Received: 3 January 2021 | Revised: 26 August 2021 | Accepted: 11 October 2021

DOI: 10.1002/nag.3293

RESEARCH ARTICLE

WILEY

Machine-learning-enabled discrete element method: Contact detection and resolution of irregular-shaped particles

Zhengshou Lai[1,2] | **Qiushi Chen**[3] | **Linchong Huang**[4]

[1] School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China

[2] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

[3] Glenn Department of Civil Engineering, Clemson University, Clemson, SC 29634, USA

[4] School of Aeronautics and Astronautics Engineering, Sun Yat-sen University, Shenzhen 518107, China

Correspondence
Qiushi Chen, Glenn Department of Civil Engineering, Clemson University, Clemson, SC, 29634, USA.
Email: qiuchi@clemson.edu

Funding information
Fundamental Research Funds for the Central Universities, Grant/Award Number: 19lgpy289; Shenzhen Natural Science Foundation, Grant/Award Number: JCYJ20190807162401662; China Postdoctoral Science Foundation, Grant/Award Number: 2019M663240; Hong Kong Scholars Program, Grant/Award Number: 2020; National Natural Science Foundation of China, Grant/Award Numbers: 51909289, 51978677; Research Grants Council of Hong Kong, Grant/Award Number: 16207319

Abstract

This paper presents a machine learning (ML)-enabled discrete element method (DEM) for the computational mechanics of irregular-shaped particles. ML-enabled DEM, as with most conventional DEMs, encompasses four main steps in one typical calculation cycle, namely, (1) the detection and resolution of contacts, (2) the evaluation of contact behavior, (3) the calculation of particle motion, and (4) the updating of particle geometric descriptions. Unlike conventional DEMs, the proposed method constructs and employs neural networks to detect particle contacts and resolve contact geometric features. Neural networks take particle geometric descriptors as inputs and output the contact status and contact geometric features. Using two-dimensional elliptical particles as an example, the performance of the ML-enabled DEM is investigated through five numerical experiments and compared with analytical solutions or conventional DEM methods. A sixth numerical experiment involving irregular-shaped particles is also presented to showcase the potential and applicability of the proposed method for other particle shapes. ML-enabled DEM can accurately capture the trajectory and energy evolution of individual particles, the fabric characteristics of dense packing, and the mechanical behavior of packing under large loads, while demonstrating computational efficiency over conventional methods.

KEYWORDS
contact detection and resolution, discrete element method, machine learning, neural network, particle shape

1 | INTRODUCTION

Originally proposed by Cundall and Strack,[1] the distinct element method (DEM, and also often referred to as discrete element method) is a type of numerical method for modeling discontinuous materials. It has gained popularity in a wide range of geomechanical problems, such as fracturing processes of rock masses,[2] shearing behaviors of soils or sands,[3,4] tunnel face failure,[5] landslide,[6] and so forth. In DEM, the motion of every particle is formulated based on the Newton's law of motion and solved using an explicit time integration scheme. The properties and responses of the particles can

be thus explicitly considered at a fundamental level. Another method in the context of discontinuous mechanics is the discontinuous deformation analysis (DDA),[7] which is derived from the principle of minimum potential energy and solves the material force and deformation using an implicit time marching scheme. Compared with DEM, DDA has a more physically meaningful mechanism of energy dissipation but might suffer from the issues of numerical instability and computational inefficiency.[8,9] A detailed comparison of DDA and DEM can be found in Khan.[8] Owing to the advantage of DEM at the fundamental level, DEM is often coupled with the continuum mechanics-based methods (e.g., finite element method (FEM)[10] and material point method (MPM)[11]) to develop the so-called multiscale approach. A typical hierarchical multiscale coupling scheme is to employ the FEM or MPM to solve a boundary value problem while using the DEM to derive the required nonlinear material responses at each Gauss integration or material point. It hence utilizes the advantages of both the continuous approaches and discontinuous approaches, and avoids or mitigates their respective drawbacks.[10]

Critical to the implementation of DEM is the detection and resolution of interparticle contacts. Contact detection refers to identifying the contacts between particles and contact resolution refers to computing the contact geometric features such as contact overlap, contact normal, and contact point, which is required by contact models to further evaluate contact forces.[12–14] Generally, the algorithms for contact detection and resolution are associated with the particle models adopted to describe the particle geometries. For simplicity and computational efficiency, particles are typically modeled using spheres (3D) or circles (2D) in many DEM studies, even though real particles are generally irregular-shaped.[15–18] The simplification of an irregular-shaped particle to a spherical or circular particle may underestimate the translational and rotational resistances between particles, and thus, could result in spuriousness in the bulk mechanical responses, including the fabric anisotropy, the force transmission mechanisms, and the deformation characteristics.[3,19,20] In view of the significance of particle shape effects, developing irregular-shaped particle models and integrating realistic particle shapes into DEM simulations have been heated topics in recent years. For example, particle models have been developed based on ellipsoids,[21] superellipsoids,[19] poly-ellipsoids,[22] poly-superellipsoids,[23] superquadrics,[24,25] cylinders,[26] polyhedron,[27] nonuniform rational basis splines,[12,28] and Fourier series,[13] to list a few. These particle models have significantly improved the fidelity of a DEM simulation.

Despite the considerable efforts on particle models, there remain significant challenges. First, existing particle models still only cover a small portion of the various particle shapes that can be encountered in nature or engineering applications.[29] Second, with irregular particle shapes, the corresponding contact detection and resolution algorithms generally require considerable implementation efforts and are the most challenging part of a particular model. In addition to these challenges, another important aspect of a particle model is the computational efficiency. Previous studies show that the procedure of contact detection and resolution generally takes up to 80% running time of a DEM simulation.[30] The situation would be worse for nonspherical particles. For a nonspherical particle represented by a continuous function or a series of discrete functions, the contact detection and resolution problem often involves a iteration-based numerical optimization method, such as Newton's method,[12,13] the Levenberg–Marquardt method,[23] the Nelder–Mead simplex algorithm,[20] and the Gilbert–Johnson–Keerthi algorithm.[31,32] The optimization process could be computationally expensive and limit the application of these particle models to simulate practical problems. All these aforementioned challenges and limitations call for the continuous development of complex particle shape models and more general contact detection and resolution algorithms.

Mathematically, contact detection and resolution algorithms take particle positions and geometries as inputs, and the outputs of the algorithms are the contact status and the corresponding contact geometric features should a contact exists. In this work, we explore machine learning (ML), specifically deep learning based on neural networks, as a viable approach towards contact problems. ML encompasses a set of tools and algorithms that are now becoming popular in almost all scientific and technological fields.[33] It serves as a black box that establishes a relationship between the inputs and outputs by learning the underlying mechanisms in the training data.[34] Recently, there have been growing efforts in integrating ML into computational mechanics, mainly in the field of continuum mechanical methods, such as FEM. For example, ML has been utilized to identify material parameters in constitutive laws,[35,36] to automate the creation and validation of physically meaningful traction–separation models,[37] and to enhance the numerical integration of FEM stiffness matrices.[38] The application of ML in the field of discrete and particle-based methods such as DEM is a topic much less explored, and a few pioneering works can be found in recent studies.[39–47] In these studies, the DEM is referred to as a physical engine, where ML is utilized to perceive and reason the relationships between objects.[40,41] ML models are constructed with the physical properties (e.g., mass, position, shape, and friction) of the objects at a series of previous time steps taken as inputs, and the ML models return the positions of the objects at the next time step.[42] As in the computer science community, an important feature of these models is that they put an emphasis on scene prediction, whereas the mechanical responses

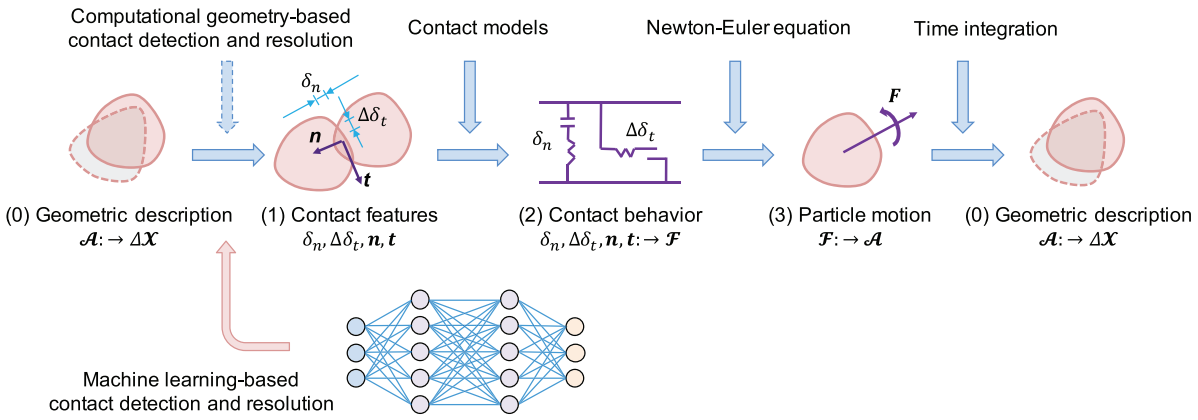


FIGURE 1 Schematic illustration of a typical calculation cycle in discrete element method (DEM) and the integration of machine learning (ML) for contact detection and resolution

(e.g., the contact forces and packing stresses) in the scene are less well explored. In addition, these studies have mainly focused on coarse-grained dynamics problems with very sparse object collisions. The numerical stability in the case of a dense granular medium under quasi-static equilibrium has not been discussed.

In this work, we propose a ML-enabled DEM, in which ML is utilized to solve the contact detection and resolution problem. Compared with previous studies of ML in discrete and particle-based methods, the proposed ML-enabled DEM has two critical and novel features. The first is the explicit consideration of irregular-shaped particles and the contacts in-between, and the second is the applicability of the proposed method for granular media with dense contacts. For irregular particle shapes, we adopt the single-particle approach, which is different from a previous work,[47] where the composite-particle approach with spheres is used to model arbitrary particle shapes and deformable materials. For contact detection and resolution, we propose and use ML. With the contact detection and resolution resolved by ML, contact models and time integration methods of the conventional DEM are directly incorporated. The developed model is used to simulate the mechanical responses of granular media under different loading conditions. In this work, we limit our discussion to two-dimensional DEM, while we note the applicability of the method to three dimensions since the implementation of ML models is general.

The remainder of this paper is structured as follows. Section 2 presents an overview of the proposed ML-enabled DEM. Section 3 provides details of the methodology, including the design and architecture of the adopted ML models, the definition of inputs and outputs, and the approach to prepare training and testing data. Section 4 presents the results of the training and evaluation of the adopted ML models. Section 5 shows six numerical experiments to demonstrate the performance of the proposed ML-enabled DEM. Section 6 presents some further discussions, and finally, Section 7 summarizes the proposed method and main findings.

2 | OVERVIEW OF THE PROPOSED METHOD

Figure 1 shows the schematic illustration of the proposed ML-enabled DEM. Similar to conventional DEMs, the proposed method encompasses four main steps in one typical calculation cycle, namely, (1) the detection and resolution of contacts, (2) the evaluation of contact behavior, (3) the calculation of particle motion, and (4) the updating of particle geometric descriptions. Among them, the detection and resolution of contacts are the most challenging and computationally intensive step in a DEM calculation cycle, especially for irregular-shaped particles. Contact detection refers to identifying whether or not two particles (or a particle and a boundary) are in contact. Contact resolution refers to resolving contact geometric features that are used in evaluating contact forces. In this work, a novel ML-based approach is proposed for contact detection and resolution. Unlike common contact detection methods used in conventional DEMs, the proposed approach is general and can be implemented to work with any irregular-shaped particles and contact models.

The outputs of contact detection and resolution are the status of particle–particle (or particle–wall) contacts and contact features, which are contact model dependent. For example, the spring-based linear contact model[1] and the Hertz-based nonlinear contact model[48–51] are two widely used contact models in DEM. For the cohesive effects between contacting

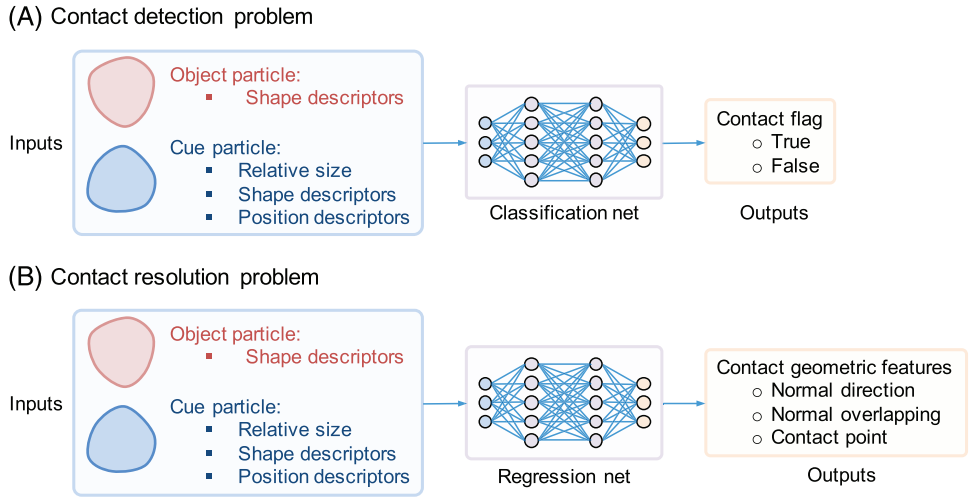


FIGURE 2 Overview of the two machine learning (ML) models for contact detection and contact resolution

particles, contact models such as the bond-particle model[52,53] and cohesion contact model[54,55] can be further incorporated. These contact features are used as inputs by the contact model to calculate contact forces (step 2 in Figure 1). Particle equations of motions (Newton–Euler equations) are then solved for particle motions and kinematics (step 3 in Figure 1), followed by the central difference time integration scheme (may also referred to as the velocity Verlet-based time integration scheme)[56–58] to update the particle geometric description (step 4 in Figure 1). There are well-established methods and algorithms from conventional DEMs that can be directly adopted for steps (2)–(4). In the following section, the novel ML-enabled contact detection and resolution approach will be presented in detail.

3 | ML-BASED CONTACT DETECTION AND RESOLUTION

In the proposed approach, ML models, specifically, artificial neural networks (ANNs), are constructed to predict the contact status and contact geometric features of particle–particle and particle–boundary contacts in DEM calculations. In this section, we describe in detail the design and architecture of ANNs, their inputs and outputs, and the preparation of data for model training.

3.1 | Design and architecture of ANNs

The problem of contact detection and resolution consists of two stages. First, given two particles (or a particle and a boundary), we need to identify if the two elements are in contact. Second, if the two elements are in contact, we need to resolve the contact geometric features (for use in the contact model). In this work, two ML models are constructed, one for contact detection and one for contact resolution. Specifically, two types of ANNs are constructed, which are termed the classification and the regression nets, as illustrated in Figure 2. The classification net is used to determine the contact status, and the regression net is used to predict the contact geometric features.

The classification and regression nets have similar architectures, which consist of an input layer, a number of hidden layers, and an output layer, as shown in Figure 3. The key differences lie in the last hidden layer and the output layer. In the classification net, the output layer has a dimension of two, namely, true and false, indicating the contact status of the two elements. The transfer function in the last hidden layer is *softmax*. In the regression net, the output layer has a dimension of one, which is the parameter to be predicted, and the transfer function in the last hidden layer is *purelin*. For all other hidden layers, the transfer function is *tansig*, which is a popular function that allows the network to learn nonlinear relationships between inputs and outputs. The number of hidden layers and the dimension of each hidden layer are problem-dependent and can be tuned based on the performance of the nets. The literature on ANNs is extensive and relatively mature, and thus the methodologies of ANN and transfer functions are not exhaustively presented here. The reader is referred elsewhere[59,60] for more details.

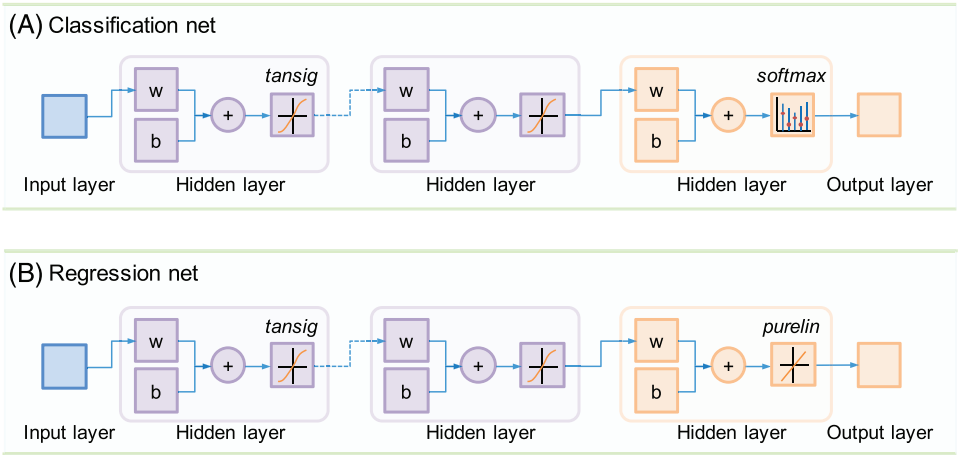


FIGURE 3 Architecture of the classification and regression nets

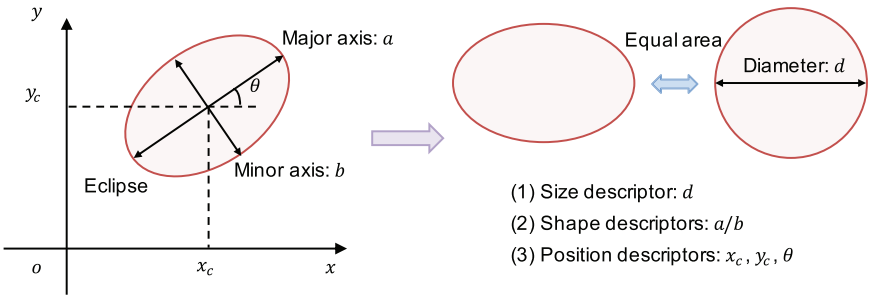


FIGURE 4 The definition of particle inputs, that is, size, shape, and position descriptors, illustrated using an elliptical particle

3.2 | Inputs and outputs of ANNs

The inputs of ANNs are the particle and boundary geometric descriptors. In DEM, particles and boundaries are the two types of basic elements. Typically, particles have a closed geometry, whereas boundaries are represented by open polylines or curves. Different approaches are available to describe a particle or a boundary in DEM, depending on the specific shape and the adopted contact detection and resolution algorithms. For example, elliptical particles can be described by a second-order polynomial function, the coordinates of the centroid, and the rotation angle from the major axis. To regularize the implementation of ML-based contact detection and resolution, we generalize the geometric description of a particle or boundary into three components, namely, size descriptor, shape descriptors, and position descriptors (see Figure 4). The size descriptor of a particle is defined as the diameter of a circle that has the same area as the particle. The shape descriptors are a set of parameters (of the associated shape function) that determine the surface of the particle. For example, the shape descriptor of an elliptical particle could be the ratio of the major axis to the minor axis of the ellipse. The position descriptors are the coordinates of the particle centroid and the rotation angle of the particle from its initial alignment. These descriptors will be taken as the inputs of the ML models.

It is worth noting that one might consider combining the size and shape descriptors into one component with a set of parameters. For instance, in the case of elliptical particles, the size and shape of a particle can be determined by using the major and minor axes, without the need to normalize them to unit size. However, in this work, we choose to separate the size and shape descriptors for two reasons. First, such separation allows us to isolate and characterize particle shapes and perform statistical analyses of shape parameters on particles of the same shape but different sizes. Second, by separating the size descriptor, it is convenient to convert the contact problem between two particles of various sizes into the contact problem between two particles of normalized sizes such that one particle has a unit size whereas the other particle is of relative size. In this way, inputs of the ML models can be regularized by using the unit size and relative size. For instance, in practical applications, the ratio of the largest to the smallest particle size in a DEM model is usually no greater than 10. Size descriptors provide a convenient way to control the range of relative particle sizes in a DEM model.

For boundary elements, as in most conventional DEMs, a boundary can be represented by open polylines or curves. Therefore, the boundary can be fully described by the parameters of the polyline function, whereas the size descriptor

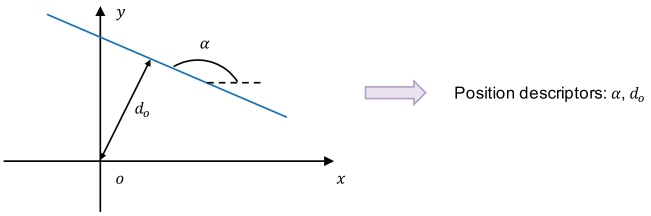


FIGURE 5 The definition of boundary inputs, that is, position descriptors, illustrated using a straight line

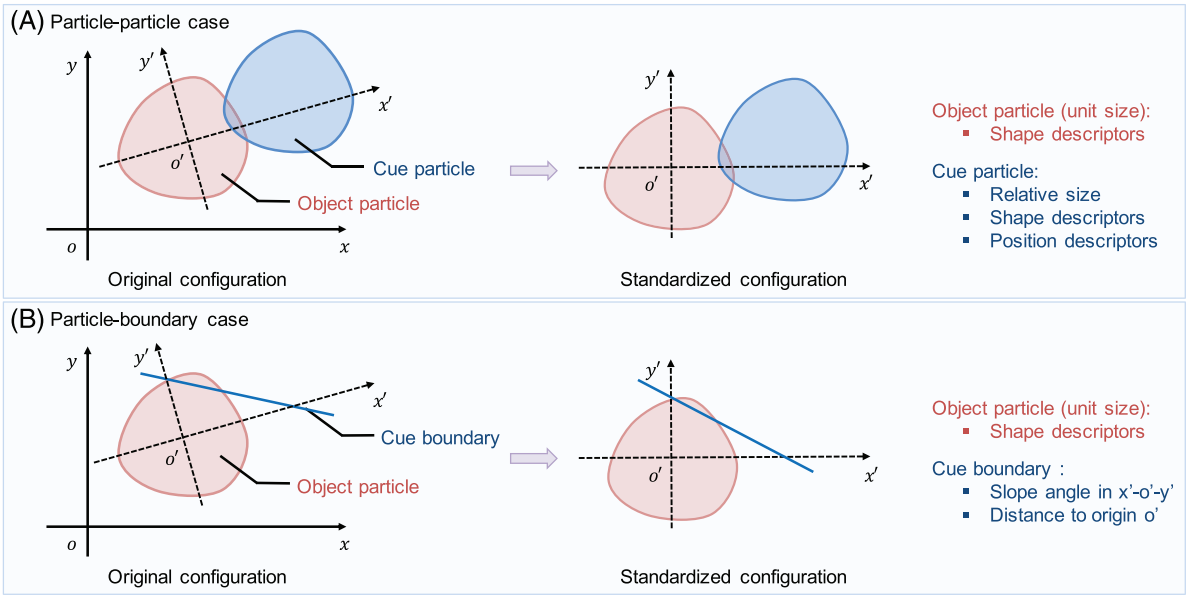


FIGURE 6 The regularization of particle and boundary geometric descriptors in the ML-based DEM

and shape descriptors are no longer needed. For example, a line boundary could be expressed as

$$x \sin \alpha - y \cos \alpha - d_o = 0 \tag{1}$$

where $\alpha \in [0, \pi)$ is the angle of the boundary line; d_o is a parameter indicating the signed distance from the origin to the line. The variables α and d_o are taken as the position descriptors, as illustrated in Figure 5.

With the definition of the particle and boundary geometric descriptions, the inputs of the ML models for the contact detection and resolution problem can be determined. An intuitive option would be to directly take all the original size, shape, and position descriptors as inputs. A limitation of this option, however, is that the position descriptors (typically the coordinates of the particle centroids) are essentially unbounded and could vary with the dimensions of a simulation domain. It is inefficient and infeasible to prepare a training data set that could cover all the possible ranges of position descriptors. Instead, in this work, we propose to transform the original configuration of the contact problem into a standardized configuration by recognizing that the relative position of two particles is usually within a limited range (i.e., less than the size of the particles' bounding circles if they are in contact). This transformation is illustrated in Figure 6, which is performed in two steps. First, the coordinate system is translated and rotated such that the centroid of the object particle is located at the origin and the rotation of the object particle is zero. Second, the coordinate system is scaled such that the object particle is of unit size. In this way, the final inputs of the particle–particle contact problem, as illustrated in Figure 6, are the shape descriptors of the object particle, and the size, shape, and position descriptors of the cue particle, all in the standardized configuration.

The outputs of ANNs are the contact status and contact geometric features. In DEM, the contact geometric features are the set of parameters required by contact models to evaluate contact forces. Currently, a general contact theory for particles of general shapes is lacking.[26] Therefore, the contact geometric features are dependent on the adopted contact model and might not be unique. Considering the widely used spring-based linear contact model,[1] the following contact geometric

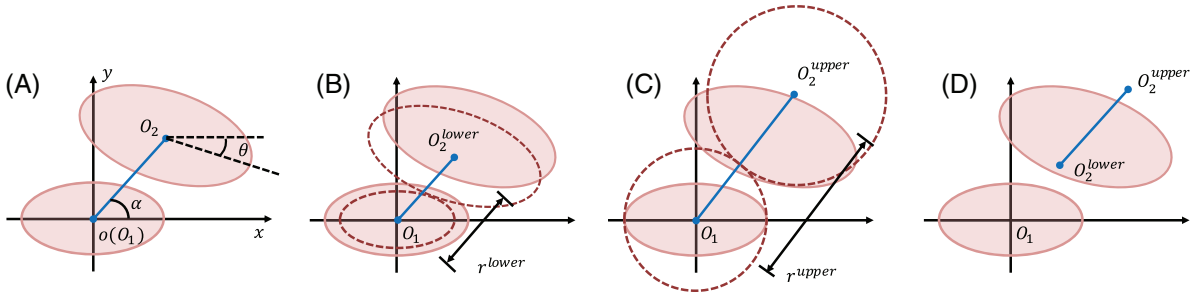


FIGURE 7 Illustration of the procedure of preparing training and testing data for particle–particle contacts

features are involved: the contact normal direction, the tangential direction, the branch vectors, the normal overlapping length, and the increment of the tangential overlapping length.[1,13] The definition of these contact geometric features are presented in Appendix A. As these contact geometric features are not completely independent from one another, the outputs in ANNs can be reduced to contact positions, normal overlapping length, and normal directions for the particle–particle case, and to contact positions and normal overlapping length for the particle–boundary case. All other contact geometric features can then be derived from these outputs.

It should be noted that if a different contact model is used, the output parameters will be modified based on the specific formulation and parameters of the contact model. In addition, in this work, we choose to split the contact geometric features into a list of independent variables and predict each of the contact geometric features using one specific regression net. An alternative way is to use a single regression net to predict all contact geometric features. This latter option is simpler in implementation, but it would require a deeper net to accommodate the complexity of the underlying relationships between the inputs and outputs. This would then decrease the computational efficiency and increase the need for training data.

3.3 | Preparing training and testing data

The performance of an ML model relies heavily on the feeding of the training data. For the contact detection and resolution problem, the training data are the contact scenarios parameterized as the particle and boundary geometric descriptors (i.e., the inputs), and the corresponding contact status and geometric features (i.e., the outputs). Hereafter, we consider a complete set of inputs and outputs of a contact scenario as one data sample in the training data set. In the following, the procedure to generate data for both particle–particle and particle–boundary contact cases is described in detail.

3.3.1 | Particle–particle case

Figure 7 illustrates the key steps in generating one data sample for the particle–particle contact case. The procedure consists of the following steps.

(1) Generate the size and shape descriptors of the two particles in the contact scenario. As discussed in Section 3.2, the object particle (particle O_1 in Figure 7A) is set to be a unit size, whereas the size of the cue particle (particle O_2 in Figure 7A) can be any random value in the specified range of relative particle sizes. For example, if the ratio of the largest to the smallest particle sizes is 10, the specified range would be 1–10. Shape descriptors are dependent on the specified particle shape templates and the range of the associated parameters. For elliptical particles, the ratio of the major axis to the minor axis is a shape parameter, and its value can be determined by selecting a random value in the specified range. To capture more realistic particle shapes, the shape descriptors can be determined based on the results of image-based particle morphology characterization and statistical analysis, for example, elsewhere[18,61–64] It is worth noting that the specific distributions of particle size and shape parameters are not required in this step; only the ranges of these descriptors matter.
(2) Generate the position descriptors of the cue particle. In DEM, the contact detection is usually implemented with two phases, namely, a broad phase and a narrow phase. The broad phase is used to quickly identify and filter out pairs of

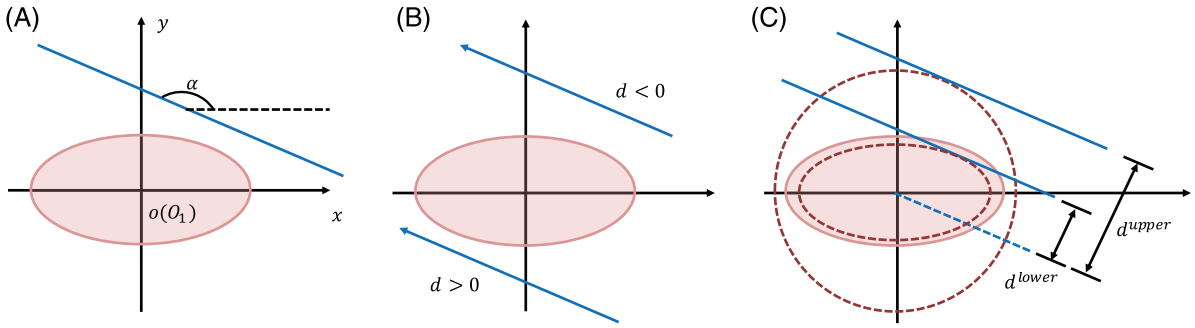


FIGURE 8 Illustration of the procedures of preparing training and testing data for particle–boundary contacts

elements that are not in contact based on fast algorithms such as the bounding circle algorithm, whereas the narrow phase performs the more expensive and exact contact detection calculations. In this work, only the narrow phase needs to be handled by the ML models. The subroutine for generating a sample of cue particle position descriptors that conforms to a narrow phase scenario are described as follows.

(a) Determine the rotation angle θ (shown in Figure 7A) by randomly selecting a value in the range from 0° to 360°.
(b) Determine the angle α (shown in Figure 7A) by randomly selecting a value in the range from 0° to 360°. Herein, α is the polar angle of position O_2 (centroid of the cue particle shown in Figure 7) in the polar coordinate system.
(c) Determine the upper bound r^{upper} of the polar radius of position O_2. Herein, the upper bound r^{upper} is defined as the maximum possible distance between the particle centroids in the narrow phase. Assuming that the bounding circle algorithm is adopted in the broad-phase contact detection, r^{upper} equals the sum of the radius of the two bounding circles, as illustrated in Figure 7B. The corresponding centroid of the cue particle is labeled as O_2^{upper}.
(d) Determine the lower bound r^{lower} of the polar radius. Herein, the lower bound r^{lower} is defined as the minimum possible distance between the particle centroids, and the corresponding centroid of the cue particle is labeled as O_2^{lower}. In a DEM simulation, the normal overlapping length is typically bounded, for example, less than 20% of the particle size. Thus, the lower bound r^{lower} could be approximated as the smallest distance between particle centroids at which the two particles, each scaled to 90% of its original size, could come in contact. With this lower bound r^{lower}, the two particles would have an overlap of approximately 20% of the particle sizes. Here, particle size is quantified in terms of the diameter of the equal-sized circle.
(e) Determine the polar radius r (of the cue particle centroid) by randomly selecting a value in the range from r^{lower} to r^{upper}. The Cartesian coordinates of the centroid of the cue particle (O_2) can then be determined with r and the polar angle α.

3. Repeat steps 1 and 2 until the required number of data samples are generated.

3.3.2 | Particle–boundary case

In the particle–boundary contact case, the procedure to generate data samples is described as follows.

(1) Determine the shape descriptors of the object particle. This step is the same as that of the particle–particle contact case.
(2) Determine the position descriptors of the boundary using the following subroutine:
   (a) Determine the angle of the boundary α by randomly selecting a value in the range from 0° to 180°.
   (b) Determine the sign of the distance d_o by randomly selecting a value of −1 or 1. A positive sign indicates that the origin is on the right side of the boundary line, and a negative sign indicates that the origin is on the left side, as illustrated in Figure 8B.
   (c) Determine the upper bound d^{upper} of the distance magnitude. Similar to the particle–particle contact case, the upper bound d^{upper} is taken as the radius of the bounding circle of the object particle.
   (d) Determine the lower bound d^{lower} of the distance magnitude. The lower bound d^{lower} is approximated as the smallest distance between the particle centroid and the boundary line at which the particle, when scaled to 90% of its original size, could come in contact (see the dashed ellipse in Figure 8C).

TABLE 1 Settings of the artificial neural network (ANN) models and training solver

Parameter	Value
Hidden layers	Three hidden layers, each having a dimension of 15
Data division	90% testing and 10% validation at random
Training function	*trainlm* with default values in MATLAB
Maximum epochs	5000
Validation checks	10

(e) Determine the distance magnitude $|d|$ by randomly selecting a value in the range from d^{lower} to d^{upper}.

3. Repeat steps 1 and 2 until the required number of data samples are generated.

4 | TRAINING AND EVALUATION OF ANNs

With the ANNs, the inputs and outputs, and the procedure of data generation defined, we use elliptical particles to illustrate the proposed method and present the results of training and evaluate the performance of trained ANNs in this section. Elliptical particles are adopted in this initial study as they are a common type of noncircular particle, and analytic contact detection and resolution algorithms are available. The aspect ratio, that is, the ratio of minor to major axis, of the particle is assumed to be 0.5. Particle sizes range from 1 to 2 after normalization.

4.1 | Data sets and training setup

Following the procedures provided in Section 3.3, the data sets for network training and testing are first prepared. In this work, the contact geometric features, defined in Appendix A, are evaluated based on computational geometric algorithms. A total of 100,000 samples of particle–boundary contacts and 400,000 samples of particle–particle contacts are collected. Each sample represents a case of contact scenario, in which particle or boundary has random size, position and rotation. The size, position, and rotation are the inputs of the ANN models. Also, each sample (i.e., a case of contact scenario) would have a contact status and a set of contact geometric features, which are the outputs of the ANN models. These samples are divided equally into training and testing data sets. With the data sets prepared, the commercial software MATLAB is used to implement, train, and test the ANN models. There are ten ANN models in total, namely, one classification net and five regression nets for particle–particle contacts, and one classification net and three regression nets for particle–boundary contacts. For ANN, the number and dimensions of hidden layers need to be determined. In this work, the same number and dimensions of hidden layers are adopted for all 10 nets. The Levenberg–Marquardt optimizer with a maximum of 5000 epochs and 10 validation checks is adopted for the training. The settings of the ANN models, and the training solver, are tabulated in Table 1. By trial and error, it is found that the ANN models with three hidden layers, with each hidden layer having a dimension of 15, and with the given 50,000 and 200,000 training samples, can provide satisfactory results of contact detection and resolution. Figure 9 shows an example of the performance of the classification net with an increasing number of training samples in the particle–boundary case. When trained with 50,000 samples, the classification net can correctly predict the contact status of more than 99% of the random contact cases. More details about the training and testing performance of the ANN models will be presented in the subsequent section.

4.2 | Results of training and testing

We first investigate the performance of the classification and regression net for particle–boundary contact cases. The performance of the classification net is quantified as the percentage of cases for which the contact status can be correctly predicted. Herein, the contact status is correctly predicted if a true contact is identified as true by the classification net, and a false contact is identified as false. We found that the trained classification net correctly predicted 99.94% of the samples in the training dataset and 99.92% of the samples in a testing data set (i.e., data set not used in the training process). Figure 10 shows two examples of incorrect predictions, namely, a true contact predicted as false, and a false contact predicted as

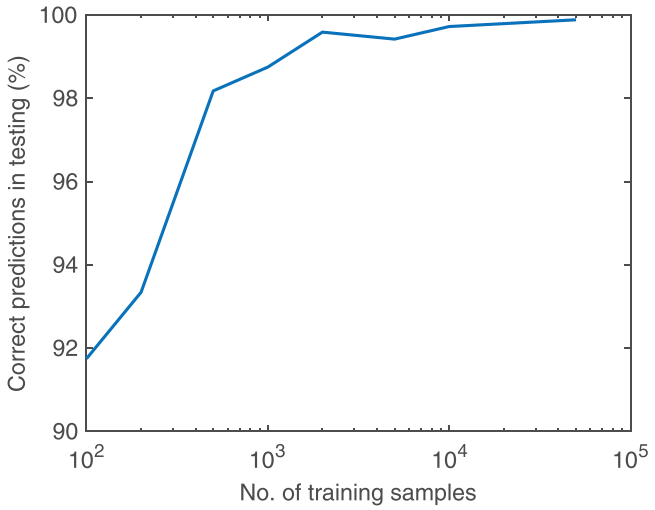


FIGURE 9 The performance of the classification net with an increasing number of training samples in the particle–boundary case

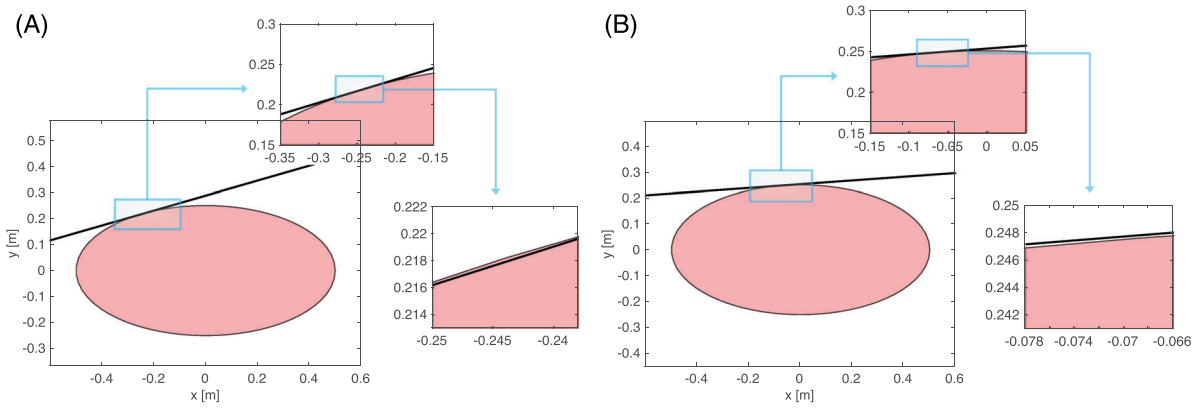


FIGURE 10 Examples of incorrect predictions in the particle–boundary contact case: (A) a true contact is predicted as false, and (B) a false contact is predicted as true by the classification net

true by the classification net. An incorrect prediction could occur when the particle overlaps the boundary with a very slight depth or when the particle has a very small clearance to the boundary.

The performance of the regression nets is quantified by the mean absolute errors (MAEs) in the predicted contact geometric features. For evaluation, the samples with true contact, and those predicted as true, are extracted. These samples are then cast into the regression nets to predict the contact geometric features. Figure 11 plots the predicted contact geometric features over the corresponding ground truths. For both the training and testing data sets, the predicted results and the ground truths match fairly well. The MAEs are on the order of 10^{-5} or smaller.

Next, the performance of the classification net and regression nets for particle–particle contacts are investigated. Based on the trained classification net, the percentage of correct predictions is approximately 99.87% for the training data set and approximately 99.67% for the testing data set. Figure 12 shows the results of the predicted contact geometric features and the ground truths, which again present a fairly good match. The MAE in the predicted overlapping length is on the order of 10^{-4}, which is greater than the value of 10^{-5} for the particle–boundary case. The MAEs in the coordinates of the contact points and the contact normals are on the order of 10^{-4} and 10^{-3}, respectively. Considering the specific data range of the contact geometric features, the MAEs overall are on the order of 10^{-4} if normalizing the data range to $[-1, 1]$. The results indicate that it is appropriate to use the same layout (i.e., the same number and dimensions of hidden layers) for every regression net, the performances of which are similar for different contact geometric features. On the other hand, the regression nets of the same layout overall perform better in the particle–boundary contact cases than in the particle–particle contact cases. This phenomenon is reasonable since the contact resolution between ellipses, as in

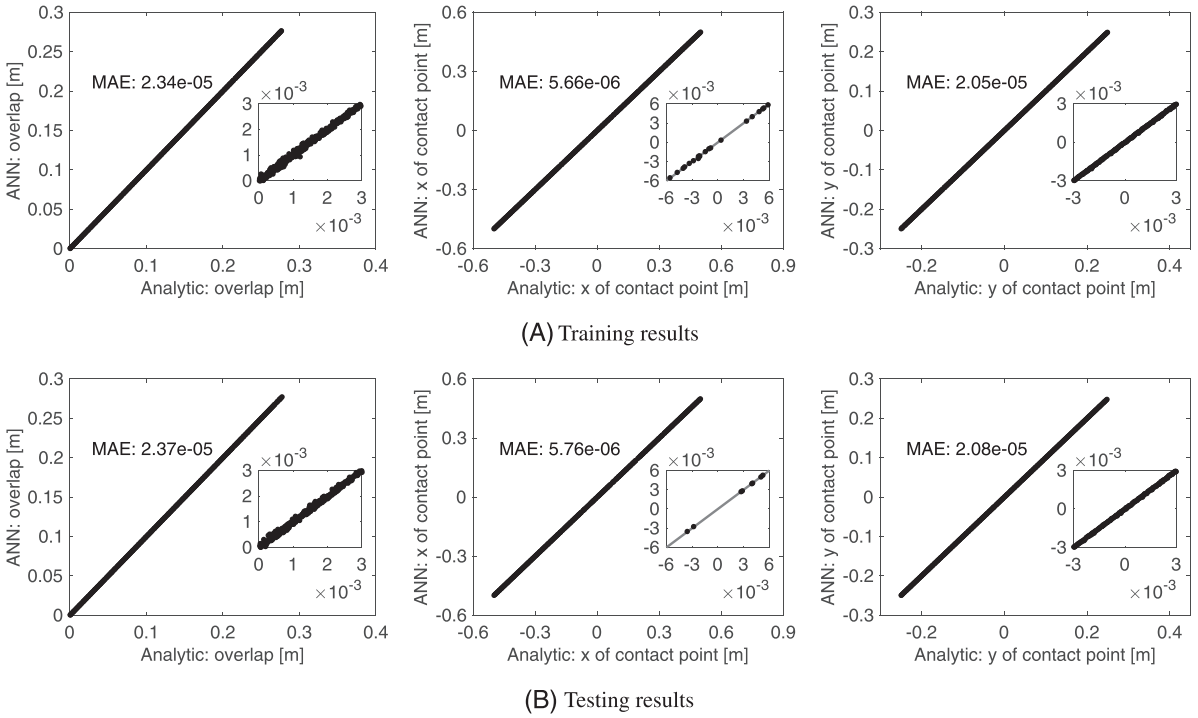


FIGURE 11 (A) Training and (B) testing results of the regression net for contact resolution of the particle–boundary case. The predicted values of the normal overlapping length and contact point coordinates are plotted over the ground truth values based on geometric algorithms. The solid blue line indicates the 1:1 curve

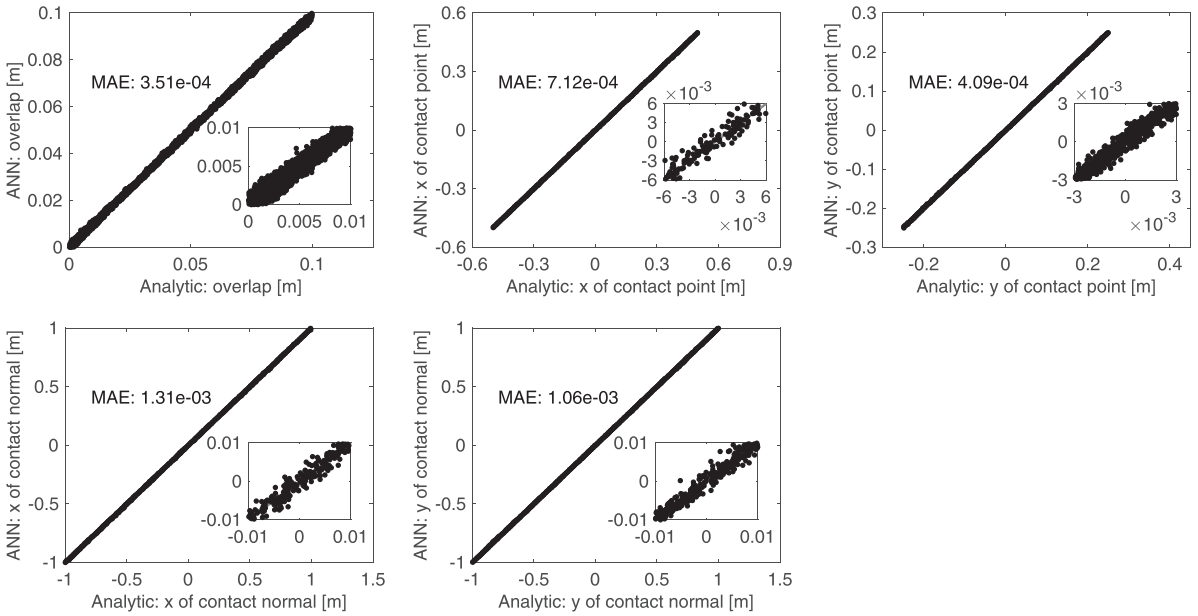
the particle–particle contact cases, is more complex than that between an ellipse and a line, as in the particle–boundary contact cases.

5 | NUMERICAL EXAMPLES

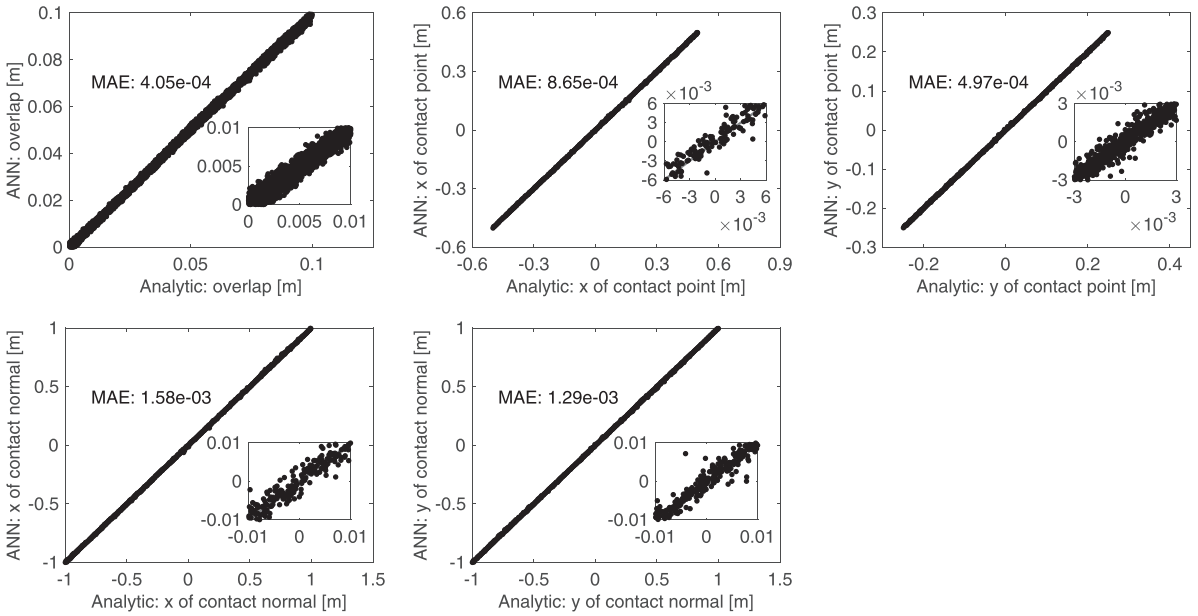
In this section, we will present six numerical experiments, including simple pendulum, rockfall, packing, oedometric compression, and angle of repose tests, to illustrate and evaluate the performance of the proposed ML-enabled DEM. The first five numerical experiments are based on the elliptical particles that have been described in the previous section. The last experiment involves arbitrary irregular-shaped particles to showcase the potentials and applicability of the proposed method for other particle shapes. The results of the proposed ML-enabled DEM and the conventional DEM will be compared. For the convenience of discussion, hereafter, we will use NetDEM to denote the proposed ML-enabled DEM, which uses neural networks for contact detection and solution; and GeoDEM to denote the conventional DEM, which uses geometric algorithms for contact detection and solution. It should be noted that there is no training and testing involved during the runtime of these example DEM simulations. The training and testing are performed *offline* and are at one-time cost, to obtain and test the classification and regression nets. When the classification and regression nets are well-trained and tested, they are then integrated into the DEM to detect contact and resolve contact geometric features, and besides that, the NetDEM runs through the similar procedures as the conventional GeoDEM does (as discussed in Section 2).

5.1 | Experiment 1: simple pendulum

The first numerical experiment considers the simple pendulum test for a single particle. As the angular displacement of a pendulum has a well-established analytical solution, the purpose of this experiment is to verify the implementation of the ANN models and the DEM code. To simulate the simple pendulum test in the DEM, a particle is suspended from fixed support by a rope (considered as a linear spring). The rope provides centripetal forces, which are calculated from the increment of the distance between the hanging and release points multiplied by the spring stiffness. The particle is released with an



(A) Training results



(B) Testing results

FIGURE 12 (A) Training and (B) testing results of the regression net for contact resolution of the particle–particle case

initial small angular displacement and is allowed to swing freely back and forth under the influence of gravity. To involve a colliding event, two scenarios are considered. In the first scenario, the particle collides with a wall when the angular displacement becomes zero, as shown in Figure 13A, and in the second scenario, the particle collides with another particle, as shown in Figure 14A. As for the specific model setup, the particle has a major axis of 0.1 m and a minor axis of 0.05 m, and the particle density is 2700 kg/m^3. The particle is initially placed with an angular displacement of 20°, and the length of the hanging rope is 1.0 m. The normal contact stiffness and tangential contact stiffness are both set to 1.0×10^7 N/m, and the spring stiffness of the hanging rope is 1.0×10^8 N/m. The damping and friction are zero. The simulations last for a complete pendulum period with a fixed time step of 1.0×10^{-4} s. Figure 13B and Figure 14B show the angular displacements of the particles over time. Overall, the results of the DEM simulations compare fairly well with the analytical solution.

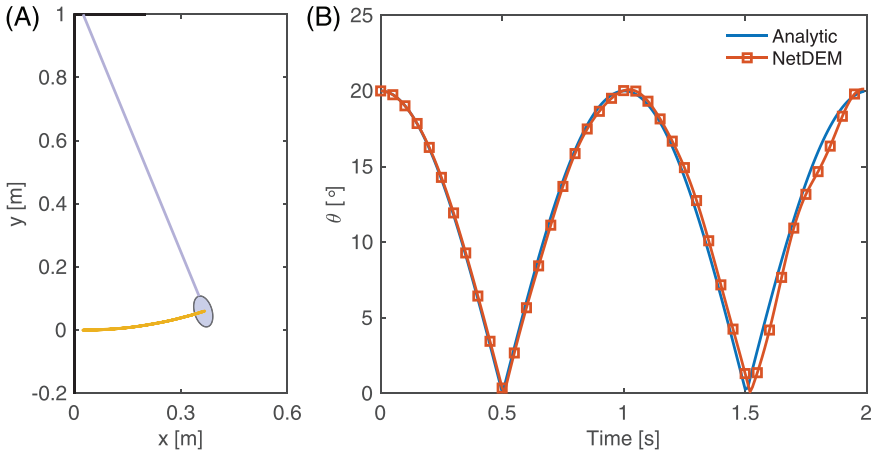


FIGURE 13 Simple pendulum test with a particle–boundary contact event: (A) simulation setup, and (B) evolution of the inclination angle θ

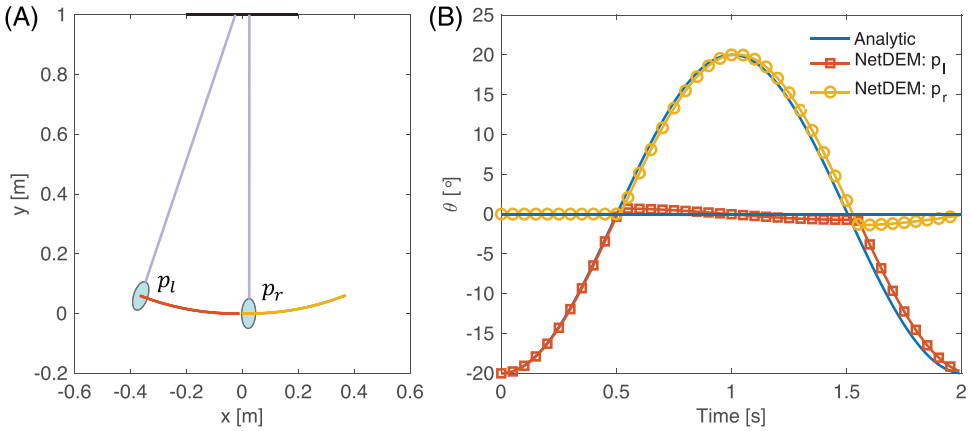


FIGURE 14 Simple pendulum test with a particle–particle contact event: (A) simulation setup, and (B) evolution of the inclination angle θ. The symbols p_l and p_r indicate the particles on the left side and right side, respectively, in (A)

5.2 | Experiment 2: rockfall

The second experiment considers the rockfall test, where a particle is released above a slope and then allowed to fall down along the slope. The model setup is illustrated in Figure 15. In this example, an elliptical particle with a major axis of 0.3 and a minor axis of 0.15 is used. The density of the particle is 2700 kg/m^3. The contact normal and tangential stiffness are 9.5×10^7 N/m, that is, 1.0×10^6 times the particle mass. The time step size is fixed at 1.0×10^{-4} s. Two simulation cases with different contact friction and contact damping values are considered. The first case considers no damping and no friction, whereas the second case considers damping of 0.2 and friction of 0.2. During the falling, the particle successively collides with the slope. The trajectory and energy evolution during a period of 2.0 s are recorded and analyzed.

The particle trajectory and energy evolution for the first simulation case (with no damping and no friction) are plotted in Figure 15. Along with the particle trajectory, the snapshots of the particle at the time when it collides with the walls are also presented. Before the beginning of the sixth collision, the particle trajectory evaluated by NetDEM fairly well matches that evaluated with GeoDEM. A notable discrepancy is observed after the seventh collision, and the discrepancy accumulates afterward. As discussed in the previous section, there is an MAE on the order of 10^{-4} in the predicted contact geometric features, which results in the discrepancy between NetDEM and GeoDEM. Interestingly, despite the deviations in the predicted trajectory, the total energy of the particle calculated from the NetDEM simulation is conserved, as shown in Figure 15A. This is likely due to the fact that the errors could be positive or negative such that their net effects on the energy are negligible.

Figure 16 shows the particle trajectory and energy evolution for the second simulation case (with damping of 0.2 and contact friction of 0.2). Again, the results of the trajectory and energy simulated from NetDEM have reasonable consistency with those simulated from GeoDEM at the very beginning, and afterward, notable discrepancies start to occur and

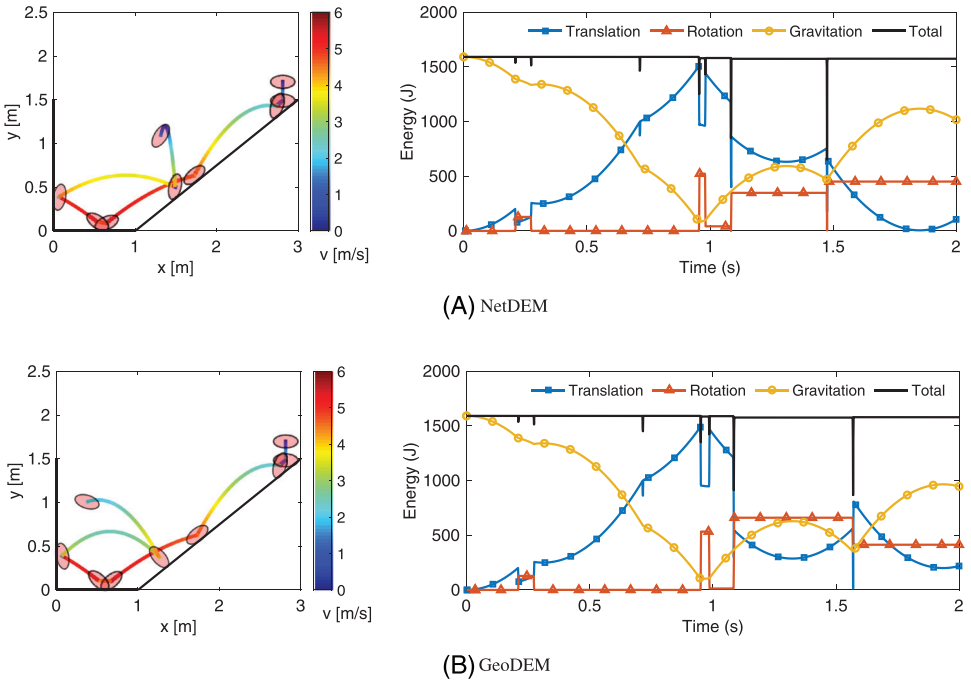


FIGURE 15 Particle trajectory and energy evolution in the rockfall test with no damping and no friction for (A) NetDEM and (B) GeoDEM. It should be noted that this test only involves a single particle, and a serial image of the particle is plotted in the figure to show the trajectory of the particle during the falling process

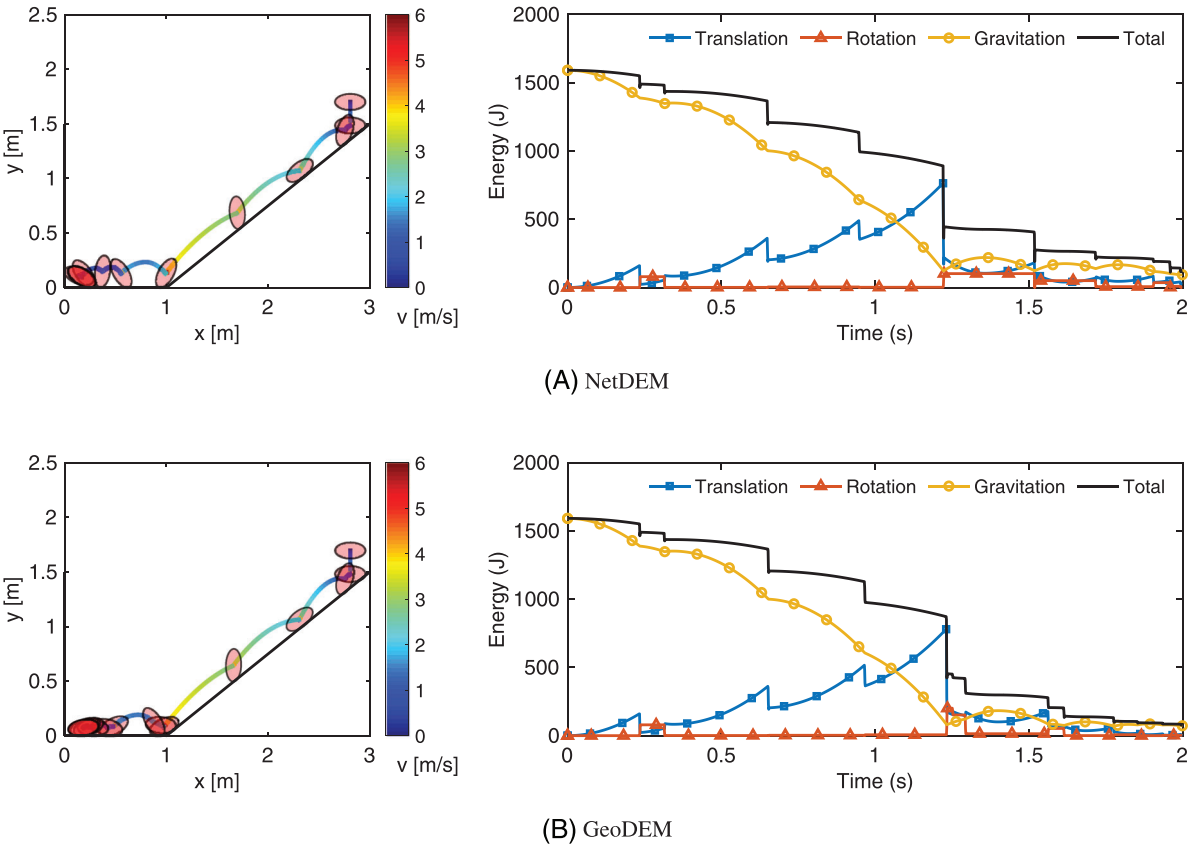


FIGURE 16 Particle trajectory and energy evolution in the rockfall test with damping of 0.2 and friction of 0.2 for (A) NetDEM and (B) GeoDEM. A serial image of the particle is plotted to show the trajectory

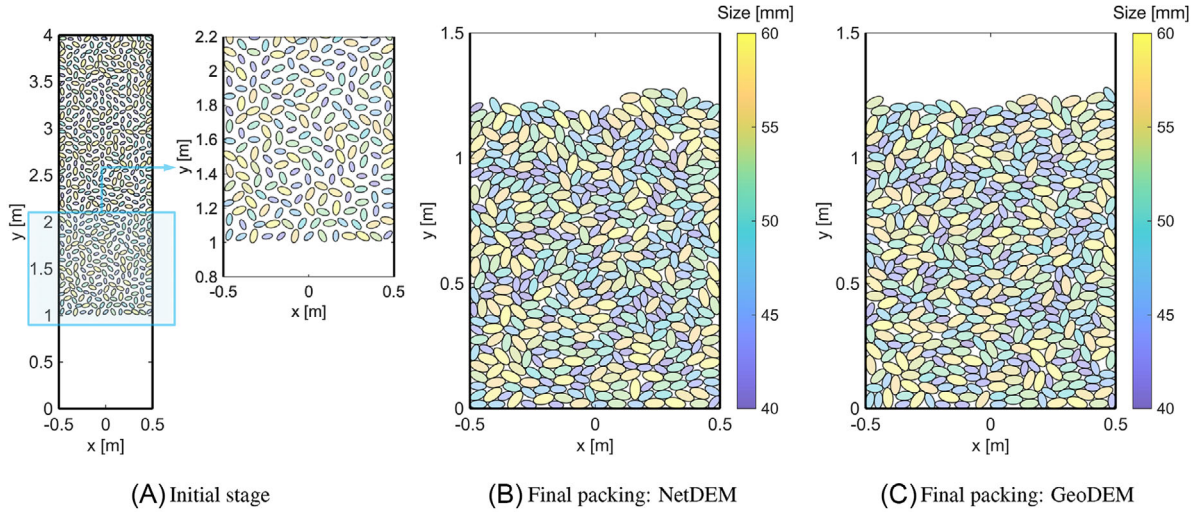


FIGURE 17 Snapshots of the packing simulation: (A) initial stage, (B) final packing based on NetDEM, and (C) final packing based on GeoDEM

accumulate as a result of the errors in the predicted contact geometric features. Nevertheless, the effects of damping and friction on the particle trajectory and energy are properly captured by NetDEM. The total energy of the particle decreases gradually with time due to the effects of damping, and it experiences a sudden decrease due to the effects of contact friction. With the dissipation of the kinetic energy in the particle, the particle gradually slows down and eventually comes to rest on the ground.

5.3 | Experiment 3: random packing

The third experiment considers the random packing test on polydispersed particles. In this example, a total of 500 elliptical particles are first randomly placed in a box region 1.0 m in width and 3.0 m in height and are then allowed to settle down under the influence of gravity, as shown in Figure 17. In particular, the position positions are obtained from a centroidal Voronoi tessellation with 500 seeds.[65,66] Thus, the particles roughly scatter evenly in the box. The particle rotation is controlled by the cross angle between its major axis and the x-axis. The rotation angles follow a uniform distribution with a range from 0° to 180°. The particle sizes follow a uniform distribution with a range from 0.04 to 0.06 m. As mentioned in Section 3.2, the equivalent size is defined as the diameter of the circle that has the same area as the particle. For all the particles, the ratio of the minor axis to the major axis is 0.5. Throughout the packing process, the following parameters are used: the contact normal and tangential stiffnesses are 1×10^7 N/m, the contact friction is 0.2, and the damping is 0.7. The simulation lasts for 5.0 s with a fixed time step of 1×10^{-4} s. Figure 17B,C compares the final particle assemblies simulated from NetDEM and GeoDEM, respectively. It can be observed that the two assemblies present similar heights, that is, similar porosities. The particles reside next to each other in the box, without any visible spuriousness such as unrealistic overlapping or floating particles.

As a quantitative analysis of the packing configuration, the alignments of the particles are calculated and plotted in Figure 18. Herein, the alignment of a particle is defined as the angle from the x-axis of the particle major axis and is bounded between 0° and 180°. For visualization purposes, the plots from 180° to 360° are also created in Figure 18 by rotating the plots from 0° to 180° by 180°. With this treatment, a particle that aligns at θ is assumed to also align at $\theta+$ 180°. The results in Figure 18 indicate that the particles present a preference for horizontal alignment. This phenomenon is expected, as the particles have a large elongation and pack freely under the influence of gravity.

In addition to the particle alignments, the coordination number of the particles is plotted in Figure 19. Based on the results of NetDEM shown in Figure 19A, the coordination number ranges from 1 to 5 with majority of them being 1 or 2. However, based on the results of GeoDEM shown in Figure 19B, the coordination number ranges from 1 to 7 with majority of them being 4 or 5. The results indicate that the particles have overall fewer contacts with other particles (or boundaries) in the NetDEM simulation than in GeoDEM simulation. With careful investigation, it is found that this phenomenon likely results from the effects of the errors in the predicted contact geometric features in NetDEM from two aspects. First,

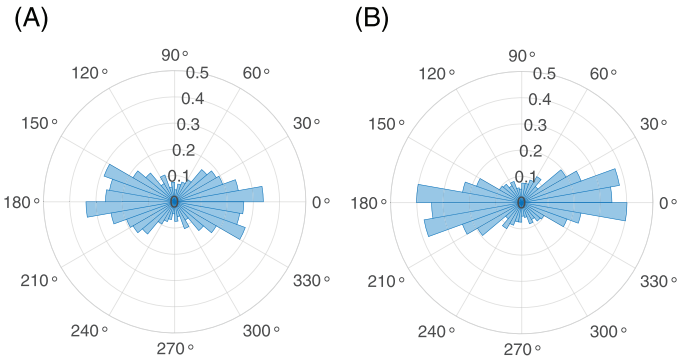


FIGURE 18 Polar histogram of the particle orientation: (A) NetDEM and (B) GeoDEM

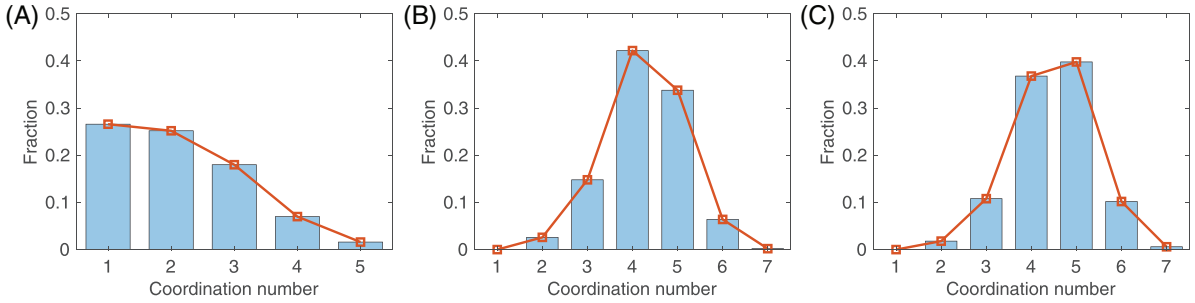


FIGURE 19 Histograms of the coordinate number: (A) NetDEM, (B) GeoDEM, and (C) NetDEM with one percent expansion in size

in the case of free packing, the overlaps between particles are fairly small, creating a situation where a true contact could be likely predicted as false, as previously demonstrated in Figure 10A. Second, although the MAE (i.e., on the order of 10^{-3} or less) in the predicted contact geometric features is minor, the error could still result in a spurious contact force that overwhelms the gravitational force of the particle. As a result, the particles experience slight oscillations around their expected equilibrium position. Figure 20 shows the evolution of the average particle translational velocity and the ratio of the kinetic energy to the contact potential energy (referred to as the relative kinetic energy hereafter) during the packing process. It can be observed that in GeoDEM, the average velocity, and the relative kinetic energy, eventually decreases to a minimal value, indicating a quasi-static equilibrium state. In NetDEM, however, the average velocity and the relative kinetic energy eventually remain vibrating around a stable constant. Due to the oscillation, two particles could be in contact at one moment while the two particles would separate at the next moment, as illustrated in Figure 21. The overall result is that the particles exhibit less contact with other particles in NetDEM than in GeoDEM. Nevertheless, it is expected that the oscillations are fairly small and would likely not affect the packing characteristics, such as the true coordination number. To confirm this aspect, the particles are expanded by one percent, and the coordination number is recalculated

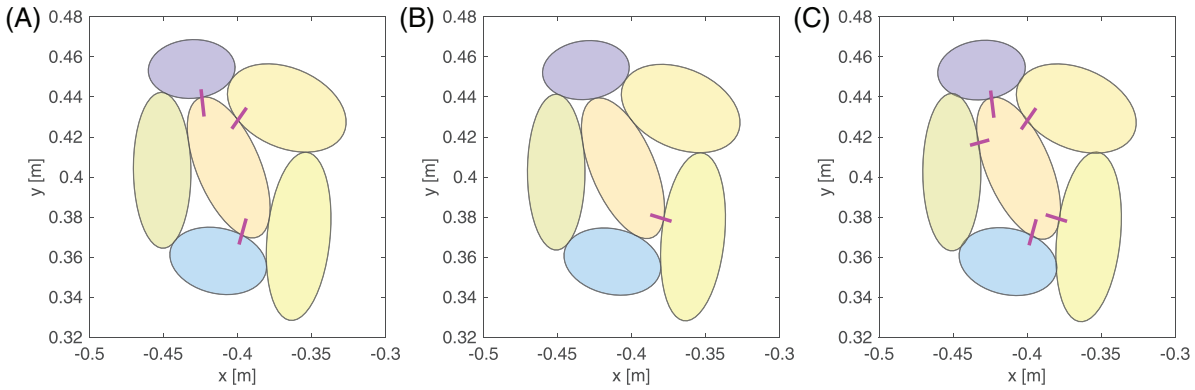


FIGURE 20 Evolution of the (A) average velocity and (B) relative kinetic energy (i.e., the ratio of the kinetic energy to the contact potential energy)

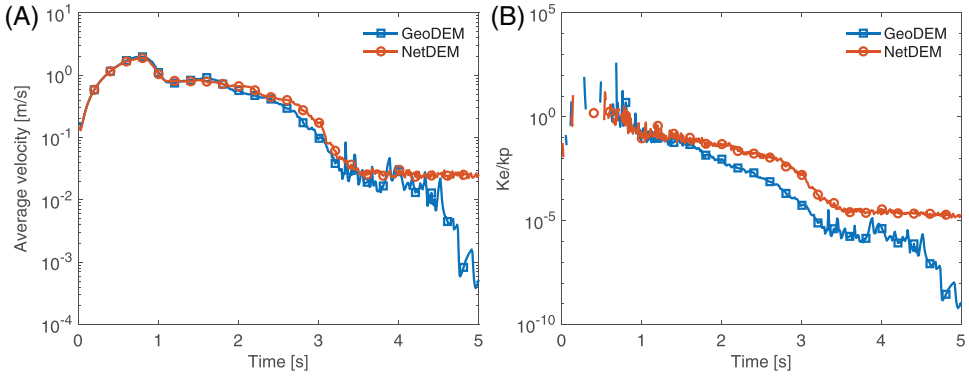


FIGURE 21 Contact statuses of a particle with its neighboring particles in the NetDEM simulation: (A) at time $t = 4.9$ s, (B) at time $t = 5.0$ s, and (C) at time $t = 5.0$ s when the particle sizes are expanded by one percent when calculating the contact statuses. The pink lines represent the contacts, and only the contacts of the center particle are plotted

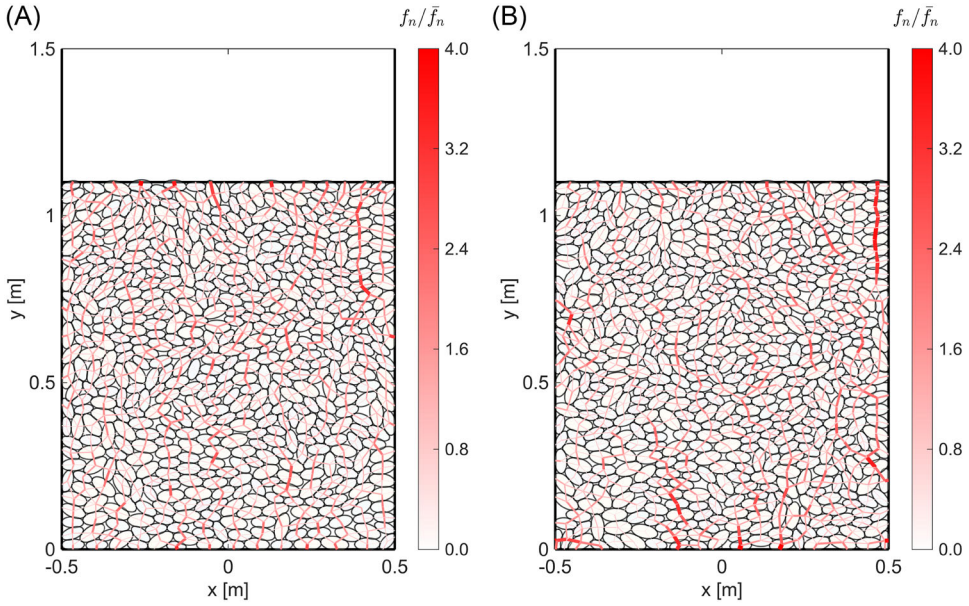


FIGURE 22 Snapshots of the oedometric compression simulation: (A) NetDEM and (B) GeoDEM. The pink lines represent the contact forces

based on the expanded particles. Figure 19C shows the distribution of the recalculated coordination number, which fairly well matches the results of GeoDEM shown in Figure 19B.

5.4 | Experiment 4: oedometric compression

The fourth experiment considers the oedometric compression test on polydispersed particles. It is noted that the purpose of the example is to demonstrate the capability of the proposed method to generate a specimen and conduct a numerical experiment on the specimen, and it is not intended to generate a specimen to meet predefined parameters or a realistic geomaterial. More importantly, this example puts an emphasis on demonstrating the capability of the proposed methods to reproduce the macroscopic mechanical behavior (e.g., stress–strain behavior) of a granular material. With the final particle assembly obtained in the previous section, a boundary wall is placed on the top of the assembly (at height $z = 1.3$ m) and is moved down at a velocity of 0.1 m/s. The particles are compressed for 2.0 s with a total compression strain of approximately 15%. The other parameters remain the same as those of the packing experiment in the previous section. After the compression, the snapshots of the particles, and the contact forces, are displayed in Figure 22.

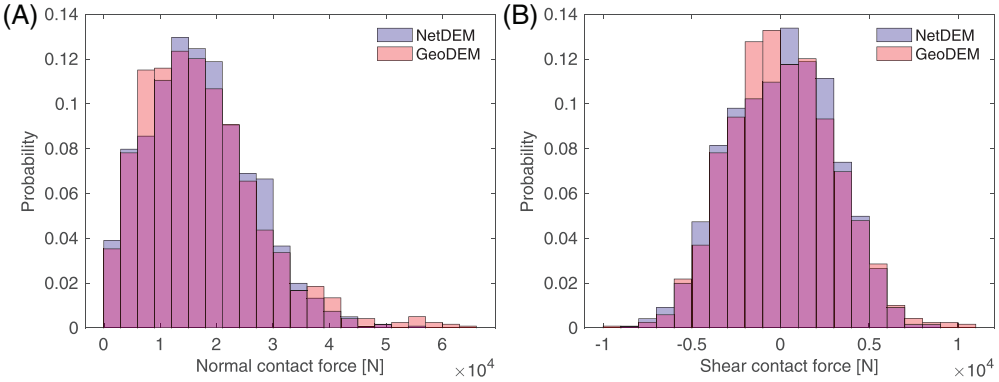


FIGURE 23 Comparison of the (A) normal contact force and (B) shear contact force distributions based on NetDEM and GeoDEM

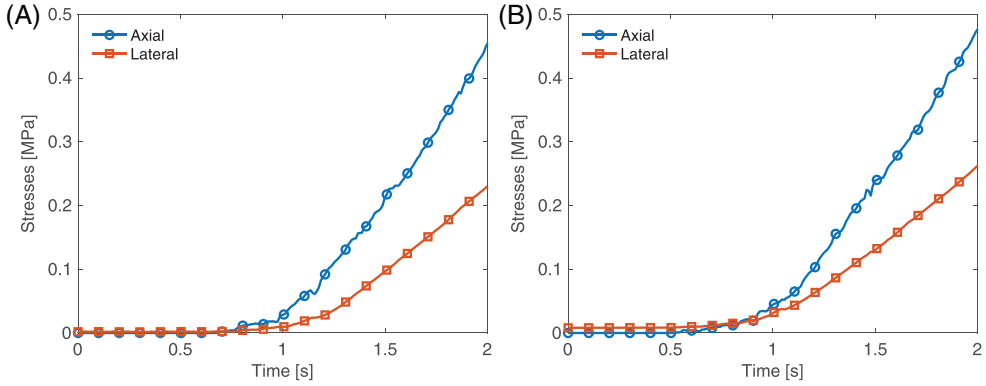


FIGURE 24 Evolution of axial and lateral stresses in the boundary walls during oedometric compression: (A) NetDEM and (B) GeoDEM

It can be observed that the spatial distribution and magnitudes (indicated by the thickness of the contact force lines) of the contact forces simulated by NetDEM and GeoDEM present a high resemblance. As a qualitative analysis, the histogram of the contact forces at all contacts is plotted in Figure 23. In both methods, the normal contact forces mainly range from 0 to 5×10^4 N, whereas the shear contact forces mainly range from 0 to 1×10^4 N, which is consistent with the specific contact friction of 0.2. The contact forces simulated by NetDEM are overall a fairly good match with those simulated by GeoDEM.

Based on the contact forces on the boundary walls, the compression stress in the top wall and the confining stress in the lateral walls can be calculated. Figure 24 shows the evolution of compression and confining stresses during the compression process. The stresses during the first 0.5 s of compression are almost zero because the particles are experiencing a great deal of rearrangement in this state, and after that, the stresses increase almost linearly with time. There is a discrepancy of approximately 20% in the results between NetDEM and the GeoDEM. The discrepancy is likely due to the randomness in the initial particle packing.

5.5 | Experiment 5: angle of repose

The fifth experiment considers the angle of repose test on polydispersed particles. With the final particle assembly obtained from the packing experiment, the right lateral wall is removed to initiate the particle angle of repose test. The parameters in this example remain the same as those of the packing experiment. The simulation lasts for 30 s, which is sufficient for the particles to reestablish a quasi-static equilibrium state. As shown in Figure 25, the average velocity of the particles almost plateaus after approximately 18 s, indicating that the particles have arrived at a quasi-static equilibrium state. Figure 26 displays the evolution of the particle assemblies with time. In this example, there is a notable discrepancy in the results of NetDEM and GeoDEM. In NetDEM, the final angle of repose is calculated to be approximately 37.5°, while in GeoDEM, the final angle of repose is approximately 20.5°. A possible explanation for the extra shear resistance in NetDEM is that the errors in the predicted contact overlapping length result in spurious surface roughness on particles. To investigate

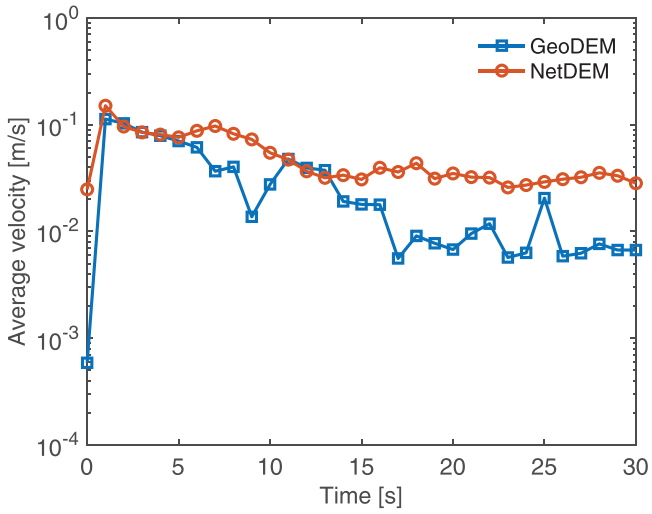


FIGURE 25 Evolution of the average velocity during the angle of repose test

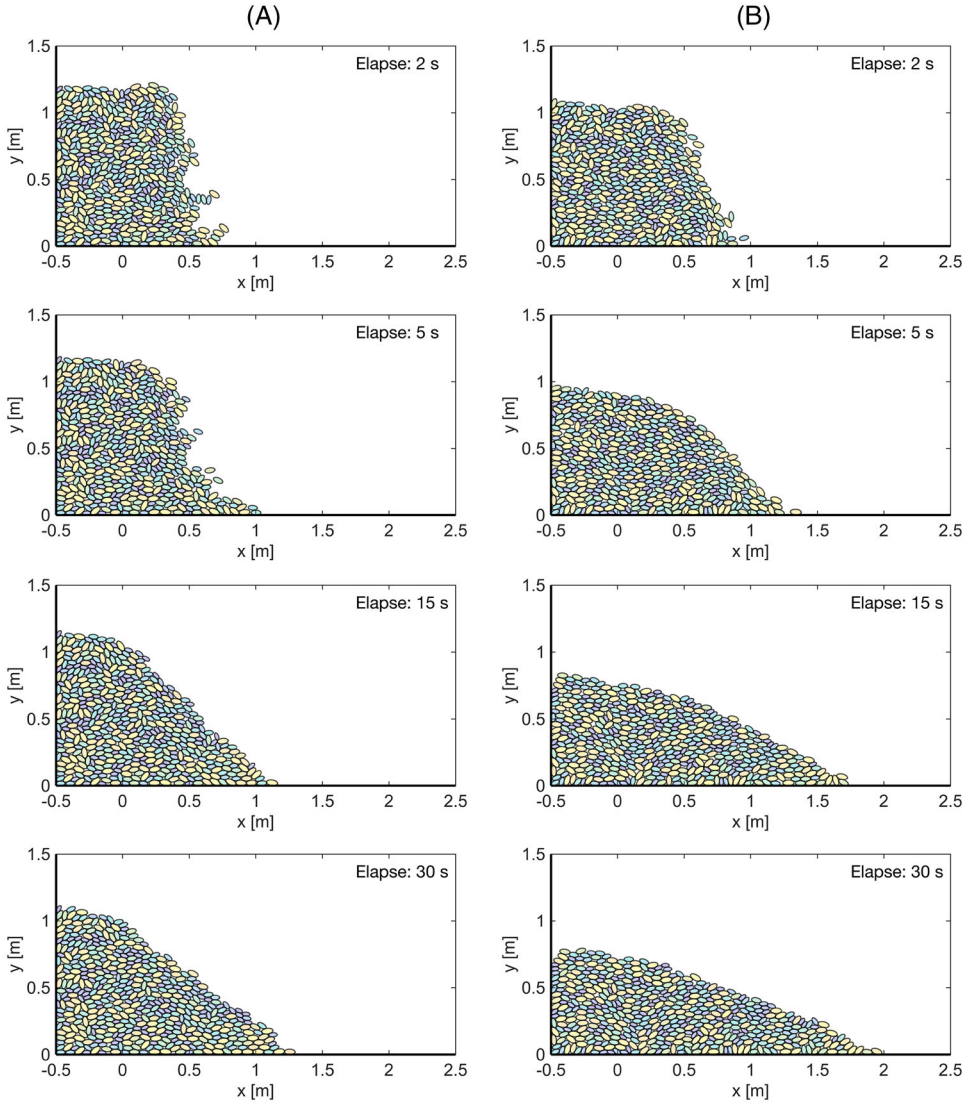


FIGURE 26 Snapshots of the angle of repose simulation: (A) NetDEM and (B) GeoDEM

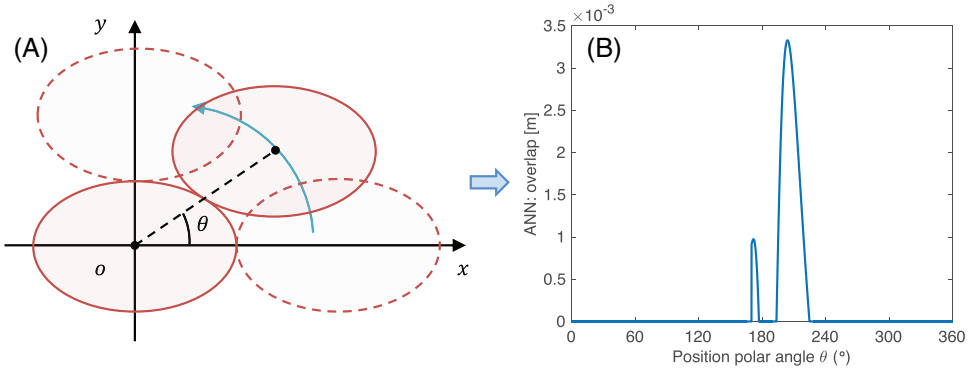


FIGURE 27 Illustration of the spurious surface roughness effects in NetDEM: (A) the experiment setup with one particle being tangent to and moving along the surface of another, (B) the results of the contact overlapping length

this aspect, a numerical test is conducted with one particle moving along the surface of another and meanwhile with the two particles being tangent with each, as sketched in Figure 27A. Analytically, the contact overlapping length of these two particles should be zero. Figure 27A shows the predicted contact overlapping length in NetDEM. Among most of the particle layouts, the contact overlapping length is zero (i.e., no contact) or next to zero, while two spikes are also observed. This phenomenon indicates that the particle may have a bump (i.e., spurious surface roughness) at the position of spikes. It is worth noting that the overlapping length profile shown in Figure 27B only corresponds to the particle layouts shown in Figure 27A. If the particles are of different rotation angles, the contact overlapping length profile (i.e., the spurious surface roughness) could be different. Mitigating the errors in the overlapping length and their effects on dynamic DEM simulations merit further exploration.

5.6 | Experiment 6: packing and compression of arbitrarily irregular-shaped particles

The last experiment considers the packing and compression tests similar to experiments 3 and 4 in the previous sections, whereas arbitrarily irregular-shaped particles are used to showcase the potentials and applicability of the proposed method for other particle shapes. An arbitrarily irregular shape template previously used in Lai and Chen[18] is taken as an example. This shape template is reconstructed from realistic soil samples based on X-ray computed tomography and image analysis. Based on this randomly selected shape template, 500 particles of the same size 0.05 m are created and then consecutively run through the packing and compression processes. Besides particle shape, all other procedures and parameters are kept the same as those used in experiments 3 and 4 in the previous sections. For GeoDEM, the Fourier series-based contact detection and resolution algorithm previously developed in Lai et al.[13] is used. For NetDEM, the design of the ANNs is kept the same as those in the elliptical particle case. The training samples are obtained based on the Fourier series-based contact detection and resolution algorithm.[13] Similar to the previous sections, the packing configuration, contact forces, and stress–strain relations are investigated, and the results are presented in Figures 28–30, respectively. Again, the NetDEM and GeoDEM show fairly consistent results, which validates the potentials and applicability of the proposed method for arbitrarily irregular-shaped particles.

6 | DISCUSSIONS

In this section, we present some further discussions about the proposed ML-enable DEM (or NetDEM). The first is a remark regarding the approaches to characterize the contact behavior. There are basically two groups of approaches to characterize the contact behavior of colliding particles, namely, the nonsmooth dynamics approach and the regularized continuous approach.[67,68] The nonsmooth dynamics approach considers particles as rigid bodies and determines the contact forces based on geometric constraints (e.g., no interparticle penetration and frictional stick–slip).[69] On the other hand, the regularized continuous approach (also referred to as the soft-particle approach) allows particles to deform at the contact zone and evaluates the contact forces as functions of indentation and compliance of the contacting surfaces.[50] The distinction in dealing with contacts further results in a difference in the time integration process in that the time

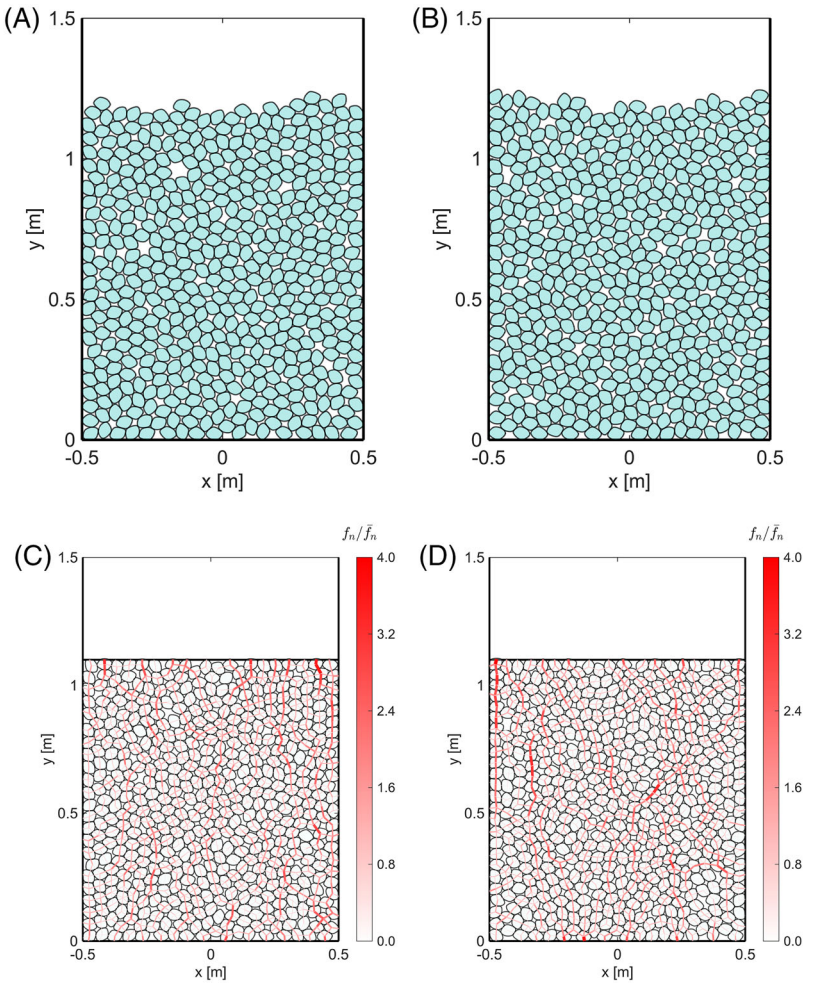


FIGURE 28 Snapshots of the packing and oedometric compression simulation with arbitrarily irregular-shaped particles: (A) the final packing in NetDEM, (B) the final packing in GeoDEM, (C) the final contact force chains in NetDEM, and (D) the final contact force chains in GeoDEM. The pink lines represent the contact forces

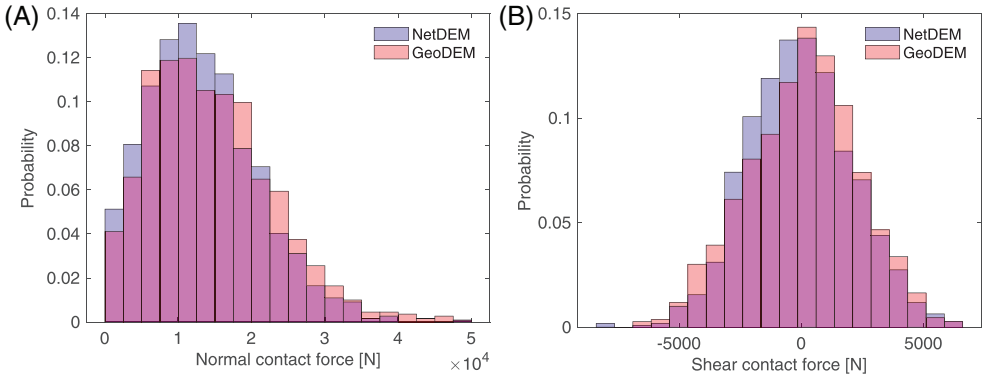


FIGURE 29 Comparison of the (A) normal contact force and (B) shear contact force distributions based on NetDEM and GeoDEM for the irregular-shaped particles case

integration in the nonsmooth dynamics approach is implicit and needs to be solved numerically, whereas the time integration in the regularized continuous approach is explicit.[68] Due to the need for an implicit solution procedure, the performance of the nonsmooth dynamics approach may deteriorate significantly in the case of dense packing. In this work, the regularized continuous approach is considered. ML models are utilized to detect contacts and resolve contact geometric features. Nonetheless, the concept of utilizing ML tools to deal with contact detection and resolution is general and could be extended to other discontinuous mechanical methods (e.g., the nonsmooth dynamics DEM or DDA) that involve interparticle contacts.[7–9,68,70]

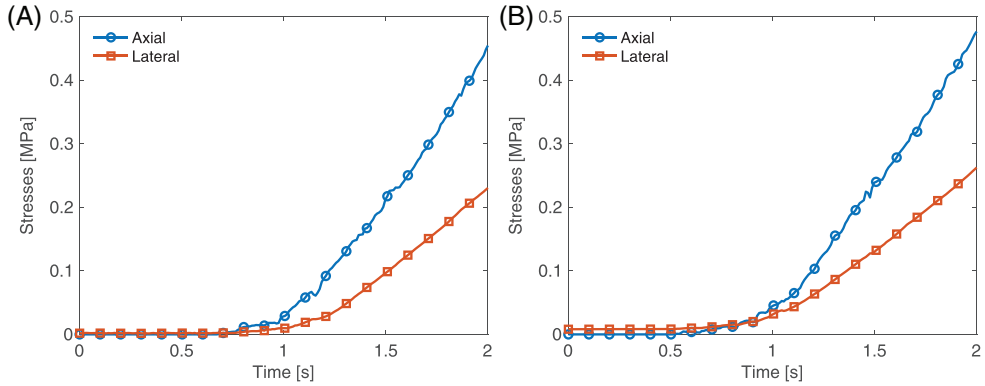


FIGURE 30 Evolution of axial and lateral stresses in the walls during oedometric compression for the irregular-shaped particles case: (A) NetDEM and (B) GeoDEM

Second, it is noted that in this work, we have adopted the geometric contact detection and resolution algorithms to prepare the training samples for the ANNs. This seems a paradox as one primary motivation of the ML-enable DEM is to avoid the tedious derivation and implementation of the geometric contact detection and resolution algorithms that are required in the conventional DEM. Herein, we would like to note that besides the geometric contact detection and resolution algorithms, there are also other algorithms available for such a task. One option is to use image analysis techniques. For example, image analysis techniques, together with photography or X-ray computed tomography, have been used to obtain realistic particle shapes[18,71,72] and compute key parameters of contact geometric features, including contact normal orientation, contact area, contact point, and so forth.[73–75] There are also attempts on the image analysis-based DEMs, such as the orientation discretization database solution[76,77] or the level set-DEM.[78,79] One main limitation with these methods is the high memory and computation demand due to the need to store and operate the large amount of particle pixels/voxels. This limitation may be overcome by integrating the image analysis techniques with ML, as the preparation of the training data is a one-time cost. With the NetDEM being trained in prior, simulations can be conducted efficiently by utilizing the ML models for contact detection and resolution. The combination of image analysis techniques and ML models is also a promising approach to develop the data-driven NetDEM, which directly links laboratory measurements with numerical simulations and thus could essentially improve the fidelity of DEM simulations. This aspect merits further exploration.

The third is about the effects of using ML for contact detection and resolution on the accuracy and stability of a DEM simulation. In NetDEM, the central difference time integration scheme is used to update particle positions and rotations. Technically, the time integration is numerically stable if the time increment being used is less than a threshold value, that is, the critical timestep.[80] For the evaluation of contact behavior, this work adopts the linear spring contact model together with the contact geometric features defined in Appendix A. This contact theory has been proved to be energy-conserving.[14] Thus, the effects of using ML for contacts on the accuracy and stability of a DEM simulation mainly resides in the errors in the predicted contact geometric features. The results of the six numerical experiments indicate that the errors in the predicted contact geometric features would not cause numerical instability issue. In addition, the errors would not exhibit a significant impact on the accuracy (in terms of packing fabric and stress–strain behavior) of a DEM simulation of quasi-static problems (e.g., random packing and compression tests). However, the errors may significantly affect the dynamics behaviors (e.g., the angle of repose test) of a particulate system.

Figure 31 illustrates the possible error sources when resolving the contact geometric features in NetDEM. The contact point influences the contact behavior mainly through the branch vector, which is required for calculating the spinning velocity of the particles and the contact moment. As investigated in Section 4, the errors in the contact geometric features overall are on the order of 10^{-3} or less. In NetDEM, particle sizes are normalized such that the larger of the two particles in contact has a unit size. Thus, the errors in the branch vector are likely on the order of 10^{-3} or less, which is fairly small. In the same manner, the errors in the contact normal are also on the order of 10^{-3} or less, as the contact normal is also of unit length. This implies that the effects of the errors in the contact point and contact normal on the contact behavior are minor. On the other hand, the effects of the errors in the contact overlapping length on the contact behavior could be significant depending on the magnitude of the external loading. Figure 32 shows the relative errors in the overlapping length for all testing samples. For the particle–particle contact case, the relative error is approximately 10^{-2} for an overlapping length greater than 2%, while the relative error could be tens or hundreds if the overlapping length is close to zero

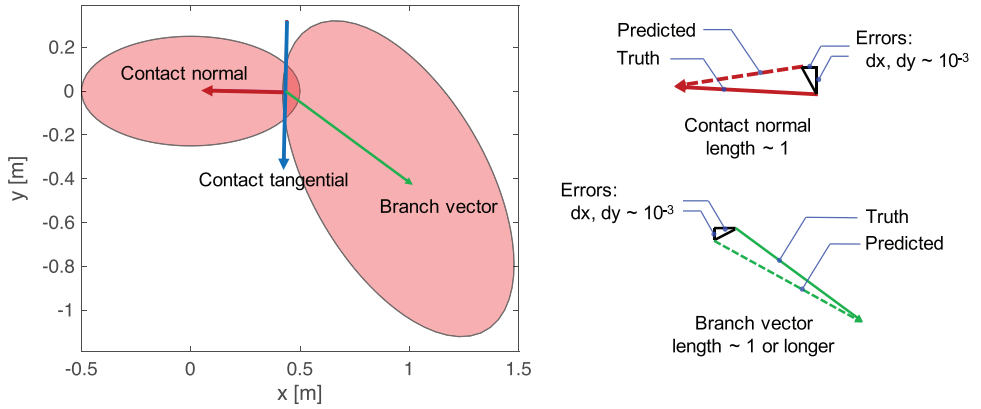


FIGURE 31 Illustration of the errors in the predicted contact geometric features

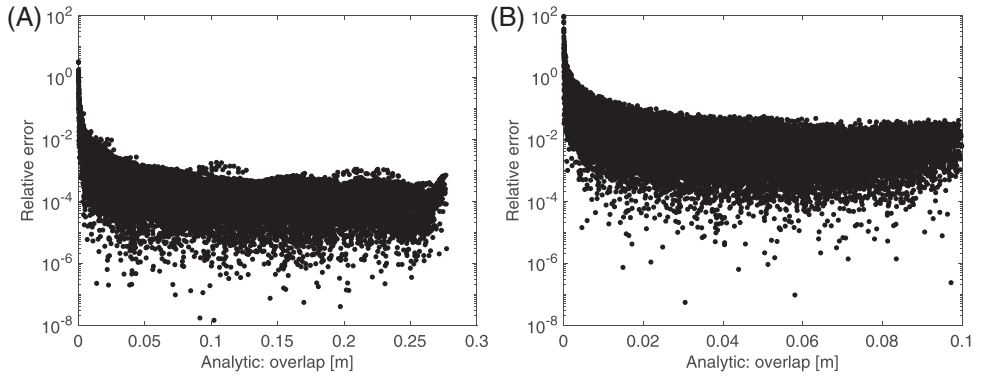


FIGURE 32 Relative errors in the contact overlapping length: (A) particle–boundary case and (B) particle–particle case

(e.g., 0.1% overlapping length). This implies that for particulate systems with high external loading, such as in the oedometric compression test, the simulation results based on NetDEM could be fairly accurate. However, for particulate systems with low external loading or dynamic responses, such as in the packing test and the angle of repose test, there could exist spurious phenomena (e.g., particle oscillation or surface roughness effects) in the simulation results based on NetDEM. Mitigating the errors in the overlapping length for small overlaps remains a big challenge and merits further exploration.

The last is about computational efficiency. The time costs of NetDEM and GeoDEM for the previous experiments are compared in Table 2. Overall, NetDEM is much more efficient than GeoDEM. For elliptical particles, such as the oedometric compression test, the time cost of NetDEM could be 25% less than that of GeoDEM. The efficiency gain could be much higher for irregular-shaped particles or 3D DEM simulations. However, it should be noted that in this work, only one type of shape is considered in a DEM simulation, and thus fairly shallow ANNs are used. If more shapes are involved, deeper

TABLE 2 Experiment specifications and computational time cost (run with single Intel i5 core and MATLAB)

Experiment	No. of particles	No. of boundaries	Physical time [s]	Computing time [s]	
				NetDEM	GeoDEM
Ellipse: simple pendulum (a)	1	2	2.0	86	88
Ellipse: simple pendulum (b)	2	1	2.0	67	68
Ellipse: rockfall	1	3	2.0	44	44
Ellipse: packing	500	4	5.0	25,020	41,760
Ellipse: compression	500	4	2.0	18,120	24,240
Ellipse: angle of repose	500	2	30.0	247,860	340,560
Irregular: packing	500	4	5.0	19,469	68,534
Irregular: compression	500	4	2.0	15,179	43,444

neural networks would be required to achieve an acceptable tolerance. This would increase the computational expenses of NetDEM, but could be mitigated with parallel computing techniques for the neural networks.

7 | SUMMARY

We present an ML-enabled framework that employs neural networks to detect and resolve particle contacts in DEM. Two types of neural networks are involved, namely, a classification network for contact detection and a regression network for contact resolution. The neural networks take the particle geometric descriptors as inputs and output the contact flag and the contact geometric features. In particular, it is proposed to characterize the particle geometric description with size, shape, and position descriptors. An approach to regularize the ranges of size, shape, and position descriptors for the ANNs is also presented. With the definitions of inputs and outputs, the procedure for generating the training and testing database of the neural network models is developed.

Taking elliptical particles as an example, the classification and regression networks are designed to have three hidden layers, and each layer has a dimension of 15. With the networks properly trained with sufficient samples, the networks could correctly predict the contact status of more than 99% of the various contact cases. The errors in contact geometric features are on the order of 10^{-4} or less for particle–boundary contact cases, and are on the order of 10^{-3} or less for particle–particle contact cases. The results of five numerical experiments have shown that the ML-enabled DEM can accurately capture the trajectory and energy evolution of individual particles, the fabric characteristics of dense packing, and the mechanical behavior of packing under large loads. Another example involving irregular-shaped particles has also been presented to showcase the potentials and applicability of the proposed method for other particle shapes. Additionally, we have discussed the possible sources of errors in the ML-enabled DEM. It is shown that the error issue mainly lies in the predicted contact overlapping length. Although the absolute errors in the predicted contact overlapping length are small in magnitude, they could have a significant effect on the contact behavior when the contact overlapping length is very small, as in the case of systems under small loads. Future studies will focus on improving the accuracy of the method for such systems. Finally, it should be noted that by using neural networks to detect and resolve contacts, the ML-enabled DEM framework is general, and the workflow and implementation are applicable to any arbitrarily irregular-shaped particles.

ACKNOWLEDGMENTS

The first author (Lai) would like to acknowledge the financial support provided by Hong Kong Scholars Program 2020, the Fundamental Research Funds for the Central Universities (19lgpy289), China Postdoctoral Science Foundation (2019M663240), the National Natural Science Foundation of China (51909289), and the Research Grants Council of Hong Kong (16207319). The third author (Huang) would like to acknowledge the financial support provided by Shenzhen Natural Science Foundation (JCYJ20190807162401662) and the National Natural Science Foundation of China (51978677).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Qiushi Chen https://orcid.org/0000-0002-0394-6710

REFERENCES

1. Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *Géotechnique*. 1979;29:47–65.
2. Lisjak A, Grasselli G. A review of discrete modeling techniques for fracturing processes in discontinuous rock masses. *J Rock Mech Geotech Eng*. 2014;6:301–314.
3. Lai Z, Chen Q. Characterization and discrete element simulation of grading and shape-dependent behavior of JSC-1A Martian regolith simulant. *Granul Matter*. 2017;19(4):69.
4. Thakur MM, Penumadu D. Triaxial compression in sands using FDEM and micro-X-ray computed tomography. *Comput Geotech*. 2020;124:103638.
5. Yin Z, Wang P, Zhang F. Effect of particle shape on the progressive failure of shield tunnel face in granular soils by coupled FDM-DEM method. *Tunn Undergr Space Technol*. 2020;100:103394.
6. Liu Z, Su L, Zhang C, Iqbal J, Hu B, Dong Z. Investigation of the dynamic process of the Xinmo landslide using the discrete element method. *Comput Geotech*. 2020;123:103561.

7. Shi G. Discontinuous deformation analysis: a new numerical model for the statics and dynamics of deformable block structures. *Eng Comput*. 1992;9:157–168.
8. Khan MS. *Investigation of Discontinuous Deformation Analysis for Application in Jointed Rock Masses* [PhD thesis]. University of Toronto; 2010.
9. Zheng F, Leung YF, Zhu J, Jiao Y. Modified predictor-corrector solution approach for efficient discontinuous deformation analysis of jointed rock masses. *Int J Numer Anal Methods Geomech*. 2019;43:599–624.
10. Guo N, Zhao J. Parallel hierarchical multiscale modelling of hydro-mechanical problems for saturated granular soils. *Comput Meth Appl Mech Eng*. 2016;305:37–61.
11. Zhao S, Zhao J, Lai Y. Multiscale modeling of thermo-mechanical responses of granular materials: a hierarchical continuum–discrete coupling approach. *Comput Meth Appl Mech Eng*. 2020;367:113100.
12. Andrade JE, Lim KW, Avila CF, Vlahinić I. Granular element method for computational particle mechanics. *Comput Meth Appl Mech Eng*. 2012;241:262–274.
13. Lai Z, Chen Q, Huang L. Fourier series-based discrete element method for computational mechanics of irregular-shaped particles. *Comput Meth Appl Mech Eng*. 2020;362:112873.
14. Feng YT. An energy-conserving contact theory for discrete element modelling of arbitrarily shaped particles: basic framework and general contact model. *Comput Meth Appl Mech Eng*. 2021;373:113454.
15. Latham JP, Munjiza A, Garcia X, Xiang J, Guises R. Three-dimensional particle shape acquisition and use of shape library for DEM and FEM/DEM simulation. *Miner Eng*. 2008;21:797–805.
16. Cox MR, Budhu M. A practical approach to grain shape quantification. *Eng Geol*. 2008;96:1–16.
17. Bagheri GH, Bonadonna C, Manzella I, Vonlanthen P. On the characterization of size and shape of irregular particles. *Powder Technol*. 2015;270:141–153.
18. Lai Z, Chen Q. Reconstructing granular particles from X-ray computed tomography using the TWS machine learning tool and the level set method. *Acta Geotech*. 2019;14:1–18.
19. Zhao S, Zhang N, Zhou X, Zhang L. Particle shape effects on fabric of granular random packing. *Powder Technol*. 2017;310:175–186.
20. Zhao S, Zhou X. Effects of particle asphericity on the macro-and micro-mechanical behaviors of granular assemblies. *Granul Matter*. 2017;19:38.
21. Lin X, Ng TT. A three-dimensional discrete element model using arrays of ellipsoids. *Géotechnique*. 1997;47:319–329.
22. Munjiza A, Peters JF, Hopkins MA, Kala R, Wahl RE. A poly-ellipsoid particle for non-spherical discrete element method. *Eng Comput*. 2009;26:645–657.
23. Zhao S, Zhao J. A poly-superellipsoid-based approach on particle morphology for DEM modeling of granular media. *Int J Numer Anal Methods Geomech*. 2019;43:2147–2169.
24. Cleary PW, Sawley ML. DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge. *Appl Math Modell*. 2002;26:89–111.
25. Podlozhnyuk A, Pirker S, Kloss C. Efficient implementation of superquadric particles in Discrete Element Method within an open-source framework. *Comput Particle Mech*. 2017;4:101–118.
26. Feng YT, Han K, Owen DRJ. A generic contact detection framework for cylindrical particles in discrete element modelling. *Comput Meth Appl Mech Eng*. 2017;315:632–651.
27. Zhao S, Zhou X, Liu W, Lai C. Random packing of tetrahedral particles using the polyhedral discrete element method. *Particuology*. 2015;23:109–117.
28. Lim KW, Andrade JE. Granular element method for three-dimensional discrete element calculations. *Int J Numer Anal Methods Geomech*. 2014;38:167–188.
29. Lai Z, Huang L. A polybézier-based particle model for the DEM modeling of granular media. *Comput Geotech*. 2021;134:104052.
30. Williams JR, O'Connor R. Discrete element simulation and the contact problem. *Arch Comput Meth Eng*. 1999;6:279–304.
31. Wachs A, Girolami L, Vinay G, Ferrer G. Grains3D, a flexible DEM approach for particles of arbitrary convex shape—part I: numerical model and validations. *Powder Technol*. 2012;224:374–389.
32. Rakotonirina AD, Delenne JY, Radjai F, Wachs A. Grains3D, a flexible DEM approach for particles of arbitrary convex shape—part III: extension to non-convex particles modelled as glued convex particles. *Comput Particle Mech*. 2019;6:55–84.
33. Gkeka P, Stoltz G, Farimani AB, et al. Machine Learning Force Fields and Coarse-Grained Variables in Molecular Dynamics: Application to Materials and Biological Systems. *Journal of Chemical Theory and Computation*. 2020;16(8):4757–4775. https://doi.org/10.1021/acs.jctc.0c00355
34. Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. *Annu Rev Fluid Mech*. 2020;52:477–508.
35. Wang K, Sun W. A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. *Comput Methods Appl Mech Eng*. 2018;334:337–380.
36. Capuano G, Rimoli JJ. Smart finite elements: a novel machine learning application. *Comput Methods Appl Mech Eng*. 2019;345:363–381.
37. Wang K, Sun W. Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning. *Comput Methods Appl Mech Eng*. 2019;346:216–241.
38. Oishi A, Yagawa G. Computational mechanics enhanced by deep learning. *Comput Methods Appl Mech Eng*. 2017;327:327–351.

39. Fragkiadaki K, Agrawal P, Levine S & Malik J. Learning visual predictive models of physics for playing billiards. In: The 6th International Conference on Learning Representations (ICLR). 2016.
40. Wu J, Yildirim I, Lim JJ, Freeman B, Tenenbaum J. Galileo: perceiving physical object properties by integrating a physics engine with deep learning. In: *Advances in Neural Information Processing Systems*. 2015:127–135.
41. Battaglia P, Pascanu R, Lai M, Rezende D, Kavukcuoglu K. Interaction networks for learning about objects, relations and physics. In: *Advances in Neural Information Processing Systems*. 2016:4502–4510.
42. Chang MB, Ullman T, Torralba A & Tenenbaum JB. A compositional object-based approach to learning physical dynamics. In: 5th International Conference on Learning Representations (ICLR). 2017.
43. Lerer A, Gross S & Fergus R. Learning physical intuition of block towers by example. In: *Proceedings of the 33rd International Conference on Machine Learning, PMLR*, 2016:430–438.
44. Li W, Azimi S, Leonardis A & Fritz M. To fall or not to fall: A visual approach to physical stability prediction. arXiv preprint arXiv:1604.00066 2016.
45. Mottaghi R, Bagherinezhad H, Rastegari M, Farhadi A. Newtonian scene understanding: unfolding the dynamics of objects in static images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016:3521–3529.
46. Wu J, Lu E, Kohli P, Freeman B, Tenenbaum J. Learning to see physics via visual de-animation. In: *Advances in Neural Information Processing Systems*. 2017:153–164.
47. Mrowca D, Zhuang C, Wang E, et al. Flexible neural representation for physics prediction. In: *Advances in Neural Information Processing Systems*. 2018:8799–8810.
48. Hertz H. Ueber die Beruhrung fester elastischer Korper (On the contact of elastic solids). *J Reine Angew Math*. 1882;92:156–171.
49. Mindlin RD. Compliance of elastic bodies in contact. *J Appl Mech ASME*. 1949;16:259–268.
50. Johnson KL. *Contact Mechanics*. UK: Cambridge University Press; 1987.
51. Lai Z, Chen L. A semianalytical Hertzian frictional contact model in 2D. *Appl Math Modell*. 2021;92:546–564.
52. Potyondy DO, Cundall PA. A bonded-particle model for rock. *Int J Rock Mech Min Sci*. 2004;41:1329–1364.
53. Potyondy DO. The bonded-particle model as a tool for rock mechanics research and application: current trends and future directions. *Geosystem Eng*. 2015;18:1–28.
54. Richefeu V, El Youssoufi MS, Peyroux R, Radjai F. A model of capillary cohesion for numerical simulations of 3D polydisperse granular media. *Int J Numer Anal Methods Geomech*. 2008;32:1365–1383.
55. Luding S. Cohesive, frictional powders: contact models for tension. *Granul Matter*. 2008;10:235.
56. Belytschko T, Yen H, Mullen R. Mixed methods for time integration. *Comput Methods Appl Mech Eng*. 1979;17:259–275.
57. Martys NS, Mountain RD. Velocity Verlet algorithm for dissipative-particle-dynamics-based models of suspensions. *Phys Rev E*. 1999;59:3733.
58. O'Sullivan C, Bray JD. Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme. *Eng Comput*. 2004;21:278–303.
59. White H. *Artificial Neural Networks: Approximation and Learning Theory*. USA: Blackwell Publishers, Inc.; 1992.
60. Hassoun MH. *Fundamentals of Artificial Neural Networks*. USA: MIT Press; 1995.
61. Mollon G, Zhao J. 3D generation of realistic granular samples based on random fields theory and Fourier shape descriptors. *Comput Methods Appl Mech Eng*. 2014;279:46–65.
62. Zhou B, Wang J. Random generation of natural sand assembly using micro X-ray tomography and spherical harmonics. *Géotech Lett*. 2015;5:6–11.
63. Zheng J, Hryciw RD. An image based clump library for DEM simulations. *Granul Matter*. 2017;19:26.
64. Sun Q, Zheng J. Realistic soil particle generation based on limited morphological information by probability-based spherical harmonics. *Comput Particle Mech*. 2020;8:1–21.
65. Mollon G, Zhao J. Fourier–Voronoi-based generation of realistic samples for discrete modelling of granular materials. *Granul Matter*. 2012;14:621–638.
66. Huang L, Huang S, Lai Z. On the optimization of site investigation programs using centroidal Voronoi tessellation and random field theory. *Comput Geotech*. 2020;118:103331.
67. Machado M, Moreira P, Flores P, Lankarani HM. Compliant contact force models in multibody dynamics: evolution of the Hertz contact theory. *Mech Mach Theory*. 2012;53:99–121.
68. Lim KW, Krabbenhoft K, Andrade JE. A contact dynamics approach to the granular element method. *Comput Methods Appl Mech Eng*. 2014;268:557–573.
69. Wriggers P, Zavarise G. Computational contact mechanics. In: *Encyclopedia of Computational Mechanics*. USA: John Wiley & Sons Ltd.; 2004.
70. Zheng F, Zhuang X, Zheng H, Jiao Y, Rabczuk T. A robust potential-based contact force solution approach for discontinuous deformation analysis of irregular convex polygonal block/particle systems. *Acta Geotech*. 2021;16:679–697.
71. Xu W, Liu G, Yang H. Study on the mechanical behavior of sands using 3D discrete element method with realistic particle models. *Acta Geotech*. 2020;15:2813–2828.
72. Zheng J, He H, Alimohammadi H. Three-dimensional Wadell roundness for particle angularity characterization of granular soils. *Acta Geotech*. 2021;16:133–149.

73. Sun Q, Zheng J, He H, Li Z. Particulate material fabric characterization from volumetric images by computational geometry. *Powder Technol*. 2019;344:804–813.
74. Dadda A, Geindreau C, Emeriault F, Du RSR, Filet AE, Garandet A. Characterization of contact properties in biocemented sand using 3D X-ray micro-tomography. *Acta Geotech*. 2019;14:597–613.
75. Wiebicke M, Andò E, Viggiani G, Herle I. Measuring the evolution of contact fabric in shear bands with X-ray tomography. *Acta Geotech*. 2020;15:79–93.
76. Džiugys A, Peters B. An approach to simulate the motion of spherical and non-spherical fuel particles in combustion chambers. *Granul Matter*. 2001;3:231–266.
77. Dong K, Wang C, Yu A. A novel method based on orientation discretization for discrete element modeling of non-spherical particles. *Chem Eng Sci*. 2015;126:500–516.
78. Kawamoto R, Andò E, Viggiani JE. Level set discrete element method for three-dimensional computations with triaxial case study. *J Mech Phys Solids*. 2016;91:1–13.
79. Kawamoto R, Andò E, Viggiani G, Andrade JE. All you need is shape: predicting shear banding in sand with LS-DEM. *J Mech Phys Solids*. 2018;111:375–392.
80. Otsubo M, O'Sullivan C, Shire T. Empirical assessment of the critical time increment in explicit particulate discrete element method simulations. *Comput Geotech*. 2017;86:67–79.

How to cite this article: Lai Z, Chen Q, Huang L. Machine-learning-enabled discrete element method: Contact detection and resolution of irregular-shaped particles. *Int J Numer Anal Methods*. 2021;1–28. https://doi.org/10.1002/nag.3293

APPENDIX A: CONTACT GEOMETRIC FEATURES OF LINEAR CONTACT MODEL

In this section, the definition of the contact geometric features of the spring-based linear contact model is presented. These contact geometric features will be taken as the outputs of the ML models in the contact detection and resolution problem.

A.1 | Particle–particle case

In the particle–particle contact case, the contact geometric features, as illustrated in Figure A.1, are defined as follows.

- Tangential direction $\boldsymbol{t}$: the direction of vector $\overrightarrow{I_1I_2}$, where points I_1 and I_2 are the intersections of the particle surfaces.
- Normal direction $\boldsymbol{n}$: the tangential direction rotated clockwise by 90°. For ease of implementation, we take the convention that the contact normal should be directed to the object particle, that is, particle O_1. In the case that the contact normal is opposite to this convention, both the obtained contact normal and tangential directions are reversed.
- Branch vectors $\boldsymbol{b}$: namely, vectors MO_1 and MO_2; the vectors going from the contact point M to the particle centroids O_1 and O_2, respectively, where the contact point M is defined as the midpoint of surface intersections I_1 and I_2.
- Normal overlapping length δ_n: the length of line segment N_1N_2, where points N_1 and N_2 are the intersections of the line that passes through the contact points parallel to the normal direction with the particle surfaces. It should be noted that the line would intersect a particle at two points, where the point inside the other particle should be used.

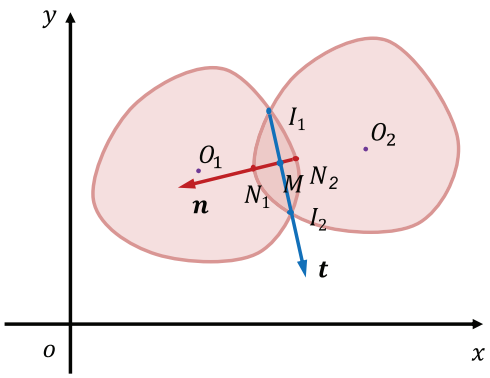


FIGURE A.1 Definition of the contact geometric features for the particle–particle case

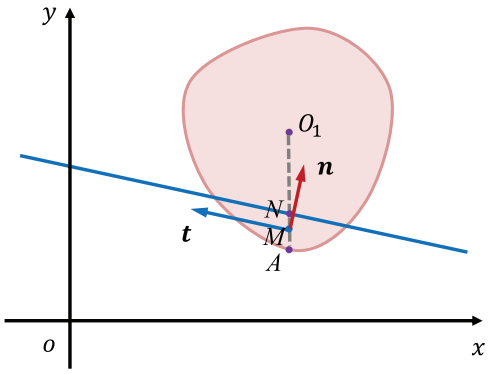


FIGURE A.2 Definition of contact geometric features for the particle–boundary case

- Increment of tangential overlapping $\Delta\delta_t$: calculated as $\Delta\delta_t = v_t \Delta t$, where v_t is the relative tangential velocity of one particle with respect to the other particle, and Δt is the current time increment. The relative tangential velocity v_t is calculated as $v_t = (\boldsymbol{v}^p - \boldsymbol{v}^q) \cdot \boldsymbol{t} - (\omega^p l_n^p + \omega^q l_n^q)$, where $\boldsymbol{v}$ and ω are the translational velocity and rotational velocity of particles, respectively, $l_n = |\boldsymbol{b} \cdot \boldsymbol{n}|$ is the length of the branch vector projected onto the contact normal direction, and $\boldsymbol{t}$ is the contact tangential direction. The superscripts $[\cdot]^p$ and $[\cdot]^q$ in the equation indicate the variables belonging to particles p and q, respectively. The subscripts $[\cdot]_n$ and $[\cdot]_t$ indicate the normal and tangential components of the variable, respectively.

A.2 | Particle–boundary case

In the particle–boundary contact case, the contact geometric features, as illustrated in Figure A.2, are defined as follows.

- Tangential direction $\boldsymbol{t}$: the direction parallel to the boundary line.
- Normal direction $\boldsymbol{n}$: the tangential direction rotated clockwise by 90°. Similar to the particle–particle contact case, we take the convention that the contact normal should be directed to the object particle, that is, particle O_1. In the case that the contact normal is opposite to this convention, both the obtained contact normal and tangential directions are reversed.
- Branch vector $\boldsymbol{b}$: namely, vector MO_1; the vector going from the contact point M to the particle centroid O_1. Herein, the contact point M is defined as the midpoint of points A and N, where point A is the point on the particle surface that has the furthest distance to the boundary line, and point N is the intersection of line AO_1 and the boundary line.
- Normal overlapping length δ_n: the distance from point A to the boundary line.
- Increment of tangential overlapping $\Delta\delta_t$: also calculated as $\Delta\delta_t = v_t \Delta t$, where v_t is the relative tangential velocity of the particle with respect to the boundary. The relative tangential velocity v_t in the particle–boundary contact case is calculated as $v_t = (\boldsymbol{v}^p - \boldsymbol{v}^b) \cdot \boldsymbol{t} - (\omega^p l_n^p + \omega^b l_n^b)$, where $\boldsymbol{v}$ and ω are the translational and rotational velocities, respectively; l_n^b is the distance from the contact point to the boundary rotational center projected onto the contact normal direction; l_n^p is the length of the branch vector projected onto the contact normal direction; and $\boldsymbol{t}$ is the contact tangential direction. The superscripts $[\cdot]^p$ and $[\cdot]^b$ indicate the variables belonging to the particle and the boundary, respectively.