.06	63.77	56.89
Memory Throughput (%)	15.93	15.58	35.67	92.75	88.52	91.88
TFLOP/s (Single-precision)	4.14	3.47	9.7	0.31	0.34	0.31
L2 Hit Rate (%)	66.56	62.43	91.37	52.86	56.16	47.87
No Eligible (%)	67.81	58.97	20.7	96.88	96.58	96.74
Average Active Threads Per Warp (%)	18.21	19.23	19.07	32	32	32
Registers Per Thread	115	96	64	40	40	48
Theoretical Occupancy (%)	33.33	41.67	66.67	100	100	83.33
Achieved Occupancy (%)	25.84	30.86	63.1	86.17	85.49	71.41

Table 8
Profiling data for the main kernels in DualSPHysics.

Performance metrics	KerMDBC	KerInteract	KerInteract(SPS)	KerSymPre	KerSymCorr	KerVerlet
Duration (ms)	0.23	2.37	3.37	0.236	0.416	0.276
Compute Throughput (%)	37.87	83.6	76.86	62.26	85.54	60.27
Memory Throughput (%)	8.34	33.97	34.48	92.88	66.46	92.62
TFLOP/s (Single-precision)	3.37	10.435	10.5	0.126	0.14	0.133
L2 Hit Rate (%)	60.23	87.08	88	51.32	40.67	44.5
No Eligible (%)	70.18	15.84	22.58	96.65	96.53	96.86
Average Active Threads Per Warp (%)	20.16	21.67	18.23	32	31.85	32
Registers Per Thread	96	48	64	33	29	39
Theoretical Occupancy (%)	41.67	83.33	66.67	100	100	100
Achieved Occupancy (%)	30.43	78.41	62.72	81.98	92.01	82.54

The relationship between GPU memory usage and number of total particles is presented in Fig. 20, with comparison against the reported results by the Peng et al. [25] using the open-source solver LOQUAT and results by Huang et al. [27] using their in-house solver modified from DualSPHysics (v.4.0). It can be seen from Fig. 20 that the three solvers all show a linear relation between memory usage and the particle number. The new solver developed in this work achieves substantially lower memory consumption than current state-of-the-art. For a simulation with 1 million particles, LOQUAT requires around 0.48 GB of GPU memory, the solver by Huang et al. [27] needs GPU memory allocation of about 0.35 GB, while GeoDualSPHysics requires only 0.23 GB. GeoDualSPHysics demonstrates 52% less memory usage than LOQUAT and 34% less than Huang et al.'s solver, demonstrating the potential of GeoDualSPHysics for the application with larger scale and finer resolutions.

The computational efficiency of GeoDualSPHysics is measured using the performance metric frames per second (FPS), defined as the number of computational steps executed per second. Fig. 21 compares the FPS achieved by GeoDualSPHysics on an NVIDIA GeForce RTX 4090 and an NVIDIA Tesla V100, alongside the reported FPS values from Huang et al. [27] on an NVIDIA Quadro RTX 6000 for the same test case. The FPS by GeoDualSPHysics on Tesla V100 shows nearly one order of magnitude higher than the reported FPS by Huang et al. [27] on Quadro RTX 6000. Although this comparison involves simulations on two different GPU devices, it is noted that the adopted Tesla V100 has a comparable FP32 performance with Quadro RTX 6000. Furthermore, the

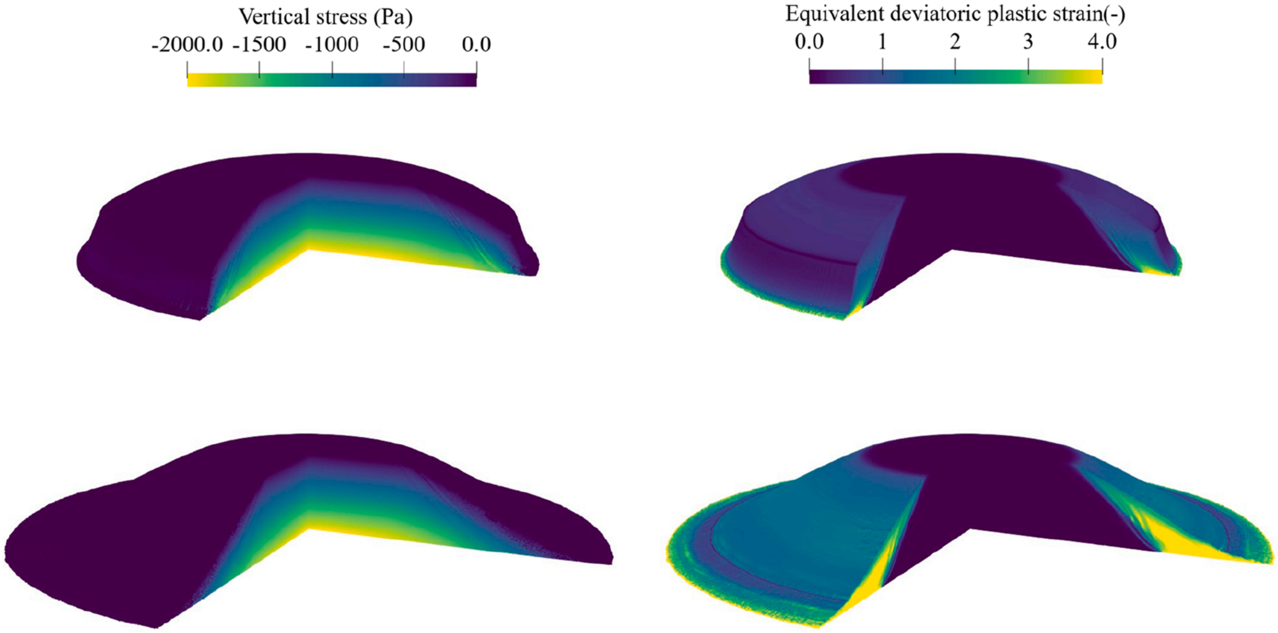


Fig. 23. Simulation results of the granular flow with 86.7 million particles.

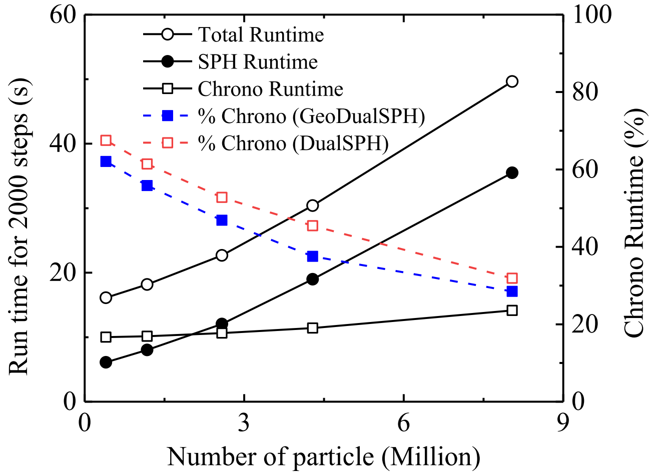


Fig. 24. Run time of GeoDualSPHysics and Project Chrono for different particle resolutions.

GeoDualSPHysics simulation on RTX 4090 achieves approximately 2.6 × higher FPS than that on the Tesla V100, indicating the performance of GeoDualSPHysics scales well with the advance in GPU devices. Specifically, for a 1.19 million particle simulation, GeoDualSPHysics achieves FPS rate of 171.1 on the Tesla V100 and 445.9 on the GeForce RTX 4090, while for the 15 million particle simulation, it reaches FPS of 12.9 on the Tesla V100 and FPS of 32.3 on the RTX 4090. These results represent a substantial enhancement over reported FPS in the literature for similar-scale simulations. These results demonstrate the advances of GeoDualSPHysics in both computational speed and memory efficiency.

Another evaluation is conducted to assess the GPU speedup of GeoDualSPHysics over the multi-core CPU. The OpenMP-based CPU implementation follows the same computational workflow as the GPU to make them comparable. For a practical simulation time of large particle number on the CPU, the simulations run for 2,000 computational steps at the particle number ranging from 0.06 to 15 million with a fixed timestep ($\Delta t = 3 \times 10^{-6}$ seconds) that satisfies the CFL condition at the finest resolution. The CPU simulation uses an Intel Xeon Gold 6248R with 16 threads allocated.

The GPU speedup over CPU (16 threads), calculated as the ratio of FPS achieved by GPU to FPS by CPU, is plotted in Fig. 22. The results show that the speedup increases rapidly with the number of particles for simulations with fewer than 2 million particles. Beyond this point, the speedup growth rate slows down as the particle count further increases. For the Tesla V100, the maximum speedup reaches around 70x, benchmarked against the Xeon Gold 6248R (using 16 threads), while the RTX 4090 achieves a maximum speedup of 180x. This finding highlights the advantage of GPU-based acceleration. It also implies that GPU acceleration in GeoDualSPHysics could outperform serial CPU computation by up to three orders of magnitude.

The specific performance metrics of the key CUDA kernels in GeoDualSPHysics are investigated using the GPU profiling tool, NVIDIA Nsight Compute. The simulation of granular column collapse with dp = 0.002 m on RTX 4090 at the first timestep is profiled. The CUDA kernels that take the main computation, including particle interaction, time integration, and boundary treatment, are analysed. Tables 7 and 8 respectively list the results of GPU profiling for main kernels in GeoDualSPHysics and DualSPHysics.

The profiling results show that CUDA kernels KerMDBC and KerInteract in GeoDualSPHysics achieve slightly lower occupancy compared to their counterparts in DualSPHysics. This reduction occurs because GeoDualSPHysics introduces additional arrays for stress tensor calculations, requiring more registers per thread for floating-point operations. To enable fair comparison, the DualSPHysics solver is tested with the resolution of sub-particle stress (SPS) option in fluid simulation, which similarly requires stress tensor computation. The comparison between KerInteract in GeoDualSPHysics and KerInteract(SPS) in DualSPHysics demonstrates nearly identical performance metrics. Although efforts have been made to retain the performance metrics of the CUDA kernels for time integration (i.e., KerSymPre, KerSymCorr, and KerVerlet), their execution times remain longer than those in the original DualSPHysics. This is expected, as the current implementation includes additional computations for elastic stress integration and return

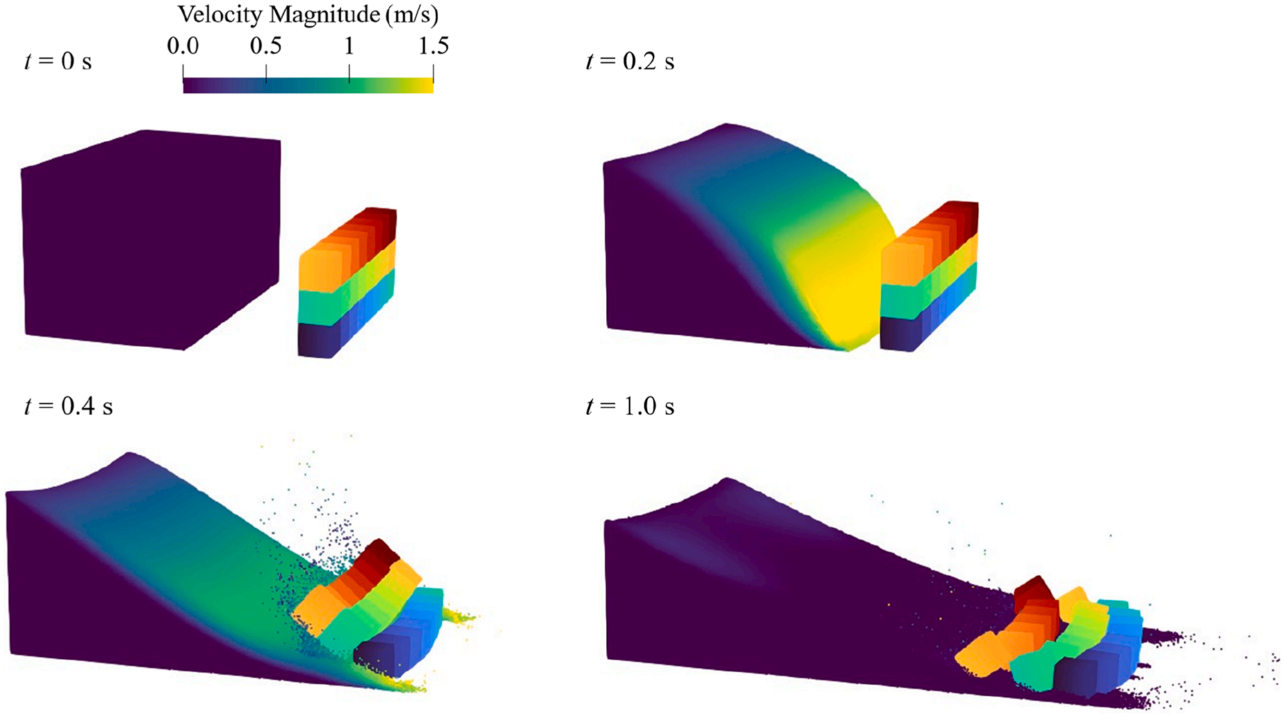


Fig. 25. Simulation results of the granular flow impacting a wall made of 24 cubic boxes.

mapping algorithm in the time integration. However, the proportion of the added cost to the overall simulation time is relatively marginal.

It is also found that the KerInteract for particle interaction accounts for the largest proportion of computation time. In comparison, the boundary treatment contributes a relatively minor computational cost: KerCDBC for cDBC requires only 3.2% of the KerInteract, while KerMDBC for mDBC accounts for 7.74%. This finding challenges the viewpoint that the boundary treatment with fixed ghost technique significantly increases computational costs.

The particle spacing has been further refined to dp $= 0.0005\ m$, with the height of granular column adjusted to 0.08 m. This configuration results in a simulation of total 86.7 million particles, including 80.7 million material particles and 6 million boundary particles, which has reached a new record for granular flow simulation ever performed on a single GPU. The simulation is conducted on the GeForce RTX 4090 GPU 24 GB, achieving FPS of 4.6 (e.g., completing 20,000 time steps in approximately 1.2 hours). The noise-free stress treatment described in section 2.5.2 is enabled in this case. Fig. 23 shows the results of the simulation with the finest resolution.

The results demonstrate that a stable, noise-free stress field has been achieved by using GeoDualSPHysics. The simulation at such a high resolution captures fine-scale details of multiple shear band formation and evolution. The capacity of simulating large numbers of particles at a practical computational speed also enables real-world applications in large scale such as landslide simulation. It is worth noting that the simulation of a finer resolution and larger number of particles (e.g., over 100 million particles) can be achieved in GeoDualSPHysics by using a GPU with greater memory capacity.

To further assess the performance of the coupling with Project Chrono, the case configuration of fluid impacting a breakwater by Martínez-Estévez et al. [35] for DualSPHysics-Chrono is adopted. This original case refers to the gravity-driven water column impacting on a block of 24 cubic boxes. In the present study, the water column is replaced with granular column with properties of $\rho = 2100$ kg/m^3, $E =$ 10 MPa, $\upsilon = 0.3$, $\varphi = 25^\circ$, $c = 0$ kPa, $\psi = 0^\circ$. The distance between the column and cubes is shortened to 0.2 m considering the limited run-out of the granular flow. The material properties of the cube box are set as soft wood. For the detailed numerical setup, readers can refer to Martínez-Estévez et al. [35] or the provided files in the attached source code.

The performance analysis was run for 2,000 computational steps at the initial spacings ranging from 0.006 m to 0.002 m with a fixed timestep ($\Delta t = 5 \times 10^{-6}$ seconds). Fig. 24 presents the total simulation runtime, the SPH runtime, the Chrono runtime, and the percentage of Chrono runtime relative to the total (% Chrono). Similar to the finding by Martínez-Estévez et al. [35], the increase of particle number directly raises the SPH runtime but has a minor influence on the Project Chrono runtime. GeoDualSPHysics shows a trend of % Chrono similar to that of DualSPHysics as particle resolutions are refined. The slightly higher % Chrono of original DualSPHysics is caused by the shorter SPH runtime for fluid simulations using DualSPHysics, as discussed in the profiling analysis.

Fig. 25 shows the full simulation results of the granular column impacting the box wall. The flow process of granular materials, impacts with rigid bodies, and multi-body collisions are captured in the simulation.

7. Conclusions

This work presents an open-source, high-performance SPH solver, GeoDualSPHysics, tailored for large deformation geomechanics modelling. The framework introduces four core developments: (1) stabilized, noise-free stress resolution for simulating geomaterials in extremely large deformations, (2) first-order consistent solid boundary modelling with two boundary treatment options: an extended modified Dynamic Boundary Condition (mDBC) and a corrected Dummy Boundary Condition (cDBC), (3) GPU-accelerated computation by leveraging the CUDA-parallelised SPH framework inherent in DualSPHysics, and (4) two-way coupling with multi-body system through integration with Project Chrono. The solver achieves practical computational speeds for systems involving tens of millions of particles, while maintaining cross-platform compatibility with both Windows and Linux environments.

Benchmarked against state-of-the-art geomechanics SPH solvers, GeoDualSPHysics demonstrates notable improvements in both memory efficiency and computational speed. The accuracy and performance of the new solver are validated through six benchmark tests spanning geotechnical failures, impact force evaluation on solid boundaries, and geomaterial-multibody system interactions. These capabilities position the solver as a versatile tool for potential applications in geohazard mitigation (e.g., landslides), geotechnical operation (e.g., soil tillage), and terramechanics (e.g., off-road vehicle mobility). The application of the developed solver to realistic engineering scenarios with complex geometries will be explored in future work. Further development will focus on improved accuracy through consistency corrections, more boundary condition options (e.g., confining stress boundary), multiscale modelling via variable-resolution schemes, coupled fluid-geotechnical-structural simulations (e.g., wave-seabed interactions), thermal-hydro-mechanical coupling, and enhanced parallelization to further broaden its engineering and scientific utility.

CRediT authorship contribution statement

Ruofeng Feng: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Jidong Zhao:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Georgios Fourtakas:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Benedict D Rogers:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

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

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Key Project #52439001), and the Research Grants Council of Hong Kong (GRF 16203123, 16208224, 16217225, CRF C7085-24G, RIF R6008-24, TRS T22-607/24N, and T22-606/23-R). The first author would like to acknowledge the use of the Computational Shared Facility at The University of Manchester.

Data availability

We have shared the code/data at the Attach File step.

References

[1] S. Zhao, J. Zhao, SudoDEM: Unleashing the predictive power of the discrete element method on simulation for non-spherical granular particles, Computer Physics Communications 259 (2021) 107670.

[2] Y.H. Shi, N. Guo, Z.X. Yang, GeoTaichi: A Taichi-powered high-performance numerical simulator for multiscale geophysical problems, Computer Physics Communications 301 (2024) 109219.

[3] A.P. Das, J. Zhao, T. Sweijen, Micromechanical modeling of triphasic granular media, Proceedings of the National Academy of Sciences of the United States of America 122 (18) (2025) e2420314122.

[4] A. Yerro, K. Soga, J. Bray, Runout evaluation of Oso landslide with the material point method, Canadian Geotechnical Journal 56 (9) (2019) 1304–1317.

[5] H. Chen, S. Zhao, J. Zhao, X. Zhou, DEM-enriched contact approach for material point method, Computer Methods in Applied Mechanics and Engineering 404 (2023) 115814.

[6] J. Yu, J. Zhao, W. Liang, S. Zhao, A semi-implicit material point method for coupled thermo-hydro-mechanical simulation of saturated porous media in large deformation, Computer Methods in Applied Mechanics and Engineering 418 (2024) 116462.

[7] J.Y. Chen, F.S. Lien, C. Peng, E. Yee, GPU-accelerated smoothed particle hydrodynamics modeling of granular flow, Powder Technology 359 (2020) 94–106.

[8] J.Y. Chen, D.L. Feng, F.S. Lien, E. Yee, S.X. Deng, F. Gao, C. Peng, Numerical modelling of interaction between aluminium structure and explosion in soil, Applied Mathematical Modelling 99 (2021) 760–784.

[9] H.H. Bui, G.D. Nguyen, Smoothed particle hydrodynamics (SPH) and its applications in geomechanics: From solid fracture to granular behaviour and multiphase flows in porous media, Computers and Geotechnics 138 (2021) 104315.

[10] R. Feng, J. Zhao, J. Chen, A 3D multiphase SPH framework for modelling soil-water interaction in rainfall-landslide-tsunami cascades, Eng. Geo. (2026) 108495, https://doi.org/10.1016/j.enggeo.2025.108495.

[11] L. Monforte, P. Navas, J.M. Carbonell, M. Arroyo, A. Gens, Low-order stabilized finite element for the full Biot formulation in soil mechanics at finite strain, International Journal for Numerical and Analytical Methods in Geomechanics 43 (7) (2019) 1488–1515.

[12] Y.F. Jin, Z.Y. Yin, Two-phase PFEM with stable nodal integration for large deformation hydromechanical coupled geotechnical problems, Computer Methods in Applied Mechanics and Engineering 392 (2022) 114660.

[13] T. Rabczuk, H. Ren, A peridynamics formulation for quasi-static fracture and contact in rock, Engineering Geology 225 (2017) 42–48.

[14] C. Yang, F. Zhu, J. Zhao, A multi-horizon fully coupled thermo-mechanical peridynamics, Journal of the Mechanics and Physics of Solids 191 (2024) 105758.

[15] C. Yang, J. Zhao, F. Zhu, R. Feng, A multi-horizon peridynamics for coupled fluid flow and heat transfer, Journal of Fluid Mechanics 1010 (2025) A66.

[16] S. Zhang, C. Zhang, X. Hu, S.D. Lourenço, A Riemann-based SPH method for modelling large deformation of granular materials, Computers and Geotechnics 167 (2024) 106052.

[17] S. Zhang, F. Wang, X. Hu, S.D. Lourenço, A unified transport-velocity formulation for SPH simulation of cohesive granular materials, Computers and Geotechnics 181 (2025) 107139.

[18] S. Zhang, D. Wu, X. Hu, C.E. Choi, S.D. Lourenço, An efficient SPH framework for modeling binary granular mixtures and implications for granular flows, International Journal for Numerical and Analytical Methods in Geomechanics 49 (3) (2025) 815–838.

[19] T.N. Hoang, H.H. Bui, T.T. Nguyen, T.V. Nguyen, G.D. Nguyen, Development of free-field and compliant base SPH boundary conditions for large deformation seismic response analysis of geomechanics problems, Computer Methods in Applied Mechanics and Engineering 432 (2024) 117370.

[20] R. Feng, G. Fourtakas, B.D. Rogers, D. Lombardi, Modelling internal erosion using 2D smoothed particle hydrodynamics (SPH), Journal of Hydrology 639 (2024) 131558.

[21] R. Feng, G. Fourtakas, B.D. Rogers, D. Lombardi, A general smoothed particle hydrodynamics (SPH) formulation for coupled liquid flow and solid deformation in porous media, Computer Methods in Applied Mechanics and Engineering 419 (2024) 116581.

[22] H. Ikari, H. Gotoh, Numerical simulation of the collapse of a bidispersed granular column using DEM and elastoplastic SPH, Computational Particle Mechanics (2025) 1–12.

[23] Y. Wang, W. Wu, A SPH Model Bridging Solid-and Fluid-Like Behaviour in Granular Materials, International Journal for Numerical and Analytical Methods in Geomechanics 49 (2) (2025) 738–755.

[24] R. Vacondio, C. Altomare, M. De Leffe, X. Hu, D. Le Touzé, S. Lind, J.C. Marongiu, S. Marrone, B.D. Rogers, A. Souto-Iglesias, Grand challenges for smoothed particle hydrodynamics numerical schemes, Computational Particle Mechanics 8 (2021) 575–588.

[25] C. Peng, S. Wang, W. Wu, H.S. Yu, C. Wang, J.Y. Chen, LOQUAT: an open-source GPU-accelerated SPH solver for geotechnical modeling, Acta Geotechnica 14 (2019) 1269–1287.

[26] E. Yang, H.H. Bui, H. De Sterck, G.D. Nguyen, A. Bouazza, A scalable parallel computing SPH framework for predictions of geophysical granular flows, Computers and Geotechnics 121 (2020) 103474.

[27] C. Huang, Y. Sun, Y. An, C. Shi, C. Feng, Q. Liu, X. Yang, X. Wang, Three-dimensional simulations of large-scale long run-out landslides with a GPU-accelerated elasto-plastic SPH model, Engineering Analysis with Boundary Elements 145 (2022) 132–148.

[28] M. Gholami Korzani, S.A. Galindo-Torres, A. Scheuermann, D.J. Williams, SPH approach for simulating hydro-mechanical processes with large deformations and variable permeabilities, Acta Geotechnica 13 (2018) 303–316.

[29] A. Amicarelli, S. Manenti, R. Albano, G. Agate, M. Paggi, L. Longoni, D. Mirauda, L. Ziane, G. Viccione, S. Todeschini, A. Sole, SPHERA v. 9.0. 0: A Computational Fluid Dynamics research code, based on the Smoothed Particle Hydrodynamics mesh-less method, Computer Physics Communications 250 (2020) 107157.

[30] H.H. Bui, T. Viet, N.G.D. Nguyen, GeoXPM: Particle-based continuum solutions for extreme geoengineering, geomechanics & geophysics applications, Tutorial Manual (2023).

[31] R. Feng, G. Fourtakas, B.D. Rogers, D. Lombardi, Large deformation analysis of granular materials with stabilized and noise-free stress treatment in smoothed particle hydrodynamics (SPH), Computers and Geotechnics 138 (2021) 104356.

[32] J.M. Domínguez, G. Fourtakas, C. Altomare, R.B. Canelas, A. Tafuni, O. García-Feal, I. Martínez-Estévez, A. Mokos, R. Vacondio, A.J. Crespo, B.D. Rogers, DualSPHysics: from fluid dynamics to multiphysics problems, Computational Particle Mechanics 9 (5) (2022) 867–895.

[33] J. O'Connor, B.D. Rogers, A fluid–structure interaction model for free-surface flows and flexible structures using smoothed particle hydrodynamics on a GPU, Journal of Fluids and Structures 104 (2021) 103312.

[34] S. Long, X. Fan, C. Li, Y. Liu, S. Fan, X.W. Guo, C. Yang, VecDualSPHysics: A vectorized implementation of Smoothed Particle Hydrodynamics method for simulating fluid flows on multi-core processors, Journal of Computational Physics 463 (2022) 111234.

[35] I. Martínez-Estévez, J.M. Domínguez, B. Tagliafierro, R.B. Canelas, O. García-Feal, A.J. Crespo, M. Gómez-Gesteira, Coupling of an SPH-based solver with a multiphysics library, Computer Physics Communications 283 (2023) 108581.

[36] C. Cen, G. Fourtakas, S. Lind, B.D. Rogers, A single-phase GPU-accelerated surface tension model using SPH, Computer Physics Communications 295 (2024) 109012.

[37] Y. Zhan, M. Luo, A. Khayyer, DualSPHysics+: An enhanced DualSPHysics with improvements in accuracy, energy conservation and resolution of the continuity equation, Computer Physics Communications 306 (2025) 109389.

[38] F. Ricci, R. Vacondio, J.M. Domínguez, A. Tafuni, Three-dimensional variable resolution for multi-scale modeling in Smoothed Particle Hydrodynamics, Computer Physics Communications (2025) 109609.

[39] D. Violeau, B.D. Rogers, Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future, Journal of Hydraulic Research 54 (1) (2016) 1–26.

[40] X. Zhou, D. Lu, Y. Zhang, X. Du, T. Rabczuk, An open-source unconstrained stress updating algorithm for the modified Cam-clay model, Computer Methods in Applied Mechanics and Engineering 390 (2022) 114356.

[41] J.J. Monaghan, Smoothed particle hydrodynamics, Annual review of astronomy and astrophysics 30 (1992) 543–574. *Vol. 30 (A93-25826 09-90), p. 543-574*.

[42] M. Antuono, A. Colagrossi, S. Marrone, Numerical diffusive terms in weakly-compressible SPH schemes, Computer Physics Communications 183 (12) (2012) 2570–2580.

[43] D. Molteni, A. Colagrossi, A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH, Computer Physics Communications 180 (6) (2009) 861–872.

[44] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, Journal of Computational Physics 136 (1) (1997) 214–226.

[45] R. Feng, G. Fourtakas, B.D. Rogers, D. Lombardi, Two-phase fully-coupled smoothed particle hydrodynamics (SPH) model for unsaturated soils and its application to rainfall-induced slope collapse, Computers and Geotechnics 151 (2022) 104964.

[46] A. Khayyer, Y. Shimizu, C.H. Lee, A. Gil, H. Gotoh, J. Bonet, An improved updated Lagrangian SPH method for structural modelling, Computational Particle Mechanics 11 (3) (2024) 1055–1086.

[47] S.J. Lind, R. Xu, P.K. Stansby, B.D. Rogers, Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves, Journal of Computational Physics 231 (4) (2012) 1499–1523.

[48] A. Skillen, S. Lind, P.K. Stansby, B.D. Rogers, Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body–water slam and efficient wave–body interaction, Computer Methods in Applied Mechanics and Engineering 265 (2013) 163–173.

[49] A. Khayyer, H. Gotoh, Y. Shimizu, Comparative study on accuracy and conservation properties of two particle regularization schemes and proposal of an optimized particle shifting scheme in ISPH context, Journal of Computational Physics 332 (2017) 236–256.

[50] M. Antuono, P.N. Sun, S. Marrone, A. Colagrossi, The δ-ALE-SPH model: An arbitrary Lagrangian-Eulerian framework for the δ-SPH model with particle shifting technique, Computers & Fluids 216 (2021) 104806.

[51] Y. Shimizu, H. Gotoh, Volume-conserved wavy interface boundary for δ-SPH-based numerical wave flume, Coastal Engineering Journal 67 (2) (2025) 214–231.

[52] T. Gotoh, D. Sakoda, A. Khayyer, C.H. Lee, A. Gil, H. Gotoh, J. Bonet, An enhanced total Lagrangian SPH for non-linear and finite strain elastic structural dynamics, Computational Mechanics (2025) 1–33.

[53] D. Wu, X. Tang, S. Zhang, X. Hu, Unified non-hourglass formulation for total Lagrangian SPH solid dynamics, Computational Mechanics 75 (3) (2025) 1081–1113.

[54] S. Zhang, D. Wu, S.D. Lourenço, X. Hu, A generalized non-hourglass updated Lagrangian formulation for SPH solid dynamics, Computer Methods in Applied Mechanics and Engineering 440 (2025) 117948.

[55] A. English, J.M. Domínguez, R. Vacondio, A.J.C. Crespo, P.K. Stansby, S.J. Lind, L. Chiapponi, M. Gómez-Gesteira, Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): Application to tank sloshing, dam break and fish pass problems, Computational Particle Mechanics 9 (5) (2022) 1–15.

[56] M.B. Liu, G.R. Liu, Restoring particle consistency in smoothed particle hydrodynamics, Applied Numerical Mathematics 56 (1) (2006) 19–36.

[57] J.J. Monaghan, A. Kos, N. Issa, Fluid motion generated by impact, Journal of Waterway, Port, Coastal, and Ocean engineering 129 (6) (2003) 250–259.

[58] Alessandro Tasora, Radu Serban, Hammad Mazhar, Arman Pazouki, Daniel Melanz, Jonathan Fleischmann, Michael Taylor, Hiroyuki Sugiyama, Dan Negrut, Chrono: An open source multi-physics dynamics engine, in: High Performance Computing in Science and Engineering: Second International Conference, HPCSE 2015, Soláň, Czech Republic, May 25-28, 2015, Revised Selected Papers 2, Springer International Publishing, 2016, pp. 19–49.

[59] A. Tasora, M. Anitescu, A convex complementarity approach for simulating large granular flows, Journal of Computational and Nonlinear Dynamics 5 (3) (2010) 1–10.

[60] W. Hu, M. Rakhsha, L. Yang, K. Kamrin, D. Negrut, Modeling granular material dynamics and its two-way coupling with moving solid bodies using a continuum representation and the SPH method, Computer Methods in Applied Mechanics and Engineering 385 (2021) 114022.

[61] A.C. Crespo, J.M. Dominguez, A. Barreiro, M. Gómez-Gesteira, B.D. Rogers, GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods, PloS one 6 (6) (2011) e20685.

[62] H.H. Bui, R. Fukagawa, An improved SPH method for saturated soils and its application to investigate the mechanisms of embankment failure: Case of hydrostatic pore-water pressure, International Journal for numerical and analytical methods in geomechanics 37 (1) (2013) 31–50.

[63] Z. Liu, H.A. Koyi, J.O. Swantesson, F. Nilfouroushan, Y. Reshetyuk, Kinematics and 3-D internal deformation of granular slopes: analogue models and natural landslides, Journal of Structural Geology 53 (2013) 27–42.

[64] M. Gomez-Gesteira, B.D. Rogers, R.A. Dalrymple, A.J. Crespo, State-of-the-art of classical SPH for free-surface flows, Journal of Hydraulic Research 48 (sup1) (2010) 6–27.

[65] S. Moriguchi, R.I. Borja, A. Yashima, K. Sawada, Estimating the impact force generated by granular flow on a rigid obstruction, Acta Geotechnica 4 (2009) 57–71.

[66] Z. Lei, B. Wu, S. Wu, Y. Nie, S. Cheng, C. Zhang, A material point-finite element (MPM-FEM) model for simulating three-dimensional soil-structure interactions with the hybrid contact method, Computers and Geotechnics 152 (2022) 105009.

[67] M. Pica Ciamarra, A.H. Lara, A.T. Lee, D.I. Goldman, F.I. Vishik, H.L. Swinney, Dynamics of drag and force distributions for projectile impact in a granular medium, Physical Review Letters 92 (19) (2004) 194301.

[68] C. Liu, Q. Sun, G.G. Zhou, Coupling of material point method and discrete element method for granular flows impacting simulations, International Journal for Numerical Methods in Engineering 115 (2) (2018) 172–188.

[69] Y. Jiang, M. Li, C. Jiang, F. Alonso-Marroquin, A hybrid material-point spheropolygon-element method for solid and granular material interaction, International Journal for Numerical Methods in Engineering 121 (14) (2020) 3021–3047.