An integrated DEM code for tracing the entire regolith mass movement on asteroids

Zhijun Song, Yang Yu, Stefania Soldini, Bin Cheng, Patrick Michel

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **532,** 1307–1329 (2024) https://doi.org/10.1093/mnras/stae1537
Advance Access publication 2024 June 22

An integrated DEM code for tracing the entire regolith mass movement on asteroids

Zhijun Song,[1★] Yang Yu,[1★] Stefania Soldini,[2] Bin Cheng[3] and Patrick Michel[4,5]

[1]*School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China*
[2]*Department of Mechanical, Materials and Aerospace Engineering, University of Liverpool, Liverpool L69 3BX, UK*
[3]*School of Aerospace, Tsinghua University, Beijing 100086, China*
[4]*Université Côte dAzur, Observatoire de la Côte dAzur, CNRS, Laboratoire Lagrange, Nice F-06304, France*
[5]*Department of Systems Innovation, School of Engineering, The University of Tokyo, Tokyo 113-0033, Japan*

Accepted 2024 June 14. Received 2024 June 1; in original form 2024 February 10

ABSTRACT

This paper presents a general strategy for tracking the scale-span movement process of asteroid regolith materials. It achieves the tracking of the mass movement on the asteroid at a realistic scale, under conditions of high-resolution asteroid surface topography (submeter level) and actual regolith particle sizes. To overcome the memory exponential expansion caused by the enlarged computational domain, we improved the conventional cell-linked list method so that it can be applied to arbitrarily large computational domains around asteroids. An efficient contact detection algorithm for particles and polyhedral shape models of asteroids is presented, which avoids traversing all surface triangles and thus allows us to model high-resolution surface topography. A parallel algorithm based on Compute Unified Device Architecture for the gravitational field of the asteroid is presented. Leveraging heterogeneous computing features, further architectural optimization overlaps computations of the long-range and short-range interactions, resulting in an approaching doubling of computational efficiency compared to the code lacking architectural optimizations. Using the above strategy, a specific high-fidelity discrete element method code that integrates key mechanical models, including the irregular gravitational field, the interparticle and particle-surface interactions, and the coupled dynamics between the particles and the asteroid, is developed to track the asteroid regolith mass movement. As tests, we simulated the landslide of a sand pile on the asteroid's surface during spin-up. The simulation results demonstrate that the code can track the mass movement of the regolith particles on the surface of the asteroid from local landslides to mass leakage with good accuracy.

Key words: methods: numerical – minor planets, asteroids: general.

1 INTRODUCTION

Regolith composed of grained material of different sizes is widely present on the surface of asteroids (Fujiwara et al. 2006; DellaGiustina et al. 2019; Tachibana et al. 2022). Space missions showed evidence of regolith material activities (Lauretta et al. 2019; Jawin et al. 2020; Cho et al. 2021), driven by various physical processes, such as the unstable high-spin rate induced by the Yarkovsky-O'Keefe-Radzievskii-Paddack (YORP) effect (Kaasalainen et al. 2007; Lowry et al. 2014), the meteorite bombardment (Bottke et al. 2020), and close flybys of planets (Benson et al. 2023), etc. Such activities can lead to a change in the mechanical environment of the asteroid's surface and cause global redistribution of the regolith. Local events like landslides (Jawin et al. 2020) or mass leaking (Scheeres 2015) can also be triggered by slight disturbances, which are more frequent than global events that cause global redistribution of the regolith during the life cycle of an asteroid. These surface mass movement processes can significantly impact the asteroid's topography and contribute to the formation of asteroid systems holding clues to their evolutionary history.

The movement of asteroid regolith materials is dominated by a weak but complex gravitational field and is highly coupled with the asteroid's surface topography. The mechanical environments are very different from that on Earth. Creating such a complex experimental environment in the laboratory to replicate the mass migration behaviour of the real regolith material is very challenging due to technical limitations. An alternative approach to trace the regolith movement is using numerical tools to mimic the asteroid surface environment. Since the regolith material is essentially a discrete particle system (Morbidelli & Raymond 2016), the discrete element method (DEM) is the natural choice for investigating this difficult problem.

In recent years, various soft-sphere DEM codes have yielded insights into the geological evolution of asteroids. Specifically, Sánchez & Scheeres (2011) derived an *N*-body self-gravity algorithm, combined it with soft-sphere DEM, and applied the code to the realm of planetary science to simulate asteroid regolith and rubble piles. Schwartz, Richardson & Michel (2012) combined the DEM algorithms into a parallel *N*-body gravity tree code framework,

★ E-mail: junzhisong@buaa.edu.cn (ZS); yuyang.thu@gmail.com (YY)

pkdgrav, and applied the code to multiple problems in planetary science. The results contributed to understanding asteroid regolith movements (Yu et al. 2014; DeMartini et al. 2019; Kim et al. 2023), the formation of multibody systems (Zhang et al. 2018; Agrusa et al. 2024), and the internal structure of asteroids (Zhang et al. 2021). Tancredi et al. (2012) developed a new contact force model within the framework of the open-source DEM code ESYS-Particle and used the code to study the response of regolith materials on the surface of asteroids to vibration. Cheng, Yu & Baoyin (2019) have also developed a soft-sphere DEM code DEMBODY for large-scale planetary computing, which was applied to study the reshaping process of top-shaped asteroids (Cheng et al. 2020) and the formation of groove-like structures on Phobos (Cheng et al. 2022). In addition to the soft-sphere DEM code for spherical particles, discrete element codes capable of handling non-spherical particles have also started to be applied to the realm of planetary science in recent years. Sánchez et al. (2021) developed a contact dynamics code for non-spherical particles in the open-source software LMGG90 and applied it to the study of asteroids. Wen & Zeng (2023) proposed a Compute Unified Device Architecture (CUDA) based implementation of a non-spherical discrete element for problems in planetary science. Marohnic et al. (2023) presented an implementation of non-spherical discrete elements within the framework of the existing *N*-body code, PKDGRAV, utilizing aggregates composed of bonded spheres to model non-spherical components and applied this code to simulate the YORP spin-up, tidal disruption, and the Brazil nut effect. Ferrari, Lavagna & Blazquez (2020) presented the numerical implementation of a granular *N*-body simulation code building upon the work by Ferrari et al. (2017) and applied it to identify the breakup limit and reshape region for spinning rubble-pile aggregates.

Unlike on Earth, the movement of regolith materials on an asteroid's surface might extend to a large spatial scale that is even comparable to the size of the asteroid. Particles involved in a landslide may lift off the surface and enter orbits around the asteroid under certain circumstances, causing a mass-shedding event (Scheeres 2015; Cheng et al. 2020; Sánchez & Scheeres 2020; Banik, Gaurav & Sharma 2022). Modelling and tracing such mass migration processes that involve both surface motion and orbital motion of regolith grains raise technical challenges to DEM methods in the following aspects: (i) the modelling of high-resolution surface topography of asteroids and the complex particle-surface contact force calculation; (ii) the precise computation of irregular surface gravity fields for asteroids; (iii) the interactive dynamics between the regolith particles and the asteroid; (iv) the scale-span simulation of the regolith migration, i.e. the spatial scale of the computational domain changes continuously from tens of times the particle size at the beginning of local landslides to tens of kilometers after the particles enter the orbital motion. Conventional solutions to these problems may have defects, especially in the modelling of high-resolution surface topography of asteroids and scale-span simulations of particle migration in regolith. For example, conventional cell-linked list algorithm (Yao et al. 2004; Allen & Tildesley 2017) based on uniform grid decomposition requires that the grid division region encompass the entire computational domain, resulting in exponential memory requirements as the computational domain size increases (Cai et al. 2018; Zhou, Xu & Liu 2021). Therefore, when regolith particles enter orbit, the computational domain rapidly expands as their orbits evolve (McMahon et al. 2020; Song et al. 2023), rendering the memory required for calculations prohibitive for current computers. In addition, creating a high-resolution shape model of asteroids with spherical particles is computationally intensive for computing interparticle and particle-surface interactions. Due to these limitations, current existing DEM codes are not capable of tracking the entire migration process of regolith materials on asteroid surfaces with sufficient accuracy and resolution.

This study aims at a high-fidelity methodology that enables scale-adaptive simulation of a landslide that extends to global mass migration on and over the asteroid surface. An advanced soft-sphere DEM code is developed based on the CPU–GPU (CPU, Central Processing Unit; GPU, Graphics Processing Unit) hybrid architecture. The code enables actual-size modelling of the regolith grains and a simulation of the regolith migration process from the surface of an asteroid with high-resolution topography to its vicinity. The asteroid is considered a homogeneous polyhedron in this method to reduce the computing consumption of global DEM modelling. The polyhedral model enables a high-resolution representation of the surface topography with an accurate gravitational field induced by the irregular shape. An efficient contact detection algorithm between a sphere and the asteroid's surface was proposed to calculate the contact force, which applies to high-precision surface models because the computational complexity is independent of the fineness of the triangular mesh. We noticed for a cross-scale simulation, the conventional cell-linked list usually causes a memory explosion. A novel technique is developed to improve the conventional cell-linked list algorithm (Allen & Tildesley 2017), which avoids the deficiency of exponential growth of the memory consumption and enables the code to simulate an arbitrarily large computational domain without a significant increase in computational load. The code adopts the CUDA language stream to speed up the gravitational calculation of the polyhedral method (Werner & Scheeres 1997). It implemented efficient parallel computation of the gravity from the asteroid, the interparticle gravity, and the coupled dynamics of the asteroid-particles system. Using the CPU–GPU architecture, we optimized the computational efficiency by overlapping the computations of the gravitational field of the asteroid and particle interactions, which improved the code performance significantly.

The rest of this paper is organized as follows: In section 2, we present the algorithm and implementation of our code in detail, covering the calculation of diverse forces acting on the particles and the techniques to improve the computational efficiency. In section 3, benchmarking tests are performed to illustrate the accuracy of our code and test its capability to model the mass migration process of the regolith of asteroids. Finally, in Section 4, we summarize this study and give a discussion on the potential applications of the code.

2 METHOD

2.1 Soft-sphere discrete element method

The discrete element method tracks the mechanical behaviour of granular materials by solving the equation of motion for each particle, which gives the qualitative results of the mechanical response of the material. A core function of the DEM is to accurately calculate the external force exerted on a single particle. The external forces can be categorized into short-range (e.g. contact force and cohesive force) and long-range (e.g. gravity) interactions (Zhang et al. 2018).

The linear spring-damping model based on a Hooke-type relation (hereafter referred to as the linear model) and the non-linear spring-damping model based on the Hertz theory for the normal direction and the Mindlin–Deresiewicz contact theory for the tangential direction (hereafter referred to as the non-linear model) are widely employed to calculate short-range contact forces between particles in DEM. The linear model, being simple and efficient, is easily implemented in code. However, its contact model parameters

Table 1. Intermediate variables used to calculate the contact forces (Zhang et al. 2017).

Variable	Symbols	Equation
Contact normal direction	$\boldsymbol{n}$	$(\boldsymbol{r}_j - \boldsymbol{r}_i)/\lvert\boldsymbol{r}_j - \boldsymbol{r}_i\rvert$
Normal overlap	δ_{N}	$R_i + R_j - \lvert\boldsymbol{r}_i - \boldsymbol{r}_j\rvert$
Tangential sliding displacement since time t_0	$\boldsymbol{\delta}_{\mathrm{S}}$	$\int_{t_0}^{t} \boldsymbol{u}_{\mathrm{S}}(\tau)\mathrm{d}\tau + \boldsymbol{\delta}_{\mathrm{S0}}$
Normal relative velocity	$\boldsymbol{u}_{\mathrm{N}}$	$[(\boldsymbol{v}_j - \boldsymbol{v}_i)\cdot\boldsymbol{n}]\boldsymbol{n}$
Tangential relative velocity	$\boldsymbol{u}_{\mathrm{S}}$	$\boldsymbol{v}_j - \boldsymbol{v}_i - [(\boldsymbol{v}_j - \boldsymbol{v}_i)\cdot\boldsymbol{n}]\boldsymbol{n} + l_i(\boldsymbol{n}\times\boldsymbol{\omega}_i) + l_j(\boldsymbol{n}\times\boldsymbol{\omega}_j)$
Contact lever arms of particle i	l_i	$(R_i{}^2 - R_j{}^2 + \lvert\boldsymbol{r}_i - \boldsymbol{r}_j\rvert^2)/(2\lvert\boldsymbol{r}_i - \boldsymbol{r}_j\rvert)$
Contact lever arms of particle j	l_j	$\lvert\boldsymbol{r}_i - \boldsymbol{r}_j\rvert - l_i$
Twisting relative angle velocity	$\boldsymbol{\omega}_{\mathrm{T}}$	$[(\boldsymbol{\omega}_j - \boldsymbol{\omega}_i)\cdot\boldsymbol{n}]\boldsymbol{n}$
Rolling relative angle velocity	$\boldsymbol{\omega}_{\mathrm{R}}$	$[(l_i + l_j)/(l_i l_j)]\boldsymbol{n}\times\boldsymbol{u}_{\mathrm{R}}$
Rolling relative velocity	$\boldsymbol{u}_{\mathrm{R}}$	$[(l_i l_j)/(l_i + l_j)](\boldsymbol{\omega}_j - \boldsymbol{\omega}_i)\times\boldsymbol{n} + 0.5[(l_j - l_i)/(l_i + l_j)]\boldsymbol{u}_{\mathrm{S}}$
Twisting angular displacements since time t_0	$\boldsymbol{\delta}_{\mathrm{T}}$	$\int_{t_0}^{t} \boldsymbol{\omega}_{\mathrm{T}}(\tau)\mathrm{d}\tau + \boldsymbol{\delta}_{\mathrm{T0}}$
Rolling angular displacements since time t_0	$\boldsymbol{\delta}_{\mathrm{R}}$	$\int_{t_0}^{t} \boldsymbol{\omega}_{\mathrm{R}}(\tau)\mathrm{d}\tau + \boldsymbol{\delta}_{\mathrm{R0}}$
Effective radius	R_{e}	$R_i R_j/(R_i + R_j)$

Note. $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the position vector of particles i and j, respectively; R_i and R_j the radii of particles i and j, respectively. τ is the time variable of integration. $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the velocity of particles i and j, respectively; ω_i and ω_j are the angular velocity of particles i and j, respectively;

cannot be directly calculated from the mechanical properties of the materials themselves; rather, they need to be determined based on a comprehensive consideration of the material properties and the intensity of particle interactions, introducing artificial uncertainty into DEM simulation results. The non-linear model, on the other hand, is derived from the theory of elastic mechanics, and its parameters can be obtained from the physical properties of materials. Nevertheless, the complete theory of Hertz and Mindlin–Deresiewicz requires intricate computational processes, leading to the adoption of simplified models in actual DEM modelling processes to enhance computational efficiency. Research has indicated that under appropriate material parameters, the linear model gives even better results than the simplified non-linear model at both macroscopic and microscopic scales (Di Renzo & Di Maio 2004). Furthermore, while the parameters of the non-linear model can be derived from the physical properties of materials, the lack of clarity regarding the physical properties of asteroid surface materials often also renders it impossible to derive accurate contact model parameters based on material properties. For these reasons, this study employs the linear spring-damping model described by Zhang et al. (2017) to calculate the contact forces and torques between particles. To facilitate the following discussion, we only list the basic equations here and more details about the derivation can be found in Zhang et al. (2017, 2018). The contact forces in the normal and tangential directions between particles i and j are given by

$$\begin{cases} \boldsymbol{F}_{\mathrm{N}} = -k_{\mathrm{N}}\delta_{\mathrm{N}}\boldsymbol{n} + C_{\mathrm{N}}\boldsymbol{u}_{\mathrm{N}} \\ \boldsymbol{F}_{\mathrm{S}} = \min(k_{\mathrm{S}}\boldsymbol{\delta}_{\mathrm{S}} + C_{\mathrm{S}}\boldsymbol{u}_{\mathrm{S}}, \mu_{\mathrm{S}}\lvert\boldsymbol{F}_{\mathrm{N}}\rvert\boldsymbol{\delta}_{\mathrm{S}}/\lvert\boldsymbol{\delta}_{\mathrm{S}}\rvert), \end{cases} \tag{1}$$

where, k_{N} and k_{S} are the spring stiffness that depends on Young's modulus of the material; C_{N} and C_{S} are plastic damping coefficients, which characterize the energy dissipation rate of the material; μ_{S} is the sliding friction coefficient.

The twisting and rolling torques are calculated as

$$\boldsymbol{M}_{\mathrm{T}} = \begin{cases} k_{\mathrm{T}}\boldsymbol{\delta}_{\mathrm{T}} + C_T\boldsymbol{\omega}_{\mathrm{T}}, \lvert k_{\mathrm{T}}\boldsymbol{\delta}_{\mathrm{T}}\rvert < M_{\mathrm{T,max}} \\ M_{\mathrm{T,max}}\boldsymbol{\delta}_{\mathrm{T}}/\lvert\boldsymbol{\delta}_{\mathrm{T}}\rvert, \lvert k_{\mathrm{T}}\boldsymbol{\delta}_{\mathrm{T}}\rvert \geq M_{\mathrm{T,max}} \end{cases} \tag{2}$$

$$\boldsymbol{M}_{\mathrm{R}} = \begin{cases} k_{\mathrm{R}}\boldsymbol{\delta}_{\mathrm{R}} + C_{\mathrm{R}}\boldsymbol{\omega}_{\mathrm{R}}, \lvert k_{\mathrm{R}}\boldsymbol{\delta}_{\mathrm{R}}\rvert < M_{\mathrm{R,max}} \\ M_{\mathrm{R,max}}\boldsymbol{\delta}_{\mathrm{R}}/\lvert\boldsymbol{\delta}_{\mathrm{R}}\rvert, \lvert k_{\mathrm{R}}\boldsymbol{\delta}_{\mathrm{R}}\rvert \geq M_{\mathrm{R,max}}, \end{cases} \tag{3}$$

where k_{T} and k_{R} are the twisting stiffness and rolling stiffness, and C_{T} and C_{R} are the twisting and rolling damping coefficients, respectively. $M_{\mathrm{T,max}}$ and $M_{\mathrm{R,max}}$ are the critical torque in the twisting and rolling direction, which depend on the shape parameter β, the static friction coefficients for twisting and rolling, μ_{T}, and μ_{R}. The intermediate variables used to calculate the contact forces and torques in Equations (1)–(3), are listed in Table 1. From these equations, it is evident that a total of 12 model parameters need to be determined for the calculation of contact forces. Among these, the stiffness in twisting and rolling directions can be determined by the stiffness in the normal and tangential directions, along with the shape parameter. The viscous damping coefficients in twisting and rolling directions can be determined by the plastic damping coefficients in the normal and tangential directions, along with the shape parameter. The plastic damping coefficients in the tangential and normal directions can be analytically derived from the linear spring-dashpot-slider model based on the coefficients of restitution in their respective directions (Asmar et al. 2002). Moreover, to ensure consistency between the resonant frequencies of normal and tangential motions, k_{S} can be set to $(2/7)k_{\mathrm{N}}$. k_{N} can be determined by anticipating the maximum interparticle overlap (Mishra 2003). To strike a balance between the precision and efficiency of contact force computations, it is quite common to establish the maximum overlap at 1 per cent of the particle radius as the criterion for determining k_{N}. The dependency relationships between the contact model parameters are listed in Table 2. From Table 2, it can be seen that the dynamic response of the particle system will be governed by six independent parameters: namely, the normal coefficient of restitution ε_{N}, tangential coefficient of restitution ε_{S}, sliding friction coefficient μ_{S}, static friction coefficients for twisting and rolling, μ_{T} and μ_{R}, and the shape parameter β. It should be noted that the particles referred to in this paper are all spherical particles; hence, the shape parameter mentioned here is a dimensionless parameter. Introducing this parameter in the rolling friction model of spherical particles aims to approximate the dynamic behaviouur of non-spherical particles using spherical particles. Essentially, it can be considered as a part of rolling friction.

Owing to the exceedingly low-gravity levels on the asteroid's surface, the cohesive force between particles becomes a strong factor that should be counted in the model. Therefore, the cohesion model (Zhang et al. 2018) that combines the granular bridge idea (Sánchez & Scheeres 2014, 2016) and a hypothetical contact area has been incorporated into our code, aiming to realistically reproduce the movement of regolith on asteroids. When two particles with radius R_i and R_j are in contact, the cohesive force between particles can be

Table 2. Contact model parameters in the linear spring-dashpot model (Asmar et al. 2002; Mishra 2003; Zhang et al. 2017).

Parameter	Symbols	Expression
Twisting stiffness	k_{T}	$2k_{\mathrm{S}}(\beta R_{\mathrm{e}})^2$
Rolling stiffness	k_{R}	$k_{\mathrm{N}}(\beta R_{\mathrm{e}})^2$
Twisting viscous damping coefficient	C_{T}	$2C_{\mathrm{S}}(\beta R_{\mathrm{e}})^2$
Rolling viscous damping coefficient	C_{R}	$C_{\mathrm{N}}(\beta R_{\mathrm{e}})^2$
Twisting critical torque	$M_{\mathrm{T,max}}$	$\mu_{\mathrm{T}}\beta R_{\mathrm{e}}\mu_{\mathrm{S}}\lvert\boldsymbol{F}_{\mathrm{N}}\rvert$
Rolling critical torque	$M_{\mathrm{R,max}}$	$\mu_{\mathrm{R}}\beta R_{\mathrm{e}}\lvert\boldsymbol{F}_{\mathrm{N}}\rvert$
Normal plastic damping coefficient	C_{N}	$-2\ln\varepsilon_{\mathrm{N}}\sqrt{\frac{k_{\mathrm{N}}m_{ij}}{\pi^2+(\ln\varepsilon_{\mathrm{N}})^2}}$
Tangential plastic damping coefficient	C_{S}	$-2\ln\varepsilon_{\mathrm{S}}\sqrt{\frac{k_{\mathrm{S}}m_{ij}}{\pi^2+(\ln\varepsilon_{\mathrm{S}})^2}}$
Reduced mass of particles i and j	m_{ij}	$m_{ij}=m_i m_j/(m_i+m_j)$
Normal spring stiffness	k_{N}	$m(\frac{v_{max}}{0.01R})^2$

Note. v_{max} represents the estimated maximum speed of the particle; R is the radii of particles. m denotes the mass of the particle.

modelled as

$$\boldsymbol{F}_{\mathrm{C}} = 4c(\beta\frac{R_iR_j}{R_i+R_j})^2\boldsymbol{n}, \tag{4}$$

where c represents the interparticle microscopic cohesion strength, measured in Pascal (Pa). Once the two particles are separated, the cohesive force will return to zero.

The tree code introduced by Barnes & Hut (1986) is employed to calculate self-gravity, significantly enhancing computational efficiency. We utilize a second-order leapfrog integrator, a common choice in the soft-sphere discrete element method, to solve the particle motion equations (Schwartz et al. 2012). It is important to highlight that the soft-sphere model incorporates damping and friction between particles (see Equations 1–3), which might appear to counter the benefits of leapfrog integration. However, given the small time-steps employed during simulations, integration accuracy remains sufficiently ensured.

2.2 Memory optimization for the scale-span computation

The contact force between particles depends on the degree of overlap between them, so accurately calculating the contact force between particles hinges on precise contact detection. A straightforward approach is the all-pairs algorithm, which is simple but the complexity of $O(N^2)$ makes it impractical for scenarios involving a large number of particles. As an example of the planetary simulation, observations suggest that a typical mass-loss on an asteroid involves approximately $10^6 \sim 10^7$ kg of regolith material (Moreno & Licandro 2014), equating to about $N = 100\,000$ particles with a density of 3350 kg m^{-3} and a diameter of 0.2 m. Implementing the all-pairs algorithm for contact detection in this scenario would require 1 billion loops for each time-step, making the computational cost unaffordable.

Two typical algorithms are available for improving contact detection efficiency: the spatial tree algorithm (Cao & Wang 2023) and the cell-linked list algorithm (Allen & Tildesley 2017). The spatial tree algorithm can better adapt to non-uniform particle distributions and large computational domains (Williams & O'Connor 1999). However, when utilizing the spatial tree algorithm for contact detection, the tree structure must be updated at every computational step, which is a highly time-consuming operation, especially for frequently varying the spatial positions of particles, a scenario encountered in this study. This significantly reduces the efficiency of particle contact detection. In contrast, the cell-linked list algorithm can ensure that particles outside neighbour cells need multiple time-steps to potentially contact the target particle by setting the cell size slightly larger than the particle diameter (Welling & Germano 2011). This enables the cell-linked list algorithm to update the neighbour list every tens or even hundreds of time-steps, thereby significantly reducing the time required for contact detection and enhancing the overall efficiency of the code. Since the relative displacement between particles is very small during awaiting neighbour list updates (typically, only a fraction of a particle radius moves within tens of time-steps), even if the self-gravity tree adopts the same update interval as the neighbour list, it does not significantly affect the accuracy of self-gravity calculations. Therefore, this study adopts the cell-linked list algorithm as the contact detection algorithm between particles.

The cell-linked list algorithm takes advantage of the short-range feature of the contact force and confines the searching field of a target particle to a limited number of cells. For illustration, the computational domain is first divided into uniform cells with an edge length denoted as L (Fig. 1). Particles are then assigned to respective cells according to their $x-$, $y-$, $z-$positions. The algorithm determines whether a target particle contacts its neighbour particles by evaluating the centre distances between it and all those in the adjacent cells (see e.g. Allen & Tildesley 2017). For a two-dimensional/three-dimensional case, a cell has 8/26 adjacent cells, respectively. This method reduces the computational complexity of contact detection from $O(N^2)$ to $O(N)$ (Yao et al. 2004; Zhou et al. 2021).

Taking Fig. 1(b) as an illustration, the diagram of the data storage structure of the conventional cell-linked list method based on uniform grid division to construct a neighbour list is shown in Fig. 2. While constructing the neighbour list, the address list, which acts as a pointer to particle information and is indexed by cell number, must be stored in memory. Thus, the space complexity of the conventional cell-linked list method is $O(N_{\mathrm{L}}^3)$ in a three-dimensional system, where the N_{L} is the number of cells in a single dimension. The complexity is acceptable for a limited computational domain. However, in a planetary scenario, the particles get dispersed quickly and soon the particle cloud extends to a vast region governed by the gravitational influence of the asteroid. As illustrated in Fig. 1, when the regolith particles enter the orbit close to the asteroid, similar to the restricted three-body system (Baoyin, Chen & Li 2010), their movement becomes unstable or even chaotic due to the disturbance caused by the complex gravitational field and other types of forces (Scheeres 1994; Yu et al. 2018, 2019; Song et al. 2023). And then the large spatial scale computational domain is needed for encompassing all particles.

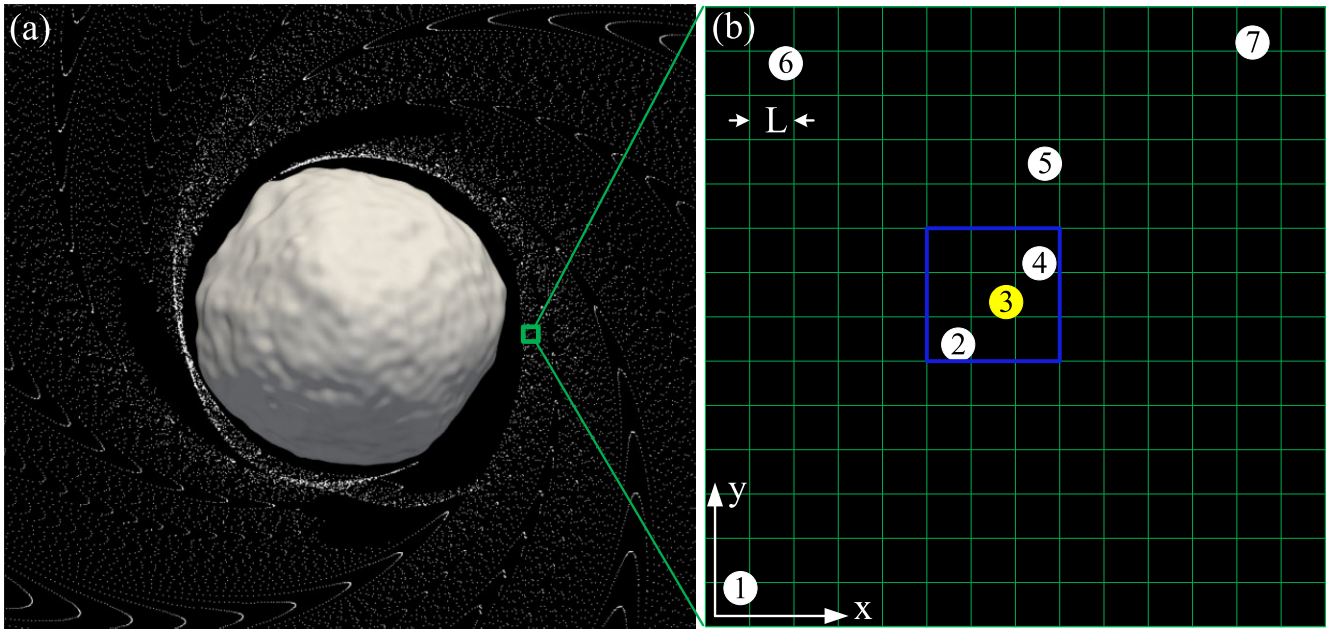

Figure 1. Distribution of the particle cloud in the planetary scenario. (a) Uneven distribution of the particle cloud on the vicinity of the asteroid. White point represents the particles; large white solid represents the asteroid. (b) Illustration of particle neighbours. The circles with numbers represent particles and their numbers and the blue wireframe represents the range the neighbour cells of the target particle 3.

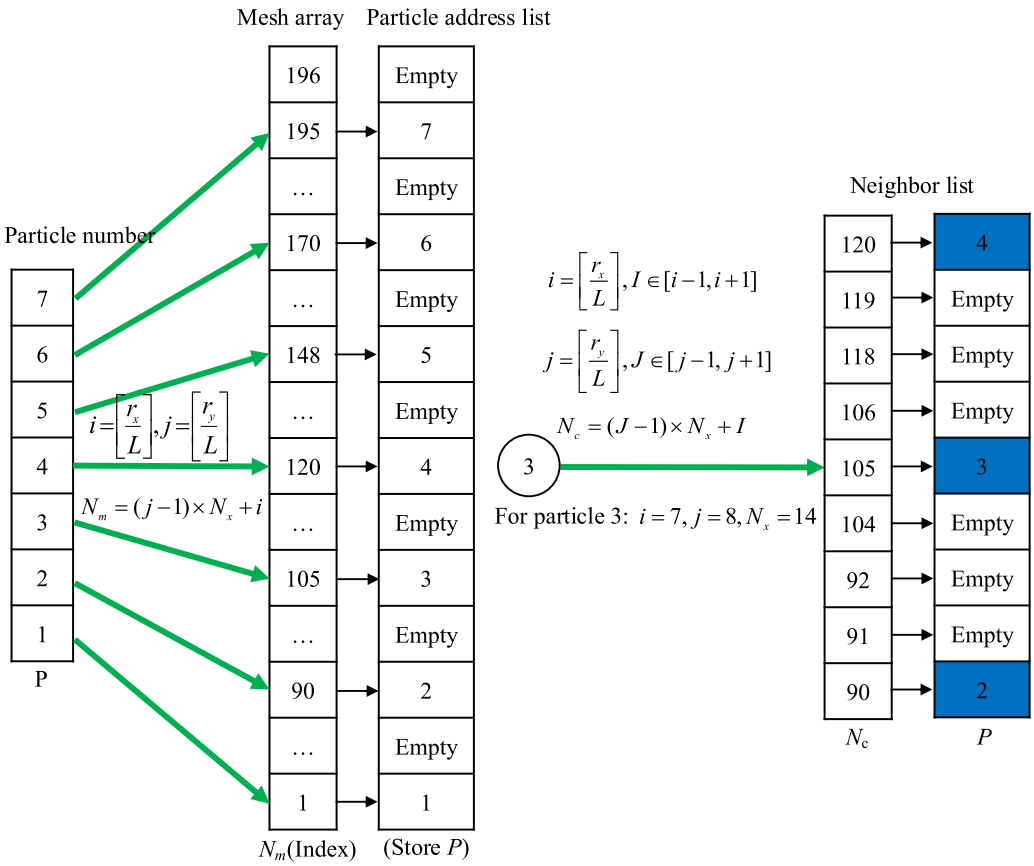

Figure 2. Scheme of the data structure of the conventional cell-linked list method. r_x and r_y indicate $x-$ and $y-$positions of the particle; L represents the cell length; N_x indicates the number of cells in the x direction. Particles are assigned to respective cells according to their $x-$, $y-$positions (left); list N_m represents the cell number used as the index of the particle address list. Constructing the neighbour list for particle 3 by retrieving particle address list (right); list N_c represents the neighbour cell number index that needs to be retrieved; numbers in the blue-filled boxes represent the numbers of neighbour particles.

On the other hand, in order to enhance the efficiency of contact detection, the grid size is usually set at 2–3 times the particle radius (Welling & Germano 2011). For a planetary scenario, the computational domain often differs by orders of magnitude compared to the regolith particle size. Consequently, a huge number of grid cells will be generated and stored in the programme, which challenges the memory management of the DEM code. Considering the asteroid (101955) Bennu, which roughly has a Hill radius of 35 km (McMahon et al. 2020), and a typical particle size of 0.1 m, at least 10^{15} grids (in the three-dimensional case) are required to encompass this computational domain. Storing these grids in integral data type requires over 10^6 GB of memory. Therefore, the conventional cell-linked list method encounters challenges in effectively processing these particular physical scenarios.

We noticed in a planetary scenario like Fig. 1, that due to the uneven distribution of the particles, only a small fraction of cells contain particles. This implies that the conventional cell-linked list method stores a significant number of empty cells, resulting in substantial memory waste. Therefore, if only the information of grids containing particles is saved rather than all cells, the memory footprint could be reduced sharply, i.e. the number of restored cells is no longer related to the computational domain; instead, it becomes related to the number of particles, which significantly conserves the memory. For planetary scenarios, it removes the limits on the computational domain, making a global whole-process simulation feasible.

Building upon the aforementioned motivations, we have refined the conventional cell-linked list method by utilizing the principles of hash mapping. In contrast to its conventional counterpart, our enhanced approach integrates merely one modulo operation and one multiplication operation. This refinement presents a straightforward, easily implementable solution to effectively mitigate the challenge of exponential memory expansion inherent in the conventional cell-linked list method and with negligible impact on contact detection efficiency, particularly when the computational domain significantly deviates from the cell size by orders of magnitude. Again using the particle distribution in Fig. 1 as an example, the schematic and data structure of the improved cell-linked list method for building a neighbour list is shown in Fig. 3. Additionally, a flowchart outlining the contact detection process is presented in Fig. 4 to complement this advancement.

Similar to the conventional cell-linked list method, our algorithm initially partitions the computational domain into a uniform grid cell of size L and allocates particles to specific cells based on their positions, as depicted in Figs 3 and 4. In contrast to the conventional approach, the improved cell-linked list method deviates from using the cell number directly as the index for storing particle address lists. Instead, it executes a modulo operation on the cell number, utilizing the resulting remainder as the index for storing these lists. During the subsequent neighbour search, the particles within the address list corresponding to the index of the cell where the target particle is located and its 26 adjacent cells (or eight adjacent cells in the two-dimensional case) will be examined in detail, as shown in Fig. 4.

None the less, employing the remainder as the index inevitably induces hash collisions, whereby multiple particles from distinct cells are mapped to the same memory address. For instance, as illustrated in Fig. 3(a), particles 2 and 6, located in different cells, are both directed to an address indexed at 10 for storage post the modulo operation. Consequently, during the detailed contact detection between the target particle and particles in adjacent cells, it is necessary to calculate the distances between the target particle and particles both in adjacent cells and in non-adjacent cells that map to the same address index as the target particle's neighbour cells. This process introduces additional time-consuming square root operations compared to the traditional cell-linked list method. For example, for target particle 3, its neighbour list is shown in Fig. 3(b) based on the particle distribution in Fig. 1(b). As seen in Fig. 1(b), particle 6 is not located in the neighbour cells of target particle 3. However, due to the modulo operation, particle 6 is mapped to the same address list index as particle 2, which is a neighbour of particle 3, as shown in Fig. 3. Consequently, during detailed contact detection for particle 3, it is necessary to calculate the distance between particle 3 and particle 2, located in its neighbour cells, as well as the distance between particle 3 and particle 6, which is not in the neighbour cells.

To avoid additional distance computations, a new 'Quotient list' is introduced to store the quotient of the cell number of the particles when dividing the particles into cells. The 'Quotient list' in Fig. 3(a) is a new integer list that stores the quotient obtained from the modulo operation on the cell number. This allows us to use this integer term to restore the cell number of the checked particle. If the restored cell number does not belong to the neighbour cells of the target particle, it indicates that the particle is not a potential contact object for the target particle. As a result, the necessity to compute the distance relative to the target particle is obviated, thereby the time-intensive square root operation typically involved in the distance calculation procedure is also avoided.

Taking particle 3 as an example, as illustrated in Fig. 1(b) and Fig. 3(b), the process of neighbour search begins by calculating the cell number associated with particle 3 and its neighbour cells. Following this, a modulo operation is performed on the eight neighbour cell numbers to obtain their remainders. These remainders serve as indices to access the 'Particle address list' stored in Fig. 3(a) alongside their respective integer terms. A verification process is conducted to confirm if 'Quotient $\times K +$ remainder' corresponds to the respective neighbour cell number, thus confirming the particle's actual cell (within the neighbour cell or not). Here is a detailed testing case. Considering neighbour grid number 90, the Remainder is 10 after undergoing a modulo operation. Upon retrieving the 'Particle address list' and the 'Quotient list' in Fig. 3(a), the corresponding particles are 2 and 6, and their 'Quotient' are 4 and 8, respectively. For particle 2, satisfying $4 \times 20 + 10 = 90$, which confirms it as a neighbour particle that requires further distance check. And for particle 6, $8 \times 20 + 10 \neq 90$, indicating that further detection is unnecessary.

By using the algorithm as described above, we efficiently reduce the memory requirement and make the improved cell-linked list method feasible for simulations of scale-span scenarios, which is much needed for the numeric study of asteroid systems. Note the proposed algorithm uses the remainder of the cell number as the index to save the particle address list, which is essentially an incomplete hash mapping. Hence, hash collisions are almost inevitable in the algorithm. The number of hash collisions will affect the efficiency of particle contact detection. The number of hash collisions occurring depends on the number of particles N and the size of the hash table K.

To investigate how the ratio between K and N affects the efficiency of contact detection, we utilized the particle distribution depicted in Fig. 15(d) of Section 3.2 as initial conditions, employing the same parameter settings as in that case. We then conducted a statistical analysis of the time required for contact detection over 10 000 time-steps under various ratios of K and N. The results, depicting the average time spent on contact detection per time-step as a function of the ratio between K and N, are presented in Fig. 5. It is important to note

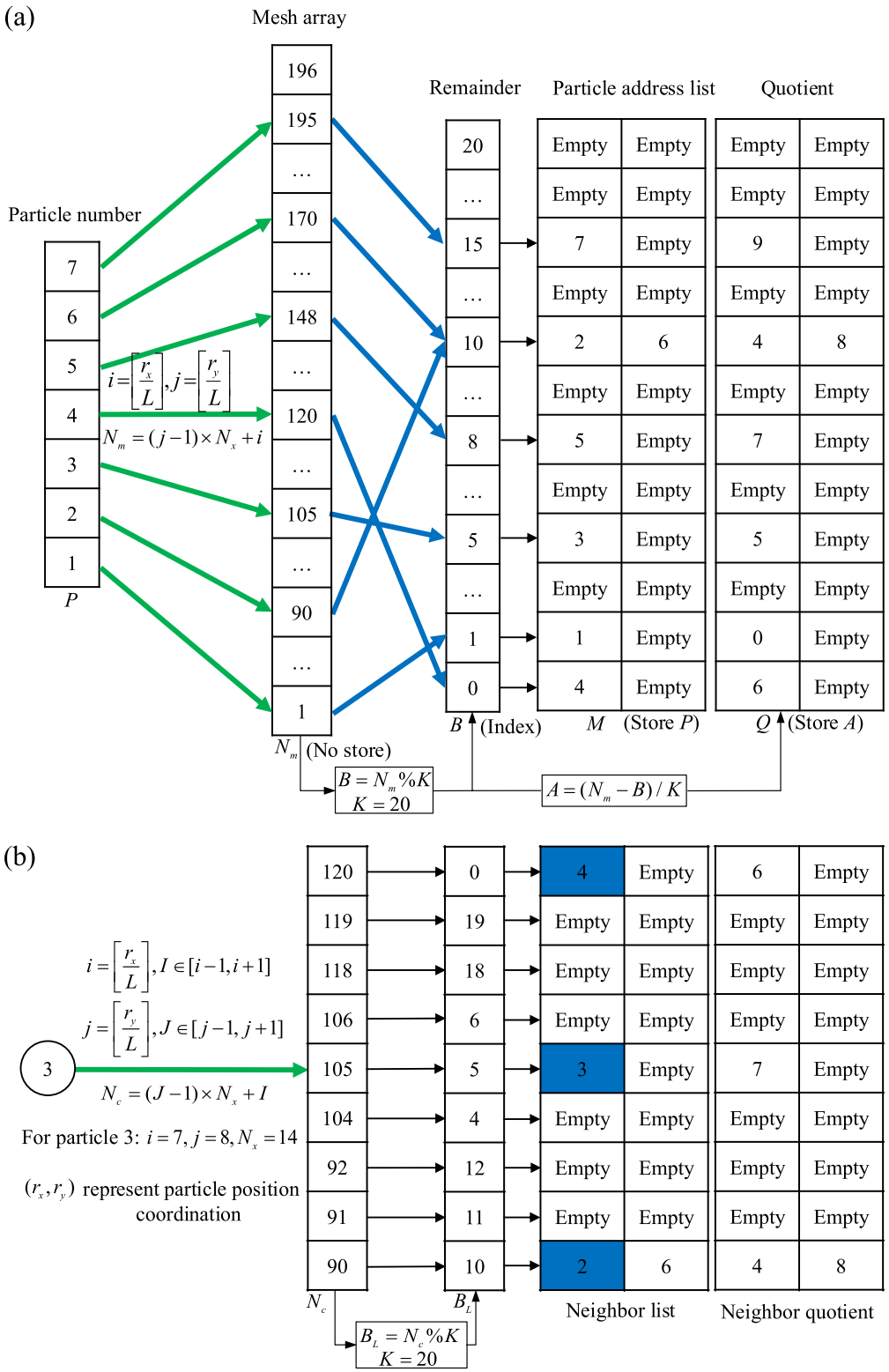

Figure 3. Diagram of the data structure of the improved cell-linked list method. (a) Binned the particle into the cells; list N_m represents the cell number used to execute a modulo operation; list B represents the remainder used as the index of the particle address list M and quotient list Q. (b) Searched particle neighbours; list N_c represents the neighbour cell number that needs to execute a modulo operation; list B_L represents the remainder used as the index to retrieve the particle address list M and quotient list Q; K is the size of the hash table; numbers in the blue filled boxes represent the numbers of neighbour particles and target particles.

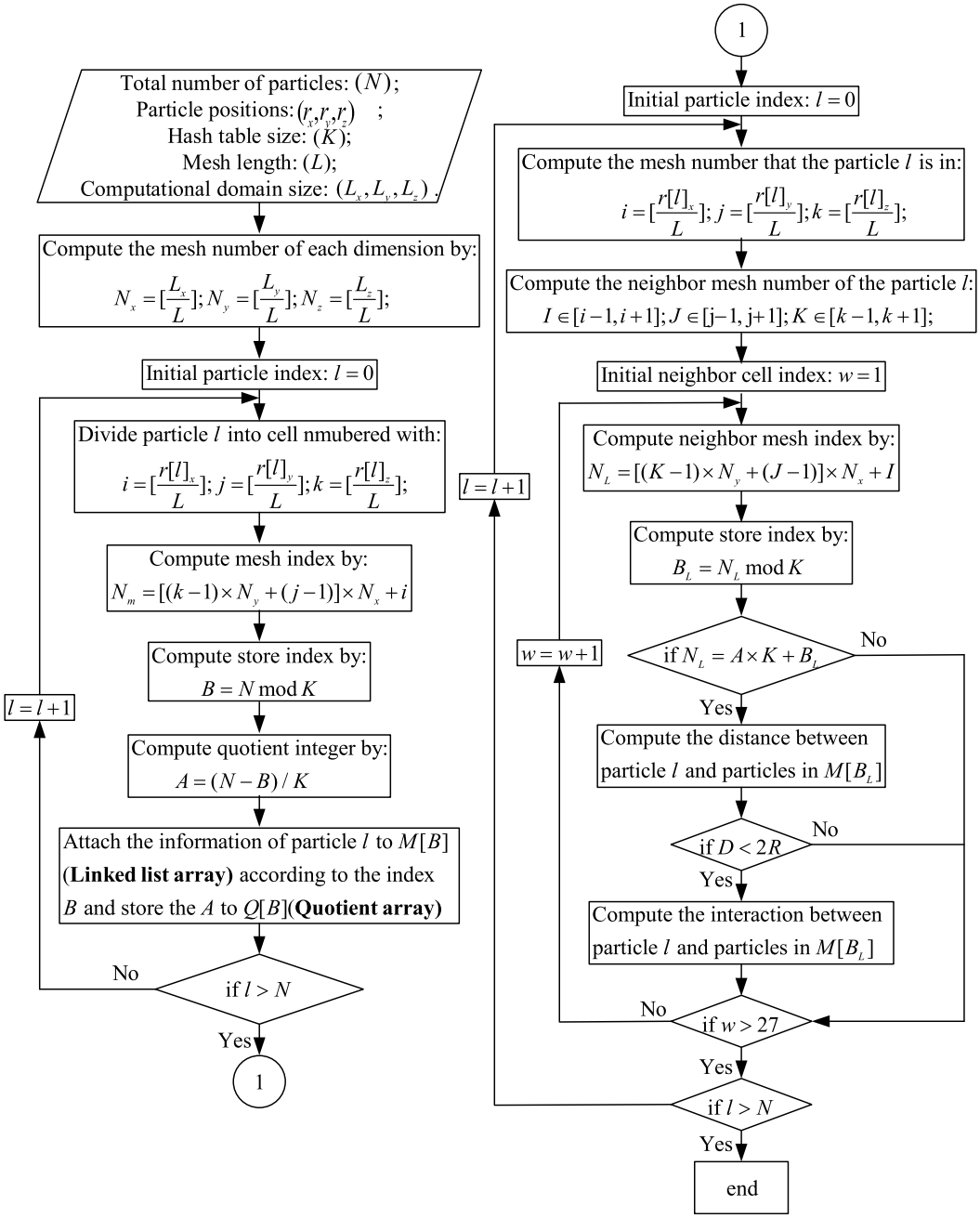

Figure 4. Neighbour detection flow chart of the improved cell-linked list method.

that, to eliminate any unforeseen impacts of compiler optimizations on efficiency tests, the results shown in Fig. 5 are obtained without compiler optimizations. Although the computational efficiency shown is significantly lower than that of the actual computational code, it does not affect our determination of the ratio between K and N.

Fig. 5 demonstrates that as the ratio of K to N increases, the time required for contact detection initially decreases rapidly, reaching a minimum of around $K/N = 10$, before increasing again and stabilizing at around 70 ms. Clearly, as the ratio of K to N increases, the occurrence of hash collisions decreases, leading to a reduction in the number of loops required for contact detection and consequently an increase in efficiency. However, as the ratio of K to N continues to increase, the particle address list becomes sparser, resulting in reduced cache utilization by the computer during the loop process. This subsequently counteracts the efficiency increase associated with the decrease in the number of loops, leading to a subsequent decrease in the efficiency of contact detection. The actual calculation results show that K is 10 times the number of particles N, i.e. it meets the requirement to conserve memory but does not notably impact the efficiency of contact detection.

2.3 Contact forces between particles and the asteroid surface

It is necessary to calculate the contact force and torques between the regolith particles and the asteroid's surface, especially for the scenarios of surface landslides and regolith migration. In our code, the polyhedron consisting of triangular surfaces is used to represent

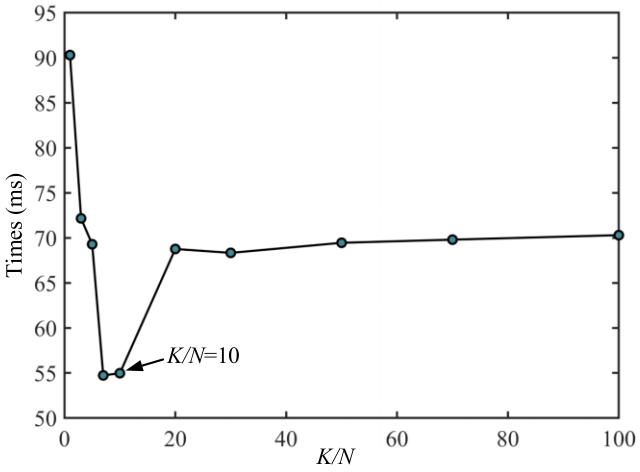

Figure 5. Contact detection time as a function of the ratio of K and N.

the actual shape of an asteroid, as shown in Fig. 6. The contact force between a particle and the surface of an asteroid depends on the degree of overlap between the particle and the surface triangle, the relative velocity, and the mechanical properties of the particle. Accurately determining these quantities requires precise detection of whether the particle is in contact with the asteroid's surface. For such a polyhedron, the most straightforward contact detection method is to assess the contact of the target particle with each triangle of the polyhedron. However, this approach requires traversing all triangles during the contact detection for each target particle. This is an excessive computational cost for high-resolution topography, which typically comprises hundreds of thousands of triangles. To improve the efficiency of contact detection between particles and asteroid surfaces, we propose an effective contact detection algorithm inspired by the concept from Gao, Cheng & Yu (2022), specifically tailored for scenarios where any ray emanating from the centre of the asteroid intersects the asteroid surface at only one point (Gao et al. 2022). Constrained by this condition, this algorithm will only apply to roughly convex-shaped asteroids, such as top-shaped asteroids Bennu and Ryugu. More general shape-insensitive algorithms will be developed in future work.

Specifically, the method first checks whether the distance between the position of the target particle and the centre of the asteroid exceeds the sum of the radius of the asteroid's envelope sphere and the radius of the particle. If this condition is met, indicating the particle is not in contact with the asteroid's surface, further detection is unnecessary. Otherwise, detailed contact detection needs to be performed between the particle and the triangles. To avoid traversing all triangles during detailed contact detection, the vertex coordinates of the original polyhedron representing the asteroid are interpolated to generate a new polyhedron with uniformly distributed triangles along the longitude θ and latitude ϕ directions with a certain interval $(\Delta\theta, \Delta\phi)$. The triangles in this new polyhedron are renumbered sequentially, as depicted in Figs 6(a) and (b). Together with the coordinate conversion relationship between the Cartesian coordinate and spherical coordinate system and equation (5), we can easily determine which triangle the target particle is located near, as shown in Fig. 6(c).

$$\begin{cases} n_\theta = \text{floor}(\dfrac{\theta}{\Delta\theta}) \\ n_\phi = \text{floor}(\dfrac{\phi}{\Delta\phi}) \end{cases} \tag{5}$$

Since the particle is improbable to span cross a triangle wider than its diameter to come into contact with the asteroid surface outside the neighbour triangle, the particle with a diameter smaller than the triangle size is likely to only come into contact with the triangle containing the projection point of the target particle on the asteroid surface, as well as the 12 neighbour triangles of this triangle (as indicated by the green-marked and orange-marked triangle in Figs 6b and c). Therefore, a more detailed contact detection is performed involving the 12 neighbour triangles of the initial triangle (the orange-marked triangle in Figs 6b and c) to pinpoint the exact contact point. On the other hand, when the particle diameter is larger than the triangle size, the number of triangle layers for detailed contact detection needs to be expanded outward based on the relationship between the particle size and the triangle size. For example, when

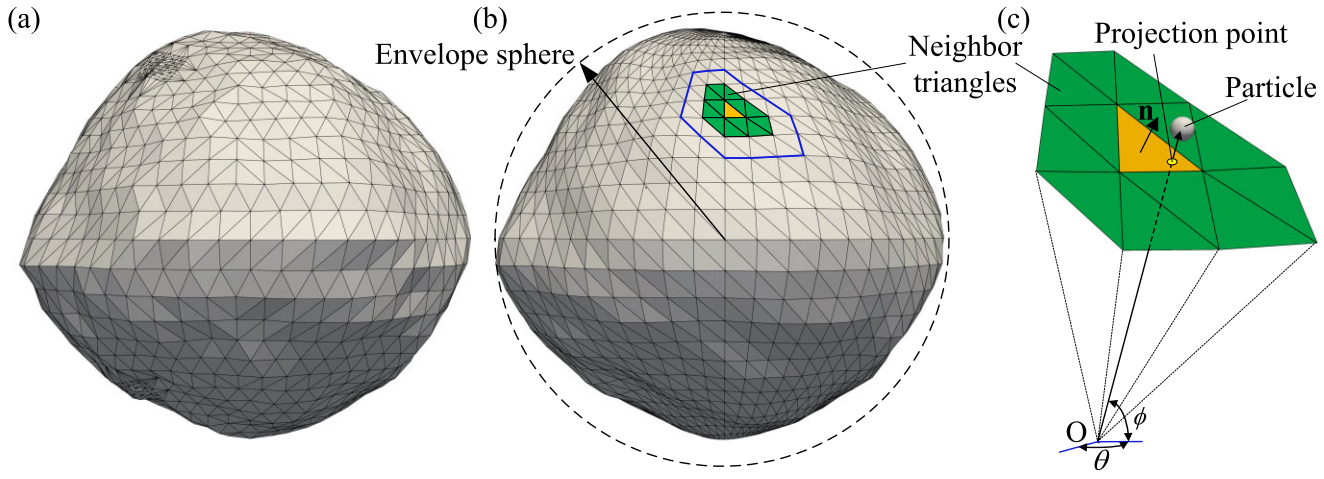

Figure 6. Diagram illustrating the object particle and corresponding triangle and its neighbour triangles that require detailed detection. (a) Original polyhedron shape. (b) New polyhedron shape with uniformly distributed triangles. (c) Object particle and corresponding triangle and its neighbour triangles. The orange triangle represents the triangle where the projection point of the target particle is located and the green triangles represent the neighbour triangle of the target particle with diameter smaller than the triangle size. Triangles surrounded by blue lines indicate neighbour triangles of target particles with a diameter ranging from 1 to 2 times the size of the triangle.

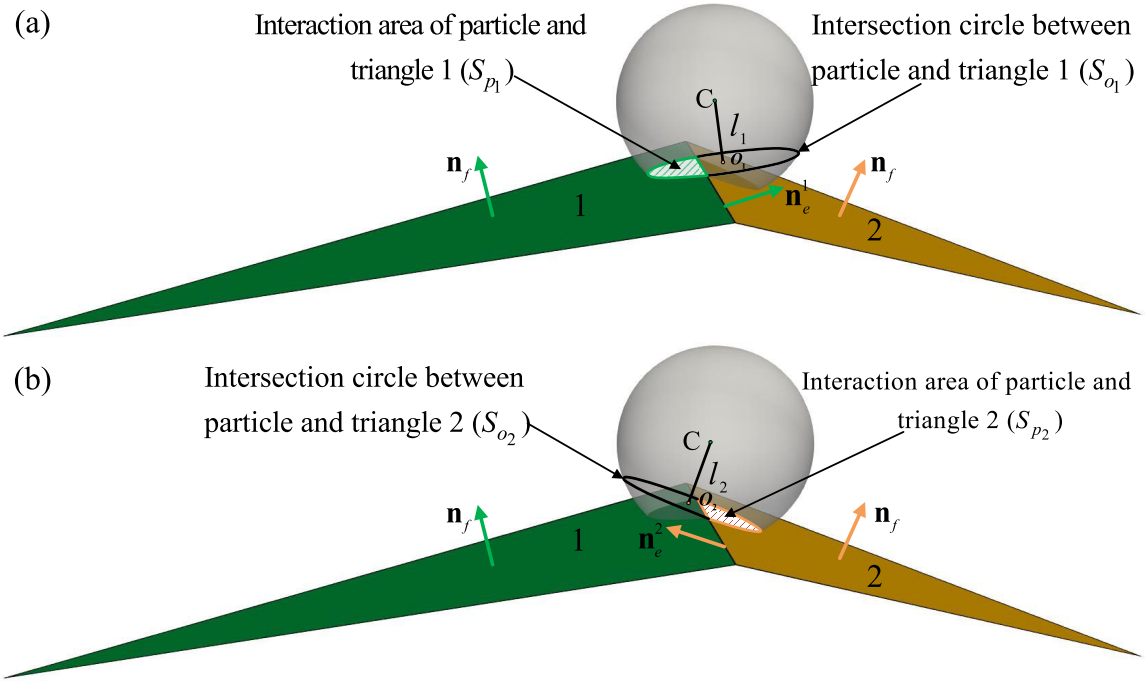

Figure 7. Illustration of the intersection area of the particle and triangle and the intersection circle between the particle and the extended plane of the triangle. (a) Intersection of the particle and triangle 1. (b) Intersection of the particle and triangle 2. $\boldsymbol{n}_{\rm f}$ is the normal vector of triangle and $\boldsymbol{n}_{\rm e}^{i} i \in (1, 2)$ represents unit normal vector that is perpendicular to the edge e and the $\boldsymbol{n}_{\rm f}$. The C is the centre of the particle.

the particle size is between 1 and 2 times the size of the triangle, two layers of triangles need to be expanded outward (as shown in the blue-lined area in Fig. 6b). Similar to the determination of neighbour cells in the cell-linked list approach, the number of neighbour triangles can simply be calculated by a predetermined numbering rule. Furthermore, in the polar regions, since any triangle will be adjacent to other polar region triangles, as illustrated in Fig. 6(b), all triangles of the polar region must be detected in the detailed contact detection phase when the target particle is located in the polar region.

The perpendicular distance l between the centre of the particle sphere C and the plane in which the triangle is located, as shown in Fig. 7, is calculated in a more detailed contact detection phase between a particle and the triangle. If l is less than the radius of the particle, it indicates that the particle may be in contact with this triangle, and the contact point vector $\boldsymbol{r}_{\rm O} = \boldsymbol{r}_{\rm C} - l\boldsymbol{n}_{\rm f}$ ($\boldsymbol{r}_{\rm C}$ represents the position vector of the centre of the particle sphere C). However, the contact situation between a sphere and a triangular is complex. Calculating the contact force simply using the contact location and the overlap can result in discontinuities in the resultant force as the particles cross the edge of the triangle (Zhou et al. 2021). To maintain a smooth resultant force as the particle crosses the boundary of the triangle, an area coefficient is introduced to calculate a weighted average of the forces exerted by all contacted triangles as illustrated in Fig. 7. Consequently, the resultant force $\boldsymbol{F}_{\rm Ac}$ on the target particle by the triangle can be derived from equation (6).

$$\boldsymbol{F}_{\rm Ac} = \sum_{i=1}^{N_{\rm tri}} \left(\frac{S_{\rm P}}{S_{\rm O}}\right)_i \boldsymbol{F}_{\rm ti}, \tag{6}$$

where $N_{\rm tri}$ represents the number of triangles in contact with the target particle, and $S_{\rm O}$ indicates the area of the intersection circle between the target particle and the extended plane of the triangle. $S_{\rm P}$ represents the area of this intersection circle inside the triangle. Since the value of $S_{\rm P}$ cannot be obtained analytically (Strobl, Formella & Pöschel 2016; Zhou et al. 2021), the values of $S_{\rm O}$ and $S_{\rm P}$ are given here numerically using the algorithm proposed by Strobl et al. (2016). $\boldsymbol{F}_{\rm ti}$ in equation (6) denotes the contact force exerted by the ith surface on the target particle, including the normal force, the tangential force, the cohesive force, the twisting torque, and rolling torque, and can be calculated by employing the similar contact law described by Zhang et al. (2017, 2018). In our code, the motion state and mechanical parameters of the particle j in the contact law (see Equations (1)–(3)) are replaced with those of the contact triangle to calculate $\boldsymbol{F}_{\rm ti}$.

2.4 CUDA acceleration of the gravity computing

Accurate calculation of the gravitational field on and near the surface of asteroids is a crucial foundation for modelling the motion of regolith materials on the surface of asteroids. The methods to calculate the gravitational field of irregular asteroids include the harmonic approach (Takahashi & Scheeres 2014) based on series expansion, the mascon approach (Geissler et al. 1996), and the polyhedron approach (Werner & Scheeres 1997), etc. Of the aforementioned methods, the harmonic approach fails to achieve convergence within the Brillouin sphere, thus rendering it unsuitable for simulating material movement on asteroid surfaces. Similarly, the mascon approach encounters a rapid decline in calculation accuracy as particles near or traverse the surface, rendering it inadequate for studying material movement on asteroids (Werner & Scheeres 1997; Park, Werner & Bhaskaran 2010). Conversely, the polyhedron approach circumvents these limitations, establishing itself as the preferred method in our code for modelling the irregular gravitational field of asteroids. Here, we provide a concise overview of the primary equations of the

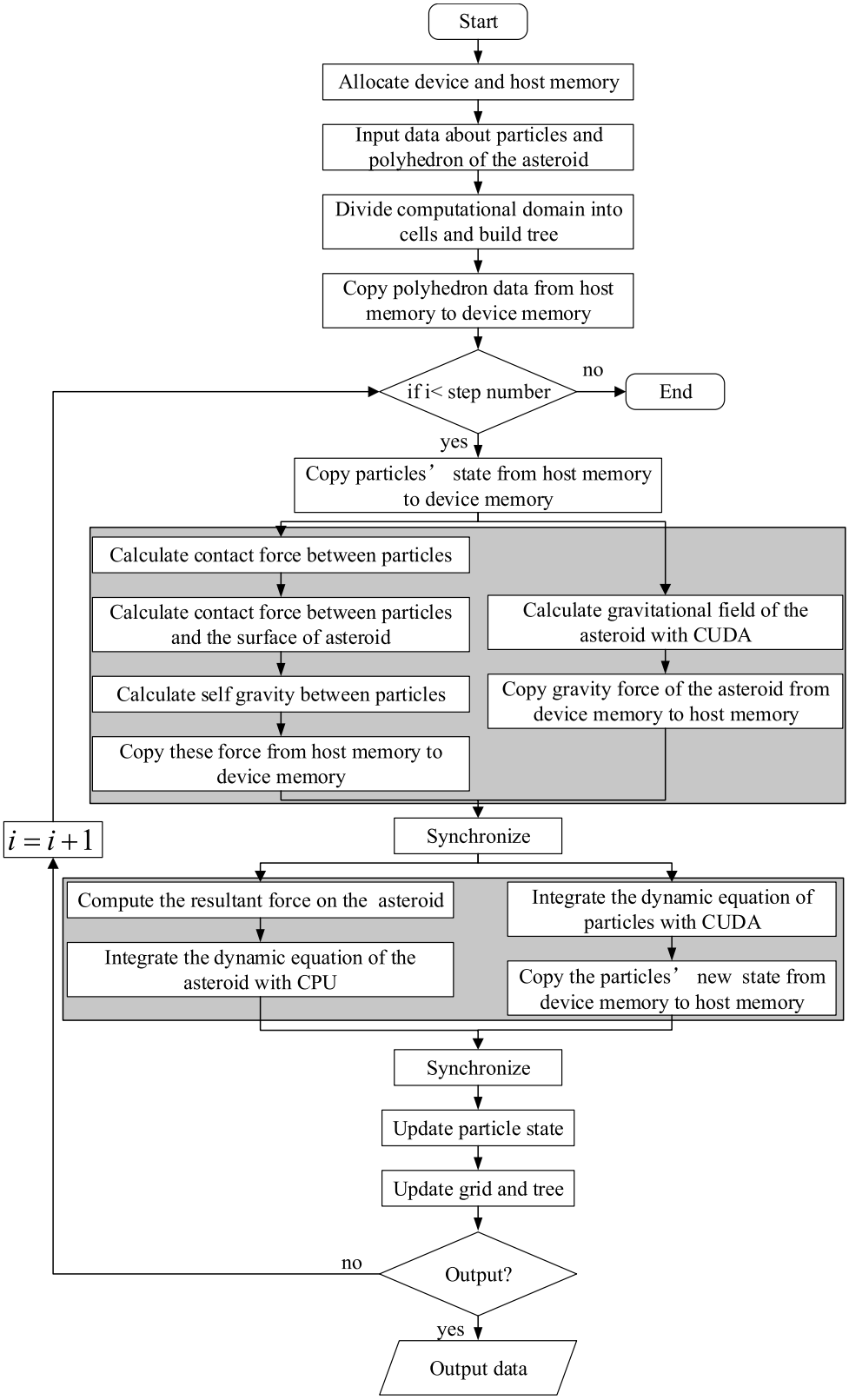

Figure 8. Flowchart of the overall code. The shaded portion represents the steps of overlapping calculations.

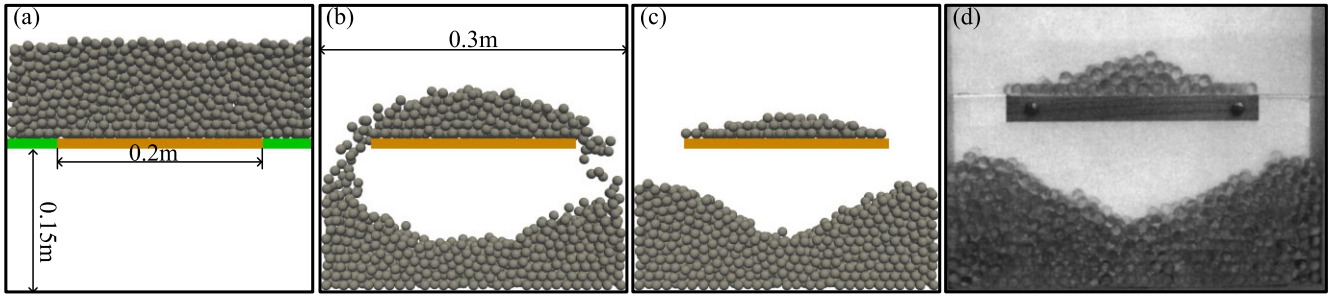

Figure 9. Simulation and reproduction of the hopper collapse experiment. (a) The initial state of particles, where the plates on both sides are removable plates and the plate in the centre is a fixed plate. (b) The particles fell from both sides under gravity after removing the two side plates. (c) The stagnant zone formed on the centre plate after discharge. (d) The experimental results (Zhou et al. 1999).

polyhedron approach, while comprehensive derivations can be found in Werner & Scheeres (1997).

Let e represent the edges that form the polyhedron, and let f denote the faces of the polyhedron. Let $\boldsymbol{n}_{\mathrm{f}}$ be the unit normal vector outwards-pointing from the face f, and let $\boldsymbol{n}_{\mathrm{e}}^{f}$ be the unit normal vector outwards-pointing that is perpendicular to both the edge e and the $\boldsymbol{n}_{\mathrm{f}}$, as shown in Fig. 7. Let $\boldsymbol{p}_i$ be the vector of the ith polyhedron vertices. The symbols G and ρ represent the gravitational constant and the bulk density of the asteroid, respectively. The gravitational force on the particle at position $\boldsymbol{r}$ by the asteroid using the polyhedron approach can be expressed as

$$\boldsymbol{F}_{\mathrm{g}}(\boldsymbol{r}) = -\mathrm{G}\rho\Big(\sum_{\mathrm{e}\in\mathrm{edges}} \mathsf{E}_{\mathrm{e}}\boldsymbol{x}_{i\in\mathrm{e}}L_{\mathrm{e}} - \sum_{\mathrm{f}\in\mathrm{faces}} \mathsf{F}_{\mathrm{f}}\boldsymbol{x}_{i\in\mathrm{f}}\omega_{\mathrm{f}}\Big) \tag{7}$$

where,

$$\boldsymbol{x}_i = \boldsymbol{r} - \boldsymbol{p}_i, \tag{8}$$

$$\mathsf{E}_{\mathrm{e}} = \boldsymbol{n}_A \otimes \boldsymbol{n}_{\mathrm{e}}^{A} + \boldsymbol{n}_B \otimes \boldsymbol{n}_{\mathrm{e}}^{B}, \tag{9}$$

$$\mathsf{F}_{\mathrm{f}} = \boldsymbol{n}_{\mathrm{f}} \otimes \boldsymbol{n}_{\mathrm{f}}, \tag{10}$$

$$L_{\mathrm{e}} = \ln\frac{x_i + x_j + e_{ij}}{x_i + x_j - e_{ij}}, \tag{11}$$

$$\omega_{\mathrm{f}} = 2\arctan\frac{\boldsymbol{x}_i \cdot (\boldsymbol{x}_j \times \boldsymbol{x}_k)}{x_i x_j x_k + x_i(\boldsymbol{x}_j \cdot \boldsymbol{x}_k) + x_k(\boldsymbol{x}_i \cdot \boldsymbol{x}_j)}, \tag{12}$$

where, $x_i = ||\boldsymbol{x}_i||$, and e_{ij} represents the length of the edge contacted the vertex i and j.

Equation (7) shows that the calculation of the gravitational force on each particle exerted by the asteroid requires traversing all edges and faces of the polyhedron, which is a very time-consuming process for a single-core CPU. Fortunately, taking into account the computational process and data for calculating the gravitational field of the asteroid on particles using equation (7) are independent of each other for different particles, the parallel computation can be utilized to significantly improve the efficiency of the gravity calculation.

CUDA is a parallel computing platform and programming model released by NVIDIA, which takes advantage of the multicore features of GPU to significantly improve the parallel performance of a computer programme. We propose a GPU-accelerated parallel algorithm to calculate the gravitational force exerted on particles based on equation (7). According to Equations (9)–(11), the calculation process requires loop calling of the face normal vector $\boldsymbol{n}_{\mathrm{f}}$, edge normal vector $\boldsymbol{n}_{\mathrm{e}}^{f}$, and length of edge e_{ij} of the polyhedron. Saving these data in shared memory in advance can greatly improve memory access speed. However, due to the constrained size of the shared memory of CUDA (typically 48 KB shared memory per block), it is not possible to directly copy all the above data into the shared memory. To speed up the memory access and consequently improve computational efficiency, we serialize certain computations to facilitate the necessary data reuse. Specifically, we introduce the concept of computational tile, i.e. dividing the overarching loop into smaller segments. Consequently, the data is partitioned into subsets that can be loaded into the shared memory. The algorithm in Appendix A gives the pseudo-code of the method in detail. As shown in lines 8–13, 23–30, and 45–53, we collaboratively load the subsets of the information related to the polyhedron into the shared memory and utilize this data for computation, in which $VertexTile$, $FaceTile$, and $EdgeTile$ denote the sizes of the vertex, face, and edge tile, respectively. The tiles can be computed based on the number of vertices, edges, and faces of the polyhedron.

By using GPU to compute the polyhedral gravity of asteroids, the code can fully exploit the advantages of heterogeneous computing, i.e. it enables the overlap computation between the GPU and CPU, thereby reducing the total computational time of the code. Fig. 8 shows a flowchart that details the heterogeneous CPU–GPU architecture designed to achieve time overlap between long-range and short-range force computations. This approach effectively offsets the computational load of the polyhedral gravity field.

2.5 Interactive dynamics between the particles and the asteroid

In order to track the delicate evolution of an asteroid with regolith material moving across its surface, we establish the motion equation of the asteroid, which is solved together with the DEM equations. The resultant force acting on the asteroid includes the contact force and torque between the particles and the asteroid, as well as the gravitational force and torque exerted by the particles on the asteroid. As a result, the resultant force on the asteroid $\boldsymbol{F}_{\mathrm{RA}}$ can be represented by

$$\boldsymbol{F}_{\mathrm{RA}} = -\sum_{i=1}^{N}(\boldsymbol{F}_{\mathrm{Ac}}^{i} + \boldsymbol{F}_{\mathrm{g}}^{i}), \tag{13}$$

where $\boldsymbol{F}_{\mathrm{Ac}}^{i}$ is the contact force between the particle i and the surface of the asteroid, $\boldsymbol{F}_{\mathrm{g}}^{i}$ is the gravity of the asteroid acting on the particle i, N is the number of particles. The expression for the contact torque acting on the asteroid can be expressed as

$$\boldsymbol{M} = -\sum_{i=1}^{N}[(\boldsymbol{r}_i - l_i\boldsymbol{n}_{\mathrm{f}}) \times \boldsymbol{F}_{\mathrm{Ac}}^{i} + (\boldsymbol{r}_i - \boldsymbol{r}_{\mathrm{ast}}) \times \boldsymbol{F}_{\mathrm{g}}^{i}] \tag{14}$$

where l_i indicates the perpendicular distance between the centre of the particle sphere C and the plane in which the triangle is located, as

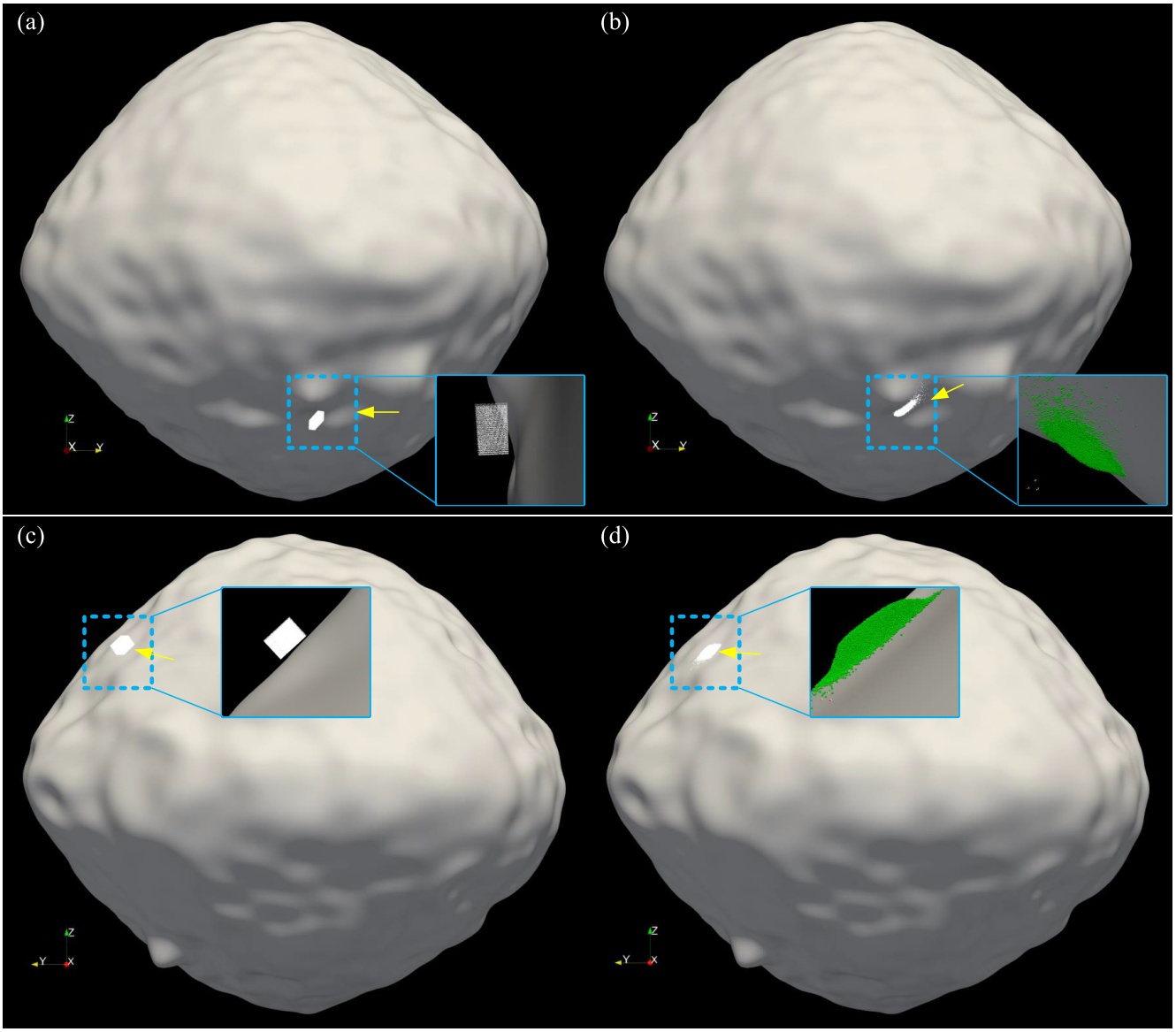

Figure 10. Initial state of the particles and the final resting sand pile on Bennu's surface formed by these particles. Panels (a) and (b) correspond to the initial state and the final resting sand pile of sand-pile 1, respectively. Panels (c) and (d) correspond to the initial state and the final resting sand pile of sand-pile 2, respectively. The direction of the arrow represents the view direction of the enlarged view.

shown in Fig. 7. Consider the asteroid as a rigid body and substitute Equations (13) and (14) to the Newton–Euler equation, the motion state of the asteroid can be determined by

$$\begin{cases} \boldsymbol{F}_{\mathrm{RA}} = m_{\mathrm{ast}} \ddot{\boldsymbol{r}}_{\mathrm{ast}} \\ \boldsymbol{M} = \mathsf{I} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathsf{I} \cdot \boldsymbol{\omega}) \end{cases} \tag{15}$$

where m_{ast} indicates the mass of the asteroid; $\ddot{\boldsymbol{r}}_{\mathrm{ast}}$ represents the acceleration of the centre of mass of the asteroid; I is the product of inertia of the asteroid; $\dot{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$ are the angular acceleration and angular velocity of the asteroid, respectively.

3 CODE VALIDATION AND APPLICATION

Based on the proposed method, we created a DEM code that can accurately simulate the motion of asteroid regolith materials, incorporating high-resolution surface details and natural-scale regolith particles. In this section, we conducted simulations to test the code performance and to demonstrate its capabilities. The first simulation focuses on validating the accuracy of the contact model implemented in our code by comparing the results with experimental data. Subsequently, we choose the target asteroid of OSRIS-REx mission Bennu as an example to simulate the mass migration of regolith material on its surface by setting a hypothetical high-spin rate. This simulation aims to demonstrate the code's performance in terms of efficiency, accuracy, and reliability while executing a scale-span simulation, which exhibited the intricate dynamics of the asteroid-regolith system during a full-cycle evolution from local landslides to post-shedding processes.

3.1 Verification of the contact model

The particle flow within the hopper exhibits a high complexity, which usually involves physical processes such as collapse, flow, heap

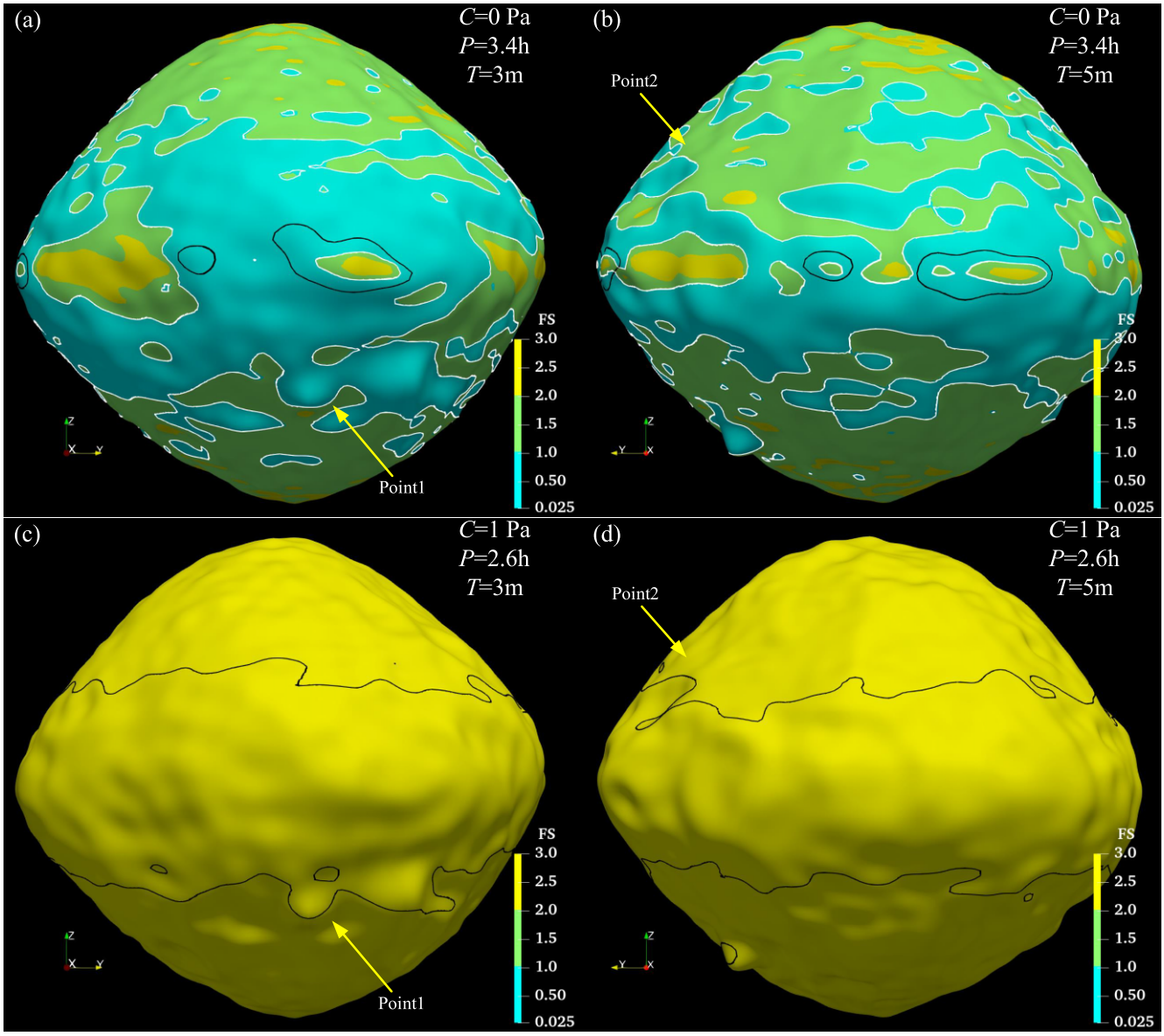

Figure 11. Failure pattern distribution of Bennu. The colour bars indicate the magnitude of FS, and the white solid line indicates the critical value of landslide failure at FS = 1. The black solid line indicates the critical value for fission failure, and the area surrounded by the black solid line indicates the area where fission failure will occur.

formation, etc. Zhou et al. (1999) conducted the hopper discharging experiment and utilized this experiment to validate the contact model they proposed in their paper. Given that this experiment involves both interparticle contacts and particle-surface contacts (the brown plate and surrounding walls in Fig. 9), and our contact model also incorporates these interactions, we employ the experiment results of Zhou et al. (1999) as a benchmark to verify the contact model described in the previous section. Using the geometrical model by Zhou et al. (1999), we initially generate a random packing of 5 mm radius particles with zero velocities above a wooden plate, and drop these particles freefall under the Earth's gravity ($g = 9.8\,\mathrm{m\,s^{-2}}$) to form the initial granular bed. When the kinetic energy of the particle system dropped to near zero, we removed the excess particles from the top layer, retaining the same number of particles, 1154, as used in the experiment. Using this approach, we generated three distinct initial heaps. Subsequently, we removed the small wooden plates on both sides, allowing the particles to discharge from both sides by gravity, as shown in Fig. 9.

Given that the glass sphere particles used in the experiment are not perfect spheres, to model the imperfect spherical particles in the experiments with the perfect spherical particles in the DEM, we set a very small shape parameter in the simulation, i.e. $\beta = 0.025$. The sliding friction coefficient and particle density were set to 0.4 and $2500\,\mathrm{kg\,m^{-3}}$, respectively, which are the same as the simulation parameters used by Zhou et al. (1999). Since Zhou et al. (1999) directly employed the damping coefficient rather than the coefficient of restitution in their parameter settings (see table 2 in Zhou et al. 1999), the coefficients of restitution used in our simulations are derived based on their damping coefficient by using the relationship between the damping coefficient and the coefficient of restitution listed in Table 2. Based on the equations listed in Table 2, the derivation of the coefficient of restitution from the damping

Table 3. Parameter used to simulate the movement process of sand-pile 1 and sand-pile 2 after failure on Bennu's surface.

ID	Sand-pile	μ_s	μ_T	μ_R	β	ε_N	ε_S	R (m)	C (Pa)	P_0 (h)
1	1	0.55	1.3	1.05	0.5	0.55	0.55	0.1	0	3.4
2	2	0.55	1.3	1.05	0.5	0.55	0.55	0.1	0	3.4
3	1	0.55	1.3	1.05	0.5	0.55	0.55	0.1	1	2.6
4	2	0.55	1.3	1.05	0.5	0.55	0.55	0.1	1	2.6

coefficient is influenced by the relative velocity of the particles. We adopted a particle relative velocity of 0.1 m s^{-1}, which roughly corresponds to the average kinetic energy per particle during the hopper discharging process (see fig. 15 in Zhou et al. 1999), to derive our coefficient of restitution from the damping coefficient reported by Zhou et al. (1999). Therefore, the tangential and normal coefficients of restitution were set to 0.5217 and 0.7099, respectively. Cleary (2000) has demonstrated that the precise integration of a collision process requires 20 to 50 time-steps. For the linear spring-dashpot-slider model, the single contact time can be derived analytically (Nagurka & Shuguang Huang 2004). Conservatively, we assume that each contact lasts 50 time-steps and set the time-step to 2 per cent of the single contact time, i.e. $\Delta t = 1 \times 10^{-6}$ s.

The simulated and experimental results are shown in Fig. 9. When the small wooden plates on both sides were removed, the particles fell from both sides of the centre plate under gravity and formed an M-shape region on the lower side of the centre plate. The friction and the small shape parameter of the spherical particles lead to remaining particles on the centre plate, forming a stagnant zone. In our simulations, the numbers of particles remaining in the stagnant zone are 126, 128, and 131, which are in good agreement with the experimental values of 128 ± 3, indicating that the contact model adopted in this code can capture the complex particle flow process.

3.2 Application to landslides on Bennu

Asteroid Bennu is one of the handful of asteroids that have been explored in detail by spacecraft so far. The onboard camera has collected unprecedented high-resolution images, revealing its topographic map and the size composition of the regolith layer (Jawin et al. 2020). In this study, we employed the shape model of Bennu with a 75-cm resolution for a hypothetical spin-up scenario. Then, four numerical experiments are presented to illustrate the modelling capabilities of the code. Specifically, in terms of the spatial scale of the computational domain, these simulations confirm the code's ability to handle the continuous expansion of the computational domain from the meter level to the kilometer level. Regarding high-resolution modelling of asteroid shapes, the simulations showcase the code's ability to track the evolution of regolith at submeter high-resolution surfaces. In terms of particle quantity, simulations 2 and 4 demonstrate our code's capacity to model the movement of regolith particles on the order of 10^6 kg using realistic particle sizes, equivalent to the mass ejected during a typical active event on an active asteroid (Hainaut et al. 2014). Furthermore, these simulations also highlight our code's accuracy in tracking the coupling effects between the mass movement of regolith and the rotational state of asteroids.

3.2.1 *Initial setups*

To simulate the mass movement of regolith material on Bennu's surface, we constructed two resting sand piles consisting of 25 088 and 77 312 spherical particles with a radius of 0.1 m on Bennu's surface. The first sand-pile is positioned amidst two boulders, while the second one is situated on the flat surface near the crater, as shown in Fig. 10. Particle material parameters can be referenced from Yu et al. (2014) and Zhang et al. (2021). Specifically, we set the coefficients of restitution, ε_N, and ε_S, to 0.55, and the friction coefficients for the tangential, rolling, and twisting direction (μ_S, μ_R, μ_T) to (0.5, 1.05, 1.3). Accounting for the irregular shape of regolith particles on the surface, we set the shape parameter β to 0.5. Bennu's density was set at 1194 kg m^{-3}. The sand-piles were created by releasing these particles from a random height of 1–5 m and a random initial velocity of less than 0.01 m s^{-1} on to Bennu's surface, allowing them to settle under its gravity. The settling continued until the relative kinetic energy dropped to near zero, as illustrated in Fig. 10.

3.2.2 *Landslide on Bennu's surface*

Sánchez & Scheeres (2020) demonstrated the presence of two distinct failure patterns exhibited by regolith particles on asteroid surfaces, taking into account the cohesive strength of these particles, i.e. the landslide failure and fission failure. In cases where there is almost no cohesion strength, the regolith material on the surface of asteroids tends to fail through landslides and mass flow from the mid-latitude region toward the equator. Conversely, for high-cohesion strength, failure typically occurs at the fission failure region of the surface through shedding the regolith material. Consequently, accurately simulating these two distinct modes of regolith failure serves as a crucial benchmark for assessing the effectiveness of our code.

We employ the Factor of Safety (FS) analysis method (Iverson, Reid & LaHusen 1997), which has previously been utilized by Barnouin et al. (2022) to assess the stability of regolith material on Bennu's surface. This method outlines the conditions under which regolith on Bennu's surface may experience landslide failure. According to the principles of slope stability analysis, when the resisting forces acting on particles exceed the driving forces, particles are prone to undergo landslides on the surface of asteroids. For regolith material with specific parameters, such as a slope angle ψ, a macroscopic cohesion strength C, a friction angle ϕ, a depth-averaged total unit weight γ_t, and a regolith thickness that could fail, note as T, the expression for the Factor of Safety is as follows:

$$\mathrm{FS} = \frac{\tan\phi}{\tan\psi} + \frac{C}{\gamma_t T \sin\psi}. \tag{16}$$

If the value of FS < 1, the driving forces exceed the resisting forces, and the regolith materials are prone to failure.

A regolith particle on or beneath the surface of the regolith will be subject to the asteroid gravity, the centrifugal force, the overbearing pressure from the regolith above its location, and the contact forces of the surrounding media acting on the grain. These forces are closely related to the coordination number of the particle system and the loading history, making it difficult to provide their analytical expression. To derive the analytical expressions for these forces and thereby establish the fission failure conditions for regolith particles, we assume an orderly layered arrangement of particles within the regolith and neglect tangential contact forces between particles in the same layer. Consequently, the support forces acting on a particle are dominated by the contact forces from the particle layers directly above and below it. Based on the above assumptions, we can conveniently derive an analytical expression for the overbearing pressure caused by the weight of the regolith above the target particle (for detailed derivation, see Appendix B). Subsequently, the support force from the regolith below the particle can be analytically determined. When the support force from the particles beneath the

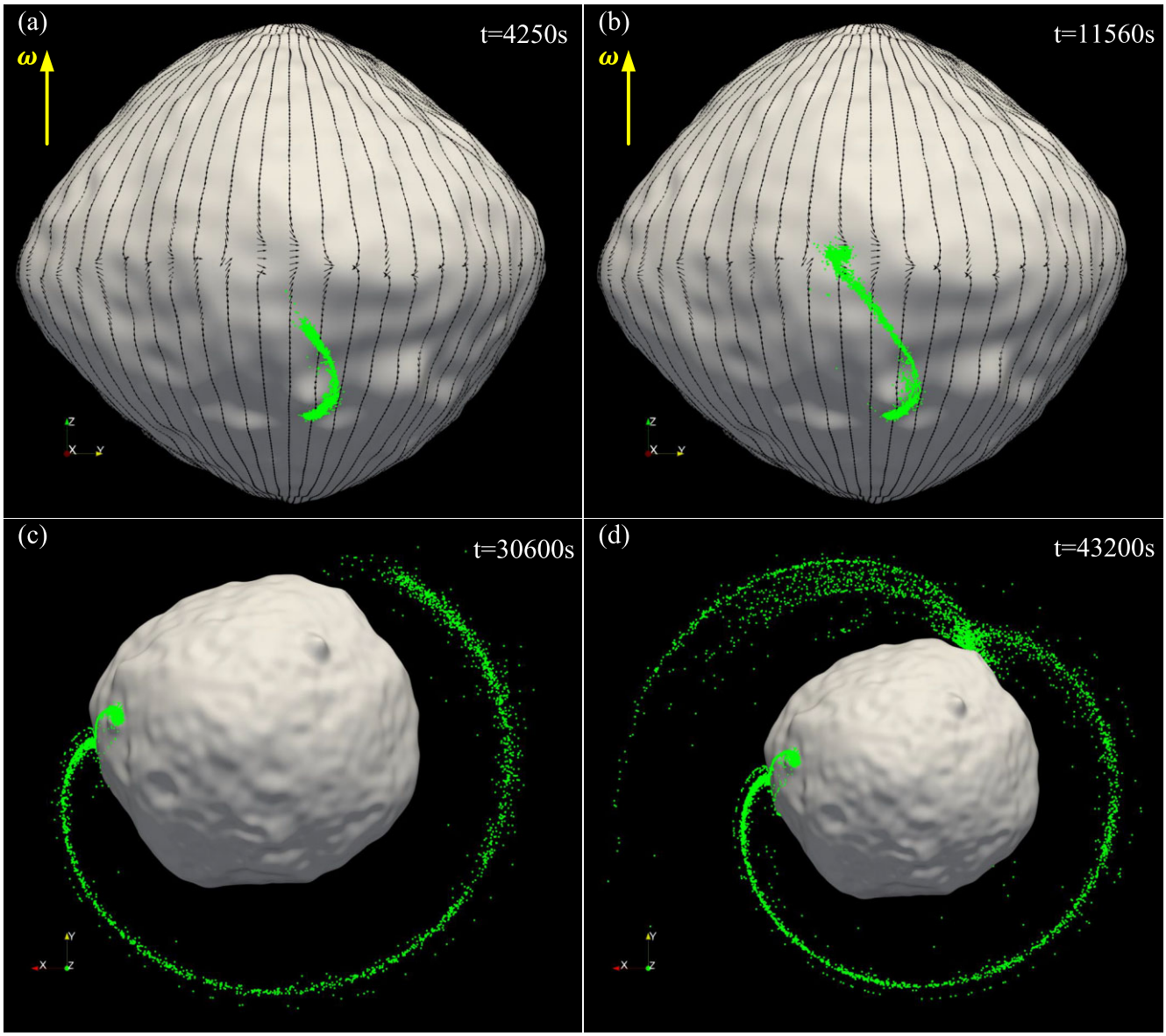

Figure 12. Snapshots of the evolution of sand-pile 1. Where the cohesive strength of the particles is 0 Pa and Bennu's spin period is 3.4 h. Each snapshot shows the distributions of particles at different simulated times. The black arrows in panels (a) and (b) indicate the tangential component of the resultant force resulting from centrifugal and gravitational forces, and the angular velocity direction is indicated by the yellow arrows in panels (a) and (b). The green dots represent regolith particles. The time cost of this 43 200 s simulation was 320 819 s. A movie of this evolution is available on the web version.

regolith particles is less than or equal to zero, the regolith undergoes fission failure. Thus, the fission failure condition of the regolith on the surface of the asteroids can be represented as:

$$(\boldsymbol{F}_{\mathrm{C}} + \frac{T}{2R} \cdot (\boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}})) \cdot \boldsymbol{n}_{\mathrm{f}} \geq 0, \tag{17}$$

where $\boldsymbol{F}_{\mathrm{g}}$, $\boldsymbol{F}_{\mathrm{ct}}$, and $\boldsymbol{F}_{\mathrm{C}}$ represent the gravitational force, centrifugal force, and cohesive force acting on a single particle, respectively, while T represents the thickness from the target particle to the regolith surface, as illustrated in Fig. B1; R is the radius of the regolith particle; The coefficient $\frac{T}{2R}$ describes the cumulative effect of the gravitational and centrifugal forces of the overlying layers of material at a specific depth under the surface of the regolith.

Equation (17) provides an approximate criterion for the fission failure of a single particle. However, our subsequent numerical experiments focus more on whether the entire sand-pile or most of it undergoes fission failure on the asteroid's surface, rather than the failure of an individual particle. To use equation (17) for selecting the spin period for numerical experiments on the fission failure of a sand-pile on the asteroid's surface, we assume that when most particles in contact with the asteroid's surface experience fission failure, the entire sand-pile will undergo large-scale failure (even though the upper or inner parts of the sand-pile may fail beforehand). Therefore, by setting the T value to the thickness corresponding to most particles in contact with the asteroid's surface, equation (17) can be used to evaluate the spin period required for the numerical experiments. In this paper, we set the value of T as the volume of particles in the sand-pile divided by the surface area occupied by the sand-pile on the asteroid.

Using formulas (16) and (17), we illustrate the distribution of failure modes for the regolith near sand-pile 1 and sand-pile 2. This analysis is conducted for the following parameters: $\beta = 0.5$, $R =$

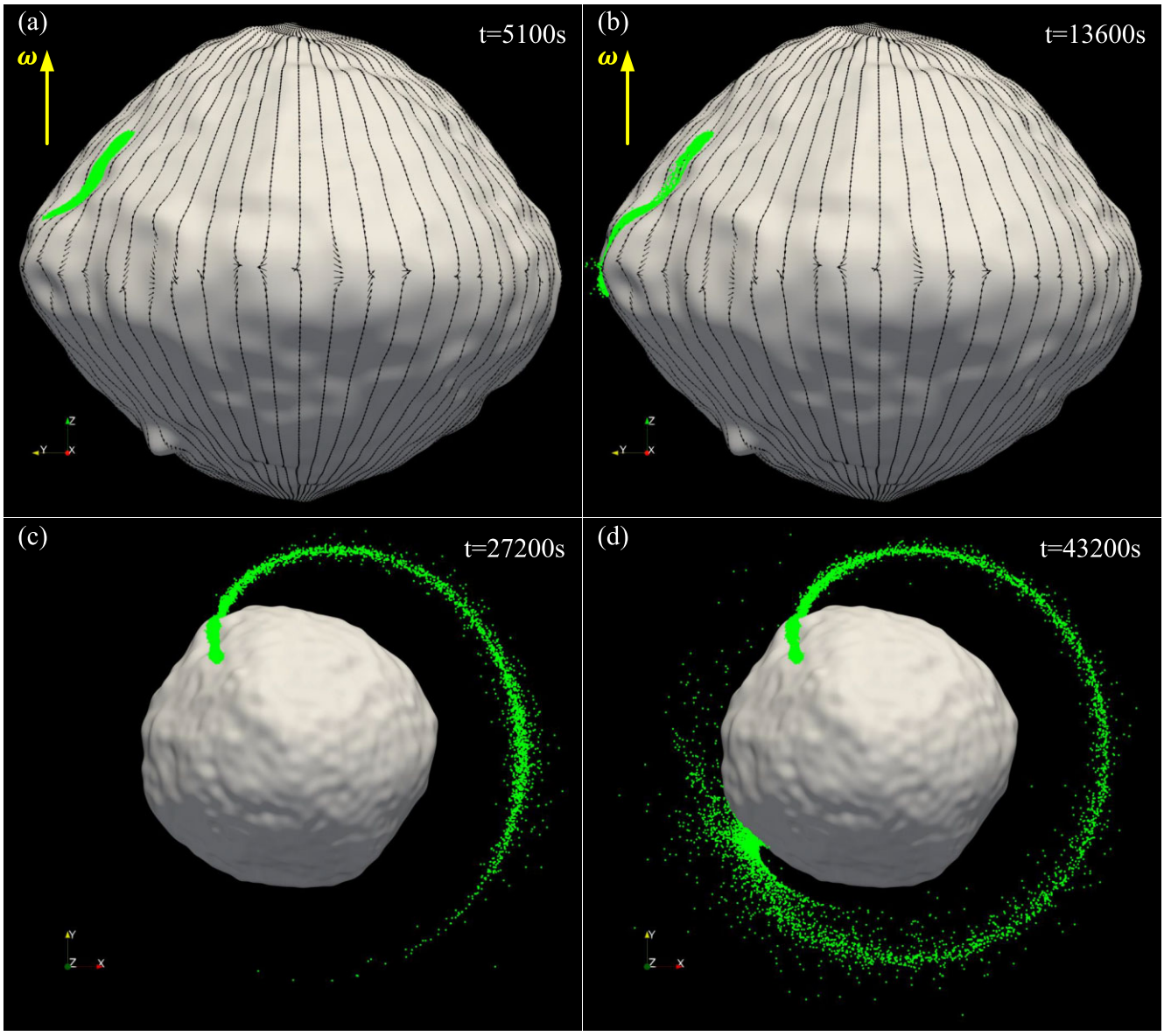

Figure 13. Snapshots of the evolution of sand-pile 2. Where the cohesive strength of the particles is 0 Pa and Bennu's spin period is 3.4 h. Each snapshot shows the distributions of particles at different simulated times. The black arrows in panels (a) and (b) indicate the tangential component of the resultant force resulting from centrifugal and gravitational forces, and the angular velocity direction is indicated by the yellow arrows in panels (a) and (b). The green dots represent regolith particles. The time cost of this 43 200 s simulation was 609 882 s. A movie of this evolution is available on the web version.

0.1 m, a friction angle of $\phi = 40^{\circ}$ (Barnouin et al. 2022), macroscopic cohesion strengths of $C = 0$ and $C = 1$ Pa, spin periods of 3.4 and 2.6 h, and regolith thicknesses of $T = 3$ m and $T = 5$ m for sandpiles 1 and 2, respectively. The corresponding distribution of failure modes is presented in Fig. 11.

Fig. 11 illustrates that when the macroscopic cohesion strength $C = 0$ Pa and the spin period of 3.4 h, for the given parameter set above, both sand-pile 1 and sand-pile 2 are situated in the vicinity of regions where landslide failures occur. When the macroscopic cohesion strength $C = 1$ Pa and the spin period of 2.6 h, for the same parameter set, both sand-pile 1 and sand-pile 2 are positioned near regions where fission failures occur. Consequently, combining the insights from Fig. 11, two parameter sets listed in Table 3 are used to simulate the movement process of sand-pile 1 and sand-pile 2 after failure on Bennu's surface, respectively. The normal spring constant $k_{\mathrm{N}} = 1.0103 \times 10^{4}\,\mathrm{N\,m^{-1}}$ and the time-step $\Delta t = 0.0017$ s are set to keep particle overlaps smaller than 1 per cent of the minimum particle radius and integrate the particle contacts precisely. All simulations are performed on a desktop computer with a Ryzen 5950X 16-core processor at 3.4 GHz and an NVIDIA GPU card GeForce RTX 2080.

It is important to emphasize that equations (16) and (17) are approximate formulas designed to provide a rough estimation of the failure conditions. Due to the intricate interaction processes involving both interparticle interactions and interactions between particles and the asteroid surface, accurately defining the failure criterion of regolith material on asteroid surfaces requires precise modelling of granular material failure processes (Zhang et al. 2018).

The subsequent evolution of sand pile 1 and sand pile 2 posterior to the failure with a cohesive strength of 0 Pa and a spin period of 3.4 h, is

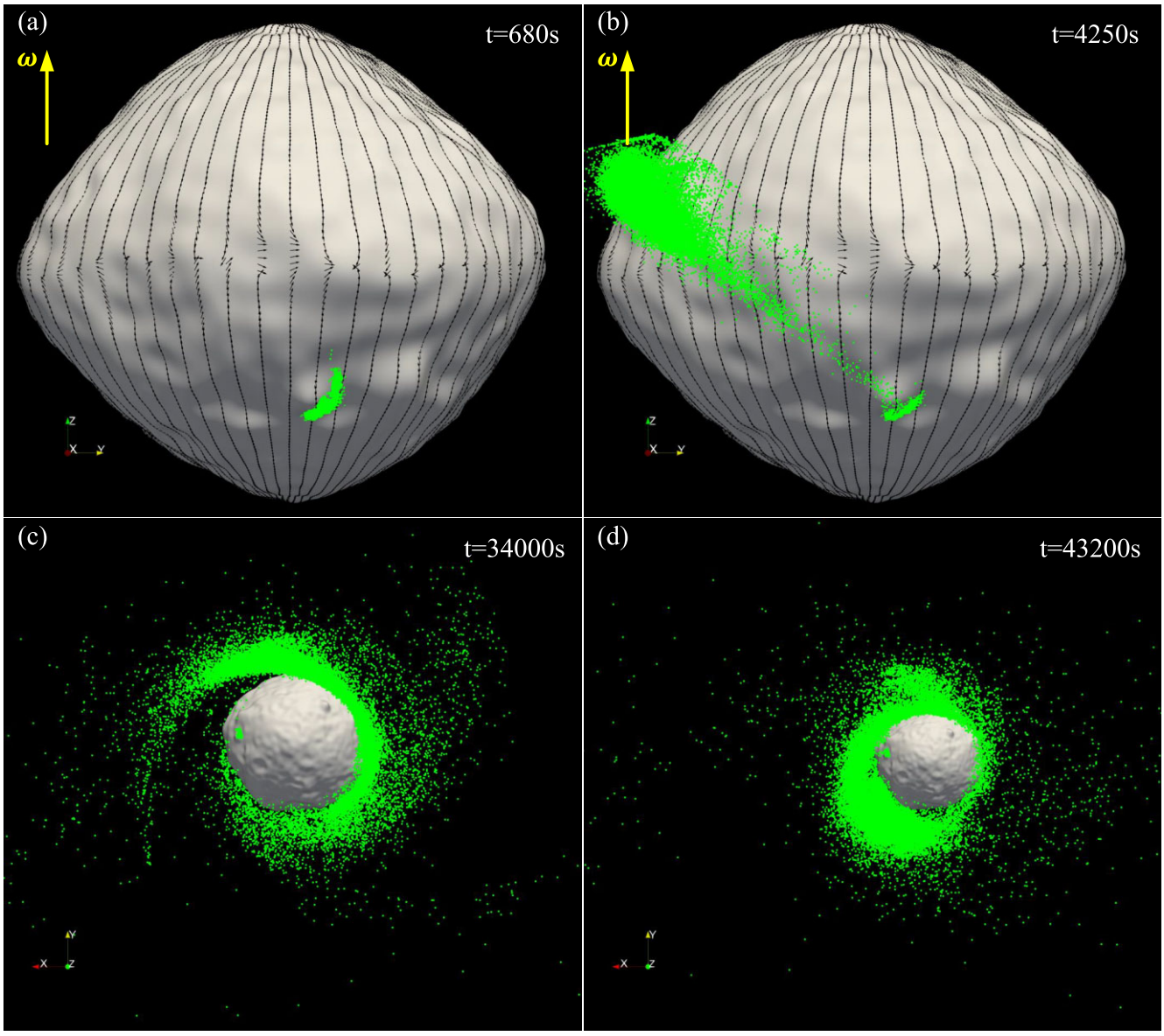

Figure 14. Snapshots of the evolution of sand-pile 1. Where the cohesive strength of the particles is 1 Pa and Bennu's spin period is 2.6 h. Each snapshot shows the distributions of particles at different simulated times. The black arrows in panels (a) and (b) indicate the tangential component of the resultant force resulting from centrifugal and gravitational forces, and the angular velocity direction is indicated by the yellow arrows in panels (a) and (b). The green dots represent regolith particles. The time cost of this 43 200 s simulation was 301 967 s. A movie of this evolution is available on the web version.

illustrated in Figs 12 and 13, respectively. As expected, during a landslide failure, the particles initiate a flow from Bennu's mid-latitudes toward the equatorial region, subsequently shedding from the equator and entering orbit. This result aligns with the insights derived from the theoretical analysis by Scheeres (2015). Once these particles enter orbit, they are trapped in the potential well of the asteroid for a long-term due to their low-relative velocities, and the subsequent evolution depends on the gravitational field of the asteroid and the interaction between the particles. Moreover, as depicted in Figs 12(a) and (b), when particles flow from Bennu's mid-latitude region towards the equatorial region, their flow direction significantly deviates from the tangential component of the resultant force resulting from centrifugal and gravitational forces. This deviation is attributed to the influence of the Coriolis effect. Figs 13(a) and (b) clearly show that particles from sand-pile 2 flowed into the underlying crater and overflowed as the mass accumulated. This result indicates that the migration of regolith material from mid-latitudes to the equator effectively fills craters along its path, resulting in the degradation of craters in the low-latitude regions of fast-spinning asteroids. This conclusion is consistent with recent observations of Didymos from the Double Asteroid Redirection Test (DART) mission.

The evolution of sand-pile 1 and sand-pile 2 posterior to the failure is presented in Figs 14 and 15, respectively, where the cohesive strength is 1 Pa and the spin period is 2.6 h. Different from the case of a period of 3.4 h, particles of sand-pile 1 entered the orbit from near the initial position shortly after the simulation began. For sand-pile 2, as indicated in Fig. 11, it was initially situated at a slightly greater distance from the fission failure region compared to sand-pile 1. As a result, sand-pile 2 went through a brief sliding phase before particles eventually flowed into the fission failure region (as shown in

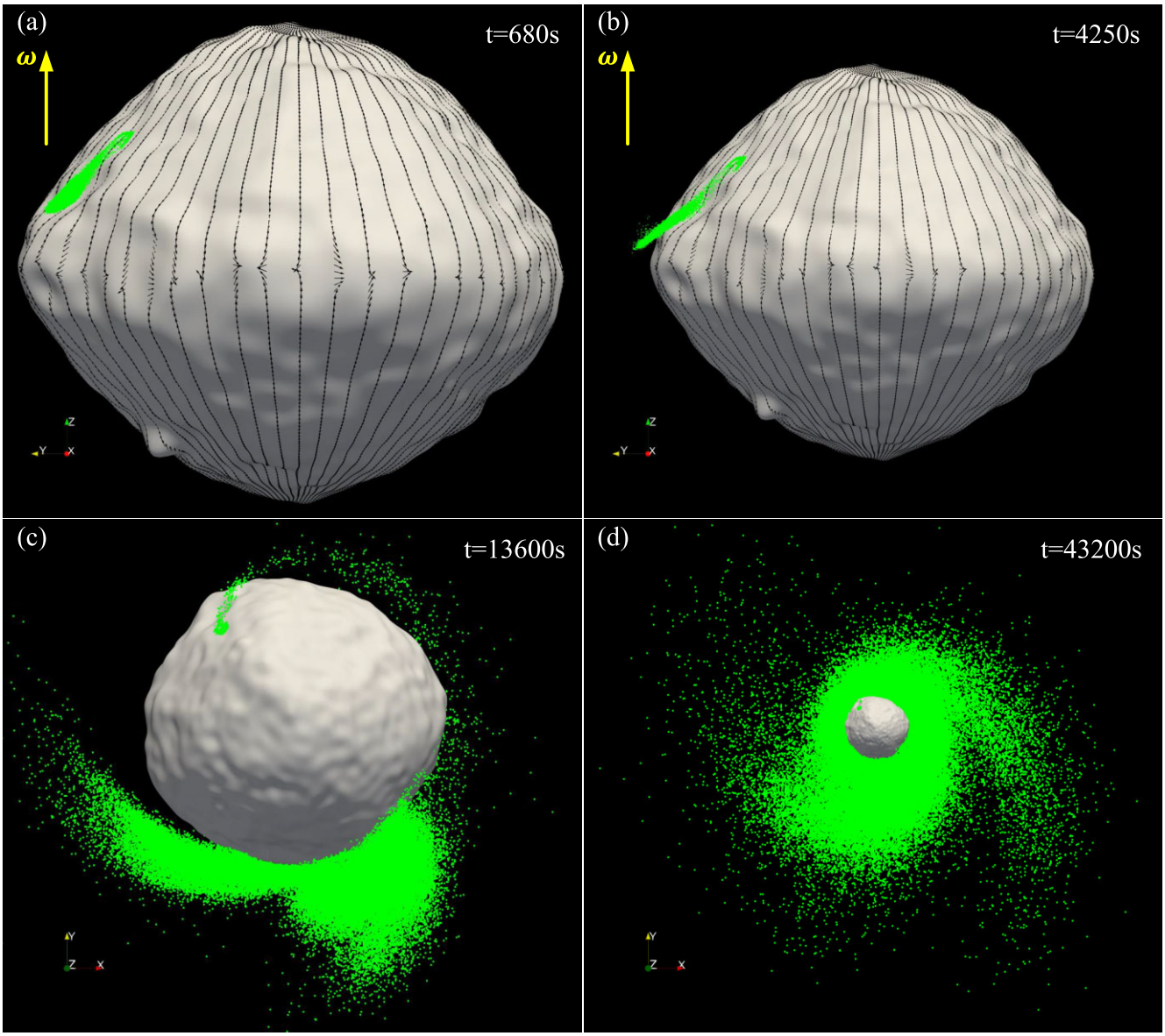

Figure 15. Snapshots of the evolution of sand-pile 2. Where the cohesive strength of the particles is 1 Pa and Bennu's spin period is 2.6 h. Each snapshot shows the distributions of particles at different simulated times. The black arrows in panels (a) and (b) indicate the tangential component of the resultant force resulting from centrifugal and gravitational forces, and the angular velocity direction is indicated by the yellow arrows in panels (a) and (b). The green dots represent regolith particles. The time cost of this 43 200 s simulation was 478 129 s. A movie of this evolution is available on the web version.

Fig. 15b). Subsequently, it rapidly underwent fission failure and shed from the asteroid's surface to enter orbit. The subsequent evolution of sandpiles 1 and 2 is illustrated in Figs 14 and 15. Furthermore, upon comparing Figs 12 and 14, it is evident that the scale of the mass migration enlarges with an increase in the rotation rate.

To further validate the accuracy of the code, we illustrate the variations in Bennu's spin rate, the mechanical energy, and the conservation quantities of the system during the evolution of sand-pile 1 on the asteroid's surface, where the cohesion strength is 1 Pa and the initial spin period is 2.6 h (the case as illustrated in Fig. 14). This result is shown in Fig. 16 and illustrates the common tendency among all four simulations. As particles from sand-pile 1 gradually enter orbits around the asteroid and expand to a wide region, as shown in Fig. 14, the spin rate of the asteroid decreases gradually to conserve angular momentum in the system consisting of both the particles and the asteroid, as shown in Figs 16(a) and (c). The mechanical energy of the system is gradually dissipated by the friction and collision behaviour between the particles and the surface of the asteroid, as shown in Fig. 16(b). Figs 16(c) and (d) show the conservation of angular momentum and linear momentum of the system in three component directions, respectively. The total angular and linear momentum of the system comprises the angular and linear momentum of all particles and the asteroid. As shown in Figs 16(c) and (d), the value of the angular momentum and linear momentum of the particles and the asteroid consistently maintain a complementary relationship during the evolution. Therefore, the system consisted of the particles and the asteroid conserves both linear momentum and angular momentum, which proves that the computational results of our code are reliable.

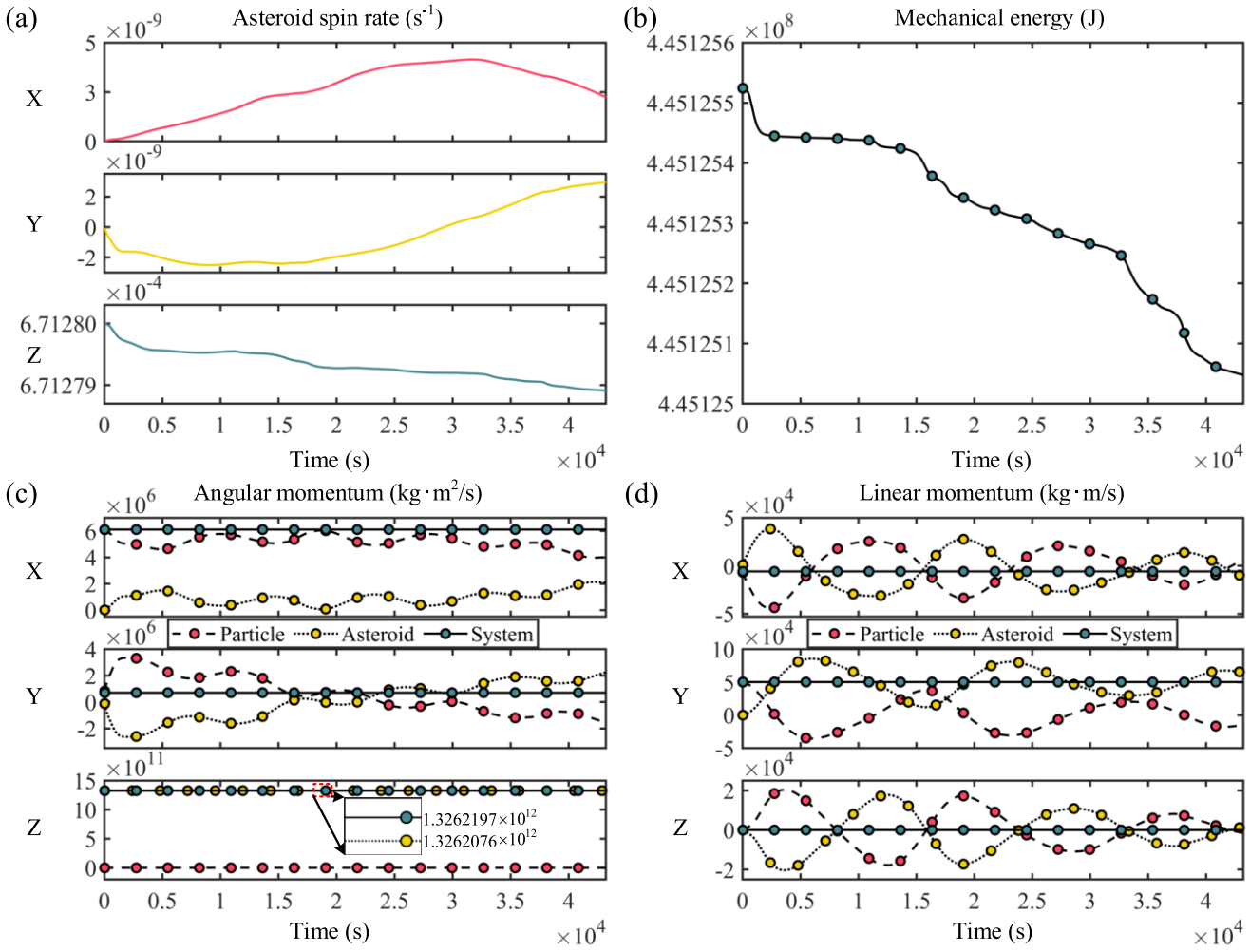

Figure 16. Variation of the asteroid's spin rate, the mechanical energy, and the conservation quantities of the system during the evolution of sand-pile 1 on the asteroid's surface. (a) Spin rate of the asteroid with respect to the body-fixed frame (*X*-, *Y*-, *Z*-direction). (b) Mechanical energy of the system. (c) Variation of angular momentum of particles, the asteroid, and the system with respect to the inertial frame origin in the inertial frame (*X*-, *Y*-, *Z*-direction). The detail enlarged view of the component of angular momentum in the *Z*-direction for the asteroid and the system is illustrated in the inset at the bottom of panel (c). (d) Variation of linear momentum of particles, the asteroid and the system with respect to the origin of the inertia frame in the inertial frame (*X*-, *Y*-, *Z*-direction). Where the system is composed of the particles and the asteroid.

4 CONCLUSION AND DISCUSSION

In this paper, we report progress on numerical techniques to simulate the asteroid's surface geological processes. A general strategy is proposed to track the scale-span evolution of the asteroid regolith materials with high-resolution surface topography and actual sizes of regolith grains. Using this strategy, a specific DEM code is developed for processing the asteroid surface activities (landslides, shedding, low-energy bombardments, etc.) under the mechanical environment of the asteroid surface, including the micro and irregular gravity, the interparticle and particle-surface interactions, and the coupled dynamics between the asteroid and the moving regolith mass. To conquer the memory exponential expansion caused by the enlarged computational domain, we improved the conventional cell-linked list method, which is compatible with simulations of variable spatial scale. In order to process long-range landslides of the regolith materials over the surface with high fidelity, we propose an efficient algorithm for detecting contacts between particles and the asteroid surface. This algorithm avoided the global traversal process when identifying contacts between particles and the triangles constituting the asteroid surface, and enhanced the computational efficiency in the case of high-resolution surface topography, significantly. Besides, we proposed a CUDA-accelerated parallel algorithm for the polyhedral method and leveraged the advantages of CPU–GPU heterogeneous computing to overlap the computations of the long-range and short-range interactions, thereby offsetting the computational load of the polyhedral gravity field. Using benchmark tests, we confirm that our code accurately models the mass movement processes of regolith materials on the asteroid surface. These processes encompass landslide failure, mass shedding, and orbital motion, all at a realistic scale and with resolutions that match actual particle sizes and surface topography.

The technique proposed in this paper primarily focuses on continuous mass movement on the surface of asteroids and the subsequent evolution, aiming to narrow the gap between numerical simulations and the actual surface environment of asteroids. The simulation presented in Section 3.2 highlights some key differences between our study and recent papers on modelling the mass movement on asteroids using the discrete element method. The techniques reported

in recent papers have almost exclusively used discrete particles (whether spherical or polyhedral) to construct the overall structure of asteroids. It facilitates the coding but also causes limitations on the code capacity: the particle size ratio is strictly limited in the discrete element method, and the huge computational cost makes it difficult to model the realistic particle sizes and surface topography resolution. For instance, Agrusa et al. (2024) recently employed the DEM to model the process of the regolith shedding from the asteroid surface and subsequently accumulating to generate small satellites. Compared to their model, our approach demonstrates advantages in modelling mass movement on the surface of asteroids. Their simulations employ particle sizes between 4 and 16 m, leading to a surface topography resolution exceeding 4 m, as these particles directly form the surface of the asteroid. In our simulations, particle sizes are 0.1 m, and the surface topography is at the submeter level. Therefore, our approach allows for modelling particle movement on the asteroid's surface with a higher resolution in terms of both particle size and surface topography resolution. Besides, our approach can model the gravitational field on the surface of the asteroid more accurately than the mascon approach. However, using a rigid polyhedron to model the asteroid's shape in our approach also constrains its capability to simulate the overall deformation and internal structural failure processes of the asteroid, even though it achieves high-resolution surface topography.

The proposed strategy and code have great potential in both theoretical and application studies. For the scenario of planetary research, it serves as a powerful tool for high-resolution investigating the surface activities on asteroids, such as the possible effects of Apophis' 2029 close encounter with Earth on Apophis' surface and the shedding of the regolith on the surface of fast-rotating asteroids, which could offer insights into fine geological processes that shape the landscapes of the asteroid. For the engineering design in asteroid surface exploration missions, it provides a reliable way to model the tricky maneuvers of the spacecraft landing on the asteroid surface. For instance, the proposed method can be used to predict the global cruise trajectory of a hopping lander with high fidelity, offering robust theoretical support for the design of asteroid surface exploration missions.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referee for the carefully reading and insightful comments to make this better. This work was supported by the National Natural Science Foundation of China under Grant 12272018.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Agrusa H. F. et al., 2024, Planet. Sci. J., 5, 54
Allen M. P., Tildesley D. J., 2017, Computer Simulation of Liquids, 2 edn. Oxford University Press, Oxford
Asmar B., Langston P., Matchett A., Walters J., 2002, Comput. Chem. Eng., 26, 785
Banik D., Gaurav K., Sharma I., 2022, Proc. R. Soc., 478, 20210972
Baoyin H.-X., Chen Y., Li J.-F., 2010, Res. Astron. Astrophys., 10, 587
Barnes J., Hut P., 1986, Nature, 324, 446
Barnouin O. S. et al., 2022, J. Geophys. Res.: Planets, 127, e2021JE006927
Benson C. J., Scheeres D. J., Brozović M., Chesley S. R., Pravec P., Scheirich P., 2023, Icarus, 390, 115324
Bottke W. F. et al., 2020, J. Geophys. Res.: Planets, 125, e2019JE006282
Cai R., Xu L., Zheng J., Zhao Y., 2018, Powder Technol., 340, 321
Cao J., Wang M., 2023, IEEE Access, 11, 44696
Cheng B., Yu Y., Baoyin H., 2019, MNRAS, 485, 3088
Cheng B., Yu Y., Asphaug E., Michel P., Richardson D. C., Hirabayashi M., Yoshikawa M., Baoyin H., 2020, Nat. Astron., 5, 134
Cheng B., Asphaug E., Ballouz R.-L., Yu Y., Baoyin H., 2022, Planet. Sci. J., 3, 249
Cho Y. et al., 2021, J. Geophys. Res.: Planets, 126, e2020JE006572
Cleary P. W., 2000, Powder Technol., 109, 83
DeMartini J. V., Richardson D. C., Barnouin O. S., Schmerr N. C., Plescia J. B., Scheirich P., Pravec P., 2019, Icarus, 328, 93
DellaGiustina D. N. et al., 2019, Nat. Astron., 3, 341
Di Renzo A., Di Maio F. P., 2004, Chem. Eng. Sci., 59, 525
Ferrari F., Tasora A., Masarati P., Lavagna M., 2017, Multibody Syst. Dyn., 39, 3
Ferrari F., Lavagna M., Blazquez E., 2020, MNRAS, 492, 749
Fujiwara A. et al., 2006, Science, 312, 1330
Gao Y., Cheng B., Yu Y., 2022, Ap&SS, 367, 84
Geissler P., Petit J.-M., Durda D. D., Greenberg R., Bottke W., Nolan M., Moore J., 1996, Icarus, 120, 140
Hainaut O. R. et al., 2014, A&A, 563, A75
Iverson R. M., Reid M. E., LaHusen R. G., 1997, Ann. Rev. Earth Planet. Sci., 25, 85
Jawin E. R. et al., 2020, J. Geophys. Res.: Planets, 125
Kaasalainen M., Ďurech J., Warner B. D., Krugly Y. N., Gaftonyuk N. M., 2007, Nature, 446, 420
Kim Y., DeMartini J. V., Richardson D. C., Hirabayashi M., 2023, MNRAS, 520, 3405
Lauretta D. S. et al., 2019, Science, 366, eaay3544
Lowry S. C. et al., 2014, A&A, 562, A48
Marohnic J. C., DeMartini J. V., Richardson D. C., Zhang Y., Walsh K. J., 2023, Planet. Sci. J., 4, 245
McMahon J. W. et al., 2020, J. Geophys. Res.: Planets, 125, e2019JE006229
Mishra B., 2003, Int. J. Miner. Process., 71, 73
Morbidelli A., Raymond S. N., 2016, J. Geophys. Res.: Planets, 121, 1962
Moreno F., Licandro J., 2014, ApJ, 781, 118
Nagurka M., Huang S., 2004, in Proc. 2004 Am. Control Conf. Vol. 1. IEEE, Boston, MA, USA, p. 499
Park R. S., Werner R. A., Bhaskaran S., 2010, J. Guid. Control Dyn., 33, 212
Sánchez P., Scheeres D. J., 2011, ApJ, 727, 120
Sánchez P., Scheeres D. J., 2014, Meteorit. Planet. Sci., 49, 788
Sánchez P., Scheeres D. J., 2016, Icarus, 271, 453
Sánchez P., Scheeres D. J., 2020, Icarus, 338, 113443
Sánchez P., Renouf M., Azéma E., Mozul R., Dubois F., 2021, Icarus, 363, 114441
Scheeres D., 1994, Icarus, 110, 225
Scheeres D., 2015, Icarus, 247, 1
Schwartz S. R., Richardson D. C., Michel P., 2012, Granular Matter, 14, 363
Song Z., Yu Y., Cheng B., Lv J., Baoyin H., 2023, A&A, 672, A172
Strobl S., Formella A., Pöschel T., 2016, J. Comput. Phys., 311, 158
Tachibana S. et al., 2022, Science, 375, 1011
Takahashi Y., Scheeres D. J., 2014, Celest. Mech. Dyn. Astron., 119, 169
Tancredi G., Maciel A., Heredia L., Richeri P., Nesmachnow S., 2012, MNRAS, 420, 3368
Welling U., Germano G., 2011, Comput. Phys. Commun., 182, 611
Wen T., Zeng X., 2023, AJ, 166, 194
Werner R. A., Scheeres D. J., 1997, Celest. Mech. Dyn. Astron., 65, 313
Williams J. R., O'Connor R., 1999, Arch. Comput. Methods Eng., 6, 279
Yao Z., Wang J.-S., Liu G.-R., Cheng M., 2004, Comput. Phys. Commun., 161, 27
Yu Y., Richardson D. C., Michel P., Schwartz S. R., Ballouz R.-L., 2014, Icarus, 242, 82
Yu Y., Michel P., Hirabayashi M., Schwartz S. R., Zhang Y., Richardson D. C., Liu X., 2018, AJ, 156, 59

Yu Y., Michel P., Hirabayashi M., Richardson D. C., 2019, MNRAS, 484, 1057
Zhang Y. et al., 2017, Icarus, 294, 98
Zhang Y., Richardson D. C., Barnouin O. S., Michel P., Schwartz S. R., Ballouz R.-L., 2018, ApJ, 857, 15
Zhang Y., Michel P., Richardson D. C., Barnouin O. S., Agrusa H. F., Tsiganis K., Manzoni C., May B. H., 2021, Icarus, 362, 114433
Zhou Y., Wright B., Yang R., Xu B., Yu A., 1999, Physica A: Stat. Mech. Appl., 269, 536
Zhou Q., Xu W.-J., Liu G.-Y., 2021, Comput. Geotech., 138, 104371

APPENDIX A:

Algorithm 1 Polyhedral method based on CUDA parallel acceleration

Require: Polyhedral geometric information P about the asteroid and particle position vector $\boldsymbol{r}$
Ensure: $\boldsymbol{F}_g$

Allocate shared memory for a float precision array $\boldsymbol{p}_{\mathrm{v}}$ of size 512×3;
Allocate shared memory for a float precision array $\mathbf{n}_f$ of size 512×3;
Allocate shared memory for a float precision array $\mathbf{n}_e^A$ of size 512×3;
Allocate shared memory for a float precision array $\mathbf{n}_e^B$ of size 512×3;
Allocate shared memory for a float type array E_l of size 512;
Allocate shared memory for a int type array F_{num} of size 512×4;
Allocate local memory for $\boldsymbol{x}_{\mathrm{v}}$ and x_{v};
for $i = 0$ to $VertexTile$ **do**
  $j = i * blockDim.x + threadIdx.x$;
  if $j < P.NumVertex$ **then**
    $\boldsymbol{p}_{\mathrm{v}}[threadIdx.x] = P.\boldsymbol{Vert}[j]$; ▷ Copy the vertex position from host memory to the device;
  end if
  Synchronize threads;
  for $k = 0$ to $blockDim.x$ **do**
    $j = i * blockDim.x + k$;
    if $j < P.NumVertex$ **then**
      $\boldsymbol{x}_{\mathrm{v}}[j] = \boldsymbol{r} - \boldsymbol{p}_{\mathrm{v}}[k]$ ▷ Equation 8
      $x_{\mathrm{v}} = \|\boldsymbol{x}_{\mathrm{v}}\|$
    end if
  end for
end for
$\boldsymbol{F} = 0$
for $i = 0$ to $FaceTile$ **do**
  $j = i * blockDim.x + threadIdx.x$;
  $Synchronize \quad threads$;
  if $j < P.NumFace$ **then**
    $\mathbf{n}_f[threadIdx.x] = P.\mathbf{n}_f[j]$;
    $F_{num}[threadIdx.x] = P.F_{num}[j]$; ▷ Copy the face information from host memory to the device;
  end if
  $Synchronize \quad threads$;
  for $k = 0$ to $blockDim.x$ **do**
    $j = i * blockDim.x + k$;
    if $j < P.NumFace$ **then**
      $\mathrm{v}_i = F_{num}[k][0]; \mathrm{v}_j = F_{num}[k][1]; \mathrm{v}_k = F_{num}[k][2]$;
      $tmp1 = \boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_i] \cdot (\boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_j] \times \boldsymbol{x}_{\mathbf{v}}[\mathrm{v}_k])$;
      $tmp2 = x_{\mathrm{v}}[\mathrm{v}_i]x_{\mathrm{v}}[\mathrm{v}_j]x_{\mathrm{v}}[\mathrm{v}_k] + x_{\mathrm{v}}[\mathrm{v}_i](\boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_j] \cdot \boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_k]) + x_{\mathrm{v}}[\mathrm{v}_k](\boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_i] \cdot \boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_j])$;
      $\omega_f = 2\arctan(tmp1/tmp2)$; ▷ Equation 12
      $\boldsymbol{F}_f = \mathbf{n}_f[k] \otimes \mathbf{n}_f[k]$; ▷ Equation 10
      $\boldsymbol{F}_f = \omega_f \boldsymbol{F}_f \cdot \boldsymbol{x}_{\mathrm{v}}[\mathrm{v}_i]$;
      $\boldsymbol{F} += \boldsymbol{F}_f$;
    end if
  end for
end for
$\boldsymbol{E} = 0$
for $i = 0$ to $EdgeTile$ **do**
  $j = i * blockDim.x + threadIdx.x$;
  $Synchronize \quad threads$;
  if $j < P.NumEdge$ **then**
    $\mathbf{n}_e^A[threadIdx.x] = P.\mathbf{n}_e^A[j]$;
    $\mathbf{n}_e^B[threadIdx.x] = P.\mathbf{n}_e^B[j]$;
    $E_l[threadIdx.x] = P.E_l[j]$; ▷ Copy the edge information from host memory to the device;
  end if
  Synchronize threads;
  for $k = 0$ to $blockDim.x$ **do**
    $j = i * blockDim.x + k$;
    if $j < P.NumEdge$ **then**
      $e_i = f_{num}[k][0]; e_j = f_{num}[k][1]; e_k = f_{num}[k][2]; e_l = f_{num}[k][3]$; ▷ e_i,e_j represent the vertex number of edge k, e_k,e_l represent the face number of edge k
      $\boldsymbol{E_e} = \mathbf{n}_f[e_k] \otimes \mathbf{n}_e^A[k] + \mathbf{n}_f[e_l] \otimes \mathbf{n}_e^B[k]$; ▷ Equation 9
      $L_e = \ln[(x_{\mathrm{v}}[e_i] + x_{\mathrm{v}}[e_j] + E_l[k])/(x_{\mathrm{v}}[e_i] + x_{\mathrm{v}}[e_j] - E_l[k])]$; ▷ Equation 11
      $\boldsymbol{E} += \boldsymbol{E_e} \cdot \boldsymbol{x}_{\mathrm{v}}[e_i] * L_e$;
    end if
  end for
end for
$\boldsymbol{F}_g = -G\rho(\boldsymbol{E} - \boldsymbol{F})$; ▷ Equation 7

APPENDIX B: THE DETAILED DERIVATION OF EQUATION (17)

Based on the simplified assumptions mentioned in Section 3.2.2, the forces acting on the regolith particle on or beneath the surface of the regolith are illustrated in Fig. B1. The surface regolith particle is supported by the underlying layer, and when it is in a static situation, the following condition applies:

$$\boldsymbol{F}_{\mathrm{b}} + \boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}} = 0. \tag{B1}$$

The support force $\boldsymbol{F}_{\mathrm{b}}$ exerted by the underlying layer consists of both the contact force $\boldsymbol{F}_{\mathrm{J}}$ and the cohesive force $\boldsymbol{F}_{\mathrm{C}}$, therefore:

$$\boldsymbol{F}_{\mathrm{J}} + \boldsymbol{F}_{\mathrm{C}} + \boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}} = 0. \tag{B2}$$

When the support force from the underlying regolith layer is less than or equal to zero, the particle undergoes fission failure. Therefore, the fission failure condition of the surface regolith particle can be represented as:

$$(\boldsymbol{F}_{\mathrm{C}} + (\boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}})) \cdot \boldsymbol{n}_{\mathrm{f}} \geq 0. \tag{B3}$$

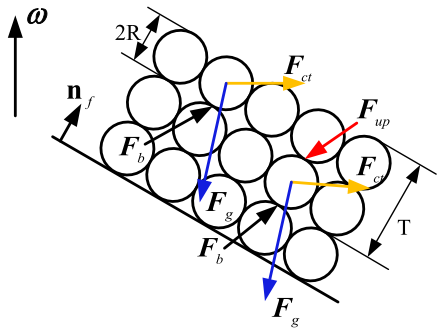

Figure B1. Forces acting on the surface and internal regolith particles. $\boldsymbol{F}_{\mathrm{g}}$ represents the gravitational force exerted on the particle by the asteroid; $\boldsymbol{F}_{\mathrm{ct}}$ is the centrifugal force acting on the particle; $\boldsymbol{F}_{\mathrm{b}}$ denotes the support force from the underlying layer (including the contact force and the cohesive force); $\boldsymbol{F}_{\mathrm{up}}$ denotes the pressure caused by the weight of the overlying layers of material at a specific depth under the surface of the regolith; $\boldsymbol{n}_{\mathrm{f}}$ represents the local normal direction; ω indicates the angular velocity vector of the asteroid; R is the particle radius, and T is the depth of the regolith that could fail.

For the internal regolith particle, as shown in Fig. B1, when it is in equilibrium, it satisfies the following equation:

$$\boldsymbol{F}_{\mathrm{b}} + \boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}} + \boldsymbol{F}_{\mathrm{up}} = 0. \tag{B4}$$

The pressure caused by the weight of the regolith above the target particle $\boldsymbol{F}_{\mathrm{up}}$ can be determined using Newton's third law, combined with equations B1 and B4, using a recursive approach. Specifically, for a regolith particle located at a thickness of T, when it is in equilibrium, the pressure force exerted by the upper regolith layer can be described by the following equation:

$$\boldsymbol{F}_{\mathrm{up}} = (\frac{T}{2R} - 1)(\boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}}). \tag{B5}$$

The support force $\boldsymbol{F}_{\mathrm{b}}$ exerted by the underlying layer is:

$$\boldsymbol{F}_{\mathrm{b}} = -\frac{T}{2R}(\boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}}). \tag{B6}$$

Therefore, the condition for fission failure of the internal regolith is:

$$(\boldsymbol{F}_{\mathrm{C}} + \frac{T}{2R} \cdot (\boldsymbol{F}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{ct}})) \cdot \boldsymbol{n}_{\mathrm{f}} \geq 0. \tag{B7}$$

This paper has been typeset from a TEX/LATEX file prepared by the author.