



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Seyedan, Seyedmohammadjavad; Sołowski, Wojciech Tomasz

From solid to disconnected state and back: Continuum modelling of granular flows using material point method

Published in: Computers and Structures

DOI: 10.1016/j.compstruc.2021.106545

Published: 15/07/2021

Document Version Publisher's PDF, also known as Version of record

Published under the following license: CC BY

Please cite the original version:

Seyedan, S., & Sołowski, W. T. (2021). From solid to disconnected state and back: Continuum modelling of granular flows using material point method. *Computers and Structures*, *251*, 01-20. Article 106545. https://doi.org/10.1016/j.compstruc.2021.106545

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Computers and Structures 251 (2021) 106545

Contents lists available at ScienceDirect

Computers and Structures

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

From solid to disconnected state and back: Continuum modelling of granular flows using material point method

Seyedmohammadjavad Seyedan*, Wojciech T. Sołowski

Department of Civil Engineering, Aalto University, Espoo, Finland

ARTICLE INFO

Article history: Received 29 January 2021 Accepted 17 March 2021

Keywords: Granular flow Continuum mechanics Material point method

ABSTRACT

This paper introduces Granular algorithm enhancing Material Point Method (MPM) that allows for simulation of granular flows in the quasi-static state, the moderate flow state and the disconnected flow state. The paper first shows the shortcomings of MPM algorithms in modelling the different states of granular flows. Next, it proposes Granular MPM, an enhancement that explicitly introduces the different states of granular flow into MPM and defines the rules for the transition between those states. Subsequently, the paper gives the algorithm and implementation for Granular MPM. The provided algorithm can enhance the common versions of MPM, including original MPM, Generalised Interpolation Material Point and Convected Particle Domain Interpolation. Finally, the paper evaluates Granular MPM and compares it with other available formulation based on: (i) an examination of the behaviour of granular points on a slope, (ii) a simulation of a granular flow over two disconnected inclined surfaces, (iii) a simulation s are significantly more realistic when compared to the results obtained by other available MPM formulations. The results also indicate that Granular MPM allows for more accurate replication of stated state flows and reduces the grid dependency of MPM when modelling the disconnected flow state, as the initial contact is independent from the grid size.

© 2021 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Granular flows are common in industry and nature. Accurate numerical modelling of granular flows can help, among others, in design of silo, chute, protective barriers against avalanches and penetration problems involving granular materials. Yet, it is challenging to create an efficient numerical method for modelling of the granular flows due to large or extreme deformations of the material and changes in its constitutive behaviour. For example, in a silo discharge a granular material experiences solid-like, flow-like and disconnected behaviour simultaneously.

During shearing, grains of a granular material move. If the grains are in continuous contact during the movement, they interact based on friction, leading to a frictional interaction mechanism. Otherwise, the grains may shortly collide during shearing, leading to a collisional interaction mechanism [1]. Overall, there are four possible states of granular material during shearing, linked to four types of behaviour [1-4]:

- (i) Quasi-static state: A quasi static state begins with the initial shearing of a granular body and continues up to the critical state, where no further change of volume occurs upon shearing. In the quasi-static state, the grains are densely packed and continuously interact with each other. The frictional interactions dominate, while the collisional interactions are negligible.
- (ii) Moderate granular flow: A moderate flow occurs when the shear rate is high enough to create a dependency between the shear rate and the shear stress. Typically, flows inside a silo and in a deep chute are of this category. During the moderate flow, both the frictional and the collisional interactions take place. The collisional interactions lead to energy dissipation (i.e. heat and sound), but no bounce back happens (zero restitution assumption) and the grains remain densely packed. The moderate flow is a steady condition characterised by moderate level of the kinetic energy. The grains cannot push each other away but cannot get very close to each other either. Therefore, the change in the packing fraction, which is percentage of space filled with grains, is small and density of the granular material remains close to the density in the critical state. In the higher flow rates, a





An reventored scored Computers & Structures Edité - Binder - Binder - Marijekyekes

^{*} Corresponding author at: Department of Civil Engineering, Aalto University, 02150 Espoo, Finland.

E-mail address: seyedmohammadjavad.seyedan@aalto.fi (S. Seyedan).

shearing dilation happens in the flow and reduces the packing fraction of the grains to some extent.

- (iii) Collisional state: A granular flow is in the collisional state when the shear rate of the flow is very high. In the collisional flows, the frictional interactions are rare while many collisional interactions occur. These collisional interactions result in energy dissipation as heat, sound and bounce back (nullifying zero restitution assumption). Therefore, the material experiences a significant density reduction and very fast deformations, leading to a gas like behaviour.
- (iv) Disconnected state: A granular flow is in the disconnected state when the grains lose the frictional contacts while the shear rate is not high, and the collisional interactions are not frequent enough to create the collisional flow. The Disconnected state typically occur when a sudden change happens at the flow environment - for example, when grains moves out of a silo orifice, they enter the disconnected state.

The granular material may either be modelled as a continuum, or as a body of separated grains. Considering the complexity of grain shapes and sizes, the latter option could be complicated and is not yet suitable for large problems. As such, this paper aims to use methods based on continuum mechanics.

The continuum mechanics calculations need to account for: (i) large deformations and displacement, (ii) the changes in constitutive behaviour of materials in different states and (iii) the modelling of low density / disconnected granular materials. Due to those requirements, some of the common numerical methods based on the continuum mechanics (e.g. Finite Element and Finite Difference methods) are not the best choice for modelling granular flows as they cannot easily cope with very large / extreme deformations, even in their advanced finite strain formulations. Material Point Method (MPM) is a numerical method based on continuum mechanics that copes well with very large deformations, since its algorithm eliminates mesh entangling problem [5,6] and is well suited for modelling of granular flows [7].

Most MPM research focus on the quasi static to moderate granular flows and do not investigate the disconnected state. To the authors' knowledge, Dunatunga & Kamrin [3] first applied MPM to model different states of granular flow. Their framework consists of constitutive models for different states of granular flows and a switching mechanism between them. Using this framework, [3] replicates silo drainage, incline chute flows and collapse of granular column. Later, [8] uses the previous framework for modelling impact and penetration of projectile into dry granular medium and compares results with experimental data. Similarly, Redaelli et al. [9] investigates solid–fluid transitions in granular flows with MPM using a novel constitutive model. This constitutive model replicates the behaviour of flows in different stages and the transition between them.

The listed approaches lead to large and unphysical deformations when using more advanced version of MPM [8]. As such, they are all limited to the earlier versions of MPM, which do not update deformation of material points during simulations [3,8,9]. Furthermore, these investigations do not examin MPM capability in modelling various stages of the granular flows, but simply assume that it is the case. An examination of this assumption is now overdue.

This paper examines the existing MPM methods capability of modelling different stages of granular flows and demonstrates their shortcomings. To deal with these shortcomings, the paper introduces Granular MPM, an extension to the MPM algorithm that incorporate the different states of granular flow into MPM modelling. Unlike previous methods, we show that Granular MPM can be used with the more recent MPM algorithms which update the deformations of the material points and we demonstrate that Granular MPM has considerable advantages over other MPM formulations for modelling granular flows.

2. MPM modelling of granular flows

2.1. Governing equations of MPM

In the explicit formulations of MPM, the material points store all the simulation data. During each time step the data from the material points is transferred to the grid nodes, where the equation for conservation of momentum is solved. The main distinctions between the explicit versions of MPM are in: (i) the equations transferring the data from the material points to the grid nodes and returning it from the grid nodes to the material points and (ii) the definition of the material point domain and its evolution. A general explicit MPM algorithm first transfers data from the material points to the grid nodes at the start of each time step:

$$m_i = \sum_p \phi_{ip} m_p \tag{1}$$

$$\mathbf{P}_{i} = \sum_{p} \phi_{ip} m_{p} \mathbf{v}_{p} \tag{2}$$

$$\mathbf{f}_{i}^{int} = \sum_{p} \nabla \phi_{ip} \boldsymbol{\sigma}_{p} V_{p} \tag{3}$$

where m_i , P_i and f_i^{int} are the nodal mass, the momentum and the internal forces, respectively. Symbols m_p , v_p , σ_p and V_p refer to the material points mass, their velocity, stress and volume, respectively. Function ϕ_{ip} and its gradient $\nabla \phi_{ip}$ transfer data between the material points and grid nodes. The definition of ϕ_{ip} and $\nabla \phi_{ip}$ depends on the version of MPM. Now, the algorithm updates the nodal values:

$$\mathbf{f}_{i} = \mathbf{f}_{i}^{\text{ int}} + \mathbf{f}_{i}^{\text{ ext}} \tag{4}$$

$$\mathbf{P}_{i} = \mathbf{P}_{i} + \mathbf{f}_{i} dt \tag{5}$$

where $\mathbf{f_i}$ is the nodal force vector, $\mathbf{f_i}^{\text{ext}}$ is the external force, which includes all tractions and body forces, and dt is the time step of calculation. At this point, the boundary conditions are applied to the updated nodal values. For example, the MPM algorithm sets the nodal force and momentum to zero for the grid nodes that have fixed boundary conditions. Finally, the algorithm returns the data to the material points and update their time dependant characteristics:

$$\mathbf{x}_{\mathbf{p}} = \mathbf{x}_{\mathbf{p}} + \sum_{i} \phi_{ip} \frac{\mathbf{P}_{i}}{m_{i}} dt$$
(6)

$$\mathbf{v}_{\mathbf{p}} = \mathbf{v}_{\mathbf{p}} + \sum_{i} \phi_{ip} \frac{\mathbf{f}_{i}}{m_{i}} dt$$
(7)

$$\boldsymbol{L}_{p} = \sum_{i} \nabla \phi_{ip} \frac{\boldsymbol{P}_{i}}{m_{i}} \tag{8}$$

$$\mathbf{F}_{p} = (\mathbf{I} + \mathbf{L}_{p}dt)\mathbf{F}_{p}$$
(9)

$$V_{p} = \det\left(\mathbf{F}_{p}\right) V_{0p} \tag{10}$$

$$\dot{\varepsilon}_{\mathbf{p}} = \sum_{i} \frac{1}{2} \left[\nabla \phi_{ip} \frac{\mathbf{P}_{i}}{m_{i}} + \left(\nabla \phi_{ip} \frac{\mathbf{P}_{i}}{m_{i}} \right)^{\mathrm{T}} \right] dt$$
(11)

$$\boldsymbol{\sigma}_{\mathrm{p}} = \boldsymbol{\sigma}_{\mathrm{p}} + \boldsymbol{\dot{\sigma}_{\mathrm{p}}} \tag{12}$$

where $\mathbf{x}_{\mathbf{p}}, \mathbf{L}_{\mathbf{p}}, \mathbf{F}_{\mathbf{p}}, V_{0p}, \dot{\epsilon}_{\mathbf{p}}$ and $\dot{\mathbf{\sigma}_{\mathbf{p}}}$ are the material points position, the gradient velocity, the deformation gradient, the initial volume, the change of strain and the change of stress, respectively. Change of stress ($\dot{\mathbf{\sigma}_{\mathbf{p}}}$) depends on $\dot{\epsilon}_{\mathbf{p}}$ and the algorithm calculates it using the constitutive model for each material point, with a standard stress integration algorithm. Finally, **I** is the identity tensor.

In this research, we use the Update Stress Last (USL) algorithm of Sulsky et al. [10] since, to our knowledge, it is the most common version, used e.g. in Uintah software [11]. Furthermore, it is more efficient than the Update Stress First (USF) method [12].Versions of MPM may define $\phi_{ip},\,\nabla\phi_{ip},$ material point domain and the domain evolution differently. For example, the original MPM defines material points as mere points with internal variables [10] and uses a linear tent function and its gradients for calculation of ϕ_{in} and $\nabla \phi_{in}$. Generalized Interpolation Material Point Method (GIMP, [7]) introduces particle characteristic functions that define the space occupied by the material points. GIMP also generalized the definitions of ϕ_{ip} and $\nabla \phi_{ip}$ and in cpGIMP version tracks the changes of the material points domains while assuming the shape of each domain remains rectangular. Finally, Sadeghirad et al. [13] introduced Convected Particle Domain Interpolation (CPDI) technique and enhanced GIMP with capability to model the nonrectangular domains of material points, with the domains being a parallelogram in the CPDI1 version and quadrilaterals in CPDI2 [13,14]. Note that the simulations of this paper use CPDI1 formulation (noted as CPDI), since it is, to our best knowledge, more common while CPDI2 may have domain distortion issues [15]. Nevertheless, the algorithms of section 3 and the appendix are equally suitable for CPDI2.

2.2. Constitutive framework for granular flows

The differences in constitutive behaviour of granular material in each state need to be considered in a constitutive model/models used for each flow stage. In the quasi-static state, the granular material exhibits a complex, though rather well investigated and understood behaviour. The granular material cannot or can hardly support tensile stresses as its shear strength is related to the mean stress. In the initial stage of shearing the material response is stiffer, resulting in a combination of very small reversible and irreversible (elastic and plastic) strains. The initial volumetric and the shear strains may significantly reduce the shearing stiffness of granular body [16,17]. After this initial small strain behaviour, further shearing leads to simultaneous changes in volumetric and shear strains and stresses. These shear and volumetric strains are typically regarded as having both elastic and plastic components. The elastic part is initially more significant, but as the shearing continues the plastic part become dominant. The increase of the volumetric and shearing strains continues up to a critical point at which the material cannot change volume upon pure shearing anymore. Furthermore, the change of packing fraction in quasi static state is a function of shear strain and largely independent of shear rate [2].

The constitutive behaviour of a granular material in the moderate flow phase is simpler than in the quasi static state [2]. However, unlike the previous, it is shear-rate dependent as the collisional interactions between the grains during the moderate flow lead to an energy loss and a rate sensitivity [2]. That can be modelled, for example, following the findings of Bagnold [18] who has shown a quadratic dependency between the shear stress and the shear rate of a granular flow. Several experiments and simulations have verified Bagnold observation in dry granular media and showed that the shear rate has a unique role in the determination of shear stresses [19–22]. The kinetic theory of granular gases defines the constitutive behaviour in the disconnected state. Redaelli et al. [1] use the kinetic theory to estimate the stresses caused by the collisional interactions. The examples in that study show that the collisional stresses are very small, unless the rate of shearing is very high. In the same way, Dunatunga and Kamrin [3] use the kinetic theory to formulate an equation of state for the disconnected stage of the flow. Based on this equation, they estimate the disconnected state to be stress-free for several granular flows [3,8].

In this paper, to illustrate the capabilities of the proposed algorithms, we use very simple constitutive models describing the behaviour of the material in various states. Most examples employ a non-associated Mohr-Coulomb constitutive model extended with Bagnold rate dependency for approximating the material behaviour both in the quasi-static and the moderate flow states. Furthermore, we use the stress-free approximation of Dunatunga and Kamrin [3] for the disconnected state.

The next two subsections investigate two problems in modelling moderate and disconnected states related to the MPM algorithm itself, not the constitutive model used in calculations. They show that a proper constitutive framework for granular flows, though necessary, is not sufficient by itself to accurately simulate granular flows in MPM.

2.3. Modelling of moderate flows

As discussed in section 1, the packing fraction remains relatively unchanged when the material is in the moderate flow state, when shear rate is small or moderate. Yet, the density of the granular material points, calculated by dividing the mass of material points to their volume $\frac{m_p}{V_p}$. changes significantly in MPM modelling. Fig. 1 illustrates an example investigating this problem:

where the granular materials move down a slope. We use a non-associated Mohr-Coulomb with dilation angle of zero for the granular material, which means that the plastic volume change should be zero. Table 1 provides further parameters used in the simulations. This example models the slope aligned to the horizon and adjusts the gravity to capture the slope inclination. All the simulations are plane strain, use square grid of 10 mm, fixed boundary conditions on all the sides of the flow media and 120 granular material points. Each grid cell contains 4 uniformly distributed material points.

Fig. 2 shows that the average density reduces significantly in both original MPM and CPDI simulations. Furthermore, considering the disconnected state by switching a material point to stress-free when the density of that material point becomes lower than a predefined critical density [3] does not fully prevent the fall in the average density. Upon examination of Fig. 2, it is also clear that the reduction of density continues during the flow, so the unchanging density of moderate flow has not been captured. This example illustrates that existing MPM formulations do not replicate the unchanging density during moderate flows, even with use of the stress-free formulation and suggests that existing MPM algorithms have problem in replicating the moderate flows.

In reality, when the density of the moderate flows falls below the critical density, the grains loose frictional contacts and stresses. These stress-free grains cannot push each other away so density reduction stops while some shearing deformations might occur. However, MPM has no means for replicating the loss of frictional contacts within a material point. That is one of the reasons why it is unable to capture the constant density of the material during moderate flow.



Fig. 1. Schematic of granular flow on a slope.

Table 1Parameters of simulation and granular materials.

| Simulations | Grid spacing(mm) | | dt (s) | | $t_{total}(s)$ | Inclination (°) |
|-------------------|------------------|-----|--------|------|--|--|
| | 10 | | 7.0e-6 | | 0.25 | 30 |
| Granular material | E(GPa) | v | φ(°) | ψ(°) | $\rho_{initial}\left(\!\tfrac{kg}{m^3}\!\right)$ | $\rho_{critical} \left(\tfrac{kg}{m^3} \right)$ |
| | 1.0 | 0.3 | 30 | 0 | 2250 | 2235 |



Fig. 2. Variation of average density for granular material points of Fig. 1.

2.4. Interactions during disconnected flows

As discussed in section 1, the frictional contacts between grains of the granular material are lost in the disconnected state, while the collisional interactions of the grains remain negligible. Therefore, there should be no interaction among the disconnected material points in MPM. Dunatunga and Kamrin [3,8] implicitly assume that a stress-free constitutive behaviour prevents interaction with the disconnected material points, however we will show that it is not the case.

Let's consider a 1D example shown in Fig. 3 (a). The material point 1 is disconnected and have no stress and velocity ($\mathbf{v}_1^p = \mathbf{0}$, $\sigma_1^p = \mathbf{0}$). The other material points move to the right side with dif-

ferent velocity $(|\mathbf{v}_2^{\mathbf{p}}| \neq |\mathbf{v}_3^{\mathbf{p}}| \neq |\mathbf{v}_4^{\mathbf{p}}|)$ and are not stress-free. There are no further external forces, tractions and boundary conditions.

In Fig. 3 (a), node 3 is a point of interest since the disconnected material point 1 and the material point 2 both contribute to the calculation of its nodal values. In all the versions of MPM, the material points 1 and 2 contribute to the calculation of node 3 mass and momentum (Fig. 3 (b)). Furthermore, the point 2 has a non-zero stress that leads to calculation of a non-zero internal force for node 3 (Eq. (3) and Fig. 3 (c)) regardless of the material point 1 constitutive model. Consequently, the updated force of node 3 is non-zero (Eq. (5)) and changes the momentum of node 3 after updating (Eq. (6)). Transferring the updated momentum and force of the node 3 to the disconnected material point 1 (Eq.



Fig. 3. 1D example of material point disconnection.

(7) and Fig. 3 (d)) changes the velocity of the material point 1 and leads to the displacement of it. The changes of the material point velocity and position would not happen in absence of the other material points. Therefore, there are interactions with disconnected material points.

Although different versions of MPM handle material points in Fig. 3 differently, the identified mechanism for disconnected interactions is present in all of them. In the original MPM, the material points only transfer data to the nodes of the grid cell they are positioned in. As a result, the material points 1 and 2 will stop interacting after they stops sending data to a common grid node. In uGIMP, the material points can contribute to the neighbouring grid cell as well. Therefore, the material point 3 in Fig. 3 might also contribute to the nodal value of the node 3 and increase the unphysical interactions. Furthermore, the disconnected material point 1 and the material point 2 can interact even after they move to another cell. Nevertheless, the domain size of the material points does not change in uGIMP and the material points will stop interacting as they move further away. In cpGIMP and CPDI, the domain of material points can change. Therefore, the domain of the disconnected material point 1, which is stress-free and has no internal resistance, grows and the material point 1 and 2 will keep interacting. This behaviour is unphysical and prevent any use of cpGIMP and CPDI in modelling of the disconnected state. It should be noted that this explains the large and unphysical deformations of [8] when using CPDI. Furthermore, the simulations in [3] suffer from the disconnected material points interactions. Although, these interaction are less noticeable since [3] only use original MPM.

Note that introduction of a contact algorithm alone will not solve the problem, as the disconnected state may occur between material points of the same material once the density drops. Hence, next section introduces Granular MPM, which allows for fully disconnected material points of the same material, may be seen as an algorithm combining fracture and contact algorithms with physical considerations related to the flow of granular materials.

3. Granular MPM

This section defines criteria for differentiating between different stages of granular flows. Furthermore, it introduces Granular MPM, an extension of MPM for modelling granular flows with low to moderate shear rate as a solution for dealing with the issues demonstrated in subsections 2.3 and 2.4.

3.1. Criteria for identifying the states of granular material

Granular MPM identifies the state of granular material based on the following criteria:

- i. Material points are in the quasi-static state when their density is higher than a critical density ($\rho_{critical}$) and they are close enough to the other material points.
- ii. Material points are in the moderate state of flow when their density is equal to the critical density and they are close enough to the other material points.
- iii. Material points are in disconnected state when they are not close enough to other material points. The density of disconnected material points remains unchanged during this state.

We use 4 main assumptions for defining the previous criteria. First, we assume a unique value for the critical density for given granular material as in reality the variation of this density is small and insignificant. Second, we assume that packing fraction in moderate flows does not change so the material remains at the critical density. Such assumption is reasonable since the study focuses on flows with low to moderate shear rate and, for example, [2] uses a similar assumption in modelling of the moderate granular flow. Third, we neglect the possible changes of density in the disconnected state. This decision was motivated by the fact that the interaction of grains in the disconnected state is limited to collisional interaction, which is relatively small for the flows with low to moderate shear rate. Furthermore, the disconnected state is usually a temporary state. Hence, in simulations when the state of granular material returns to moderate flow, the material points volumetric and shear strains can be taken as unchanged after being shortly in the disconnected state. Fourth, the collisional flow are uncommon in real life and usually only observed in laboratories [4]. As such, we consider this stage out of the scope of this research.

3.2. Capturing the moderate flow

In order to address the problems highlighted in subsection 2.3, Granular MPM modifies material points in the state of moderate flow when their density falls lower than the critical value. The algorithm first changes the stress tensor and sets the material point to stress-free. Furthermore, it changes the deformation gradient of these material points.

In this paper, we amend the gradient deformation by returning it to the gradient of the previous step. This return ensures to keep the density of moderate flow around the critical value. It also prevents overestimation of rotation and shearing deformations by removing their changes in the current step. Nevertheless, this amendment might lead to underestimation of the rotation and shearing deformation in the granular material points (subsection 4.2 provides further details). Other alternatives could be to reduce the changes in the rotation and shearing deformation, which may require further breaking the granular material points into smaller ones to avoid domains being too distorted. These alternatives are beyond the scope of this paper and could be investigated in future research.

The modification is introduced after the final transfer of data to the material points. A simple density check identifies the granular material points that require the modifications. This check controls the density and applies the modifications as follows:

If
$$\rho_p < (\rho_{cr} - 10L)$$

$$\boldsymbol{\sigma}_{\mathrm{p}} = \boldsymbol{0} \tag{13}$$

 $\mathbf{F}_{p} = \left(\mathbf{I} + \mathbf{L}_{p} dt\right)^{-1} \mathbf{F}_{p}$ (14)

$$V_{p} = det(\mathbf{F}_{p})V_{0p} \tag{15}$$

endif

where ρ_{cr} is the critical density of the material point and ρ_p is density of material point. TOL is a constant value that ensures the check applies only to material points with density lower than critical value and was assumed to be 1.0e-6 $\frac{kg}{m^3}$ in this study. L_p and V_{0p} are the gradient velocity and the initial volume of the material points.

3.3. Preventing interaction of disconnected material points

Granular MPM implements parallel grids to prevent interactions with disconnected material points. This solution is inspired by [23], which is an extension for modelling of explicit cracks in MPM. In [23], the material points on the two sides of a crack contribute to parallel grids so they will not interact. Similarly, Granular MPM moves the disconnected material points to parallel grids and run the MPM algorithm for each grid separately. Therefore, the disconnected material points do not interact with the others.

The decision to which parallel grid given point belongs requires a test to determine whether the material points are close enough to interact (and share a grid) or not. To facilitate the check, Granular MPM defines a domain of interaction around each material point. If the domains of interaction of two material points intersect, the two points can interact and should belong to the same grid. Similarly, a material point is disconnected if there is no other material point with an intersecting domain of interaction at the time of the check.

We define the domain of interaction as domain of material point scaled up to the critical volume (V_{cr}), which is the highest volume a material point can have and is equals to $\frac{P_{cr}}{m_p}$. Fig. 4 (a) shows two examples of scaling domains of material point (solid lines) to find domains of interaction (dashed lines). As this figure illustrates, the domain of material point and the domain of interaction are concentric and proportional. The domain of interaction is a line in 1D MPM variants, a constant square in 2D original MPM and uGIMP, a shape changing constant volume rectangle in 2D cpGIMP

and a shape changing constant volume parallelograms in 2D CPDI. In the Fig. 4 (b), material points 1 and 2 have domains smaller than their domain of interaction. The domains of material point 3 and 4 have reached their domain of interaction and cannot grow anymore. The domains of points 5–6 and 7–8 are non-overlapping, while their domains of interaction have an overlapping part as such they should interact and share common grids.

3.4. Algorithm of granular MPM

The algorithm of Granular MPM implements the procedure for identifying connected and disconnected material points and assigning them to the correct grids. After the algorithm assigns the material points to parallel grids, it runs the calculations on each grid separately. This algorithm can be implemented into any of the existing explicit MPM algorithms.

Each step of Granular MPM algorithm has three main tasks and an optional pre-task. First, the algorithm checks for the interaction between granular material points (pre-task and task 1). Then, it defines the grids based on the necessary interactions (task 2). Next, it runs the MPM algorithm for each grid separately, checks the state of the material points to use correct constitutive model and applies the possible adjustments of the gradient deformations, as given in section 3.2 (task 3).

Task 1 checks the domain of interaction of a material point for possible intersections with all other material points. This is a simple but inefficient way for determining necessary interactions. In order to improve the efficiency of the task 1, the algorithm of Granular MPM introduces an optional pre-task. This pre-task finds the granular material points that send data to at least one common grid node. When two material points have a common grid node, they are connected through this node and have the potential to interact in the current time step. Therefore, the task 1 can be limited to these material points and the number of checks reduce significantly.

Fig. 5 illustrates a schematic of the pre-task and the task 1 for the Granular original MPM. In this figure, the material points are shown as filled black circles, the continuous thin lines represent the grid and the domain of interaction for the material point are marked with thick dashed lines. The pre-task uses the current positions of the material points to find that the material point 5 can interact with the material point 4, 6, 8 and 9 in the current step. The output of the pre-task is the **connection** array. The 5th row of the **connection** array has index of the material points 4, 5, 6, 8 and 9. The material point 7 is not in the 5th row of the **connec**tion array since the pre-task algorithm utilizes the original MPM. In a pre-task for any GIMP or CPDI version, the material point 7 should be in the 5th row of the connection array. Then, the task 1 checks for the intersection between the domain of interaction of material point 5 with the domains of material point 4, 5, 6, 8 and 9. The output of the task 1 is the interaction array. The 5th row of interaction has index of the material points 4, 5 and 9.



Fig. 4. 1D and 2D material point domains and domains of interaction.



Fig. 5. Schematic of pre-task and task 1 of Granular original MPM.

The pre-task and task 1 do the similar checks for the other material points and write other rows in the **connection** and **interaction** arrays. For example, the 6th row of the **interaction** array contains the material points 6, 7 and 8. The appendix provides a simple Matlab algorithm for the pre-task and task 1.

Task 2 of the algorithm determines the grids based on the interaction array of task 1. This task loop over all the material point. When it finds a material point that has not been assigned to any grid, it moves this material point and all material points that interact with it to a new grid, ensuring that all the material points that interact with each other are assigned to the same grid. For example for material points of Fig. 5, it first finds the material point 1 has not been assigned to any grid. Then, it moves material point 1 and 2 to the first grid. Next, it starts a loop that adds material point 3, 7, 6 and 8 to the first grid. The algorithm continues this process until all the material points are assigned to one and only one grid. This task determines 3 parallel grids for material points of Fig. 5. Material points 1, 2, 3, 6, 7 and 8 are in the first grid. Material points 4, 5 and 9 are in the second one. Finally, the disconnected material point 10 is in the third grid. The appendix provides further explanation and a Matlab algorithms for the task 2.

Task 3 runs the given version of MPM algorithm for each grid. The main difference from the given MPM algorithm is the density control check for all the granular material points scheduled after the run. Furthermore, this task applies the appropriate constitutive model for a material point based on the state of the flow. The appendix provides 2 separate Matlab algorithms for reading and rewriting of the material points data for given grids. Finally, this research use the Update Stress Last (USL) algorithm of Sulsky et al. [10]. Nevertheless, Granular MPM can use other algorithms of stress update.

To summarise, Granular MPM considers the unique physics of granular flows, allows for grain continuum separation into pieces and combination back into continuum material. The algorithm allows for different constitutive models for each state of the material and limits the deformation of the material in moderate and disconnected flows. As such, it provides a much more realistic replication of granular flows than previous methods ([3,8,9]).

4. Results and discussions

This section compares capabilities of MPM and Granular MPM in modelling of the granular flows. The shown examples also investigate the limitations of computational frameworks for granular flows. The examples compare 3 types of algorithms:

- 1- The conventional version of MPM algorithms, ignoring the change of the constitutive behaviour in the disconnected state. In this section, the simulations denoted original MPM and CPDI are of this type.
- 2- Stress-free MPM [3] that uses conventional versions of MPM but considers the change of constitutive behaviour in the disconnected state. The constitutive behaviour of a material point is stress-free when density of the material point becomes smaller than critical. Those simulations are denoted stress-free original MPM, stress-free cpGIMP and stress-free CPDI.
- 3- Granular MPM that uses the proposed Granular MPM algorithm of subsection 3.4. Those simulations are denoted Granular original MPM, Granular uGIMP, Granular cpGIMP and Granular CPDI.

4.1. 1D disconnected state

This example compares the stress-free and the Granular MPM modelling of the disconnected state. Fig. 6 shows a 1D setup of material points. The bold numbers denote the material points and circled numbers indicate the grid nodes, spaced 10 mm apart. Material points 1 and 2 have the same initial density of $1500 \frac{\text{kg}}{\text{m}^3}$, corresponding to the domain length of 10 mm. This example assumes the critical density is $1400 \frac{\text{kg}}{\text{m}^3}$ so the domain of interaction



Fig. 6. 1D setup of dense and disconnected and material points.

length is 10.71 mm. The domains of interaction of material point 1 and 2 (marked with brackets) intersect thus point 1 and 2 are in a quasi-static state. These points move leftwards and have uneven velocities. Material point 3 has just switched into the disconnected state. Its domain of interaction is 10.71 mm long and have a corresponding density of 1399.99 $\frac{\text{kg}}{\text{m}^3}$, just below the critical density. This material point moves rightwards. Finally, the material points 4 and 5 represent another body of granular material. These material points have density of $1500 \frac{\text{kg}}{\text{m}^3}$, corresponding to the domain length of 10 mm and their domain of interactions intersect, indicating the quasi-static state.

The example assumes a simple linear elastic constitutive model for the material points with density higher than critical density. In addition, it assumes stress-free behaviour for the material point with density lower than critical. See Table 2 for the summary of material parameters.

The Granular MPM algorithm considers the material point 3 to be disconnected, stress-free and moves it to a parallel grid, which ensures that it will not interact with the other material points. On the other hand, the stress-free algorithm only considers that the material point 3 is stress-free. Fig. 7 illustrates the velocity evolution in time of the material point 3 for the Granular and stress-free procedure using the original MPM and cpGIMP interpolations.

Upon examination of Fig. 7, the stress-free original MPM algorithm calculates a slight change of velocity for the material point 3 in the beginning of modelling while the algorithm of Granular MPM calculates no change of velocity in the beginning of simulations. The velocity change in the stress-free original MPM is the result of interaction between the material point 2 and 3 and stops when they stop transferring data to a common node. There is no interaction in Granular original MPM as it assigns different grid to material point 2 and 3. Furthermore, the stress-free original MPM calculates that velocity of the material point 3 starts changing at time equal to 0.00392 s when the material point 3 and 4 both start sending data to the grid node 6. This interaction in stress-free original MPM depends on the relative position of material points on the grid and lead to a grid dependent solution. The Granular original MPM calculates the change of velocity for material point 3 later, at 0.0067 s, when this point domain of interaction starts touching the domain of interaction of point 4, and the grids of material point 3 and 4 merge. Therefore, the Granular MPM solution prevents the unwanted disconnected-solid interactions and the initiation of points interaction is not grid dependent. Furthermore, Fig. 7 shows that the stress-free cpGIMP algorithm continues to change the velocity for the material point 3, due to interaction on grid node 4, indicating that the stress-free condition is not sufficient to ensure the disconnected state well. On the other hand, the Granular cpGIMP considers different grid for material point 2 and 3 so no interaction between them occur and the velocity of

Table 2Parameters of example one.

| E(kPa) | $\rho_{initial}$ | $\rho_{critical}$ | dt | Grid spacing |
|--------|-----------------------|-----------------------|---------|--------------|
| 100 | $1500 \frac{kg}{m^3}$ | $1400 \frac{kg}{m^3}$ | 0.00002 | 10 mm |

the material point 3 remains unchanged until the grid of material point 3 and 4 merge.

Finally, Fig. 8 shows the displacement of material point 3 using the stress-free and the Granular original MPM and cpGIMP. As expected, the stress-free original MPM and the stress-free cpGIMP predict quite different displacements for the material point 3. Nevertheless, the Granular original MPM and cpGIMP predict same displacements for this material point until 0.0067 s, with later differences in point 3 behaviour caused by the differences between original MPM and cpGIMP interpolations.

4.2. Granular avalanche

The second example models a flow of a granular material on two inclined planes. Fig. 9 shows a schematic of the problem where the granular material initially moves down the higher plane due to gravity, with a starting velocity of 1 m/s, falls to the lower plane and stops by the wall. The example allows to compare the capabilities of MPM, stress-free MPM and Granular MPM algorithms in modelling of granular flows using original MPM and CPDI.

The simulation describes the granular material by the Mohr-Coulomb model enhanced with Bagnold strain rate dependency to account for dependency between the shear stress and the shear rate, as in [24]. This constitutive model yield surface equation is:

$$F = (\sigma_1 - \sigma_3) - (\sigma_1 + \sigma_3)\sin\phi - 2(\eta\dot{\gamma}^2)\cos\phi = 0$$
(16)

where σ_1 is the maximum principal stress, σ_3 is the minimum principal stress, ϕ is the friction angle, $\dot{\gamma}$ is the rate of simple shear and η is a parameter describing dependency of the shear failure stress on the shear rate. The constitutive model uses a nonassociated flow rule with the plastic potential surface G defined as:

$$\mathbf{G} = (\sigma_1 - \sigma_3) - (\sigma_1 + \sigma_3)\sin\psi = \mathbf{0} \tag{17}$$

where ψ is the dilation angle. This constitutive model is relatively basic, but sufficient to show differences related to the choice of MPM algorithms in modelling granular flows. The simulations also assume stress-free behaviour for the granular material points with density lower than critical in stress-free original MPM, stress-free CPDI, Granular original MPM and Granular CPDI.

This example models the slopes as horizontal and adjusts the gravity angle to capture the slope inclination. All the simulations use square grid of 100 mm, fixed boundary conditions on all the sides of the flow media and total of 160 material points to simulate the granular material (4 per cell). Furthermore, the planes shown in Fig. 9 are replicated using boundary conditions. Table 3 shows the parameters of the granular material and the simulation in this example.

Fig. 10 shows that the average density of the material points (computed based on volumetric strain in the material points) in the original MPM reduces significantly in the simulations. Similarly, there is a considerable reduction in the average density in the stress-free original MPM simulation, even though there is some increase in the material density during the final depositing stage. Despite that density increase, in the stress-free simulation the density of almost all the material points reduce below the critical state density, resulting in the material points to be stress-free during



Fig. 7. Material point 3 velocity for the Granular and stress-free procedures using original MPM and cpGIMP.



Fig. 8. Material point 3 displacement for the Granular and stress-free algorithms using original MPM and cpGIMP interpolation.



Fig. 9. Schematic of second example slopes and initial position of granular material.

S. Seyedan and W.T. Sołowski

Table 3

Parameters of second example.

| Simulation | Grid spacing(m 100 | Grid spacing(mm) 100 | | | dt (s) 1.0e-5 | | Inclination (°) 15 |
|-------------------|-----------------------|-------------------------|------|------|------------------|--|---|
| Granular material | E(GPa) | ν | φ(°) | ψ(°) | η | $\rho_{\text{initial}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ | $\rho_{\text{critical}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ |
| | 1.0 | 0.3 | 30 | 0 | 0.15 | 2250 | 2235 |



Fig. 10. Variation of average density for granular material points using original MPM.



Fig. 11. Snapshots of second example simulations at different times using original MPM, stress-free original MPM and Granular original MPM. The videos for these simulations are available in supplemental materials of the paper.

deposition. Finally, Fig. 10 shows that there is a relatively constant average density of the material in the Granular original MPM simulation. The density of the material in the Granular original MPM initially reduces slightly, down to the critical density and then increase slightly during the depositing stage preventing the material points to be stress-free.

Fig. 11 illustrates positions of the original MPM, the stress-free original MPM and the Granular original MPM material points at different times. At t = 0.45 s, the differences are slight: the simulations with original MPM and stress-free original MPM predict slightly higher velocity of the material (check the supplemental materials for the velocities), with the front material points move faster than the other parts, while the back part of the granular material moves slower. This is due to the transfer of velocity and momentum from the material points in the back to those in the front of the original MPM and stress-free original MPM. On the contrary, the interaction control of the Granular MPM algorithm limits the transfer of velocity and momentum and reproduces the steady condition of a moderate flow. At t = 1.2 s, the front material points of the original MPM and the stress-free original MPM have already reached the wall at the end of the lower slope while the material points of Granular original MPM are still moving towards it. At the wall, the front material points in both the original MPM and the stress-free original MPM have density lower than the critical value. As such, the downward velocity of the front

material points in the original MPM translates into stress and leads to bounce back of these material point. On the other hand, stressfree material points remain at the wall and deposit there. At t = 2.1 s, the granular material points in all the simulations have reached to the end of lower slope. In the original MPM, some material points have bounced back above the other material points leading to a very loose packing. The bounce back of the material points is much less pronounced in the other two simulations. In the stress-free simulation, the material points are deposited in a disordered way since the stress-free material points cannot produce compressive stresses to push the neighbouring material points and create an ordered packing. In contrast, the material points at the end of the Granular original MPM simulation show a uniform packing due to their ability to produce compressive stresses. At t = 3.0 s, the top material points in the original MPM simulation have remained in the unrealistic loose packing. The material points in the stress-free and Granular original MPM are somewhat similarly distributed in the final stage of simulation, although the stress-free simulation leads to a looser and nonuniform packing compared to the Granular original MPM simulation.

Fig. 12 shows the snapshots of the material points position for the same problem and at the same times, but obtained with CPDI, stress-free CPDI and Granular CPDI. In Fig. 12, each material point has a deformable parallelogram domain, which is updated during



Fig. 12. Snapshots of second example simulations at different times using CPDI, stress-free CPDI and Granular CPDI. The videos for these simulations are available in supplemental materials of the paper.

simulation. At t = 0.45 s, the front and back of the flowing material in CPDI and the stress-free CPDI have gone through shearing and volume increase, due to the transfer of velocity and momentum from the back to the front. On the other hand, the material points became disconnected in the Granular CPDI simulation since the Granular MPM algorithm notices when the material points reach the critical density and assigns these material points to separate grids. At t = 1.2 3s, the front material points in the CPDI and the stress-free CPDI have reached the end, while in the Granular CPDI the material points are still moving towards it. Subsequently, in the CPDI simulation, the front material points bounce back, deform and go through considerable volume increase. In the stress-free CPDI, the material points do not bounce back and have lower yet unphysical volume increase. In comparison, in the Granular CPDI simulation, the deformations of the material points are relatively small and they have separated from each other due to Granular CPDI parallel grids. At t = 1.72 s, the material points in all the simulations are close to the end of lower slope and have started slowing down. Some of the material points in the CPDI simulation have very large volume and are close to the upper boundary of the grid. After some time steps, the CPDI simulation stops because corner of a material point moves out of the grid and causes error in calculation. In the stress-free CPDI, few material points at the upper boundary are very distorted and have large volume. However, the material points in the Granular CPDI have no significant deformations and exhibit no volume increase. At t = 2.4 s, the stress-free CPDI simulation has several material points with significant volume increase which will moves out the grid after some time steps and cause the stress-free CPDI simulation to stop. In contrast, the material points of Granular CPDI are rather uniformly deposited and allow the simulation to end without experiencing any numerical instability.

As subsection 3.2 highlights, Granular MPM removes the changes of rotation and shearing deformation of granular material points when their density falls lower than the critical value. In order to investigate the effects of removing them, the example runs an alternate Granular CPDI simulation of granular avalanche that keeps the rotation and shearing deformations but removes the volumetric ones. The simulation achieves these by changing the equation (14) to:

$$\mathbf{F}_{p} = \sqrt{\frac{V_{cr}}{V_{p}}} \mathbf{F}_{p}$$
(18)

Fig. 13 compares the final states of the granular avalanche when: (i) removing the deformation of the steps if $\rho_{\text{p}} < \rho_{\text{cr}}$ and (ii) keeping the rotation and shearing deformations of the steps but removing the volumetric ones if $\rho_p < \rho_{cr}.$ The material points in the dashed ellipse in Fig. 13 (ii) experience considerable and unphysical shearing leading to entangling. These material points remain relatively unsheared if the rotation and shearing deformations are removed when the density falls lower than the critical (i). On the other hand, the material points in the continuous ellipses in Fig. 13 do not experience significant shearing. Comparing these material points suggests that removing the shear deformations when density falls below the critical value results in separation of material points. This breaks force chains between the material points and while it allows them to separate, it could be unphysical. In conclusion, both of the methods have some disadvantages and a more realistic solution could be investigated in the future. The solution should prevent the overestimation of shearing deformation while preserving the force chains between the material points.

4.3. Silo filling

This example models a silo filling problem. Such a problem includes the three granular flow stages. First, the grains start a



Fig. 13. Final states of the granular avalanche when: (i) removing the deformation if $\rho_p < \rho_{cr}$ (ii) keeping the rotation and shearing deformations of the steps but removing the volumetric ones if $\rho_p < \rho_{cr}$.

disconnected free fall upon release from the top. Then, they reach to current level of grains, start interacting and slow down. Finally, they almost stop and move to a quasi-static state where the packing changes due to continuing increase of stress from the upper grains. The modelling of a silo filling is challenging since it should capture the transition between the different stages as well as changes of the packing fraction in the bottom.

This example models a flat bottom silo in 2D and fills it from the top part. The silo in this example is based on the flat bottom silo of Wójcik et al. ([25]) with the height of 7.5 m and the width of 2.7 m, see Fig. 14. The silo material is modelled by a linear elastic model, with stiffness reduced and wall thickness increased, when compared to the original silo, due to numerical reasons. The silo is fixed at the sides. The fill material is replicated using the same elastoplastic model described in subsection 4.2. The simulations use



Fig. 14. Schematic of the silo in the third example.

0.3 m square grid and the material points with mass and spacing initially corresponding to square of 0.15 m. Every 0.17 s, the simulations release 3 material points in the middle of 3 adjacent grid cells at top of the silo. Fig. 14 illustrates the first series of the released material points into the silo. The release of the material points continues during all the simulations and leads to the accumulation of 504 material points in total. Table 4 shows the parameters used in this example.

Table 4

Parameters of the third example.

Fig. 15 shows the positions of material points at different times computed with the original MPM, the stress-free original MPM and the Granular original MPM, with the colour of each point representing the velocity of that material point in $\frac{m}{2}$.

At t = 4.125 s, the material points of the original MPM simulation have almost filled the silo, since the very first falling material points interact with the other falling material points over the grid nodes, transfer momentum to them and prevents them from

| runaneters of the third end | inpiei | | | | | | | |
|-----------------------------|------------------|------------------|------|--------|-------|--|---|--|
| Simulation | Grid spacing(mm) | Grid spacing(mm) | | | dt(s) | | $t_{total}(s)$ | |
| | 300 | | | 5.0e-5 | | 28.56 | | |
| Silo | E(GPa) | | | v | | $\rho_{\text{initial}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ | | |
| | 1.0 | | | 0.3 | | 2250 | | |
| Granular Material | E(GPa) | v | φ(°) | ψ(°) | η | $\rho_{\text{initial}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ | $\rho_{\text{critical}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ | |
| | 1.0 | 0.3 | 30 | 0 | 0.15 | 2250 | 2235 | |
| | | | | | | | | |



Fig. 15. Snapshots of third example at different times using original MPM, stress-free original MPM and Granular original MPM with colour of points represents the velocity of the material point in $\frac{m}{3}$. The videos for these simulations are available in supplemental materials of the paper.

reaching to the bottom of the silo. As such, the material points fill the silo rapidly with unphysically few material points. Comparing the stress-free and the Granular original MPM simulation, the material points at the bottom of the silo have considerable different positions, velocities and packings. In the stress-free original MPM, the material points at the bottom are not close, some of them have considerable remaining velocity (in range of 4 m/s to 6 m/s) and are irregularly packed. On the other hand, the material points in the Granular original MPM simulation are close, their velocity is low (in range of 0 m/s to 1 m/s) and are regularly packed. The interactions control of the Granular original MPM is the main reason for this different behaviour as it prevents interaction among the material points unless they are close enough to be on the same grid. Furthermore, the volume control of the Granular algorithm results in the material points to have compressive stresses at the bottom, which allows the granular material points to push away the neighbouring ones and create an ordered packing.

At t = 9.9 s, the material points of original MPM have filled the silo, with the simulation failing shortly afterwards. In the stress-free original MPM, the material points have irregular placements, high velocities and low packing while the material points of the Granular original MPM have regular placement, low velocities and tight packing.

At t = 27.79 s, the material points have filled the silo in the simulation of stress-free original MPM while the same number of material points have filled only half of the silo in the Granular original MPM. Furthermore, the stress-free silo has a considerably higher packing fraction in the top of silo than its bottom. In contrast, the packing is relatively constant in the Granular original MPM simulation.

The same simulation has been repeated using CPDI, stress-free CPDI and Granular CPDI, with results given in Fig. 16. In this figure, each point represents position of a material point, the colour of the

points demonstrates velocity and the boundaries around the points corresponds to the domains of the material points.

At t = 2.64 s, the simulations are already quite different. In the CPDI calculation, the interactions of material points cause some of granular material points to deform significantly and move to the sides of silo. In the stress-free CPDI simulation, a similar process happens but the granular material points do not move to the sides that much and experience considerable volume change. Finally, in the Granular CPDI simulation, no volume changes take place.

At t = 4.95 s, the computation with CPDI algorithm leads to a silo filled with material points that have experienced considerable volume increase and are irregularly packed. The stress-free simulation predicts that granular material points are in the bottom of silo, some of them being very distorted and their domains overlapping. In contrast, the domains of the material points in the Granular CPDI simulation are not distorted or overlapping and are filling the silo well.

At t = 6.31 s, the CPDI-simulated silo has lower packing in the bottom than in the top part. In the stress-free simulation, the granular material points are significantly distorted, have high velocities and have breached the boundary condition of the model. As a result, the stress-free CPDI simulation fails shortly after this point. However, the Granular CPDI simulation continues without any problem.

At t = 14.32 s, the CPDI material points are distorted, overlapping and have breached the boundary conditions of the model. In contrast, the granular material points continue to deposit in Granular CPDI simulation replicating the silo filling.

This example shows the Granular MPM extension can model challenging simulations that the other available MPM formulations cannot capture. The Granular MPM can also use the more recent versions of MPM, such as CPDI, that are more accurate and correct than the original MPM.



Fig. 16. Snapshots of third example simulations at different times using CPDI, stress-free CPDI and Granular CPDI with colour of points represents the velocity of the material point in $\frac{m}{2}$. The videos for these simulations are available in supplemental materials of the paper.



Fig. 17. Schematic of the experiments (after [26]).



Fig. 18. Schematic of the model.

4.4. Estimation of granular flow impact force

The final example replicates laboratory experiments by Moriguchi et al. [26] who investigated flow and impact forces of Toyoura sand. At the beginning of their experiments, Moriguchi et al. [26] placed 50 kg of Toyoura sand on top of a flume. Each experiment released the sand, photographed its flow and recorded the load cell measurements of the normal impact force at the bottom (see Fig. 17). Although the experiments were performed with different slope angles, [26] only provides information on flows surface of 45° experiment. Therefore, we only replicate this 45° experiment numerically.

The numerical simulation models the experiment assuming 2D plane strain and Granular uGIMP scheme, see Fig. 18 for the schematics. In the model, the inclination of the flume is modelled by changing the gravity direction, while the coating of the flume with the sand is approximated by adding a layer of sand material fixed to the lower boundary. The boundaries of numerical domain are fixed, with the grid being 0.01 m squares filled with the material points with mass and spacing corresponding to 0.01 m (i.e. 1 material point per grid cell). To compare the force on the wall, the stresses are integrated using the rigid material option in

Computers and Structures 251 (2021) 106545

Uintah. Furthermore, we compare the flow in the simulation to the experimental flow surfaces.

Toyoura sand is modelled by elasto-plastic constitutive model described in subsection 4.2 with parameters shown in Table 5.

Fig. 19 compares the experimental flow surfaces with the simulation at different times. The snapshots are simulated with the Granular uGIMP model and the black lines represent the estimated flow surface of the experiment. The figure shows some differences in the quasi-static stages of flow, where the flow initiates and stops. The likely reason is that the quasi-static behaviour of sand is complex and the assumed simple elasto-plastic constitutive model cannot capture it very well. For example, the sand experiences a considerable reduction of stiffness after the opening of the box, not replicated by the model. Therefore, the Granular uGIMP model slightly overestimates the progress of the flow at t = 0.4 s. On the other hand, the model captures the flow surface of the experiment well in the middle later, in the moderate flow phase.

Fig. 20 shows that the Granular uGIMP replicates the impact force of the experiment well. Nevertheless, the model miscalculates arrival time of the granular flow slightly. This could be due to the simple elasto-plastic constitutive model that underestimates the flow velocity (kinetic energy) leading to miscalculation of the arrival time.

5. Conclusions

The paper first demonstrates problems in modelling of the moderate and disconnected states of granular flows with existing MPM algorithms. Then, it proposes Granular MPM, an algorithm for simulating the granular flows, applicable to all the common versions of MPM. Finally, the paper illustrates the capabilities of Granular MPM over four examples and compares it to the previous MPM algorithms.

The shown examples confirm that Granular MPM enhancement improves the ability of material point method to accurately replicate of the granular material behaviour for all the versions of explicit MPM. The proposed algorithm also reduces the grid dependency, as the start of material points interaction is no longer grid dependent. In addition, the granular MPM algorithm allows for replication of the steady state during moderate flows of granular materials, where the packing fraction remains unchanged and the transfer of momentum between the grains stops. The Granular MPM also defines criteria for the transition between the different stages of flow as well as the changes of the packing fraction during the flow, improving realism.

The proposed algorithm needs further validation against wide range of experimental data. To that end, the ongoing research aim is to implement Granular MPM formulation into 3D and adapt existing contact algorithms to use with Granular MPM. The future aim is to simulate realistically large-scale problems with much larger amounts of material points and ultimately make the Granular MPM computations easily available to scientists and practicing engineers.

| Table ! | 5 |
|---------|---|
|---------|---|

| Parameters | of | the | fourth | example |
|-------------|----|-----|--------|----------|
| 1 arameters | O1 | unc | iouiui | champic. |

| Simulation | Grid spacing(mm) | | | dt(s) | | $t_{total}(s)$ | t _{total} (s) | |
|-----------------|---|-----|------|------------------|------|--|---|--|
| | 10 | | | Adjusted by Uint | ah | 2 | | |
| Rigid load cell | $ ho\left(rac{\mathrm{kg}}{\mathrm{m}^3} ight)$ 7000 | | | | | | | |
| Toyoura sand | E(kPa) | v | φ(°) | ψ(°) | η | $\rho_{\text{initial}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ | $\rho_{\text{critical}}\left(\frac{\text{kg}}{\text{m}^3}\right)$ | |
| | 5.2 | 0.3 | 31 | 0 | 0.15 | 1379 | 1140 | |



Fig. 19. Flow surface of the model and the experiment.



Fig. 20. Impact force of the experiment and Granular uGIMP simulation.

Declaration of Competing Interest

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

Acknowledgements

The work has been partially funded by the Academy of Finland under the project 'Progressive failure and post-failure modelling of slopes with Generalized Interpolation Material Point Method (GIMP)', decision number is 286628.

Appendix. Algorithms for Granular MPM

This appendix provides explanation for implementing the Granular MPM procedure into the existing MPM codes. This appendix introduces the tasks and algorithms for adding the granular procedure into a 2D original MPM code. Furthermore, the appendix provides details on how the algorithms should be changed for using other versions of MPM.

2D Granular original MPM

This part explains the 3 tasks of a Granular original MPM algorithm. The Granular original MPM first examines the interaction between the material points and decides which of them should take place (pre-task and task 1). Then, it defines the parallel grids based on the necessary interactions (task 2). Next, it reads the data of material points in each grid, runs the original

MPM for that grid, rewrites the data of the material points for the current grid (task 3) and repeats the task 3 until all the grids are completed.

Task 1: Examining interaction between material points

The pre-task and task 1 in Granular original MPM check connection and interaction between the material points and determine the material points that should interact. Table A1 and Table A2 show goal, input, output and algorithm for the pre-task and the task 1.

Task 2: Determining grids

The task 2 in the Granular MPM algorithm determines the grids based on the **interaction** array of task 1. This algorithm finds a material point that has not been assigned to any grid. Then, it moves this material point and all the material points that interact with it to a new grid. Next, it makes sure that all the material points that interact with each other are assigned to the same grid. The algorithm continues this process until all the material points are assigned to one and only one grid.

The task 2 algorithm uses two while loops and a vector for determining the grids. The first while loop assigns the material points to the grids and continues until all the material points are assigned to a grid. The second while loop makes sure that the material points interacting with each other are assigned to the same grid. The second while loop continues as long as the length of the current grid changes. Furthermore, the algorithm assigns material points to a grid by adding the corresponding rows of **interaction** array to that grid. The algorithm keeps track of the assigned material points using a vector named **assign**. Table A3

Table A1

The algorithm for the pre-task.

Pre-task (Optional)

Goal: finding material points that are connected through grid.

Input: number of material points (mpCount), material points position (x_p), number of grid nodes in different directions (**gnCount**), grid dimensions in different directions (L)

Output: connection which is an array containing indexes of material points. Each row of connection has indexes of material points connected with the row number material point through grid (see Fig. 5).

for p = 1; mpCount %this loop finds the grid nodes that each material point send data to. $pElems(p) = floor(x_p(p,1)/L(1) + 1) + (gnCount (1)-1) *(floor(x_p(p,2)/L(2)));$ **Connect _temp** (p,1) = **pElems**(p) + floor (**pElems**(p)/(**gnCount** (1)-1)); **Connect** _ **temp** (p,2) = **Connect** _ **temp** (p,1) + 1; **Connect** _ **temp** (p,3) = **Connect** _ **temp** (p,2) + **gnCount** (1); **Connect** _ **temp** (p,4) = **Connect** _**temp** (p,1) + **gnCount** (1); **Connect** (p,:) = [**Connect**_temp (p,1) **Connect**_temp (p,2) **Connect**_temp (p,3) **Connect**_temp (p,4)]; end for i = 1: mpCount %for each material point, this loop finds material points connected through grid. for j = 1:4 id_n = find(Connect = = Connect (i,j)); id_p = id_n- mpCount *floor((id_n-1)/mpCount); connection {i} = [connection {i} id_p']; end connection{i} = unique(connection {i}); % removing repeated material points end

Table A2

The algorithm for the task 1.

Task 1

Goal: checking interaction between material points. Input: number of material points (mpCount), material points position (xp), initial discretization of material points in different directions (lp), critical volume of material points (V_{cr}) Output: interaction which is an array containing indexes of material points. Each row of interaction has indexes of material points interacting with the row number material point (see Fig. 5). for p = 1: mpCount %this loop calculates domain of interaction (DI) for material points. r1 = lp (p,1)/2; % initial domain of material point r2 = lp (p,2)/2; **DI** (p,1) = sqrt(0.25* (r1/r2) ***V**_{cr}(p)); % domain of interaction **DI** (p,2) = sqrt(0.25* (r2/r1) * $V_{cr}(p)$); end for i = 1: mpCount %finding min and max of DI for material point. ximax=xp (i,1) + DI (p,1); ximin=x_p (i,1) - DI (p,1); yimax= x_p (i,2) + **DI** (p,2); yimin=**x**_p (i,2) - **DI** (p,2); for j = 1:length (connection {i}) %finding min and max of DI for connected material point ximax= x_p (connection {i}(j),1) + DI (connection {i}(j),1); xjmin=x_p (connection {i}(j),1)- DI (connection {i}(j),1); yjmax=x_p (connection {i}(j),2) + DI (connection {i}(j),2); yjmin=x_p (connection {i}(j),2)- DI (connection {i}(j),2); % if domains of interaction intersect, then the material point index should be added to interaction if(ximax > xjmin) && (ximin < xjmax) && (yimax > yjmin) && (yimin < yjmax) interaction {i} = [interaction {i} connection {i}(j)]; elseif(ximax > xjmin) && (ximin < xjmax) && (yjmax > yimin) && (yjmin < yimax) interaction {i} = [interaction {i} connection {i}(j)]; elseif (xjmax > ximin) && (xjmin < ximax) && (yimax > yjmin) && (yimin < yjmax) interaction {i} = [interaction {i} connection {i}(j)]; elseif (xjmax > ximin) && (xjmin < ximax) && (yjmax > yimin) && (yjmin < yimax) interaction {i} = [interaction {i} connection {i}(j)]; end end end

shows goal, input, output and algorithm for the pre-task and the task 2.

Fig. A1 demonstrates how the algorithm of task 2 assigns the material points of Fig. 5 to the parallel grids. The inputs of task 2 algorithm are the number of material points (mpCount = 10) and the **interaction** array of task 1. Furthermore, the algorithm creates an empty **grids** array and an **assign** vector with 10 rows of 0 value (see Fig. A1.a). Each 0 in the **assign** vector represents a material point which is not yet assigned to any grid. Next, the first while loop of the algorithm starts. The first loop finds the unassigned

material points and assigns the first of them to a new grid. For the material points of Fig. 5, the loop finds that there are some unassigned material points and assigns the first of them (material point 1). In order to assign a material point, the algorithm adds the material point corresponding row from the **interaction array** to the current grid. In addition, it marks this material point as assigned by changing its row of **assign** vector to 1 (Fig. A1.b). Then, the second loop of algorithm starts and assigns the material points in the current grid if they are not already assigned. As such, the material point 2 of Fig. 5 gets assigned and adds the material point

Table A3

The algorithm for task 2.

| Ũ | | |
|--------|--|--|
| Task 2 | | |

Goal: determining grid for all the material points Input: number of material points (mpCount), interaction Output: grids which is an array containing indexes of material points. Each row of rows has indexes of interacting material points that interacts with each other (see Fig. A1) grids = cell(mpCount,1); %initializing the grids array assuming the highest possible size. gridCount = 0; % a variable for counting number of parallel grids assign = zeros(mpCount,1); %a vector for identifying material points assigned to a grid. 0 shows unassigned % material point and 1 shows assigned material point **Unassigned** = find(**assign**==0); %finding the unassigned material points while length (**Unassigned**) > 0% first while loop that ensures every material point is assigned to a grid gridCount = gridCount + 1; % assigning the first unassigned material point to a new grid and then marking it as assigned grids {gridCount} = [interaction {Unassigned(1)}]; assign(UnAssigned(1)) = 1; stop = 0; % variable for stopping the second while loop stopCounter = 0; % counter for the second while loop while stop < 1% second while loop that assigns material points in the current grid if not already assigned for j = 1:length (grids{gridCount}) if assign(grids{gridCount}(j)) < 1 grids{gridCount} = [grids{gridCount} interaction{grids{gridCount}(j)}]; assign(grids{gridCount}(j)) = 1; end end grids{gridCount} = unique(grids{gridCount}); % removing repeated indexes %controlling second while loop if stopCounter==length (grids{gridCount}) % if length of grid is same as before, end second while stop = 1: else % if not, save the length of the current grid stopCounter = length (grids{gridCount}); end end % end of the second while loop **Unassigned** = find(**assign**==0); %finding the unassigned material points end % end of the first while loop

grids = grids(~cellfun('isempty',grids)); %removing the empty cells in grids



Fig. A1. Assigning material points of Fig. 7 to parallel grids.

3 and 7 to the first grid. Similarly, these material points get assigned and add material point 6 and 8 to the first grid. The second loop of algorithm ends when the material point 1, 2, 3, 6, 7 and 8 are added to the first grid and marked as assigned (Fig. A1.c). After that, the first loop of algorithm checks if there are any unassigned material points, finds that the material point 4 is the first unassigned one and adds the 4th row of the **interaction** array to the second grid. Finally, the task 2 assigns the material point 10, which means all the material points are assigned now, thus the algorithm ends (Fig. A1.d).

Task 3: Running the original MPM

The task 3 runs the original MPM for material points of each grid based on the **grids** array, checks the state of the material points to use correct constitutive model and applies the possible adjustments of the strain field, as given in section 3.2. For each grid, the algorithm first reads the time-dependent data of that grid, runs the original MPM, applies the right constitutive models for those material points, checks for adjustments of the strain field and then rewrites the time-dependent data.

Task 3- reading data

Table A4

The algorithm for reading data of material points in task 3.

Goal: reading time-dependent data of material point for a grid.

- **Input:** grids, grid number (i_grid), material points position ($\mathbf{x}_{\mathbf{n}}$), material points velocity ($\mathbf{v}_{\mathbf{n}}$), material points stress ($\boldsymbol{\sigma}_{\mathbf{n}}$), ... (all the other time-dependent data of material points)
- Output: position of grid material points (x_{p-grid}), velocity of grid material points (v_{p-grid}), stress of grid material points (σ_{p-grid}), ... (all the other time-dependent data of grid material points)

GmpCount = length (grids{i grid}) :

 $\mathbf{x}_{\mathbf{p-grid}} = \mathbf{zeros}(\text{GmpCount}, 2); \%$ initiation of the grid material points position

v_{p-grid} = zeros(GmpCount,2);

 $\sigma_{p-grid} = zeros(GmpCount,3);$

% the initiation should continue for all the other time-dependent data of material points

for i = 1: GmpCount

 $\mathbf{x}_{\mathbf{p}-\mathbf{grid}}$ (i, 1:2) = $\mathbf{x}_{\mathbf{p}}$ (**grids**{i_grid} (i), 1:2); % reading of the grid material points position

 $\mathbf{v}_{p-\text{grid}}$ (i, 1:2) = \mathbf{v}_p (grids{i_grid} (i), 1:2);

 σ_{p-grid} (i, 1:3) = σ_{p} (grids{i_grid} (i), 1:3);

%the reading should continue for all the other time-dependent data of material points End

Table A5

. The algorithm for rewriting material points data in task 3.

Task 3- rewriting data

Goal: rewriting time-dependent data of material point for a grid.

Input: grids, grid number (i_grid), material points position ($\mathbf{x}_{\mathbf{p}}$), material points velocity ($\mathbf{v}_{\mathbf{p}}$), material points stress ($\sigma_{\mathbf{p}}$), ... (all the other time-dependent data of material points), updated position of grid material points (\mathbf{x}_{p-grid}), updated velocity of grid material points (\mathbf{v}_{p-grid}), updated stress of grid material points (σ_{p-grid}), ... (all the other time-dependent data of grid material points)

Output: material points position (\mathbf{x}_{p}), material points velocity (\mathbf{v}_{p}), material points stress (σ_{p}), ... (all the other time-dependent data of material points) GmpCount = length (grids{i_grid});

for i = 1: GmpCount x_p (grids{i_grid} (i), 1:2) = x_{p-grid} (i, 1:2); $\mathbf{v_p} (grids\{i_grid\} (i), 1:2) = \mathbf{x_{p-grid}} (i, 1:2);$ σ_{p} (grids{i_grid} (i), 1:3)= σ_{p-grid} (i, 1:3);

... %the reading should continue for all the other time-dependent data of material points

end

This task requires two separate algorithms for reading and rewriting time- dependent data based on the grids. These algorithms require the grids and all time-dependent data of material points. Table A4 presents the algorithms for reading timedependent data based on the grids. On the other hand, Table A5 shows the algorithm for rewriting the updated time-dependent data based on the grids. Furthermore, the check for the state of the material points is usually simple and depends on the structure of existing MPM code. Finally, checking for the adjustments of the strain field follows the subsection 3.2.

Granular algorithm for other 2D MPM versions

After some modifications, the other versions of MPM can also use the Granular MPM enhancement. These modifications concern the pre-task and task 1 of the procedure while the task 2 and 3 do not require any adjustments. This part provides some explanations on how these modifications should be done.

Pre-task modification

The pre-task in the Granular MPM procedure finds the material points that are connected through grid. Two material points are connected when they send data to at least one common grid nodes. In the original MPM, the material points send data to the nodes of their current cell. On the other hand, the material points in many other MPM versions can contribute to the neighbouring grid cell as well as the cell they are currently in. Therefore, the connections between the material points depend on the version of MPM and the pre-task of Granular MPM procedure should be modified based on the given version of the base algorithm.

The pre-task algorithm has two main parts. In the first part, the algorithm finds the material points that send data to each grid node. This part of the algorithm is dependent on the version of MPM thus requires modification when other versions of MPM are used. Furthermore, this part is a common step in different versions of MPM and the algorithm for it usually already exists. The second part of algorithm, uses the first part to finds material points that send data to at least one common grid nodes, which means they are connected. This second part is not dependent to the version of MPM thus this part of table A1 can be used for any other methods.

Task 1 modification

Task 1 checks the material points domains of interaction and determines the material points that should interact. The previous algorithm for task 1 (Table A1) assumes a rectangular shape for the domain of interactions. As such, the algorithm can be used for original MPM as well as GIMP versions (uGIMP and cpGIMP). On the other hand, CPDI methods have nonrectangular material point domains, which results in nonrectangular domains of interaction, and require modification in task 1 algorithm.

In this research, the CPDI version of task 1 do a step by step search for an intersection point between the domains of interaction. For each pair of domain of interaction, the algorithm starts from one side of the domain of main material point (the material point corresponding to the rows of interaction array) and checks all the sides of the other domains of interaction to find an intersection point. If such a point is found, the algorithm adds the material points to the **interaction** array and stops checking this pair. If an intersection point is not found, the algorithm checks another side of the domain of main material point. The check stops when all the sides of the main material point are checked and no intersection is found.



Fig. A2. Schematic of the 2D CPDI version of task 1.

Fig. A2 illustrates a schematic of how the 2D CPDI version of task 1 works. In this figure, the material points are numbered with **<u>a</u>** (the main material point) and **b** (the other material point), the domains of interaction for the material points are marked with thick dash lines and the corners of domains have counter clock wise indexes of 1 to 4. First, the algorithm finds the points of intersection between the side **a12** and **b12** and checks to see if this intersection point is on **a12** as well as **b12**. Since this point is not on **b12**, it is not a point of intersection between the domains and thus the algorithm continues. Next, the algorithm does the same check between a12 with b23, b34 and b41 and finds that there are no intersections between the domains. After that, the algorithm does the intersection check between a23 with b12, b23, b34 and b41, finds no intersection between the domains and continues checking. Finally, the algorithm checks the intersection point between **a34** and **b12** and finds that this point is on **a34** as well as **b12**. Therefore, this point is an intersection between the domains, the algorithm adds **b** to the **interaction** array and stops checking this pair of material points.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compstruc.2021.106545.

References

- Redaelli I, di Prisco C, Vescovi D. A visco-elasto-plastic model for granular materials under simple shear conditions. Int. J. Numer. Anal. Methods Geomech. Jan. 2016;40(1):80–104. <u>https://doi.org/10.1002/nag.2391</u>.
- [2] Kamrin K. Nonlinear elasto-plastic model for dense granular flow. Int. J. Plast. 2010;26(2):167–88. <u>https://doi.org/10.1016/j.ijplas.2009.06.007</u>.
- [3] Dunatunga S, Kamrin K. Continuum modelling and simulation of granular flows through their many phases. J. Fluid Mech. Sep. 2015;779:483–513. https://doi.org/10.1017/jfm.2015.383.
- [4] Campbell CS. Granular material flows An overview. Powder Technol. Mar. 2006;162(3):208–29. <u>https://doi.org/10.1016/j.powtec.2005.12.008</u>.
- [5] Sulsky D, Zhou S-J, Schreyer HL. Application of a particle-in-cell method to solid mechanics. Comput. Phys. Commun. May 1995;87(1-2):236-52. <u>https://doi.org/10.1016/0010-4655(94)00170-7</u>.
 [6] Sulsky D, Schreyer HL. Axisymmetric form of the material point method with
- [6] Sulsky D, Schreyer HL. Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems. Comput. Methods Appl. Mech. Eng. Dec. 1996;139(1–4):409–29. <u>https://doi.org/10.1016/S0045-7825</u> (96)01091–2.
- [7] Bardenhagen SG, Kober EM. The generalized interpolation material point method. C. - Comput. Model. Eng. Sci. 2004;5(6):477–95. <u>https://doi.org/ 10.3970/cmes.2004.005.477</u>.

- [8] S. Dunatunga and K. Kamrin, Continuum modeling of projectile impact and penetration in dry granular media, J. Mech. Phys. Solids, vol. 100, no. December 2016, pp. 45–60, Mar. 2017, doi: 10.1016/j.jmps.2016.12.002.
- [9] Redaelli I, Ceccato F, di Prisco C, Simonini P. Solid-fluid Transition in Granular Flows: MPM Simulations with a New Constitutive Approach. Procedia Eng. 2017;175:80–5. <u>https://doi.org/10.1016/j.proeng.2017.01.028</u>.
- [10] Sulsky D, Chen Z, Schreyer HLL. A particle method for history-dependent materials. Comput. Methods Appl. Mech. Eng. Sep. 1994;118(1-2):179–96. https://doi.org/10.1016/0045-7825(94)90112-0.
- [11] J. Guilkey et al., Uintah user guide, in Technical Report, SCI Institute Technical Report, 2009.
- [12] Wallstedt PC, Guilkey JE. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. J. Comput. Phys. Nov. 2008;227(22):9628–42. <u>https://doi.org/10.1016/j.jcp.2008.07.019</u>.
- [13] Sadeghirad A, Brannon RM, Burghardt J. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. Int. J. Numer. Methods Eng. Jun. 2011;86(12):1435–56. https://doi.org/10.1002/nme.3110.
- [14] Sadeghirad A, Brannon RM, Guilkey JE. Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. Int. J. Numer. Methods Eng. Sep. 2013;95(11):928–52. https://doi.org/10.1002/nme.4526.
- [15] Wang L et al. On the use of domain-based material point methods for problems involving large distortion. Comput. Methods Appl. Mech. Eng. 2019. https://doi.org/10.1016/j.cma.2019.07.011.
- [16] Seyedan S, Sołowski WT. Enhancing Constitutive Models for Soils: Adding the Capability to Model Nonlinear Small Strain in Shear. Adv. Civ. Eng. Apr. 2019;2019:1–11. <u>https://doi.org/10.1155/2019/6016350</u>.
- [17] Benz T. Small-Strain Stiffness of Soils and its Numerical Consequences. Universität Stuttgart 2007.
- [18] Bagnold RA. Experiments on a gravity-free dispersion of large solid spheres in a Newtonian fluid under shear. Proc. R. Soc. London. Ser. A. Math. Phys. Sci. Aug. 1954;225(1160):49–63. <u>https://doi.org/10.1098/rspa.1954.0186</u>.
- [19] Lois G, Lemaître A, Carlson JM. Numerical tests of constitutive laws for dense granular flows. Phys. Rev. E Nov. 2005;72(5):. <u>https://doi.org/10.1103/ PhysRevE.72.051303</u>051303.
- [20] Silbert LE, Ertaş D, Grest GS, Halsey TC, Levine D, Plimpton SJ. Granular flow down an inclined plane: Bagnold scaling and rheology. Phys. Rev. E Oct. 2001;64(5):. <u>https://doi.org/10.1103/PhysRevE.64.051302</u>051302.
- [21] da Cruz F, Emam S, Prochnow M, Roux J-N, Chevoir F. Rheophysics of dense granular materials: Discrete simulation of plane shear flows. Phys. Rev. E Aug. 2005;72(2):. <u>https://doi.org/10.1103/PhysRevE.72.021309</u>021309.
- [22] Hanes DM, Inman DL. Observations of rapidly flowing granular-fluid materials. J. Fluid Mech. Jan. 1985;150:357–80. <u>https://doi.org/10.1017/ S0022112085000167</u>.
- [23] Nairn JA. Material point method calculations with explicit cracks. C. Comput. Model. Eng. Sci. 2003;4(6):649–63. <u>https://doi.org/10.3970/ cmes.2003.004.649</u>.
- [24] S. Seyedan and W. T. Sołowski, Estimation of Granular Flow Impact Force on Rigid Wall Using Material Point Method, in: V International Conference on Particle-based Methods, 2017.
- [25] Wójcik M, Sondej M, Rejowski K, Tejchman J. Full-scale experiments on wheat flow in steel silo composed of corrugated walls and columns. Powder Technol. Apr. 2017;311:537–55. <u>https://doi.org/10.1016/i.powtec.2017.01.066</u>.
- [26] Moriguchi S, Borja RI, Yashima A, Sawada K. Estimating the impact force generated by granular flow on a rigid obstruction. Acta Geotech. Mar. 2009;4 (1):57-71. <u>https://doi.org/10.1007/s11440-009-0084-5</u>.