



ARTICLE

# A Consistent Time Level Implementation Preserving Second-Order Time Accuracy via a Framework of Unified Time Integrators in the Discrete Element Approach

Tao Xue<sup>1</sup>, Yazhou Wang<sup>2</sup>, Masao Shimada<sup>2</sup>, David Tae<sup>2</sup>, Kumar Tamma<sup>2,\*</sup> and Xiaobing Zhang<sup>1,\*</sup>

<sup>1</sup>School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China

<sup>2</sup>Department of Mechanical Engineering, University of Minnesota-Twin Cities, Minneapolis, 55455, USA

\*Corresponding Authors: Kumar Tamma. Email: ktamma@umn.edu; Xiaobing Zhang. Email: zhangxb680504@163.com

Received: 24 January 2022 Accepted: 21 April 2022

## ABSTRACT

In this work, a consistent and physically accurate implementation of the general framework of unified second-order time accurate integrators via the well-known GSSSS framework in the Discrete Element Method is presented. The improved tangential displacement evaluation in the present implementation of the discrete element method has been derived and implemented to preserve the consistency of the correct time level evaluation during the time integration process in calculating the algorithmic tangential displacement. Several numerical examples have been used to validate the proposed tangential displacement evaluation; this is in contrast to past practices which only seem to attain the first-order time accuracy due to inconsistent time level implementation with different algorithms for normal and tangential directions. The comparisons with the existing implementation and the superiority of the proposed implementation are given in terms of the convergence rate with improved numerical accuracy in time. Moreover, several schemes via the unified second-order time integrators within the framework of the GSSSS family have been carried out based on the proposed correct implementation. All the numerical results demonstrate that using the existing state-of-the-art implementation reduces the time accuracy to be first-order accurate in time, while the proposed implementation preserves the correct time accuracy to yield second-order.

## KEYWORDS

Computational dynamics; time integration; Discrete Element Method (DEM)

## 1 Introduction

In order to numerically describe the behavior of granular materials (not continuous body) represented as a collection of large number of distinct particles, the discrete/distinct element method (DEM), proposed in [1], is being widely used for its ease of implementation and versatility. In the DEM technique, each particle element can undergo large displacement and rotation, and the deformations of individual particle elements are assumed to be small, compared with their displacements; hence we can treat the elements as rigid bodies and the deformation is usually described in terms of the relative normal displacement and the relative tangential displacement between the particles in contact.



Consequently, the evaluation of those two displacements has a significant influence on both the dynamic response of particles and the numerical accuracy of the simulation.

One of the common applications for the DEM is particle packing. Various packing methods using DEM have been studied previously such as, Formation of a pile, i.e., sand piling process [2–6], pouring or depositing particles under gravity [7–12]. The DEM particle packing is also used in pharmaceutical field for powder tableting [13,14]. Another field of application using DEM is simulating particle flow; and a commonly studied example is ball mill mixer simulation; some of which include, horizontally rotating drums [15–18], tumbling ball mill [19–21]. Drum and mill are commonly used in industries for mixing, drying or coating subjects; thus, there is extensive interest among researchers in this study. Besides, the DEM is still studied and utilized in recent studies such as solid-liquid flow simulation [22], impact disruption of cohesive soft sphere simulation [23], granular shear flows of dry flexible fibers [24] and non-spherical complex granular simulation [25].

Furthermore, the DEM not only simulates granular flow, but it can also be used for solid body analysis and assessing fracture as studied in [26]. Recently, the DEM has been expanded to be able to solve multi-physics applications where granular particles interact with a continuous body, be it structural or thermodynamics. An Extended Discrete Element Method (XDEM) has been proposed in [27] which couples continuous body analysis method and DEM together without losing small scale information from commonly used volume averaging concept. The XDEM allows those microscale information from particles to accurately influence a macroscopic body. The XDEM opens up a wide variety of possible physical applications in many fields of study.

In transient dynamic simulation of the DEM type problems, contact forces and particle motion need to be evaluated and updated at every time step. The widely used time integrations in the DEM numerical simulations include the explicit Euler method, fourth-order Runge-Kutta (RK-4) method, and central difference method/velocity verlet method, etc. It is straightforward to implement these methods in the DEM simulation; however, these explicit time stepping techniques possess several numerical disadvantages in addition to the conditionally stable features; for example, the Euler methods are first-order accurate in time, RK-4 method requires additional computations of high-order derivatives of differential variables [28,29], and the central difference method is a non-dissipative scheme, i.e., it cannot introduce numerical dissipation if one needs to. Also, past practices poorly model the tangential physics of particle interactions. Recently, a unified time integration framework, under the umbrella of the so-called generalized single-step single-solve (GSSSS) family of algorithms, including not only implicit, but also explicit single-step single-solve algorithms, which can also be written in a linear three-step form, has been proposed in [30]. This single simulation tool kit, which has been originally developed in the field of computational structural dynamics, encompasses not only most existing algorithms that are second-order time accurate and are commonly used over the past 50 years or so, but it additionally inherits a wide variety of new implicit and explicit time integration schemes also with second-order time accuracy, including the explicit central difference method as a special case. To implement these time integrations in DEM and preserve the numerical accuracy of the algorithms, the contact force is required to be evaluated in a consistent way with the selected time integrations in both normal and tangential directions. According to our knowledge, most of the existing literature have been embedding the first-order accurate updating evaluation of tangential displacement into the first/second-order accurate time integrations. The specifics are shown in [Section 2](#), leading to reduced accuracy.

In this work, the main objective is to show the consistent implementation of the GSSSS family of algorithms to the simulation of granular assemblies with DEM while retaining all the advantages

of numerous methods within the present framework of algorithms. It is worthy to mention that one major drawback of the DEM technique, as well as the other particle based methods or meshless methods [31–35], is contact detection or neighboring particle search. To circumvent this issue, the linked list is employed to detect the contacting neighbor particles. In addition, Open Multi-Processor (Open MP) is implemented to reduce the simulation computational time. The outline of the rest of the paper is as follows. Section 2 presents the basic theory of DEM formulations and both the linear and the nonlinear model of contact forces and torques are presented in detail. The GSSSS unified time integrators are highlighted in Subsection 2.2 and the specifics of the proposed consistent tangential displacement evaluation is derived and compared with the existing tangential displacement evaluation technique carefully. The effectiveness of the proposed consistent tangential displacement evaluation is demonstrated and illustrated using simple numerical examples in Section 3. Finally, the conclusions are drawn and presented in Section 4. This study shows that the DEM with explicit schemes within the GSSSS family of algorithms achieves and consistently preserves second-order time accuracy; the consistent time level implementation and preserving the second-order time accuracy for the presented results for a wide variety of time integrators in the present unified framework include features such as no overshoot in displacement or no overshoot in velocity or no overshoot in both displacement and velocity for a given set of initial conditions of the particles and differs from algorithm to algorithm.

## 2 Discrete Element Method with the Explicit GSSSS Family of Algorithms

### 2.1 Basic DEM Formulations

DEM is a numerical simulation technique for a system of small “distinct” particles/elements, such as powders and granular materials. For the sake of simplicity, we assume the shape of every particle/element in a system is a disk in the two-dimensional Euclidean space  $\mathbb{E}^2$  or a sphere in the three dimensional Euclidean space  $\mathbb{E}^3$ , although DEM can treat any shapes of elements in general, see [36]. In DEM, we assume that each particle is *rigid*, i.e., no deformation; however, particles are allowed to *overlap* slightly one another at the contact points.

Let  $\mathbf{q}_i(t) : [0, \infty) \rightarrow \mathbb{R}^{n_{\text{dim}}}$  denote the position component vector of the  $i^{\text{th}}$  particle ( $i = 1, 2, \dots, n_{\text{ele}}$ ) with respect to the Cartesian coordinate frame  $\mathcal{B}$ , where  $n_{\text{ele}}$  and  $n_{\text{dim}}$  denote the number of particles/elements in the system and number of dimension, respectively. Suppose that the  $i^{\text{th}}$  and  $j^{\text{th}}$  particles are in contact at time  $t$  as shown in Fig. 1a. The overlap (penetration) between particle  $i$  and  $j$ , which is denoted by  ${}_n\delta_{ij} \in \mathbb{R}$ , is defined as

$${}_n\delta_{ij} = (r_i + r_j) - d_{ij} \quad (1)$$

where  $r_i$  and  $r_j$  denote the radii of the  $i^{\text{th}}$  and  $j^{\text{th}}$  particles, respectively; and  $d_{ij}$  is the center distance between the particles  $i$  and  $j$  and  $d_{ij} := \|\mathbf{q}_j - \mathbf{q}_i\| = \sqrt{(\mathbf{q}_j - \mathbf{q}_i) \cdot (\mathbf{q}_j - \mathbf{q}_i)}$  is the center distance between the particles  $i$  and  $j$ . We assume the overlap is sufficiently small relatively to the particle sizes. If the inequality  $d_{ij} < r_i + r_j$  is met, the overlap  ${}_n\delta_{ij} > 0$  occurs; and the contact forces between particles  $i$  and  $j$  are taken into consideration. Define the unit *normal* vector between these particles as  $\mathbf{n}_{ij} := \frac{\mathbf{q}_j - \mathbf{q}_i}{\|\mathbf{q}_j - \mathbf{q}_i\|} = \frac{\mathbf{q}_j - \mathbf{q}_i}{d_{ij}}$ ,  $\|\mathbf{n}_{ij}\| = 1$ . The translational velocity component vectors of particles  $i$  and  $j$  (at the center points  $\mathbf{C}_i \in \mathbb{E}^{n_{\text{dim}}}$  and  $\mathbf{C}_j \in \mathbb{E}^{n_{\text{dim}}}$ , respectively) with respect to  $\mathcal{B}$  are given as the (total) time derivatives of the position components vectors, i.e.,  $\mathbf{v}_i = \dot{\mathbf{q}}_i$  and  $\mathbf{v}_j := \dot{\mathbf{q}}_j$ , respectively ( $\dot{\square} \equiv d\square/dt$  denotes the total time derivative of  $\square$ ). Let  $\boldsymbol{\omega}_i$  and  $\boldsymbol{\alpha}_i$  denote the angular velocity and angular acceleration vectors of particle  $i$ , respectively (for  $i = 1, 2, \dots, n_{\text{ele}}$ ). Therefore, the velocity at point  $\mathbf{P}_i \in \mathbb{E}^{n_{\text{dim}}}$ , as shown in Fig. 1a,  $\mathbf{v}_{\mathbf{P}_i}$ , can be written as  $\mathbf{v}_{\mathbf{P}_i} = \mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i = \mathbf{v}_i + r_i \boldsymbol{\omega}_i \times \mathbf{n}_{ij}$ , where  $\mathbf{r}_i = r_i \mathbf{n}_{ij}$ . Similarly,  $\mathbf{v}_{\mathbf{P}_j}$ , the velocity of point  $\mathbf{P}_j$ , can be written as  $\mathbf{v}_{\mathbf{P}_j} = \mathbf{v}_j + \boldsymbol{\omega}_j \times \mathbf{r}_j = \mathbf{v}_j - r_j \boldsymbol{\omega}_j \times \mathbf{n}_{ij}$ , where  $\mathbf{r}_j = -r_j \mathbf{n}_{ij}$ . Hence, the relative velocity

vector at point  $\mathbf{P}_i$  with respect to point  $\mathbf{P}_j$  is obtained as

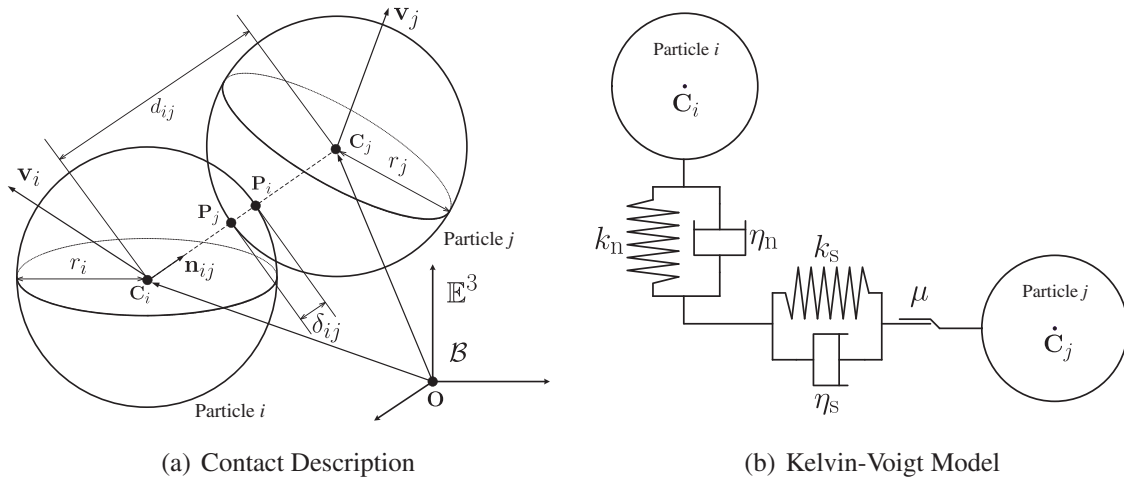
$$\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} = \mathbf{v}_{\mathbf{P}_i} - \mathbf{v}_{\mathbf{P}_j} = \mathbf{v}_{ij} + (r_i\boldsymbol{\omega}_i + r_j\boldsymbol{\omega}_j) \times \mathbf{n}_{ij} \quad (2)$$

where  $\mathbf{v}_{ij} := \mathbf{v}_i - \mathbf{v}_j$  denotes the relative translational velocities of particle  $i$  with respect to  $j$ . Since the projection of  $\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$  parallel to  $\mathbf{n}_{ij}$  is given as

$${}_n\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} = (\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij} = (\mathbf{v}_{ij} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij}, \quad (3)$$

the projection of  $\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$  perpendicular to  $\mathbf{n}_{ij}$ , which is obtained by subtracting  ${}_n\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$  from  $\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$ , can be written as

$${}_s\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} = \mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} - ({}_{n}\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j})\mathbf{n}_{ij} \quad (4)$$



**Figure 1:** (a) Pictorial illustration of particles  $i$  and  $j$  in contact; and (b) the Kelvin-Voigt model for particles  $i$  and  $j$  in contact.  $\mathbf{O} \in \mathbb{E}^3$  is an origin of a cartesian coordinate frame  $\mathcal{B}$  for  $\mathbb{E}^3$  with a right-handed orthonormal basis for the associated vector space

Note that  ${}_n\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$  and  ${}_s\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$  are the relative velocity vector of point  $\mathbf{P}_i$  with respect to point  $\mathbf{P}_j$  in the normal ( $\mathbf{n}_{ij}$ ) and tangential ( $\mathbf{t}_{ij}$ ) directions, respectively. Substituting Eqs. (2) into (4) yields:

$${}_s\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} = \mathbf{v}_{ij} - (\mathbf{v}_{ij} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij} + (r_i\boldsymbol{\omega}_i + r_j\boldsymbol{\omega}_j) \times \mathbf{n}_{ij} \quad (5)$$

$$\text{Hence, } \mathbf{t}_{ij} = \frac{{}_s\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}}{\|{}_s\mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}\|}, \quad \|\mathbf{t}_{ij}\| = 1.$$

Similarly, we can derive the relative acceleration vectors of point  $\mathbf{P}_i$  with respect to point  $\mathbf{P}_j$  in the normal ( $\mathbf{n}_{ij}$ ) and tangential ( $\mathbf{t}_{ij}$ ) directions. Since the accelerations at  $\mathbf{P}_i$  and  $\mathbf{P}_j$  are given as  $\mathbf{a}_{\mathbf{P}_i} = \mathbf{a}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_i) + \boldsymbol{\alpha}_i \times \mathbf{r}_i = \mathbf{a}_i + r_i\boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{n}_{ij}) + r_i\boldsymbol{\alpha}_i \times \mathbf{n}_{ij}$  and  $\mathbf{a}_{\mathbf{P}_j} = \mathbf{a}_j + \boldsymbol{\omega}_j \times (\boldsymbol{\omega}_j \times \mathbf{r}_j) + \boldsymbol{\alpha}_j \times \mathbf{r}_j = \mathbf{a}_j - r_j\boldsymbol{\omega}_j \times (\boldsymbol{\omega}_j \times \mathbf{n}_{ij}) - r_j\boldsymbol{\alpha}_j \times \mathbf{n}_{ij}$ , where  $\mathbf{a}_i \in \mathbb{R}^{n_{\text{dim}}}$  and  $\mathbf{a}_j \in \mathbb{R}^{n_{\text{dim}}}$  are the linear acceleration vectors of particles  $i$  and  $j$ , respectively, the relative acceleration vector at point  $\mathbf{P}_i$  with respect to point  $\mathbf{P}_j$  may be obtained as

$$\begin{aligned}
 \mathbf{a}_{\mathbf{P}_i/\mathbf{P}_j} &= \mathbf{a}_{\mathbf{P}_i} - \mathbf{a}_{\mathbf{P}_j} \\
 &= \mathbf{a}_{ij} + (r_i \boldsymbol{\alpha}_i + r_j \boldsymbol{\alpha}_j) \times \mathbf{n}_{ij} \\
 &\quad + r_i [(\boldsymbol{\omega}_i \otimes \boldsymbol{\omega}_i) - (\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_i) \mathbf{I}_{n_{\dim}}] \mathbf{n}_{ij} + r_j [(\boldsymbol{\omega}_j \otimes \boldsymbol{\omega}_j) - (\boldsymbol{\omega}_j \cdot \boldsymbol{\omega}_j) \mathbf{I}_{n_{\dim}}] \mathbf{n}_{ij}
 \end{aligned} \tag{6}$$

where  $\mathbf{a}_{ij} := \mathbf{a}_i - \mathbf{a}_j$  denotes the relative translational accelerations of particle  $i$  with respect to  $j$ , and  $\mathbf{I}_{n_{\dim}} \in \mathbb{R}^{n_{\dim} \times n_{\dim}}$  denotes the identity tensor. Hence, the relative acceleration vector of point  $\mathbf{P}_i$  with respect to point  $\mathbf{P}_j$  in the normal ( $\mathbf{n}_{ij}$ ) and tangential ( $\mathbf{t}_{ij}$ ) directions are obtained as

$${}_n \mathbf{a}_{\mathbf{P}_i/\mathbf{P}_j} = (\mathbf{a}_{\mathbf{P}_i/\mathbf{P}_j} \cdot \mathbf{n}_{ij}) \mathbf{n}_{ij} \quad \text{and} \quad {}_s \mathbf{a}_{\mathbf{P}_i/\mathbf{P}_j} = (\mathbf{a}_{\mathbf{P}_i/\mathbf{P}_j} \cdot \mathbf{t}_{ij}) \mathbf{t}_{ij} \tag{7}$$

respectively.

### Contact Forces and Torques

The resultant contact force and torque acting on particle  $i$  are defined as the summations of the contact forces and torques, respectively, acting on particle  $i$  from all particles  $j (\neq i)$  that are in contact with particle  $i$ . The contact forces due to particle(s)  $j$  can be modeled as follows.

**Linear Model:** The contact force acting on particle  $i$  from particle  $j$  can be uniquely decomposed into the contact forces in the normal direction ( $\mathbf{n}_{ij}$ ) and tangential direction ( $\mathbf{t}_{ij}$ ) as  $\mathbf{f}_{ij} = {}_n \mathbf{f}_{ij} + {}_s \mathbf{f}_{ij} \in \mathbb{R}^{n_{\dim}}$ . In DEM, the contact forces are usually modeled by the *Kelvin-Voigt model*, as shown in Fig. 1b; hence, both the normal and tangential contact forces on particle  $i$  due to particle  $j$  have the stiffness/spring and damping terms, and the friction on the surface is also considered. For the normal contact force,  ${}_n \mathbf{f}_{ij}$ , using the stiffness and damping constants in the normal direction,  $k_n$  and  $\eta_n$ , respectively, we have

$${}_n \mathbf{f}_{ij} = k_n (-{}_n \boldsymbol{\delta}_{ij}) + \eta_n (-{}_n \mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}) = - [k_n ({}_n \boldsymbol{\delta}_{ij}) + \eta_n (\mathbf{v}_{ij} \cdot \mathbf{n}_{ij})] \mathbf{n}_{ij} \tag{8}$$

where  ${}_n \boldsymbol{\delta}_{ij}$  is the relative displacement vector between particle  $i$  and  $j$  in the normal direction, i.e.,  ${}_n \boldsymbol{\delta}_{ij} \in \mathbb{R}^{n_{\dim}}$ , may be defined as the displacement while the particles  $i$  and  $j$  are in contact for the time interval,  $[t_{c_1}, t_{c_2}] \subset [0, \infty)$ , i.e.,

$${}_n \boldsymbol{\delta}_{ij} = \int_{t_{c_1}}^{t_{c_2}} {}_n \mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} dt = {}_n \boldsymbol{\delta}_{ij} \mathbf{n}_{ij} = [(r_i + r_j) - d_{ij}] \mathbf{n}_{ij} \tag{9}$$

The viscous damping coefficient  $\eta_n$  in Eq. (8) is usually selected as  $\eta_n = -2 \ln C_R \sqrt{\frac{\hat{m}_{ij} k_n}{\pi^2 + (\ln C_R)^2}}$ , where  $C_R \in [0, 1] \subset \mathbb{R}_+$  denotes the coefficient of restitution, and  $\hat{m}_{ij}$ , which is sometimes called the reduced mass, is defined as

$$\hat{m}_{ij} := [m_i^{-1} + m_j^{-1}]^{-1} = \frac{m_i m_j}{m_i + m_j} \tag{10}$$

Similarly, the tangential contact force,  ${}_s \mathbf{f}_{ij}$ , can be written as

$${}_s \mathbf{f}_{ij} = k_s (-{}_s \boldsymbol{\delta}_{ij}) + \eta_s (-{}_s \mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}) \tag{11}$$

where  $k_s$  and  $\eta_s$  are the stiffness and damping coefficients in the tangential direction, respectively. The tangential relative velocity,  ${}_s \mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j}$ , in the equation above is given by Eq. (5). The relative displacement vector between particles  $i$  and  $j$  in the tangential direction, i.e.,  ${}_s \boldsymbol{\delta}_{ij} \in \mathbb{R}^{n_{\dim}}$ , may be defined as the displacement while the particles  $i$  and  $j$  are in contact for the time interval,  $[t_{c_1}, t_{c_2}] \subset [0, \infty)$ , i.e.,

$${}_s \boldsymbol{\delta}_{ij} = \int_{t_{c_1}}^{t_{c_2}} {}_s \mathbf{v}_{\mathbf{P}_i/\mathbf{P}_j} dt \tag{12}$$

We usually take  $\eta_s \simeq \eta_n$  unless otherwise the values of the damping coefficients are critical in the simulation. If the condition  $\|s\mathbf{f}_{ij}\| > \mu\|n\mathbf{f}_{ij}\|$ , where  $\mu > 0$  denotes the coefficient of friction, is met, the “slip” between particles  $i$  and  $j$  on the surface at the contact point occurs, and therefore, the tangential contact force, given in Eq. (11), is replaced with the frictional force,  $s\mathbf{f}_{ij} = -\mu\|n\mathbf{f}_{ij}\|\mathbf{t}_{ij}$  with  $n\mathbf{f}_{ij}$ , given in Eq. (8). Once we get all the contact forces,  $\mathbf{f}_{ij}$ , acting on particle  $i$ , we can finally obtain the resultant

contact force on particle  $i$  as  $\mathbf{f}_i = \sum_{j \neq i}^{n_{\text{cont}}^i} \mathbf{f}_{ij}$ , where  $n_{\text{cont}}^i$  denotes the number of the particles that are in contact with particle  $i$  at time  $t$ . On the other hand, the resultant torque on particle  $i$  can be written in the form  $\mathbf{h}_i = \sum_{j \neq i}^{n_{\text{cont}}^i} \mathbf{r}_i \times_s \mathbf{f}_{ij}$ .

**Nonlinear Model:** There exist several nonlinear models developed for DEM in the literature to obtain more accurate, realistic simulation results. A common nonlinear model of DEM, proposed in [37], is based on the classical Hertzian contact theory and Mindlin theory [38,39]. The normal contact force,  $n\mathbf{f}_{ij}$ , based on the Hertzian contact theory for two spheres, is still defined in the same form as Eq. (8),

$$n\mathbf{f}_{ij} = -k_n(n\delta_{ij}) - \eta_n(n\mathbf{v}_{P_i/P_j}) \quad (13)$$

However, as opposed to the linear model,  $n\delta_{ij}$  and stiffness  $k_n$  are defined as

$$n\delta_{ij} = (n\delta_{ij}^{3/2}) \mathbf{n}_{ij} \quad (14)$$

$$k_n = \frac{4}{3} \sqrt{\hat{r}_{ij}} \hat{E}_{ij} \quad (15)$$

with  $n\delta_{ij}$ , defined as in Eq. (1),  $\hat{r}_{ij} := [r_i^{-1} + r_j^{-1}]^{-1} = \frac{r_i r_j}{r_i + r_j}$ , and  $\hat{E}_{ij} := \left[ \frac{1-v_i^2}{E_i} + \frac{1-v_j^2}{E_j} \right]^{-1} = \frac{E_i E_j}{(1-v_i^2)E_j + (1-v_j^2)E_i}$  where  $E_i$  and  $\nu_i$  are the Young modulus and Poisson ratio of particle  $i$ , respectively. For two spherical particles with the same radius ( $r = r_i = r_j$ ) and the same material properties ( $E = E_i = E_j$  and  $\nu = \nu_i = \nu_j$ ), the stiffness defined in Eq. (15) may be written in the form,  $k_n = \frac{E\sqrt{2r}}{3(1-\nu^2)}$ . In the case of a spherical particle  $i$  with a wall, we assume  $r_j \rightarrow +\infty$ , i.e.,  $\hat{r} \rightarrow r_i$ ; and thereby,  $k_n = \frac{4\sqrt{r_i}}{3} \frac{E_i E_{\text{wall}}}{(1-\nu_i^2)E_{\text{wall}} + (1-\nu_{\text{wall}}^2)E_i}$ , where  $E_{\text{wall}}$  and  $\nu_{\text{wall}}$  are the Young modulus and Poisson ratio of the wall. According to [37], the normal damping coefficient  $\eta_n$  in Eq. (13) is suggested to be  $\eta_n = \kappa \sqrt{\hat{m}_{ij} k_n} (n\delta_{ij}^{1/4})$ , in which  $\hat{m}_{ij}$  is as defined in Eq. (10). For  $\kappa > 0$ , which can be regarded as dependent on the coefficient of restitution between two sphere particles or between a sphere particle and a wall, it is to be determined empirically by analysts (Usually,  $\alpha \in [0.01, 1]$ ). Note that the  $n\mathbf{v}_{P_i/P_j}$  in Eq. (13) is still defined as given in Eq. (3).

On the other hand, the tangential contact force,  $s\mathbf{f}_{ij}$ , for a nonlinear model is defined, based on the Hertzian contact theory and Mindlin theory [38,39] with the assumption that “no-slip” is allowed between two sphere particles  $i$  and  $j$ , as

$$s\mathbf{f}_{ij} = -k_s(s\delta_{ij}) - \eta_s(s\mathbf{v}_{P_i/P_j}) \quad (16)$$

with

$$k_s = \left[ \frac{1}{8b_{ij}} \frac{2-\nu_i}{G_i} + \frac{2-\nu_j}{G_j} \right]^{-1}, \quad b_{ij} := \sqrt{\hat{r}_{ij} n\delta_{ij}} \quad (17)$$

where  $G_i := E_i/[2(1+\nu_i)]$  is called the shear modulus of particle  $i$ , and  $b_{ij}$  is the radius of a contact area between sphere particles  $i$  and  $j$ . For two sphere particles with the same radius ( $r = r_i = r_j$ ) and the

same material properties ( $E = E_i = E_j$  and  $\nu = \nu_i = \nu_j$ , i.e.,  $G := G_i = G_j$ ), we have  $k_s = \frac{2G}{2-\nu} \sqrt{2r} \binom{n}{\delta_{ij}}$ . In the case of the contact between sphere particle  $i$  and a wall, we assume the wall ( $r_j \rightarrow +\infty$ ) is rigid, i.e.,  $E_j \rightarrow +\infty$ , since the elastic tangential displacement of the wall due to the contact with particle  $i$  is usually negligible; hence, Eq. (17) leads to the following expression:  $k_s = \frac{8G_i}{2-\nu_i} \sqrt{r_i} \binom{n}{\delta_{ij}}$ . For most applications, we take the same value for the normal and tangential damping coefficients:  $\eta_s \simeq \eta_n$ .

## 2.2 Transient Analysis: A Consistent Time Level Implementation of Unified Time Integrators

Applying the Newton-Euler equation to particle  $i$  for  $i = 1, 2, \dots, n_{\text{ele}}$ , we get

$$\mathbf{M}_i \mathbf{a}_i = \mathbf{f}_i^{\text{appl}} \tag{18}$$

$$\mathbf{J}_i \boldsymbol{\alpha}_i = \mathbf{h}_i^{\text{appl}} \tag{19}$$

where  $\mathbf{M}_i = m_i \mathbf{I}_{n_{\text{dim}}} \in \mathbb{R}^{n_{\text{dim}} \times n_{\text{dim}}}$ , with the identity tensor  $\mathbf{I}_{n_{\text{dim}}} \in \mathbb{R}^{n_{\text{dim}} \times n_{\text{dim}}}$ , and  $\mathbf{a}_i \in \mathbb{R}^{n_{\text{dim}}}$  denote the mass tensor and the acceleration vector of particle  $i$ , respectively; and  $\mathbf{J}_i \in \mathbb{R}^{n_{\text{dim}} \times n_{\text{dim}}}$  and  $\boldsymbol{\alpha}_i \in \mathbb{R}^{n_{\text{dim}}}$  denote the inertia tensor (with respect to the center of mass) and the angular acceleration vector of particle  $i$ , respectively. The right-hand side of Eq. (18), i.e.,  $\mathbf{f}_i^{\text{appl}} \in \mathbb{R}^{n_{\text{dim}}}$ , is the applied force on particle  $i$ , and it includes any force acting on the particle, including conservative and nonconservative forces, as well as the contact forces described in the previous subsection. Likewise,  $\mathbf{h}_i^{\text{appl}} \in \mathbb{R}^{n_{\text{dim}}}$  in Eq. (19) is the applied torque vector of particle  $i$ .

In a unified format for all particles ( $i = 1, 2, \dots, n_{\text{ele}}$ ) in the system, Eqs. (18) and (19) can be written in the form  $\mathbf{M} \mathbf{a} = \mathbf{f}^{\text{appl}}$  and  $\mathbf{J} \boldsymbol{\alpha} = \mathbf{h}^{\text{appl}}$ , respectively, where  $\mathbf{M} = \text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{n_{\text{ele}}}) \in \mathbb{R}^{N \times N}$ ,  $\mathbf{J} = \text{diag}(\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_{n_{\text{ele}}}) \in \mathbb{R}^{N \times N}$ ,  $\mathbf{a} = (\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_{n_{\text{ele}}}^T)^T \in \mathbb{R}^N$ ,  $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T, \dots, \boldsymbol{\alpha}_{n_{\text{ele}}}^T)^T \in \mathbb{R}^N$ ,  $\mathbf{f}^{\text{appl}} = (\mathbf{f}_1^{\text{appl}T}, \mathbf{f}_2^{\text{appl}T}, \dots, \mathbf{f}_{n_{\text{ele}}}^{\text{appl}T})^T \in \mathbb{R}^N$ ,  $\mathbf{h} = (\mathbf{h}_1^{\text{appl}T}, \mathbf{h}_2^{\text{appl}T}, \dots, \mathbf{h}_{n_{\text{ele}}}^{\text{appl}T})^T \in \mathbb{R}^N$  with  $N := n_{\text{dim}} n_{\text{ele}}$ . Or simply,

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix}}_{*\mathbf{M}} \underbrace{\begin{pmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{pmatrix}}_{*\mathbf{a}} = \underbrace{\begin{pmatrix} \mathbf{f}^{\text{appl}} \\ \mathbf{h} \end{pmatrix}}_{*\mathbf{f}} \quad \forall t \in [0, \infty) \tag{20}$$

### Time Discretization

Consider the temporal discretization of the governing equations for a time interval  $\mathbb{I} = [t_0, t_f] \subset [0, \infty)$  split into subintervals, i.e.,  $\mathbb{I} = [t_0, t_f] = \bigcup_{n=0}^{f-1} [t_n, t_{n+1}]$ , in which  $t_0$  and  $t_f$  denote the initial and final time levels of simulation, respectively. The time step size is defined as  $\Delta t := t_{n+1} - t_n > 0$ , and it is assumed to be constant for simplicity. Temporally discretizing Eq. (20) by way of the *normalized time weighted residual methodology* [40] results in the following algorithmic framework:

$$*\mathbf{M} * \tilde{\mathbf{a}}^{n+1} = * \mathbf{f} (* \tilde{\mathbf{q}}^{n+1}, * \tilde{\mathbf{v}}^{n+1}, t_{n+W_1}) \tag{21}$$

where the algorithmic acceleration, velocity, and configuration vectors are defined as

$$\begin{aligned} * \tilde{\mathbf{a}}^{n+1} &= * \mathbf{a}^n + W_1 \Lambda_6 \Delta * \mathbf{a}^{n+1} \\ * \tilde{\mathbf{v}}^{n+1} &= * \mathbf{v}^n + \Delta t W_1 * \mathbf{a}^n, \quad * \tilde{\mathbf{q}}^{n+1} = * \mathbf{q}^n + \Delta t W_1 * \mathbf{v}^n + \frac{\Delta t^2}{2} W_2 * \mathbf{a}^n \end{aligned} \tag{22}$$



respectively, for  $n = 0, 1, \dots, f - 1$ . The associated update equations to advance from  $t = t_n$  to  $t = t_{n+1}$  are

$$\begin{aligned} {}^* \mathbf{q}^{n+1} &= {}^* \mathbf{q}^n + \Delta t {}^* \mathbf{v}^n + \frac{\Delta t^2}{2} \mathbf{a}^n + \lambda_3 \Delta {}^* \mathbf{a}^{n+1} \Delta t^2 \\ {}^* \mathbf{v}^{n+1} &= {}^* \mathbf{v}^n + \Delta t {}^* \mathbf{a}^n + \Delta t \lambda_5 \Delta {}^* \mathbf{a}^{n+1}, \quad {}^* \mathbf{a}^{n+1} = {}^* \mathbf{a}^n + \Delta {}^* \mathbf{a}^{n+1} \end{aligned} \quad (23)$$

The algorithmic parameters, also known as the DNA [discrete numerically assigned] markers, are defined with the input scalar algorithmic parameters  $(\rho_\infty^{\min}, \rho_\infty^{\max}, \rho_\infty^s)$ , denoting the minimum principal root, the maximum principal root, and the spurious root [41], respectively, together with  $\eta$ , as

$$\begin{aligned} W_1 = W_2 &= \frac{1}{1 + \rho_\infty^s}, \quad W_1 \Lambda_6 = \frac{2 + \rho_\infty^{\min} + \rho_\infty^{\max} + \rho_\infty^s - \rho_\infty^{\min} \rho_\infty^{\max} \rho_\infty^s}{(1 + \rho_\infty^{\min})(1 + \rho_\infty^{\max})(1 + \rho_\infty^s)} \\ \lambda_3 &= \frac{\eta}{(1 + \rho_\infty^{\min})(1 + \rho_\infty^{\max})}, \quad \lambda_5 = \frac{3 + \rho_\infty^{\min} + \rho_\infty^{\max} - \rho_\infty^{\min} \rho_\infty^{\max}}{2(1 + \rho_\infty^{\min})(1 + \rho_\infty^{\max})} \end{aligned} \quad (24)$$

for the  $U0(\rho_\infty^{\min}, \rho_\infty^{\max}, \rho_\infty^s)$  family, and

$$\begin{aligned} W_1 &= \frac{3 + \rho_\infty^{\min} + \rho_\infty^{\max} - \rho_\infty^{\min} \rho_\infty^{\max}}{2(1 + \rho_\infty^{\min})(1 + \rho_\infty^{\max})}, \quad W_2 = \frac{2}{(1 + \rho_\infty^{\min})(1 + \rho_\infty^{\max})} \\ W_1 \Lambda_6 &= \frac{2 + \rho_\infty^{\min} + \rho_\infty^{\max} + \rho_\infty^s - \rho_\infty^{\min} \rho_\infty^{\max} \rho_\infty^s}{(1 + \rho_\infty^{\min})(1 + \rho_\infty^{\max})(1 + \rho_\infty^s)}, \quad \lambda_3 = \frac{\eta}{2(1 + \rho_\infty^s)}, \quad \lambda_5 = \frac{1}{1 + \rho_\infty^s} \end{aligned} \quad (25)$$

for the  $V0(\rho_\infty^{\min}, \rho_\infty^{\max}, \rho_\infty^s)$  family. The  $U0$  and  $V0$  represent zero-order overshooting behaviors in the configuration and the velocity fields, respectively. Note that no two schemes within the algorithmic framework can have the same DNA markers that completely distinguish one scheme from another. Any scheme within this framework is guaranteed to be second-order accurate by approximating  ${}^* \mathbf{q}^n \approx {}^* \mathbf{q}(t_n)$ ,  ${}^* \mathbf{v}^n \approx {}^* \dot{\mathbf{q}}(t_n)$ , and  ${}^* \mathbf{a}^n \approx {}^* \ddot{\mathbf{q}}(t_{n-\phi})$  at the time step  $n$  where the acceleration time level shifting parameter is defined by  $\phi := W_1 \Lambda_6 - W_1$ ; see [42] for the details. At time  $t = t_0$ , the acceleration need to be computed from  ${}^* \mathbf{a}^0 = {}^* \mathbf{M}^{-1} {}^* \mathbf{f}({}^* \mathbf{q}^0, {}^* \mathbf{v}^0, t_0)$  where  ${}^* \mathbf{q}^0 = {}^* \mathbf{q}(t_0)$  and  ${}^* \mathbf{v}^0 = {}^* \dot{\mathbf{q}}(t_0)$  are given initial conditions. In DEM simulation, we update from  $n$  to  $n + 1$  via Eqs. (21)–(23) after evaluating the force vectors, as shown in the previous subsection, at time  $t_n$ .

#### Algorithmic Evaluation of ${}_s \delta_{ij}$ :

Suppose particles  $i$  and  $j$  are in contact during a time interval  $\mathbb{I}_c = [t_{c_1}, t_{c_2}] \subset \mathbb{I}$  and split it into subintervals,  $\mathbb{I}_c = [t_{c_1}, t_{c_2}] = \bigcup_{n=c_1}^c [t_n, t_{n+1}]$  with  $t_{\bar{c}+1} \equiv t_{c_2}$ . Following the temporal discretization as given in Eq. (21), the tangential contact force,  ${}_s \tilde{\mathbf{f}}_{ij}$ , is evaluated at  $\mathbf{q} = \tilde{\mathbf{q}}^{n+1}$ ,  $\mathbf{v} = \tilde{\mathbf{v}}^{n+1}$ , and  $\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}}^{n+1}$ . That is, the algorithmic tangential force may be written as

$${}_s \tilde{\mathbf{f}}_{ij} |_{\tilde{\mathbf{q}}^{n+1}, \tilde{\mathbf{v}}^{n+1}} = :{}_s \tilde{\mathbf{f}}_{ij}^{n+1} = k_s (-{}_s \tilde{\boldsymbol{\delta}}_{ij}^{n+1}) + \eta_s (-{}_s \tilde{\mathbf{v}}_{\mathbf{P}_i/\mathbf{P}_j}^{n+1}) \quad (26)$$

where  ${}_s \tilde{\mathbf{v}}_{\mathbf{P}_i/\mathbf{P}_j}^{n+1}$  and  ${}_s \tilde{\boldsymbol{\delta}}_{ij}^{n+1}$  are the algorithmic relative tangential velocity vector and the algorithmic relative tangential displacement vector defined as follows:  ${}_s \tilde{\mathbf{v}}_{\mathbf{P}_i/\mathbf{P}_j}^{n+1} := \tilde{\mathbf{v}}_{ij}^{n+1} - (\tilde{\mathbf{v}}_{ij}^{n+1} \cdot \tilde{\mathbf{n}}_{ij}^{n+1}) \tilde{\mathbf{n}}_{ij}^{n+1} + (r_i \tilde{\boldsymbol{\omega}}_i^{n+1} + r_j \tilde{\boldsymbol{\omega}}_j^{n+1}) \times \tilde{\mathbf{n}}_{ij}^{n+1}$ , where  $\tilde{\mathbf{v}}_{ij}^{n+1} = \mathbf{v}_{ij}^{n+1} + \Delta t W_1 \Lambda_4 \mathbf{a}_{ij}^n$ ,  $\tilde{\mathbf{n}}_{ij}^{n+1} = \frac{\tilde{\mathbf{q}}_j - \tilde{\mathbf{q}}_i}{\|\tilde{\mathbf{q}}_j - \tilde{\mathbf{q}}_i\|}$ , and

$${}_s \tilde{\boldsymbol{\delta}}_{ij}^{n+1} = \left\| \begin{matrix} \bullet \\ \bullet \\ \bullet \end{matrix} \right\| \tilde{\mathbf{t}}_{ij}^{n+1} \quad (27)$$



where

$${}_{s}\delta_{ij}^{\sim n+1} := {}_{s}\delta_{ij}^n + \Delta t W_1 \Lambda_1 {}_{s}\mathbf{v}_{P_i/P_j}^n + \Delta t^2 W_2 \Lambda_2 {}_{s}\mathbf{a}_{P_i/P_j}^n \quad (28)$$

$$\tilde{\mathbf{t}}_{ij}^{n+1} = \frac{{}_{s}\tilde{\mathbf{v}}_{P_i/P_j}^{n+1}}{\|{}_{s}\tilde{\mathbf{v}}_{P_i/P_j}^{n+1}\|} \quad (29)$$

Note that  ${}_{s}\delta_{ij}^n$  in Eq. (28) is given by

$${}_{s}\delta_{ij}^n = \|{}_{s}\delta_{ij}^n\| \mathbf{t}_{ij}^n \quad (30)$$

where

$${}_{s}\delta_{ij}^n := {}_{s}\delta_{ij}^{n-1} + \Delta t {}_{s}\mathbf{v}_{P_i/P_j}^{n-1} + \frac{\Delta t^2}{2} {}_{s}\mathbf{a}_{P_i/P_j}^{n-1} + \Delta t^2 \lambda_3 \Delta_s \mathbf{a}_{P_i/P_j}^n, \quad \Delta_s \mathbf{a}_{P_i/P_j}^n := {}_{s}\mathbf{a}_{P_i/P_j}^n - {}_{s}\mathbf{a}_{P_i/P_j}^{n-1} \quad (31)$$

$$\mathbf{t}_{ij}^n := \frac{{}_{s}\mathbf{v}_{P_i/P_j}^n}{\|{}_{s}\mathbf{v}_{P_i/P_j}^n\|} \quad (32)$$

**Remark**

1. If the contact starts at time  $t_n$ , i.e.,  $t = t_n = t_{c1}$ , we have  ${}_{s}\delta_{ij}^n = \mathbf{0}$  in Eq. (28).
2. When  ${}_{s}\tilde{\mathbf{v}}_{P_i/P_j}^{n+1} = \mathbf{0}$ , the algorithmic tangential unit vector,  $\tilde{\mathbf{t}}_{ij}^{n+1}$ , defined by Eq. (29), becomes singular. In this case, we assume that particle  $j$  is sliding on the surface of particle  $i$ , and therefore use  $\tilde{\mathbf{t}}_{ij}^{n+1} = \frac{{}_{s}\delta_{ij}^{\sim n+1}}{\|{}_{s}\delta_{ij}^{\sim n+1}\|}$  instead; and we have  ${}_{s}\tilde{\delta}_{ij}^{n+1} = {}_{s}\delta_{ij}^{\sim n+1}$ . Similarly, if  ${}_{s}\mathbf{v}_{P_i/P_j}^n = \mathbf{0}$ , use  $\mathbf{t}_{ij}^n = \frac{{}_{s}\delta_{ij}^n}{\|{}_{s}\delta_{ij}^n\|}$  instead of Eq. (32).
3. When slip occurs at the algorithmic time level  $t_{n+W_1} \in \mathbb{I}_c$ , i.e.,  $\|{}_{s}\tilde{\mathbf{f}}_{ij}^{n+1}\| > \mu \|{}_{n}\tilde{\mathbf{f}}_{ij}^{n+1}\|$ , in which  ${}_{n}\tilde{\mathbf{f}}_{ij}^{n+1}$  is the algorithmic normal contact force defined as  ${}_{n}\tilde{\mathbf{f}}_{ij}^{n+1} := -k_n [(r_i + r_j) - \|\tilde{\mathbf{q}}_j^{n+1} - \tilde{\mathbf{q}}_i^n\|] \tilde{\mathbf{n}}_{ij}^{n+1} - \eta_n (\tilde{\mathbf{v}}_{ijj}^{n+1} \cdot \tilde{\mathbf{n}}_{ij}^{n+1}) \tilde{\mathbf{n}}_{ij}^{n+1}$ , and the algorithmic tangential force  ${}_{s}\tilde{\mathbf{f}}_{ij}^{n+1}$  defined in Eq. (26) is replaced with the following algorithmic frictional force,  ${}_{s}\tilde{\mathbf{f}}_{ij}^{n+1} = k_s ({}_{s}\tilde{\delta}_{ij}^{n+1}) = -\mu \|{}_{n}\tilde{\mathbf{f}}_{ij}^{n+1}\| \tilde{\mathbf{t}}_{ij}^{n+1}$ . Therefore, the algorithmic tangential displacement vector,  ${}_{s}\tilde{\delta}_{ij}^{n+1}$ , defined in Eq. (27), should be replaced with  ${}_{s}\tilde{\delta}_{ij}^{n+1} = -\frac{\mu}{k_s} \|{}_{n}\tilde{\mathbf{f}}_{ij}^{n+1}\| \tilde{\mathbf{t}}_{ij}^{n+1}$ . Similarly, in the case of  $\|{}_{s}\mathbf{f}_{ij}^n\| > \mu \|{}_{n}\mathbf{f}_{ij}^n\|$ , employ  ${}_{s}\delta_{ij}^n = -\frac{\mu}{k_s} \|{}_{n}\mathbf{f}_{ij}^n\| \mathbf{t}_{ij}^n$  instead of Eq. (30).
4. The evaluations of  ${}_{s}\delta_{ij}^{\sim n+1}$  and  ${}_{s}\delta_{ij}^n$ , as given in Eqs. (28) and (31), respectively, are crucial and necessary to ensure the second-order accuracy of the family of algorithms in time. For the central difference method [43], which is equivalent to the U0(1, 1, 0) scheme with  $\eta = 1$ , i.e., a scheme obtained by selecting  $(\rho_\infty^{\min}, \rho_\infty^{\max}, \rho_\infty^s) = (1, 1, 0)$  with  $\eta = 1$  in Eq. (24), the algorithm simplifies to  ${}_{s}\tilde{\mathbf{f}}_{ij}^{n+1} = {}_{s}\mathbf{f}_{ij}^{n+1} = -k_s ({}_{s}\delta_{ij}^{n+1}) - \eta_s ({}_{s}\mathbf{v}_{P_i/P_j}^{n+1})$  with

$${}_{s}\delta_{ij}^{n+1} = \left\| {}_{s}\delta_{ij}^n + \Delta t {}_{s}\mathbf{v}_{P_i/P_j}^n + \frac{\Delta t^2}{2} {}_{s}\mathbf{a}_{P_i/P_j}^n \right\| \mathbf{t}_{ij}^{n+1} \quad (33)$$

$${}_{s}\mathbf{v}_{P_i/P_j}^{n+1} = {}_{s}\mathbf{v}_{P_i/P_j}^n + \Delta t {}_{s}\mathbf{a}_{P_i/P_j}^n + \frac{\Delta t}{2} \Delta_s \mathbf{a}_{P_i/P_j}^{n+1}, \quad \Delta_s \mathbf{a}_{P_i/P_j}^{n+1} := {}_{s}\mathbf{a}_{P_i/P_j}^{n+1} - {}_{s}\mathbf{a}_{P_i/P_j}^n \quad (34)$$

It is worth noting that the tangential acceleration is usually not included during the computations of the tangential contact forces, i.e., for the central difference method, for example,

$${}_s\delta_{ij}^{n+1} = \left\| {}_s\delta_{ij}^n + \Delta t_s \mathbf{V}_{\mathbf{P}_i/\mathbf{P}_j}^n \right\| \mathbf{t}_{ij}^{n+1}$$

is inconsistently used instead of Eq. (33) of the most literature; in this case, the order of accuracy of a scheme within the algorithmic framework presented reduces to only one.

5. For the evaluation of the tangential displacement  ${}_s\delta_{ij}$  in Eq. (16) for the nonlinear model in the discrete time system, the approach presented here may be applied in the same manner to ensure the second-order time accuracy of the schemes.

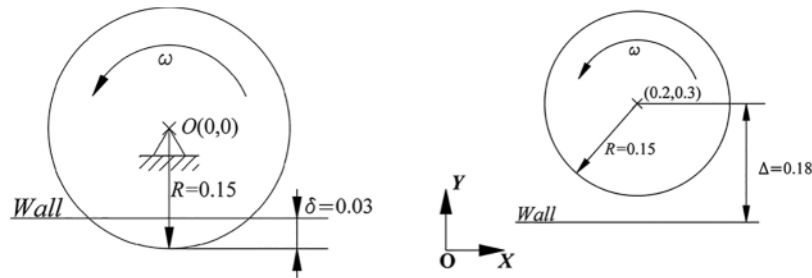
### 3 Numerical Experiments and Discussion

To verify and validate the concepts described in the previous section, two numerical experiments are given in this section. The main objective of Section 3.1 is to compare the numerical results by traditional tangential displacement evaluation and the proposed consistent evaluation method employing the Central Difference Method. It is worth noting that Central Difference Method is the most widely used explicit second-order time accurate integrator in solving dynamic problems. Therefore, we simply use it as the benchmark in the following simulations. The accuracy in time with different tangential displacement evaluations are given as well. In Section 3.2, a simple 3D granular cube falling problem is simulated with several selected second-order time accurate schemes from the GSSSS unified time integrators. The numerical performances of different schemes with different tangential displacement evaluations are discussed. The linked list algorithm is used to improve the particle searching process and parallel computing is also used in this work. In addition, the calculation time for the 3D case is provided to demonstrate the computational efficiency of the proposed method.

#### 3.1 Validations

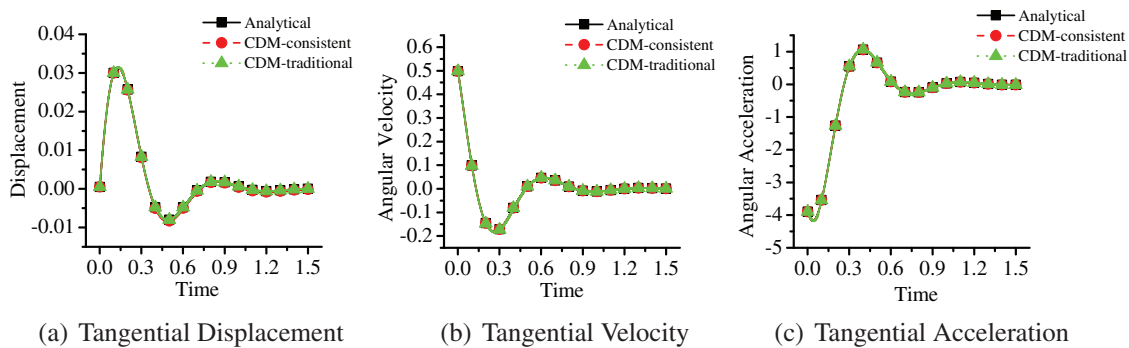
To valid our proposed approach and the DEM code, two cases are designed to test the performance of the proposed tangential displacement evolution. Case I is designed to extract the tangential displacement calculation in the general discrete element method. The influence of normal direction is avoided in Case I via fixing a two dimensional disk with only initial angular velocity and constant normal force. Case II gives a general ball falling problem with gravity and the comparisons of traditional and consistent tangential displacement evolution.

**Case I:** The equation of the tangential motion can be derived to yield a second-order ODE (ordinary differential equation) from Eq. (17), which has an analytical solution for the linear equation. To test our new proposed tangential displacement evaluation method, a very simple and specific physical problem is designed as a 2-D disk with fixed translational movement. As illustrated in Fig. 2 (left), although the normal force is involved, the tangential movement can be easily extracted. The disk is fixed at the origin  $\mathbf{O}$ , the initial angular velocity  $\omega^0$  is set as 0.5 and the tangential stiffness  $k_s$  is 1,000. Both the proposed and traditional tangential displacement evaluation method are applied with the central difference method, which has been proved to be equivalent as U0 (1, 1, 0) within the GS4-family, is used in the simulation.



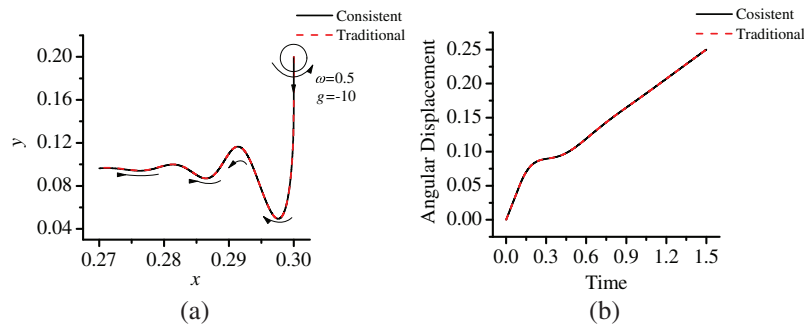
**Figure 2:** Schematic of the problem. Left: Case I and right: Case II

The comparison of the results with different tangential displacement evaluation methods is given in Fig. 3. Both the proposed consistent time level method and the traditional method show good agreement with the analytical solutions in angular position, velocity and acceleration. In this part, our interest is to simply show that the proposed method has the improved or same performance as the traditional method in the sense of capturing physical phenomena.

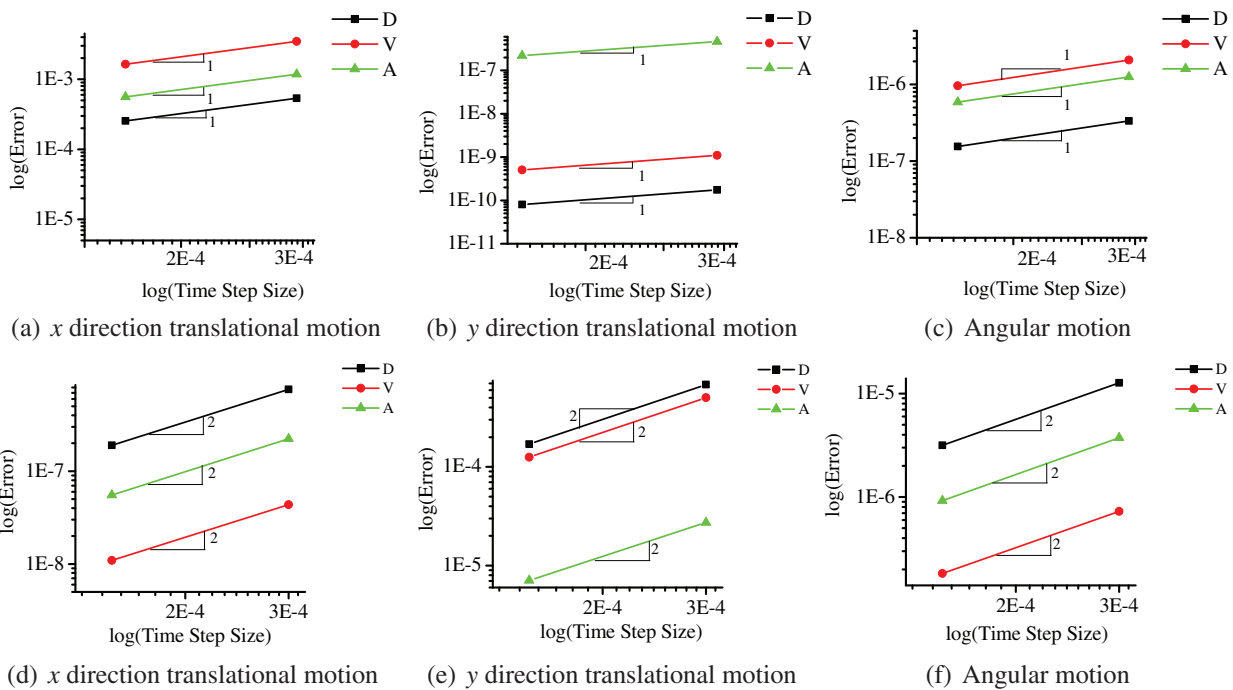


**Figure 3:** The time history of tangential motion with different tangential displacement evaluation methods

**Case II:** To complete the validation of our code, a general single ball falling problem shown in Fig. 2 (right), is simulated by different tangential displacement evaluation methods. To be consistent with the previous numerical simulation case, the central difference method is used to solve the governing equation. The time history of ball’s translational trajectory of the center point and rotational motion are given in Fig. 4, and the time convergence plots for both methods are given in Fig. 5. The top row in Fig. 5 provides the convergence plots of position/angular displacement (D), velocity (V), and acceleration (A) evaluated by the central difference scheme with the traditional tangential displacement evaluation method, while the bottom row provides the one evaluated by the GSSSS U0(1, 1, 0) with the proposed tangential displacement evaluation method. As expected, the results of both methods show excellent agreement as shown in Fig. 4. However, as Fig. 5 (top) shows, the order of convergence in time for the traditional method is only one, in sharp contrast to the proposed method which is of order two. In other words, the well-known second-order accurate central difference method obtains first-order convergence in the case when using the traditional tangential displacement evaluation method. Hence, the superiority of the proposed consistent tangential displacement evaluation over the traditional method is evident with respect to the time accuracy; it also yields improved physics, which can be observed for applications where the tangential effects dominate.



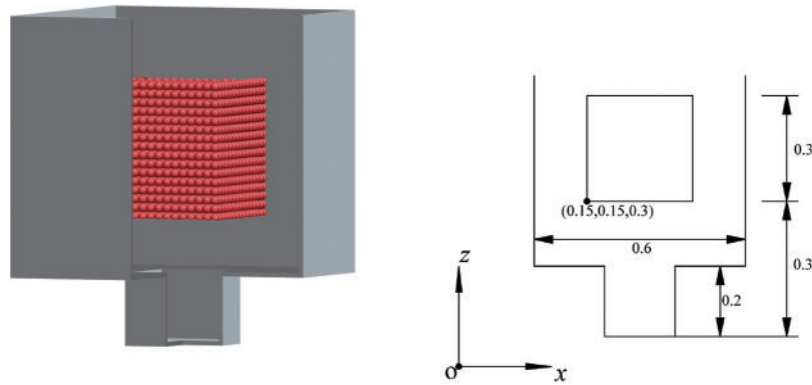
**Figure 4:** The time history of ball's motion with different tangential displacement evaluation methods: (a) Translational motion of ball's center (b) Angular displacement



**Figure 5:** Time accuracy plots of the central difference method with traditional tangential displacement evaluation method (top) and consistent tangential displacement evaluation method (bottom)

### 3.2 Numerical Experiment

**Problem Description:** The numerical example chosen in this section is that of a simple granular cube drop of particles in three-dimensional space, as shown in Fig. 6. The granular cube has dimensions of 0.3 length  $\times$  0.3 width  $\times$  0.3 height, and the container has dimensions of 0.6 length  $\times$  0.6 width  $\times$  0.8 height with a cube shape hole (0.2 length  $\times$  0.2 width  $\times$  0.2 height) at the bottom. Initially, the particles are packed together and the particle distance ( $l$ ) and diameter of particles are set to be 0.02, and the total number of particles is 3,375. The mass of each particle is 0.008. The normal and tangential stiffness are chosen as  $k_n = 1,500$  and  $k_s = 100$ , respectively.



**Figure 6:** Schematic of the problem (Numerical example)

In this section, the consistent time level implementation and preserving the second-order time accuracy for the presented results in this section are as follows for a wide variety of time integrators in the present unified framework. They yield no overshoot in displacement or no overshoot in velocity or no overshoot in both displacement and velocity for a given set of initial conditions:

- CDM-T: Central Difference Method with traditional tangential displacement evaluation. Note that the central difference method has been proved to be equivalent to the GS4-II U0-based family with  $(\rho_{\infty}^{\min}, \rho_{\infty}^{\max}, \rho_{\infty}^s) = (1, 1, 0)$ .
- CDM-C: Central Difference Method with consistent tangential displacement evaluation.
- V0(1, 1, 0)-C: GS4-II V0-based family with  $(\rho_{\infty}^{\min}, \rho_{\infty}^{\max}, \rho_{\infty}^s) = (1, 1, 0)$  and consistent tangential displacement evaluation.
- U0/V0(0, 1, 0)-C: GS4-II U0-based family or V0-based family with  $(\rho_{\infty}^{\min}, \rho_{\infty}^{\max}, \rho_{\infty}^s) = (0, 1, 0)$  and consistent tangential displacement evaluation.

#### *Numerical Setting and Computational Cost*

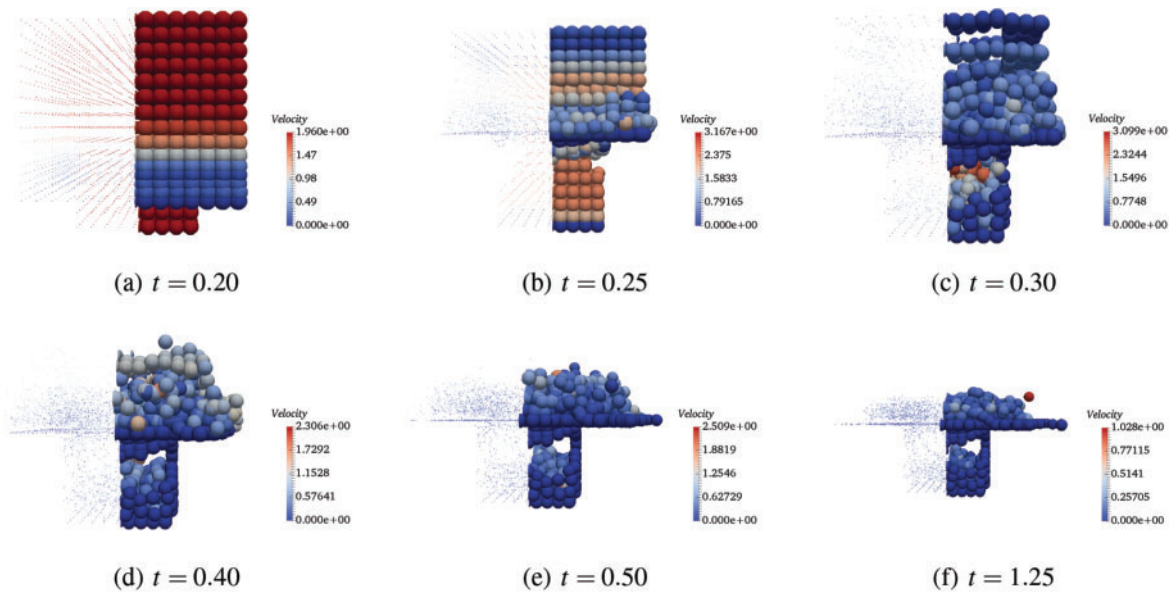
In order to capture particle contacts, the time step size used in DEM type simulations is always required to be very small. This implies that the implicit time integration schemes are not necessary in the general DEM simulations. Therefore, in this work, all the schemes listed are explicit and the time step size is  $\Delta t = 0.0001$  within the total simulation duration from  $t_0 = 0$  to  $t_f = 2$ . As mentioned before, the neighborhood particle-detecting process in any particle-based method is the most expensive step in the simulation. In this paper, the linked list algorithm is used to optimize the neighboring particle searching and decrease the computational time, and parallel computing is also applied to improve the computational performance of our code. Our 3D cases were run on a 1.80 GHz, Intel(R) Core(TM) i7-8565K CPU with 8 GB RAM. [Table 1](#) summarizes the specific timings used for all our 3D examples and shows that our proposed GSSSS family of algorithms does not increase computationally cost in comparison to the traditional algorithms, while our proposed method can obtain better computational accuracy (see [Fig. 5](#)).

**Table 1:** Parameters that vary across simulations: **n**: number of particles,  $k_n$ : normal stiffness coefficient;  $k_s$ : tangential stiffness coefficient; **Time**: average time per 100 time steps

Schemes	n	$k_n$	$k_s$	Time
CDM-C (Fig. 7)	3375	1500	100	0.740
CDM-T (Fig. 8)	3375	1500	100	0.732
V0 (1, 1, 0)-C (Fig. 9)	3375	1500	100	0.725
U0/V0 (0, 1, 0)-C (Fig. 10)	3375	1500	100	0.737

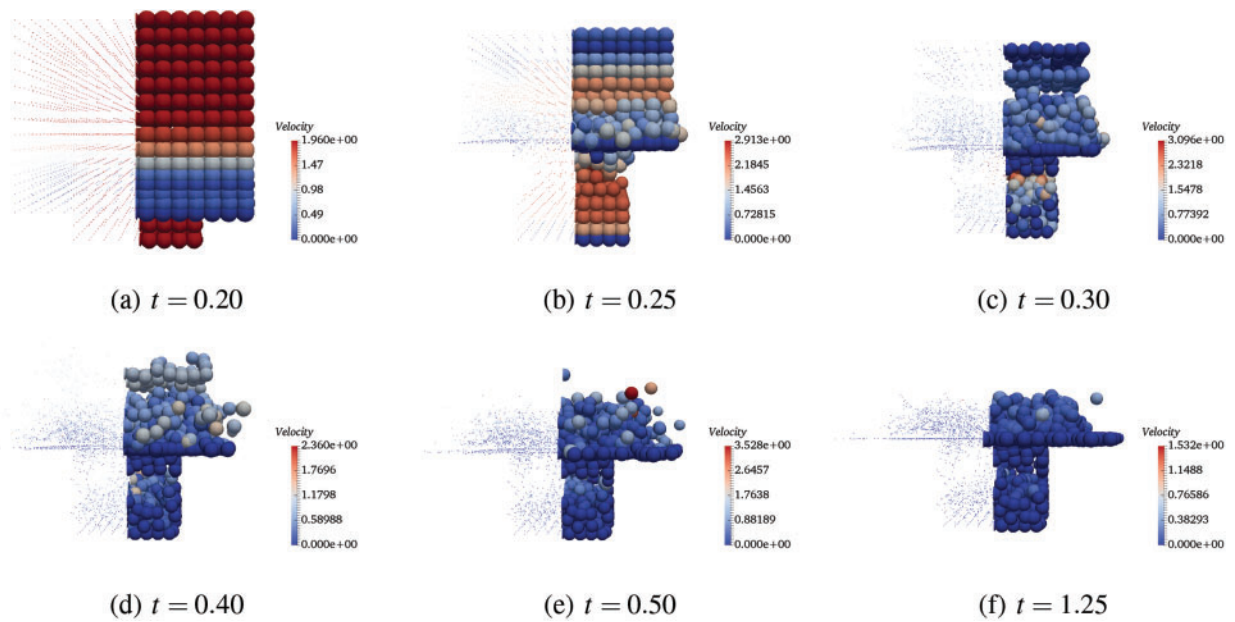
### Numerical Results and Discussion

Three-dimensional snapshots with the granular velocity field values at six different times ( $t = 0.20, 0.25, 0.30, 0.40, 0.50$  and  $1.25$ ) by different schemes are shown in Figs. 7 to 10, respectively. The specifics of particle filling process is observed from 0.2 to 0.5 and the final particle distribution is obtained at time 1.25. The process of particle filling the bottom space corresponds to the evolution of total kinetic energy of the system shown in Fig. 11 (left). As expected, the results of different schemes are very close at each time and the particle distribution shows good physical reliability of the simulation results. In order to better understand the dynamic responses of the granular cube falling process and the computational performance of different schemes, the time history of the total kinetic energy with different schemes are given in Fig. 11 (left). Because the central portion of the granular cube keeps charging into the bottom empty space, the total kinetic energy does not decrease immediately, but a short duration (from 0.17 to 0.23) of energy relaxing instead. After the bottom space is filled with particles, the total kinetic energy decreases sharply though some particles on the top layers still keep bouncing. As the frictional force and damping are considered in the particle contact model, the total kinetic energy converges to zero as time goes.

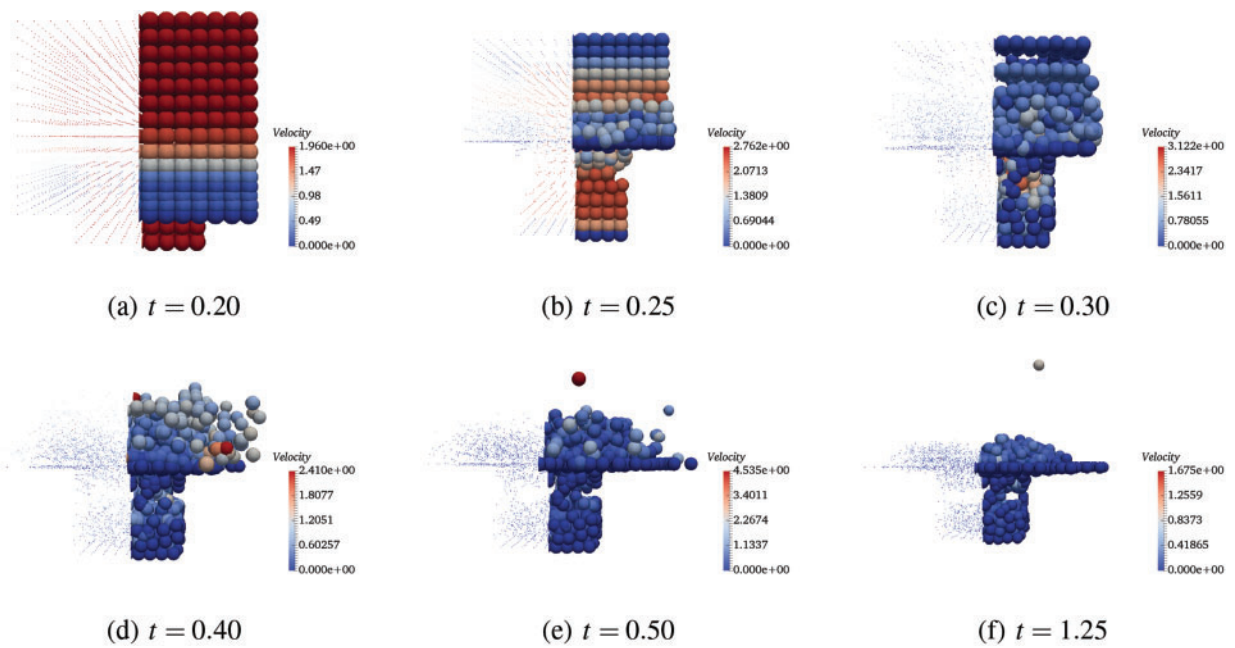


**Figure 7:** Velocity fields at different time by CDM-C



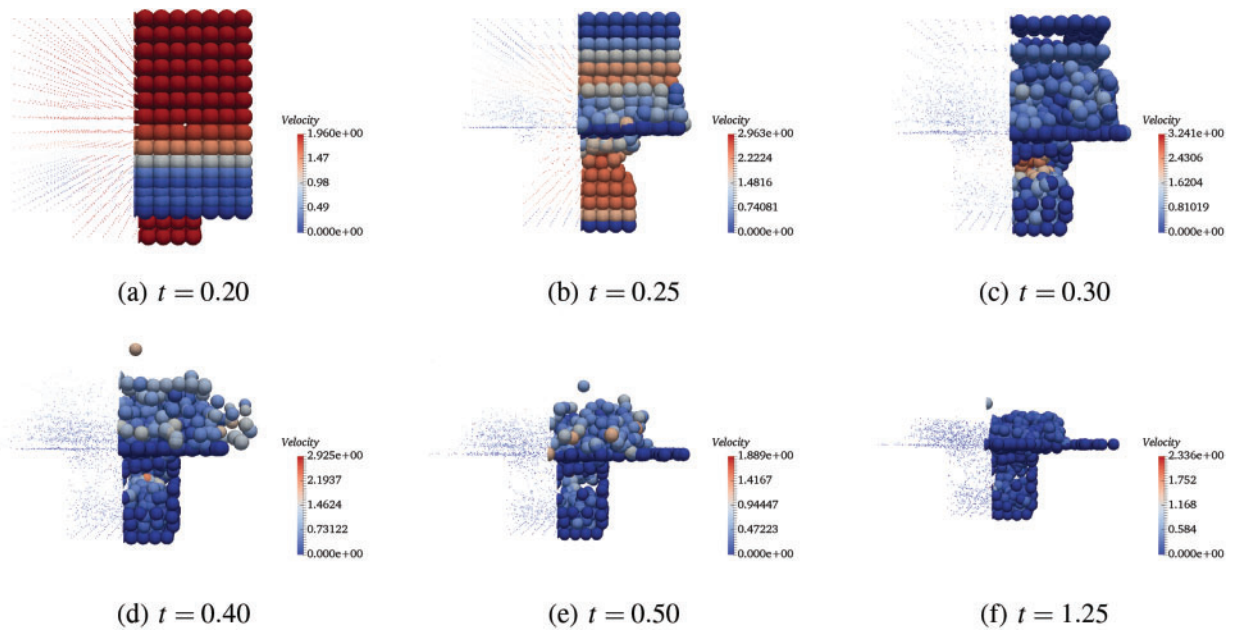


**Figure 8:** Velocity fields at different time by CDM-T

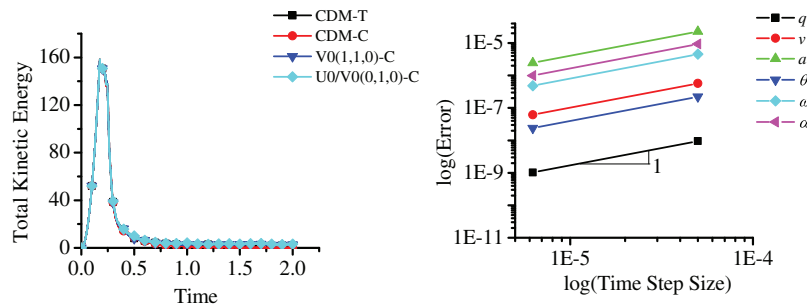


**Figure 9:** Velocity fields at different time by V0(1, 1, 0)-C





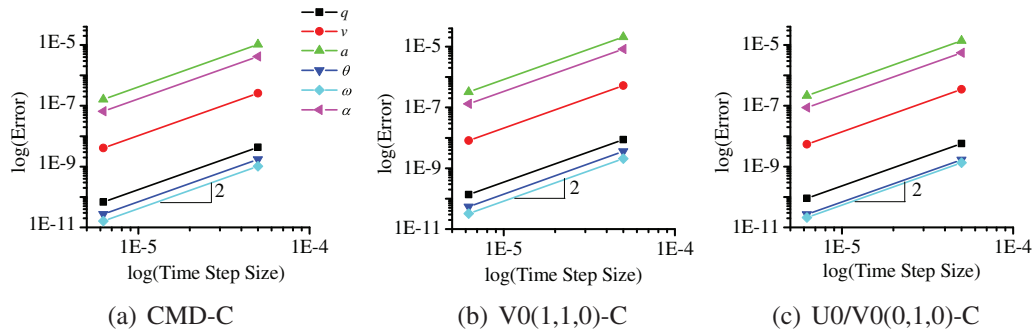
**Figure 10:** Velocity fields at different time by  $U_0/V_0(0, 1, 0)$ -C



**Figure 11:** Left: total kinetic energy. Right: time accuracy plots of central different method with traditional tangential displacement evaluation

The time accuracy for translational and rotational position, velocity and acceleration of the central difference method with the traditional tangential displacement evaluation shows only first-order time accuracy in Fig. 11 (right). However, alternately, Fig. 12 shows the second-order time accuracy for translational position, velocity and acceleration of different time integration schemes selected within the GS4-II family with the proposed consistent tangential displacement evaluation. All the schemes show the second-order time accuracy in both translational and rotational motion. It is to be noted however, that using central difference method in solving the second-order ODE, it is supposed to yield second-order accuracy in time. According to the time convergence order from different tangential displacement evaluations, it implies that the drawback via the traditional tangential displacement evaluation in DEM type method is not at the same time level as the algorithm and has significant influence on the whole system’s time accuracy. In other words, only when the tangential displacement evaluation is evaluated at a consistent time level corresponding to the selected time integration scheme within the unified framework described, the results are completely reliable and the time integration

schemes are implemented correctly and efficiently maintaining second-order time accuracy with improvement in physics.



**Figure 12:** Time accuracy plots of different selected schemes within GS4-II family:  $q$ ,  $v$  and  $a$  are translational position, velocity and acceleration, respectively.  $\theta$ ,  $\omega$  and  $\alpha$  are angular displacement, velocity and acceleration, respectively

#### 4 Concluding Remarks

In this work, a consistent tangential displacement evaluation approach is proposed under the framework of the unified second-order time integrators GS4-II family for improving discrete element method (DEM) simulations. The newly proposed approach is derived and designed to maintain the simplicity and generalization of the GS4-II family. Comparing it with the traditional tangential displacement evaluation method, the proposed consistent time level implementation approach can be easily implemented and yields the correct algorithmic tangential displacement at the same time level as in the time integrations via using the same parameters ( $\rho_{\infty}^{\min}$ ,  $\rho_{\infty}^{\max}$ ,  $\rho_{\infty}^s$ ) as the GS4-II family. The improved time accuracy of both the translational and rotational motions by using the proposed method gives second-order accuracy in time, which is consistent with the time integration schemes; whereas, the traditional second-order time integration method such as the central difference can only obtain first-order accuracy in time via using the traditional tangential displacement evaluation.

**Acknowledgement:** We appreciate all the constructive comments and suggestions from editors and reviewers.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

#### References

1. Cundall, P. A., Strack, O. D. L. (1979). A discrete numerical model for granular assemblies. *Geotechnique*, 29, 47–65. DOI 10.1680/geot.1979.29.1.47.
2. Luding, S. (1997). Stress distribution in static two-dimensional granular model media in the absence of friction. *Physical Review E*, 55, 4720–4729. DOI 10.1103/PhysRevE.55.4720.
3. Baxter, J., Tüzün, U., Burnell, J., Heyes, D. M. (1997). Granular dynamics simulations of two-dimensional heap formation. *Physical Review E*, 55(3), 3546. DOI 10.1103/PhysRevE.55.3546.
4. Zhou, Y. C., Xu, B. H., Zou, R., Yu, A. B., Zulli, P. (2003). Stress distribution in a sandpile formed on a deflected base. *Advanced Powder Technology*, 14(4), 401–410. DOI 10.1163/156855203769710636.

5. Fazekas, S., Kertész, J., Wolf, D. E. (2005). Piling and avalanches of magnetized particles. *Physical Review E*, 71(6), 061303. DOI 10.1103/PhysRevE.71.061303.
6. Li, Y., Xu, Y., Thornton, C. (2005). A comparison of discrete element simulations and experiments for ‘sandpiles’ composed of spherical particles. *Powder Technology*, 160(3), 219–228. DOI 10.1016/j.powtec.2005.09.002.
7. Yen, K. Z. Y., Chaki, T. (1992). A dynamic simulation of particle rearrangement in powder packings with realistic interactions. *Journal of Applied Physics*, 71(7), 3164–3173. DOI 10.1063/1.350958.
8. Zhang, Z. P., Liu, L. F., Yuan, Y. D., Yu, A. B. (2001). A simulation study of the effects of dynamic variables on the packing of spheres. *Powder Technology*, 116(1), 23–32. DOI 10.1016/S0032-5910(00)00356-9.
9. Latham, J. P., Lu, Y., Munjiza, A. (2001). A random method for simulating loose packs of angular particles using tetrahedra. *Geotechnique*, 51(10), 871–879. DOI 10.1680/geot.2001.51.10.871.
10. Silbert, L. E., Grest, G. S., Landry, J. W. (2002). Statistics of the contact network in frictional and frictionless granular packings. *Physical Review E*, 66(6), 061303. DOI 10.1103/PhysRevE.66.061303.
11. Munjiza, A. A. (2004). *The combined finite-discrete element method*. West Sussex, UK: John Wiley & Sons.
12. An, X. Z., Yang, R. Y., Dong, K. J., Zou, R. P., Yu, A. B. (2005). Micromechanical simulation and analysis of one-dimensional vibratory sphere packing. *Physical Review Letters*, 95(20), 205502. DOI 10.1103/PhysRevLett.95.205502.
13. Just, S., Toschkoff, G., Funke, A., Djuric, D., Scharrer, G. et al. (2013). Experimental analysis of tablet properties for discrete element modeling of an active coating process. *AAPS PharmSciTech*, 14(1), 402–411. DOI 10.1208/s12249-013-9925-5.
14. Lewis, R. W., Gethin, D. T., Yang, X. S., Rowe, R. C. (2005). A combined finite-discrete element method for simulating pharmaceutical powder tableting. *Numerical Methods in Engineering*, 62, 853–869. DOI 10.1002/(ISSN)1097-0207.
15. Buchholtz, V., Pöschel, T., Tillemans, H. J. (1995). Simulation of rotating drum experiments using non-circular particles. *Physica A: Statistical Mechanics and its Applications*, 216(3), 199–212. DOI 10.1016/0378-4371(95)00045-9.
16. Pöschel, T., Buchholtz, V. (1995). Complex flow of granular material in a rotating cylinder. *Chaos, Solitons and Fractals*, 5(10), 1901–1912. DOI 10.1016/0960-0779(94)00193-T.
17. Sakai, M., Shibata, K., Koshizuka, S. (2006). Development of a criticality evaluation method involving the granular flow of the nuclear fuel in a rotating drum. *Nuclear Science and Engineering*, 154(1), 63–73. DOI 10.13182/NSE06-A2618.
18. Portillo, P. M., Muzzio, F. J., Ierapetritou, M. G. (2007). Hybrid DEM-compartment modeling approach for granular mixing. *AIChE Journal*, 53(1), 119–128. DOI 10.1002/(ISSN)1547-5905.
19. Mishra, B. K., Rajamani, R. K. (1992). The discrete element method for the simulation of ball mills. *Applied Mathematical Modelling*, 16, 598–604. DOI 10.1016/0307-904X(92)90035-2.
20. Inoue, T., Okaya, K. (2012). Grinding mechanism of centrifugal mills-A simulation study based on the discrete element method. *Comminution*, 1994, 425. DOI 10.1016/0301-7516(95)00049-6.
21. Gudin, D., Kano, J., Saito, F. (2007). Effect of the friction coefficient in the discrete element method simulation on media motion in a wet bead mill. *Advanced Powder Technology*, 18(5), 555–565. DOI 10.1163/156855207782146625.
22. Sun, X., Sakai, M., Yamada, Y. (2013). Three-dimensional simulation of a solid-liquid flow by the DEM-SPH method. *Journal of Computational Physics*, 248, 147–176. DOI 10.1016/j.jcp.2013.04.019.
23. Schwartz, S. R., Michel, P., Richardson, D. C. (2013). Numerically simulating impact disruptions of cohesive glass bead agglomerates using the soft-sphere discrete element method. *Icarus*, 226(1), 67–76. DOI 10.1016/j.icarus.2013.05.007.
24. Guo, Y., Wassgren, C., Hancock, B., Ketterhagen, W., Curtis, J. (2015). Computational study of granular shear flows of dry flexible fibres using the discrete element method. *Journal of Fluid Mechanics*, 775, 24–52. DOI 10.1017/jfm.2015.289.

25. Guo, Y., Curtis, J. S. (2015). Discrete element method simulations for complex granular flows. *Annual Review of Fluid Mechanics*, 47, 21–46. DOI 10.1146/annurev-fluid-010814-014644.
26. Munjiza, A., Owen, D., Bicanic, N. (1995). A combined finite-discrete element method in transient dynamics of fracturing solids. *Engineering Computations*, 12(2), 145–174. DOI 10.1108/02644409510799532.
27. Peters, B. (2013). The extended discrete element method (XDEM) for multi-physics applications. *Scholarly Journal of Engineering Research*, 2, 1–20.
28. Wang, Y., Tamma, K., Maxam, D., Xue, T., Qin, G. (2021). An overview of high-order implicit algorithms for first-/second-order systems and novel explicit algorithm designs for first-order system representations. *Archives of Computational Methods in Engineering*, 28(5), 3593–3619. DOI 10.1007/s11831-021-09536-3.
29. Xue, T., Wang, Y., Aanjaneya, M., Tamma, K. K., Qin, G. (2021). On a generalized energy conservation/dissipation time finite element method for Hamiltonian mechanics. *Computer Methods in Applied Mechanics and Engineering*, 373, 113509. DOI 10.1016/j.cma.2020.113509.
30. Shimada, M., Tamma, K. K. (2013). Explicit time integrators and designs for first/second order linear transient systems. *Encyclopedia of Thermal Stresses*, 3, 1524–1530. DOI 10.1007/978-94-007-2739-7\_764.
31. Atluri, S. N., Han, Z. D., Shen, S. (2003). Meshless local petrov-galerkin (MLPG) approaches for solving the weakly-singular traction and displacement boundary integral equations. *Computer Modeling in Engineering & Sciences*, 4(5), 507–518. DOI 10.3970/cmcs.2003.004.507.
32. Han, Z. D., Atluri, S. (2004). Meshless local petrov-galerkin (MLPG) approaches for solving 3D problems in elasto-statics. *Computer Modeling in Engineering & Sciences*, 6, 169–188. DOI 10.3970/cmcs.2004.006.169.
33. Long, S., Atluri, S. N. (2002). A meshless local petrov-galerkin method for solving the bending problem of a thin plate. *Computer Modeling in Engineering & Sciences*, 3(1), 53–64. DOI 10.3970/cmcs.2002.003.053.
34. Xue, T., Zhang, X., Tamma, K. K. (2018). A two-field state-based peridynamic theory for thermal contact problems. *Journal of Computational Physics*, 374, 1180–1195. DOI 10.1016/j.jcp.2018.08.014.
35. Xue, T., Zhang, X., Tamma, K. K. (2019). A non-local dissipative lagrangian modelling for generalized thermoelasticity in solids. *Applied Mathematical Modelling*, 73, 247–265. DOI 10.1016/j.apm.2019.04.004.
36. Matuttis, H., Chen, J. (2014). *Understanding the discrete element method: Simulation of non-spherical particles for granular and multi-body systems*, 1st edition. Chichester, UK: John Wiley.
37. Tsuji, Y., Tanaka, T., Ishida, T. (1992). Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. *Powder Technology*, 71, 239–250. DOI 10.1016/0032-5910(92)88030-L.
38. Mindlin, R. D. (1949). Compliance of elastic bodies in contact. *Journal of Applied Mechanics*, 16, 259–268. DOI 10.1115/1.4009973.
39. Mindlin, R. D., Deresiewicz, H. (1953). Elastic spheres in contact under varying oblique forces. *Journal of Applied Mechanics*, 20, 327–344. DOI 10.1115/1.4010702.
40. Har, J., Tamma, K. K. (2012). *Advances in computational dynamics of particles, materials and structures*. Chichester, UK: John Wiley.
41. Tamma, K. K., Har, J., Zhou, X., Shimada, M., Hoitink, A. (2011). An overview and recent advances in vector and scalar formalisms: Space/time discretizations in computational dynamics: A unified approach. *Archives of Computational Methods in Engineering*, 18, 119–283. DOI 10.1007/s11831-011-9060-y.
42. Shimada, M., Hoitink, A., Tamma, K. K. (2015). The fundamentals underlying evaluation of acceleration computations for general dynamic applications: Issues and noteworthy perspectives. *Computer Methods in Applied Mechanics and Engineering*, 104(2), 133–158. DOI 10.3970/cmcs.2015.104.133.
43. Hughes, T. J. R. (2000). *The finite element method: Linear static and dynamic finite element analysis*. Mineola, New York: Dover Publications.