Discrete Element Methods: Basics and Applications in Engineering

Peter Wriggers and B. Avci

Introduction

Particle systems are of great importance in many industrial branches like in chemical and food industries as well as in geotechnical engineering problems. Coupling of particles with fluids are related to fluvial erosion, fluidized beds, sedimentation and transport in the blood system. Thus, the numerical simulation of particle systems is of great interest, both from practical and fundamental points of view. Therefore, the understanding, the simulation and analysis of related phenomena is significant, particularly with regard to micro movements, homogenization procedures and coupled moderate or highly concentrated particulate flows. Certainly, such problems require an accurate description of the underlying physics, but the simulation of particulate flow and movement is still a challenging task for a large number of particles.

Popular examples of pure particle methods are Molecular Dynamics (MD), see Alder and Wainwright (1957), Discrete Element Method (DEM), see Cundall and Strack (1979), and Smoothed Particle Hydrodynamics (SPH), see Gingold and Monaghan (1977). In these methods, the positions of the particles and the evolution of their quantities are described by ordinary differential equations and solved in a Lagrangian way. In the framework of DEM and SPH, the particle dynamics are obtained by applying the Newton-Euler equations and the Navier-Stokes equations, respectively. Both of these numerical techniques (DEM and SPH) utilize many common algorithms, such as neighbor search algorithms, distance compu-

P. Wriggers (✉)
Faculty of Mechanical Engineering, Institut for Continuum Mechanics,
Leibniz University Hannover, Hannover, Germany
e-mail: wriggers@ikm.uni-hannover.de

B. Avci
CADFEM GmbH, Hannover, Germany

L. De Lorenzis and A. Düster (eds.), *Modeling in Engineering Using Innovative Numerical Methods for Solids and Fluids*, CISM International Centre for Mechanical Sciences 599,
https://doi.org/10.1007/978-3-030-37518-8_1

tations among neighboring particles and force or kernel evaluations. Thus, many subroutines can be implemented, used and maintained in a single framework.

In the recent years, effort has been made to improve the existing methods and to develop new efficient numerical approaches. Among the various methods evolved so far, e.g., see Zhu et al. (2007), Zohdi (2007), Pöschel and Schwager (2005) for an overview, the approaches for the treatment of particle systems through direct numerical simulation models are of high computationally cost. Basically, direct numerical simulations can be performed by different discretization methods.

Concerning the treatment of the particle collision, the methods employed in the DEM can be classified into two main classes, which are characterized by hard particle contact (Alder and Wainwright 1957) and soft particle contact (Cundall and Strack 1979). Methods assigned to the first of the two groups are instantaneous collision models. Here, the particles separate immediately from each other in the event of collision. They undergo no deformation, so they are considered to be rigid. In the other approach the particles are treated as quasi rigid objects such that they are assumed to suffer minute deformations during the collision. The force based methods of the second group can be applied to govern the contact forces implying the particles' strength and eventually also allow to locally break a particle if very strong forces act on it. The difference of the methods for the treatment of particle contact is particularly crucial in highly concentrated systems. Here, a particle will usually collided with more than one partner at the same time. Hence some particles might be as well in a permanent contact situation with neighboring particles like it occurs in a heap of sediments or in case of agglomerated adhesive particles. In these cases, the application of hard contact models may not be suitable. In contrast, force based soft contact models are applicable both for dense and dilute systems. However, the computation of particle interactions is for soft spheres much more expensive compared to the hard particle approach since very small time steps are needed to resolve the contact interaction between the particles in time.

Other applications—like sediment transport or multiscale computation for granular materials—are related to coupling discrete elements to solids and fluids. In case of a direct numerical simulation of 3D particulate flows one has to couple fluids and particles. This can be done in different ways that span the bridge from just tracing of particles to a full interaction of particle and fluid via the tractions. The latter can be based on a complex ALE finite element scheme using a adaptive remesching of the finite elements to follow the particle movement, see e.g. Johnson and Tezduyar (1997), but is—due to the high computational effort—often limited to a small amount of particles. Another approach is the fictitious domain method which can handle much more particles in the flow. In both approaches the numerical tools of DEM and FEM are appropriately coupled by a staggered scheme. To describe the collision between particles, the soft contact approach is applied using repulsive force models. For more details see e.g. Avci and Wriggers (2012).

When coupling finite elements for solids and discrete elements there are two different possibilities. The first is a surface coupling where contact between a finite and a discrete element takes place. This can be handled by standard contact algorithms, see e.g. Wriggers (2006). On the other hand it is possible to couple finite and dis-

crete elements within a volume. Such coupling requires specific treatment of linking the movement of the finite and the discrete elements. On possibility is the Arlequin methodology, see e.g. Dhia and Rateau (2005), which was applied with respect to particle and finite element coupling in Wellmann and Wriggers (2012).

Governing Equations

The motion of a quasi rigid particle $\mathcal{P}_i$ is described by a position vector $\mathbf{X}_i$ to its center of mass and a rotation Ψ_i at time $t = t_0$. In Fig. 1 the kinematics of the movement of the particle $\mathcal{P}_i$ is depicted for different time instants. Additionally another particle $\mathcal{P}_j$ is depicted that collides with particle $\mathcal{P}_i$.

The trajectory of particle $\mathcal{P}_i$ can be deduced from the Newton-Euler equations. Consequently its translational and angular velocities, $\mathbf{u}_i = \dot{\mathbf{x}}_i$ and $\boldsymbol{\omega}_i = \frac{\mathrm{d}\Psi_i}{\mathrm{d}t}$, have to satisfy

$$M_i \frac{\mathrm{d}^2 \mathbf{x}_i}{\mathrm{d}t^2} = \rho_i \, V_i \mathbf{g} + \mathbf{F}_i \tag{1}$$

$$\Theta_i \frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} + \boldsymbol{\omega}_i \times (\Theta_i \boldsymbol{\omega}_i) = \mathbf{T}_i \, . \tag{2}$$

Therein, M_i is the mass, $\mathbf{x}_i$ the position vector at time t to the center of mass (defined as $\mathcal{M}_i$), ρ_i the mass density, $\mathbf{g}$ the gravity and V_i denotes the volume of the particle $\mathcal{P}_i$. The tensor of inertia is represented by Θ_i. Furthermore, the sum of the contact forces is stated as $\mathbf{F}_i$ The torques that are caused by $\mathbf{F}_i$ with respect to $\mathcal{M}_i$ are associated to the quantities $\mathbf{T}_i$. The traction vector $\mathbf{t}$ on $\partial\boldsymbol{\Omega}_p$ is defined by $\mathbf{t} = \sigma \mathbf{n}_f$ where $\mathbf{n}_f$ is the unit outward normal vector and $\mathbf{r}_i$ is the position vector of a point at $\partial\boldsymbol{\Omega}_p$ with respect to $\mathcal{M}_i$.

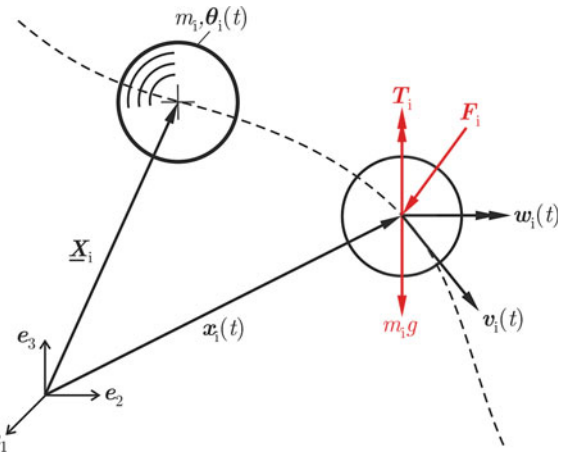


Fig. 1 Kinematics and applied forces related to a particles

Constitutive Modeling of the Particle Phase

The particles are modeled as quasi rigid spheres. To describe their collision behavior, a force based approach is used in order to govern the inter-particle forces that are deduced from repulsive models. In the sections below, the relevant concepts of the contact models are stated.

Normal Contact Model

The normal contact force acting between the colliding particles is described by a constitutive viscoelastic model. For adhesive particles being in contact, the attractive van der Waals force is additionally considered in the contact area. For the purpose of governing the elastic contact force F_e^n, the Hertzian law (Hertz 1882) constitutes a well-established model. If the particles, to be treated, have also viscous material properties a consistent phenomenological model has to be employed, see e.g., Refs. (Brilliantov et al. 1996, 2007), where the effect of viscosity is considered via an added dissipative force F_d^n. Regarding the presence of the attractive van der Waals force in the contact area, the JKR theory, see Johnson et al. (1971), provides a proper treatment of adhesion, even in the case of underwater adhesion, see e.g. Loskofsky et al. (2006). For a detailed description of the JKR model see also Maugis (1992).

If adhesion is considered, the attractive force F_a^n acts against the elastic force F_e^n so that it consequently reduces the particles' compression. Thus, one obtains for the force acting on a particle

$$F^n = F_e^n - F_a^n + F_d^n \,. \tag{3}$$

Hertz Law

When using the Hertzian contact law Hertz (1882), the elastic repulsive force is governed by

$$F_e^n = \frac{4}{3} E \sqrt{R}\, \delta^{3/2} \,, \tag{4}$$

where $\delta = \delta_i + \delta_j$ is the total particle compression which is also called the approach of the particles. The values

$$R = \left(\frac{1}{R_i} + \frac{1}{R_j} \right)^{-1} \quad \text{and} \quad E = \left(\frac{1 - \nu_i^2}{E_i} + \frac{1 - \nu_j^2}{E_j} \right)^{-1} \tag{5}$$

denote the effective radius and the effective Young's modulus of the contact pair $\mathcal{P}_i$ and $\mathcal{P}_j$, respectively. The Poisson ratio is associated with the particles as ν_i and ν_j.

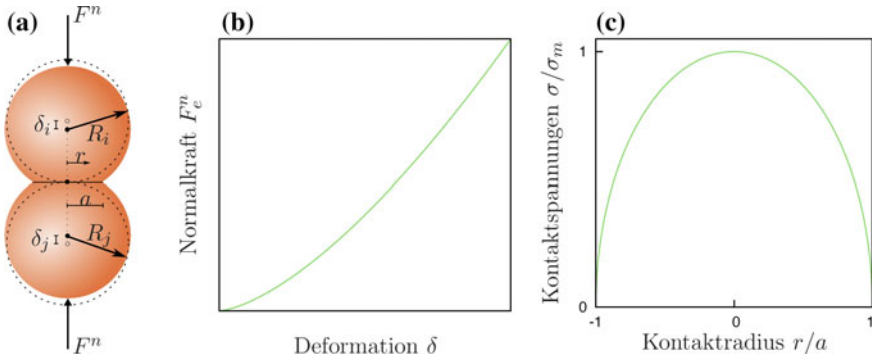


Fig. 2 Hertz contact law

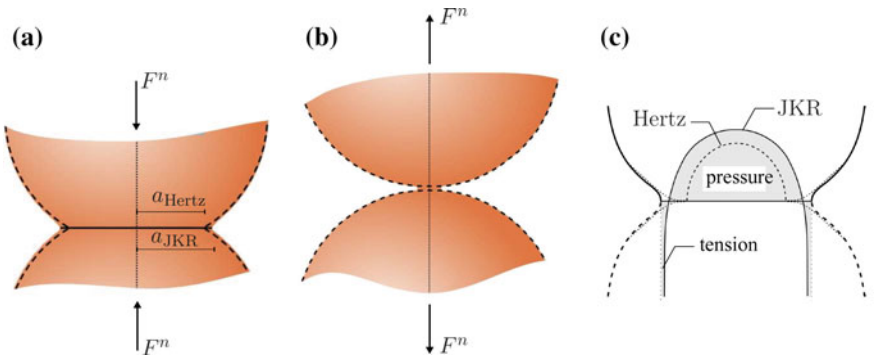


Fig. 3 Adhesion in the contact interface

As a result of the mutual compression of the particles, a circular area is formed in the contact zone. One can deduct that 1 the radius a of the shaped contact area is related to the total deformation δ via $a^2 = R\delta$.

Figure 2 depicts the deformation state that is assumed locally within the quasi rigid spheres. It also shows the relation between approach δ of the two spheres with respect to the force F^n. One can easily see the nonlinearity of the Hertz law. Furthermore the distribution of the contact pressure σ/σ_m depending on the contact radius r/a is depicted in the right part of the figure. Here $\sigma = F^n/(\pi\, a^2)$ and $\sigma_m = \max \sigma$.

Adhesion Law

According to the JKR model, see Johnson et al. (1971), it is implied that the adhesive force acts only within the contact area, see Fig. 3. Here, the work of adhesion under a liquid or just the free energy changes to separate a unit contact area of $\mathcal{P}_i$ and $\mathcal{P}_j$ in a liquid medium (l) is defined as

$$W = \gamma_{il} + \gamma_{jl} - \gamma_{ij}, \tag{6}$$

where γ describes the respective interfacial energy, see Loskofsky et al. (2006). Since the adhesive force F_a^n is opposed to the elastic force F_e^n, it reduces the elastic deformation δ_e leading to

$$\delta_e - \delta_a = \frac{a^2}{R} - \sqrt{\frac{2\pi\, W a}{E}}, \tag{7}$$

where δ_e is obtained from the Hertzian law, the second term δ_a is due to adhesion, see Maugis (1992) for details. The corresponding forces follow as

$$F_e^n - F_a^n = \frac{4Ea^3}{3R} - 2\pi a^2 \sqrt{\frac{2WE}{\pi a}}. \tag{8}$$

In case of the absence of external forces, i.e. $F^n = 0$, then $F_a^n \neq 0$ while $F_e^n = 0$ and $F_e^d = 0$. Furthermore, an equilibrium contact area a_0 is formed in the contact zone where a mutual compression δ_0 of the particles occurs

$$a_0 = \left(\frac{9\pi\, WR^2}{2E}\right)^{1/3} \quad \text{and} \quad \delta_0 = \left(\frac{3R}{4}\left(\frac{\pi\, W}{E}\right)^2\right)^{1/3}. \tag{9}$$

To pull the particles off each other, one has to apply a traction force under which they suffer minute stretching deformations forming a connecting neck around the contact zone. Once the pulling force has reached a critical level, i.e. $F^n = -F_c^n$, the contact breaks and the particles will separate. The critical force F_c^n yields

$$F_c^n = \frac{3}{2}\pi\, WR \tag{10}$$

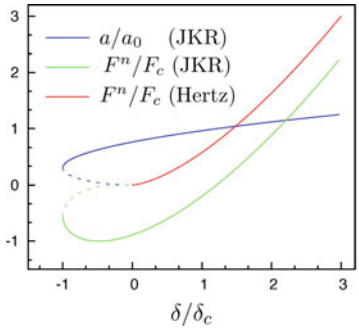
and the corresponding critical distance of the particles follows

$$\delta_c = \frac{1}{48^{1/3}} \frac{a_0^2}{R}. \tag{11}$$

Here the pulling distance can be defined as $\delta = -\delta_c$. Figure 4 demonstrates the differences in the force.

Viscous Effects in the Contact Interface Law

Viscous properties can be modeled using the system depicted in Fig. 5 for two spheres (a) and one sphere (b) being in contact wth a rigid wall. The forces in the damper d are provided below.



	δ/δ_c	a/a_0	F^n/F_c
Contact	0	$(2/3)^{2/3}$	$-8/9$
Equilibrium	$(4/3)^{2/3}$	1	0
Lift-Off	-1	$(1/6)^{2/3}$	$-5/9$

Fig. 4 Forces in the contact interface due to Hertz and the JKR model

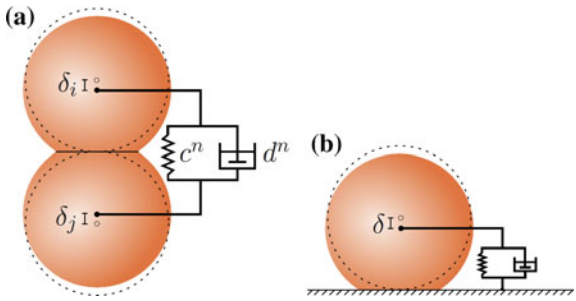


Fig. 5 Viscous effects in the contact interface

To consider the properties of material viscosity, a dissipative force is adopted according to the work of Brilliantov et al. (1996, 2007) which yields

$$F_d^n = A\dot{a}\frac{\partial}{\partial a}(F_e^n - F_a^n)\,. \tag{12}$$

From this definition, the viscous force follows as

$$F_d^n = \dot{a}A\left(\frac{4Ea^2}{R} - \frac{3}{2}\sqrt{8\pi\, WEa}\right), \tag{13}$$

where the dissipative factor[1] A is related to a constant function of material viscosity. Consequently, considering the above set of equations, one obtains the force-displacement relation for adhesive viscoelastic particles

[1]This factor A can also be used as a fitting parameter within specific simulations—like quasistatic predictions of granular material behaviour—to damp oscillations.

$$F^n(\delta) = \frac{4}{3} E \sqrt{R} \delta^{3/2} - 2R^{3/4} \delta^{3/4} \sqrt{2\pi\, W E} + A\dot{\delta} \left(2E\sqrt{R}\sqrt{\delta} - 3\sqrt{\frac{1}{2} R^{3/2} \pi\, W E \delta^{1/4}} \right). \tag{14}$$

Computation of the Approach

In the present contribution, the constitutive law given in (4) and (14) is treated analogous to the penalty method, e.g., see Wriggers (2006) and Wellmann et al. (2008). In this methodology one computes the force F_e^n between to particles $\mathcal{P}_i$ and $\mathcal{P}_j$ is computed from the approach of two particles by using the interface law (in case of the classical penalty approach this equation is $F_e^n = \varepsilon_P\, \delta$ with the spring stiffness, know as penalty parameter, ε_P). However in this case the penalty spring is nonlinear, see e.g. Eq. (4).

In all constitutive equations above the approach of the spheres or the rate of this approach has to be evaluated. For this one can compute the gap in normal direction g_n from the given current positions of two spheres

$$g^n = (R_i + R_j) - l > 0, \tag{15}$$

and define $g^n \equiv \delta$ as the mutual compression or approach of $\mathcal{P}_i$ and $\mathcal{P}_j$. With this kinematic relation the contact forces can be immediately computed by evaluating (4) and (14). In Eq. (15) $l = ||\mathbf{l}||$ is the length of the distance vector between $\mathcal{M}_i$ and $\mathcal{M}_j$ in the current configuration, where $\mathbf{l} = \mathbf{x}_i - \mathbf{x}_j$. The deformation rate $\dot{\delta}$ is computed in Eq. (16) by the projection of the relative velocity $(\mathbf{v}_i - \mathbf{v}_j)$ onto the direction of the normal unit vector $\mathbf{n}$. This yields

$$\dot{\delta} = \dot{g}^n = -(\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{n} \quad \text{where} \quad n = l/l. \tag{16}$$

We note that the direction of the contact force F^n of the respective particle is opposite to the direction of compression δ. Based on this observation, one can deduct from (14) the contact force $\mathbf{F}^n = F^n \mathbf{n}$ that contributes to the momentum equation (1).

Tangential Contact Model

The constitutive relation of Coulomb's law couples the tangential force F^t to the normal force F^n. For this the coefficient of friction has to be introduced as a constitutive parameter. The relation is not smooth since for sliding

$$F^t = \mu_G F^n \tag{17}$$

holds while

$$F^t \leq \mu_H F^n \tag{18}$$

is valid for sticking. In this relation the dynamic and the static coefficients of friction have to be introduced. The parameter μ_G stands for sliding and μ_H for stick, where $\mu_G \leq \mu_H$. The relations between normal and tangential force are depicted in Fig. 6. Here (a) shows a simplified model of the real behaviour in (b) with $\mu_G = \mu_H$. Another simplified model is depicted in Fig. 6c which differentiates between the coefficient of friction for sticking and sliding.

Within the constitutive treatment for the tangential interface force F^t, a tangential spring-dashpot element with an incorporated slider is used in order to model the tangential friction behaviour, see e.g. Cundall and Strack (1979). This model is depicted in Fig. 7 where again the difference between the tangential contact of two particles and of a particle with a rigid wall is made.

Since the tangential part of the interface force F^t is related to two states, sliding and sticking, it can be viewed like an elasto-plastic process where sliding relates to the plastic flow. For these types of problems efficient algorithms were developed in the mid 1980s. The first application to frictional contact can be found in Wriggers (1987) and has been further developed over the years, see e.g. Wriggers et al. (1990), Luding (2004) and Wriggers (2006). The idea is to algorithmically predict first a "trial" stick step followed by a slip check in the second step. Then the regularized penalty formulation for the tangential trial traction takes the form

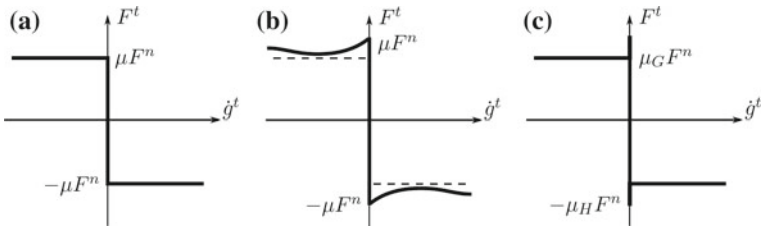


Fig. 6 Friction states with stick and sliding

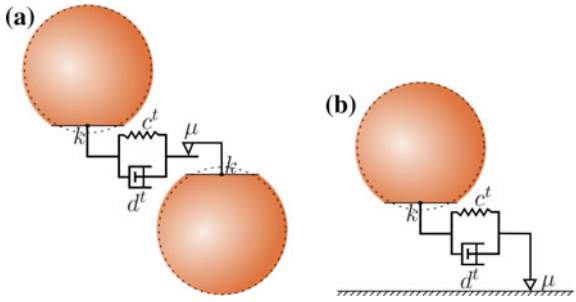


Fig. 7 Friction states with stick and sliding

$$\mathbf{F}_o^t = -(c^t \mathbf{g}^t + d^t \mathbf{v}^t). \tag{19}$$

Therein, $\mathbf{g}^t$ is the elongation of the tangential spring, c^t and d^t are the tangential spring stiffness and the tangential dissipation parameter, respectively. The tangential relative velocity at the contact point $\mathcal{C}$ is given by

$$\mathbf{v}^t = \mathbf{v}^s - (\mathbf{v}^s \cdot \mathbf{n})\,\mathbf{n} \tag{20}$$

with the relative velocity at $\mathcal{C}$

$$\mathbf{v}^s = \mathbf{v}_i^{\mathcal{C}} - \mathbf{v}_j^{\mathcal{C}}, \tag{21}$$

where the surface velocities are defined by $\mathbf{v}_i^{\mathcal{C}} = \mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i$ and $\mathbf{v}_j^{\mathcal{C}} = \mathbf{U}_j + \boldsymbol{\omega}_j \times \mathbf{r}_j$. The vectors pointing from $\mathcal{M}_i$ and $\mathcal{M}_j$ to $\mathcal{C}$ are associated with $\mathbf{r}_i = R_i(-\mathbf{n})$ and $\mathbf{r}_j = R_j\mathbf{n}$, respectively. By introducing a trial function f^{tr}, the following relation can be stated for the tangential contact

$$f^{\mathrm{tr}} =: ||\mathbf{F}_o^t|| - \mu_s ||\mathbf{F}^n|| \Rightarrow \begin{cases} \leq 0 & : \text{ Stick} \\ > 0 & : \text{ Slip.} \end{cases} \tag{22}$$

If $f^{\mathrm{tr}} \leq 0$ the contact point $\mathcal{C}$ is in the stick region and if $f^{\mathrm{tr}} > 0$ it is in slip region, thus sliding occurs in the contact area. Note that the stick case is valid again if $F_o^t < \mu_d F^n$ holds during the sliding process. If the contact point sticks, the actual tangential spring $\mathbf{g}^t$ is incremented for the succeeding time step by the relation $\Delta\mathbf{g}^t = \mathbf{v}^t \Delta t_{\mathrm{D}}$. Consequently, the new spring length is defined by

$$\overline{\mathbf{g}}^t = \mathbf{g}^t + \Delta\mathbf{g}^t\,. \tag{23}$$

Here, Δt_{D} denotes the time step of the discrete element method. However, if the contact point slides in the current time step, the tangential spring is aligned by

$$\overline{\mathbf{g}}^t = -\frac{1}{c_t}(F^t \mathbf{t} + d^t \mathbf{v}^t) \tag{24}$$

in order to fulfill Coulomb's slip condition. Therein, $\mathbf{t} = \mathbf{F}_o^t / ||\mathbf{F}_o^t||$ is the direction of the trial traction.

In two subsequent time steps, the contact area might be slightly rotated. As proposed in Luding (2004), the tangential spring is continuously projected onto the current rotated contact area at the beginning of each new time step via $\mathbf{g}^t = \overline{\mathbf{g}}^t - (\overline{\mathbf{g}}^t \cdot \mathbf{n})\mathbf{n}$. For the above approach, the tangential contact force is $F^t = ||\mathbf{F}_o^t||$ if $f^{\mathrm{tr}} \leq 0$ and $F^t = \mu_d F^n$ if $f^{\mathrm{tr}} > 0$ holds. By computing F^t, one obtains $\mathbf{F}^t = F^t\mathbf{t}$ and $\mathbf{T}^t = \mathbf{r} \times \mathbf{F}^t$ which contributes to $\mathbf{F}$ and $\mathbf{T}$ in Eq. (1) and Eq. (2), respectively.

Rolling Resistance Model

During a rolling motion of two particles over each other, the leading part of the contact area is continuously compressed and the trailing part is decompressed with respect to the rolling direction, see Fig. 8.

In case of an attractive van der Waals force in the contact area, the particles suffer an opposing torque to rolling motion, whereby a rolling resistance is generated. For instance, regarding an agglomerated straight particle chain with a lack of rolling resistance, the chain cannot resist any external impact resulting in a tangential force. As a consequence, the particles will roll smoothly over each other, whereby the chain easily bends and it will finally take a compact shape. For a detailed discussion of the rolling resistance of adhesive particles, see Dominik and Tielens (1995).

For a constitutive treatment of the rolling resistance, a model consisting of a rolling spring-dashpot-slider element is adopted in this contribution, see Iwashita and Oda (1998) and Fig. 9.

At this, the opposing torque is given by

$$\mathbf{M}_o^r = -(c^\varphi \phi + d^\varphi \dot{\phi}) = -\frac{1}{R}(c^\phi \mathbf{g}^r + d^\varphi \mathbf{v}^r). \tag{25}$$

Therein, c^φ is the rolling stiffness, d^φ relates to the rolling viscosity coefficient, ϕ to the particle rotation, $\mathbf{g}^r$ to the rolling distance and $\mathbf{v}^r$ is the rolling velocity which, according to Kuhn and Bagi (2004), can be computed using

$$\mathbf{v}^r = -R\left[(\boldsymbol{\omega}_i - \boldsymbol{\omega}_j) \times \mathbf{n} + \frac{1}{2}\left(\frac{1}{R_j} - \frac{1}{R_i}\right)\mathbf{v}^t\right]. \tag{26}$$

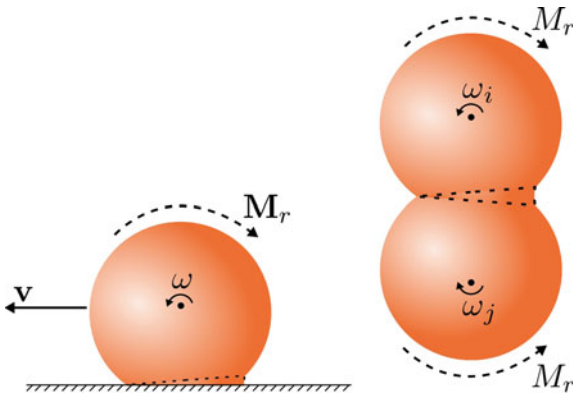


Fig. 8 Idea of rolling resistance

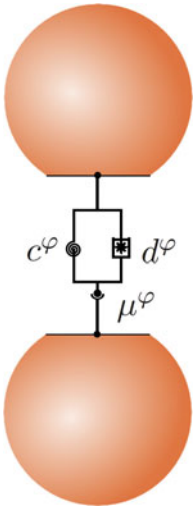


Fig. 9 Model of rolling resistance

In the context of the JKR theory one can set for c^ϕ the following expression

$$c^\varphi = 4F_c^n R\left(\frac{a}{a_0}\right)^{3/2} \qquad \text{where} \qquad d^\varphi = 0\,, \tag{27}$$

see Dominik and Tielens (1995) for details.

Introducing a trial force $\mathbf{F}_o^r = \mathbf{M}_o^r/R$, the problem of the rolling resistance can algorithmically be treated analogous to the tangential friction model. Here, one can apply as a yield criteria for the slider $M_{\max}^r$ or $\mu^r F^n$, where μ^r is a rolling friction coefficient, see Iwashita and Oda (1998).

DEM Solver

The kinematic variables of the particles are governed by the computation of the equations of motion (1) and (2). For this purpose, a time integration scheme has to be applied to solve these equations.

At the same time the contact between the particles has to be considered. This yields the different contact forces and moments between the particles. These forces and moments can be computed using the interaction laws discussed in the previous section.

Time Integration

Since DEM simulations usually need very small time steps Δt in order to resolve the constitutive laws between the particles the computational time for a larger system of particles is high. Thus time integration methods need to be considered very carefully. They have on one side to guarantee a certain accuracy and on the other side they need to be efficient and robust to solve the Newton-Euler-equations in (2). An overview with respect to methods that are used in DEM can be found in e.g. Kruggel-Emden et al. (2008).
The integration scheme is composed of three steps:

1. Predicting all the kinematic variables.
2. A force computation step is followed using the predicted variables according to Section "Constitutive Modeling of the Particle". Hence, the evolution of the translational and rotational accelerations can be computed from Eqs. (1) and (2).
3. Applying an error criteria between the predicted and calculated accelerations, the correction of the kinematic variables is ensued.

To illustrate the construction of such time integration scheme we consider a Gear algorithm in more detail. This starts with a predictor computation which is different for translation and rotation.
For the translation one computes

$$\begin{aligned}
\mathbf{x}^P(t+\Delta t) &= \mathbf{X}(t) + \dot{\mathbf{X}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{X}}(t)\Delta t^2 + \frac{1}{6}\dddot{\mathbf{X}}(t)\Delta t^3 \\
\dot{\mathbf{x}}^P(t+\Delta t) &= \dot{\mathbf{X}}(t) + \ddot{\mathbf{X}}(t)\Delta t + \frac{1}{2}\dddot{\mathbf{X}}(t)\Delta t^2 \\
\ddot{\mathbf{x}}^P(t+\Delta t) &= \ddot{\mathbf{X}}(t) + \dddot{\mathbf{X}}(t)\Delta t \\
\dddot{\mathbf{x}}^P(t+\Delta t) &= \dddot{\mathbf{X}}(t)
\end{aligned} \tag{28}$$

For the rotation the predictor is given by

$$\begin{aligned}
\boldsymbol{\omega}^P(t+\Delta t) &= \boldsymbol{\Omega}(t) + \dot{\boldsymbol{\Omega}}(t)\Delta t + \frac{1}{2}\ddot{\boldsymbol{\Omega}}(t)\Delta t^2 + \frac{1}{6}\dddot{\boldsymbol{\Omega}}(t)\Delta t^3 + \frac{1}{24}\overset{(iv)}{\boldsymbol{\Omega}}(t)\Delta t^4 \\
\dot{\boldsymbol{\omega}}^P(t+\Delta t) &= \dot{\boldsymbol{\Omega}}(t) + \ddot{\boldsymbol{\Omega}}(t)\Delta t + \frac{1}{2}\dddot{\boldsymbol{\Omega}}(t)\Delta t^2 + \frac{1}{6}\overset{(iv)}{\boldsymbol{\Omega}}(t)\Delta t^3 \\
\ddot{\boldsymbol{\omega}}^P(t+\Delta t) &= \ddot{\boldsymbol{\Omega}}(t) + \dddot{\boldsymbol{\Omega}}(t)\Delta t + \frac{1}{2}\overset{(iv)}{\boldsymbol{\Omega}}(t)\Delta t^2 \\
\dddot{\boldsymbol{\omega}}^P(t+\Delta t) &= \dddot{\boldsymbol{\Omega}}(t) + \overset{(iv)}{\boldsymbol{\Omega}}(t)\Delta t \\
\overset{(iv)}{\boldsymbol{\omega}}{}^P(t+\Delta t) &= \overset{(iv)}{\boldsymbol{\Omega}}(t)
\end{aligned} \tag{29}$$

Based on these predictions the force $\mathbf{F}^P(\mathbf{x}^P, \dot{\mathbf{x}}^P)$ and the moment $\mathbf{M}^P(\dot{\mathbf{x}}^P, \boldsymbol{\omega}^P)$ are computed which act both on a particle. Based on these quantities the translational and

rotational accelerations can be computed from (2). This yields $\mathbf{a}^* = \mathbf{F}^P/m$ and $\dot{\boldsymbol{\omega}}^* = \mathbf{M}^P/\Theta$. Now the differences $\Delta\ddot{\mathbf{x}} = \mathbf{a}^* - \ddot{\mathbf{x}}^P$ and $\Delta\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}^* - \dot{\boldsymbol{\omega}}^P$ can be determined for the corrector step in which the translations and rotations are corrected. For the translations it follows

$$\begin{aligned}
\mathbf{x}(t+\Delta t) &= \mathbf{x}^P(t+\Delta t) + c_0^t\,\mathbf{s}^t \\
\dot{\mathbf{x}}(t+\Delta t) &= \dot{\mathbf{x}}^P(t+\Delta t) + c_1^t\,\mathbf{s}^t\,\frac{1}{\Delta t} \\
\ddot{\mathbf{x}}(t+\Delta t) &= \ddot{\mathbf{x}}^P(t+\Delta t) + c_2^t\,\mathbf{s}^t\,\frac{2}{\Delta t^2} \\
\dddot{\mathbf{x}}(t+\Delta t) &= \dddot{\mathbf{x}}^P(t+\Delta t) + c_3^t\,\mathbf{s}^t\,\frac{6}{\Delta t^3}
\end{aligned} \tag{30}$$

and the rotations are obtained from

$$\begin{aligned}
\boldsymbol{\omega}(t+\Delta t) &= \boldsymbol{\omega}^P(t+\Delta t) + c_0^r\,\mathbf{s}^r \\
\dot{\boldsymbol{\omega}}(t+\Delta t) &= \dot{\omega}^P(t+\Delta t) + c_1^r\,\mathbf{s}^r\,\frac{1}{\Delta t} \\
\ddot{\boldsymbol{\omega}}(t+\Delta t) &= \ddot{\boldsymbol{\omega}}^P(t+\Delta t) + c_2^r\,\mathbf{s}^r\,\frac{2}{\Delta t^2} \\
\dddot{\boldsymbol{\omega}}(t+\Delta t) &= \dddot{\boldsymbol{\omega}}^P(t+\Delta t) + c_3^r\,\mathbf{s}^r\,\frac{6}{\Delta t^3} \\
\overset{(iv)}{\boldsymbol{\omega}}(t+\Delta t) &= \overset{(iv)}{\boldsymbol{\omega}}{}^P(t+\Delta t) + c_4^r\,\mathbf{s}^r\,\frac{24}{\Delta t^4}
\end{aligned} \tag{31}$$

with the constants

$$\mathbf{s}^t = \frac{\Delta t^2}{2}\Delta\ddot{\mathbf{x}}\,,\; c_0^t = \frac{1}{6}\,,\; c_1^t = \frac{5}{6}\,,\; c_2^t = 1\,,\; c_3^t = \frac{1}{3}$$

and

$$\mathbf{s}^r = \Delta t\,\Delta\dot{\boldsymbol{\omega}}\,,\; c_0^r = \frac{251}{720}\,,\; c_1^r = 1\,,\; c_2^r = \frac{11}{12}\,,\; c_3^r = \frac{1}{3}\,,\; c_4^r = \frac{1}{24}$$

Within this method of Gear the translations are approximated by a Taylor expansion of 3rd order while the rotations have to be modeled by a 4th order Taylor expansion. This choice is necessary to achieve conservation of momentum and moment of momentum and a good accuracy of the solution. A more complete description of the Gear algorithm can be found in Allen and Tildesley (1987), Pöschel and Schwager (2005).

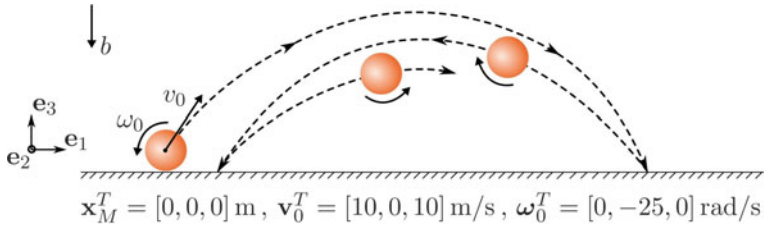


Fig. 10 Behaviour of a rigid sphere for a given velocity including frictional contact

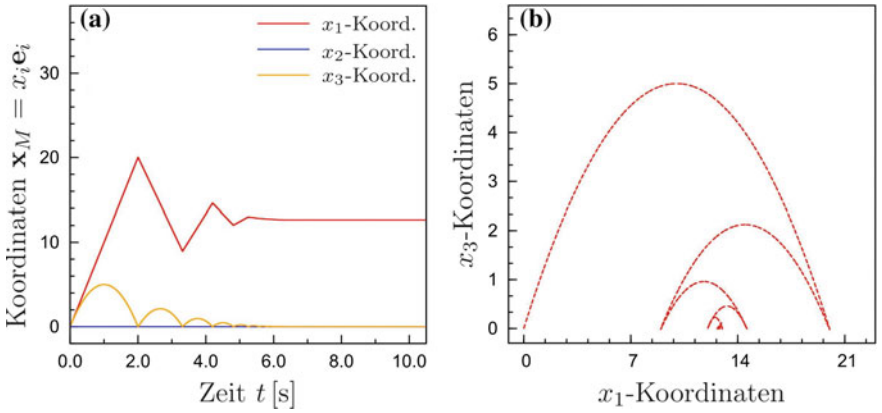


Fig. 11 **a** Temporal evolvement of the coordinates of the center point of the particle and **b** trajectory of the particle during the motion

Test Example

A specific example is considered to show that the normal and tangential contact is working correctly. Here only one sphere is considered that comes into contact with a rigid surface. It will be shown that the given formulations and the algorithms are able to reproduce the rebound of the particle, see Fig. 10. The following constants are used to compute the motion of the particle: density $\rho = 2.5\,\mathrm{kg/m^3}$, dissipation factor $A = 6 \times 10^{-4}\,\mathrm{s}$, normal stiffness $c_t = 10^6\,\mathrm{N/m}$, damping $d_t = 50\,\mathrm{Ns/m}$ and friction coefficient for stick and slip $\mu_{(G,H)} = 2.5$. The behaviour can be modeled experimentally using a bouncy ball made of rubber.

As shown in Fig. 10 the particle is subjected to an initial translations and rotational velocity v_0 und $\boldsymbol{\omega}_0$ at time $t_0 = 0$ and starts from the rigid plane. The rotational velocity is described as a backward spin.

The temporal evolvement of the numerically computed coordinates of the particle center can be found in Fig. 11a. Figure 11b depicts the trajectory of the particle center and thus shows the behaviour of the particle that jumps back and forth due to the initial velocities and the gravitational force ρb. From the diagrams in Fig. 11 one can observe that the maximal altitude of the particle is $x_{3,\max} = 5.00\,\mathrm{m}$ after that it

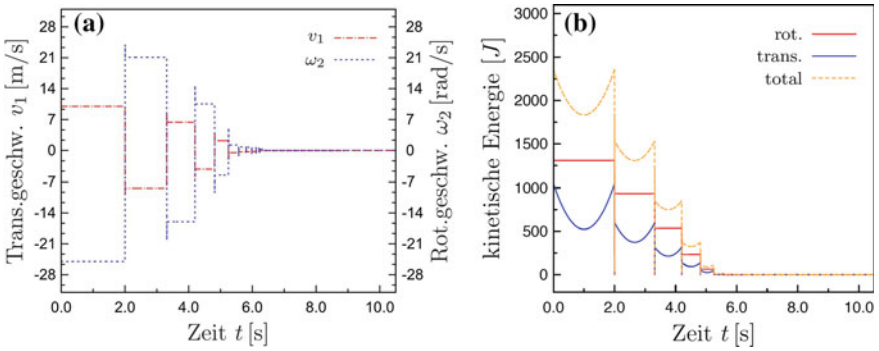


Fig. 12 **a** Temporal evolvement of the translational and rotational velocities and **b** temporal evolvement of the kinetic energy

starts to drop and contacts the rigid plane after $t_k = 2.00\,\mathrm{s}$ at $x_{1,k} = 20.00\,\mathrm{m}$. This numerical result matches exactly the analytical solutions for the trajectory[2]. When the particle impacts the rigid plane then due to tangential relative displacements a frictional force develops that is opposite to the direction of the relative velocity. This force reduces the tangential translation velocity and contributes to a change in the rotational velocity. In this case the frictional force is large enough to reverse the tangential motion and the rotation, see Fig. 12a. Thus the particle bounces back. However due to the frictional dissipation the kinetic energy is reduced such that the particle does not reach the position at t_0. The numerical computed kinetic energy is depicted in Fig. 12b with respect to time. Following the further trajectory of the particle one observes that it bounces back and forth until the kinetic energy is used up which is depicted by the horizontal line in the right part of Fig. 12b .

Search Algorithms

Besides the time integration, the computation of the contact forces is the most CPU time consuming part of a DEM simulation. Thus the contact search and detection governs the efficiency of a DEM code. WIth this in mind, the evaluation of the contact detection Eq. (15) has to be minimized to neighboring particles, since they are the only eligible contact partners. If one simply creates a loop over all N particles in a system, the number of tests amounts to $\sum_{i=1}^{N-1}(N-i) = N(N-1)/2$. Such an algorithm has the complexity of $O(N^2)$ and will result in extremely large computation time, even for small particle systems with less than a few thousand particles. If large particle

[2] Analytical solution for the trajectory:

$$\tilde{x}_{3,\max} = \frac{(v_0 \sin\alpha)^2}{2\,b}\,, \quad \tilde{x}_{1,k} = \frac{v_0^2 \sin(2\,\alpha)}{b}\,, \quad \tilde{t}_k = \frac{2\,v_0 \sin\alpha}{b}\,.$$

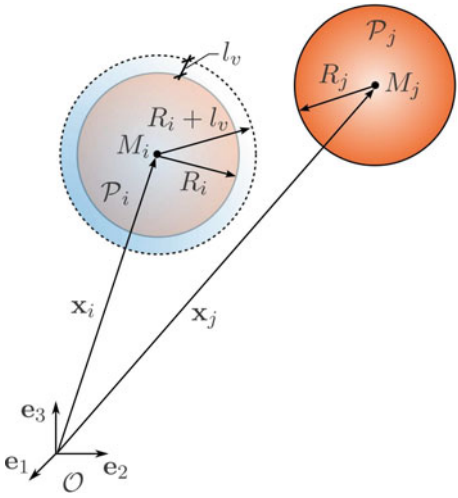


Fig. 13 Particle contact: Verlet distance

systems with several millions of particles have to be computed then an algorithm of complexity $O(N)$ is needed.

There many different algorithm that reduce the complexity of contact search. Among them are methods like space cell decomposition, combinations with binary tree search and heapsort algorithms for the global search. More advanced algorithms are the NBS algorithm, the alternating digital tree method, space filling curve technique or Double-Ended Spatial Sorting (DESS) algorithm.

Here we will look in more detail at algorithms that stem from the area of molecular dynamics. These are known as Verlet list and linked cells and can be found for the Verlet list in Verlet (1967) and for the link cells in Quentrec and Brot (1973). Here both algorithms will be combined in order to yield a fast contact search algorithm, see also Ref. (Allen and Tildesley 1987) and references therein.

Verlet lists are based on the idea that for each particle $\mathcal{P}_i \{i = 1, \ldots, N\}$ one generates a list with neighbouring particles. By this, the local contact check can be reduced to these neighbouring particles and thus is a local operation. The Verlet list can be build, based o the computation of a distance

$$g^n_{\text{Verlet}} = (R_i + l_v) + R_j - (\mathbf{x}_i - \mathbf{x}_j) \cdot \mathbf{n} \quad \begin{cases} \leq 0 \text{ no neighbour} \\ > 0 \text{ neighbour} \begin{cases} j < i \text{ list of } \mathcal{P}_i \\ j > i \text{ list of } \mathcal{P}_j \,. \end{cases} \end{cases}$$

This takes into account the Verlet distance l_v depicted in Fig. 13. The Verlet distance can be defined by

$$l_v = \frac{1}{2} \left(\text{Max} \left[R_i \right]_{i=1}^{N} + \text{Min} \left[R_i \right]_{i=1}^{N} \right) \alpha_v \,.$$

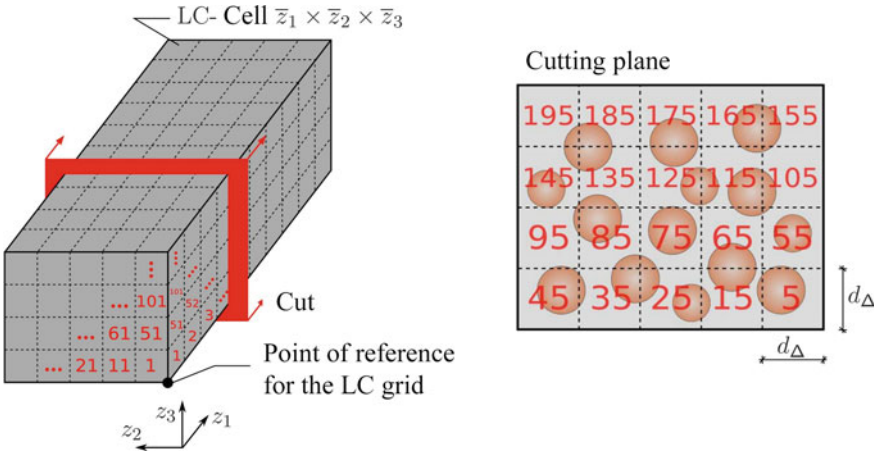


Fig. 14 Building the Verlet list with the linked cell method

where α_v is a constant that can be chosen. A good choice for this constant is $\alpha_v \approx 0.05$–0.10.

A list of neighboring particles is maintained for each particle $\mathcal{P}$ holding $l_v > |g^n|$. Once the Verlet list is built, Eq. (15) is only evaluated for neighboring pairs. Certainly, the list has to be updated at some intervals since particles can move to other locations and then have completely new neighbours. Here, a possible rebuild criteria can be defined by

$$\Delta s_{\max} = \text{Max}\left[\, ||\mathbf{x}_i - \mathbf{x}_i'|| \,\right]_{i=1}^{N} > \beta_v l_v \,.$$

In this equation the position vector $\mathbf{x}_i'$ of a particle $\mathcal{P}_i$ is used which relates to the position vector of the existing Verlet list while $\mathbf{x}_i$ is the position vector of the current time step. The factor β_v can be generally chosen in the range 0.5–0.6 which ensures that no complete penetration of a particle by another one occurs. For details see Pöschel and Schwager (2005).

In total the Verlet list method has a complexity of $O(N)$ for the computation of the local contact and thus the contact forces. However building the list is not so efficient. Due to this drawback the linked cell approach is used to construct the list. This amounts to an algorithm that overall approaches a complexity of $O(N)$. The idea of the linked cell algorithm is to divide the computational domain into cubic cells of uniform side lengths that are slightly larger than the maximal particle diameter, see Fig. 14.

After assigning the particles $\mathcal{P}_i$ to the cells with respect to their center of mass $\mathcal{M}_i$, the relevant particles for the construction of the neighbor list are referenced to the 26 surrounding cells and of course to the mid-cell which contains the considered particle $\mathcal{P}_i$.

Example: Silo Discharge

The DEM scheme is validated by means of a laboratory silo discharge experiment. Choi et al. (2004) analyzed the velocity profile in a quasi 2D silo using an image based particle tracking method. For the numerical modeling of this silo discharge, see Wellmann (2011).

The box-shaped silo of size [20 × 2.5 × 90] cm is filled with soda lime glass beads of slight polydispersity ($d = 3 \pm 0.1$ mm) using a distributed filling procedure. In total 190.782 particles are used to model the silo discharge. This leads in three dimensions to 1.144.692 particles. The rectangular orifice of [16 × 25] mm at the bottom center is opened and a steady state flow is allowed to develop before the tracking procedure starts. The tracking covers a rectangular section of [20 × 50] cm above the orifice. The flow is measured at a rate of 125 frames per second for 16.4 s. For the evaluation of the velocity profile the observation window is divided into [48 × 48] mm cells used as averaging domains. For a more detailed description of the data gathering and evaluation see Choi et al. (2004, 2005).

The elastic parameters of soda lime glass are given as $E = 71$ GPa and $\nu = 0.22$. The mass density is $\rho = 2.5$ g/cm3. The friction coefficient between dry soda lime glass beads was measured by Ishibashi et al. (1994) as $\mu = 0.162$. The gravity constant is chosen as $g = 9.81$ m/s^2. Since the particles in the vicinity of the orifice move at reasonable velocities the viscoelastic contact law described in Section "Normal Contact Model" is applied. The material constant A, see (13), to include viscoelastic effects was chosen as $A = 5.05\ 10^{-8}$ s for the soda lime glass beads. This corresponds to a restitution coefficient of 0.97 at a relative velocity of 1.18 m/s and a particle size of $d = 3.18$ mm. The material properties of the glass silo were assumed to be identical with the glass bead properties.

The stiffness of the particles and their size require a time step of the order of $\Delta t \approx 1\,\mu s$. For a total simulation time of 10 s this results in a number of time steps of the order of 10^7. Combined with a number of particles of $N \approx 2 \times 10^5$ this amounts to a large computational effort. However, numerical tests show that the system behavior is not altered significantly by reducing the stiffness of the particles to E = 0.9 GPa. This value assures that the overlap corresponding to the maximum contact force at the silo bottom is less than 1% of the particle radius. In order to preserve the dynamic particle behavior the viscoelastic constant is adopted to $A = 2.9\ 10^{-7}$ s. The stiffness reduction enables a time step of $\delta t = 10\,\mu s$ resulting in 3×10^6 steps for a simulation period of 30 s.

The initial sample, see Fig. 15, is generated using uniformly distributed particles with diameters in the range 2.9–3.1 mm and a solid fraction in the packaging of $\Phi = 0.5$. To account for the solid fraction of a denser package ($\Phi = 0.6$) the box used as package space is enlarged in the z direction to 1.08 m. In order to get realistic fabric properties the sample is settled under the influence of gravity using the DEM scheme and the above material parameters. At the point where the kinetic energy was nearly dissipated the orifice was opened starting the silo discharge simulation for 30 s. During the simulation the output was written at 0.3 s intervals, see Fig. 15. After a

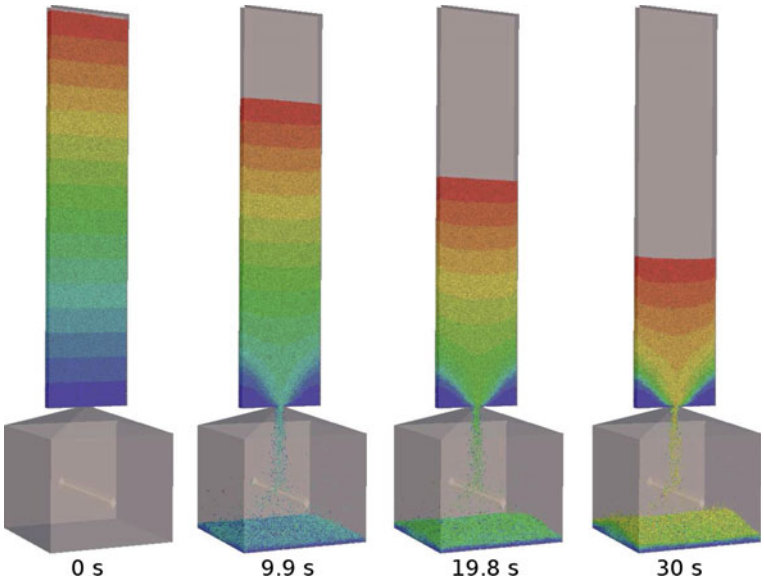


Fig. 15 Outflow of the particles from the silo at different times

short time a steady state flow develops with a mass flow rate of $Q = 132.6$g/s. This corresponds to a deviation of 6% from the experimental flow rate of Q = 141.1 g/s. Considering the uncertainties regarding the initial distribution of the particles and the container-particle friction this match is satisfactory.

In order to deduce a continuous downward velocity distribution $v(x)$ from the discrete DEM output a coarse graining scheme is applied. For this purpose the box-shaped [20.5 × 50] cm silo volume above the orifice is divided into a regular grid of [40 × 1 × 80] linear hexaeders. An ansatz v_h is defined on the hexaeder mesh and is fittes to the discrete DEM results using a volume weighted least square approach. Finally, the velocity profile is evaluated in the silo mid-plane and averaged over the data points between $t = 5$ s and $t = 20$ s. The resulting profile is compared with the experimental profile from (Choi et al. 2004) in Fig. 16.

There is a good agreement of the maximum velocity at the orifice, of the velocity gradient around the orifice, and of the run of the contour lines. The DEM scheme predicts a realistic shape of the stagnant zones at the lower silo corners. For quantitative comparisons 1D velocity profiles are evaluated at two heights sketched in the right part of Fig. 16.

The results are presented in Fig. 17. While there is a close agreement of the profiles in the vicinity of the orifice at $z = 9.1\,d$ the simulation predicts higher downward velocities at the boundaries at $z = 29.1\,d$. Note that this deviation can result basically from the different evaluation schemes: While the DEM data is averaged over the entire depth of the silo, the experimental image-based particle tracking method is based solely on the trajectories of the particles at the front window.

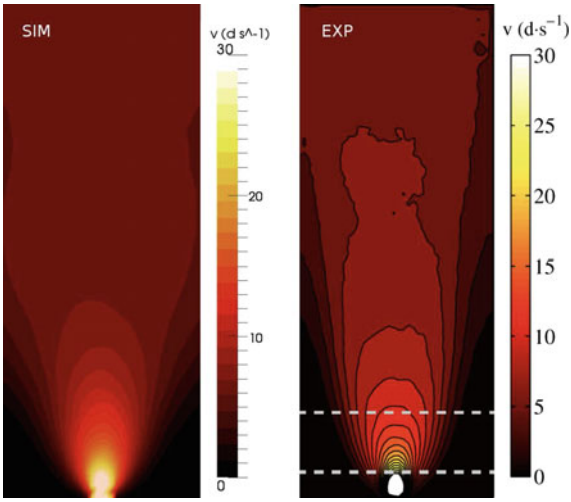


Fig. 16 Comparison of downward velocity: simulation and experiment

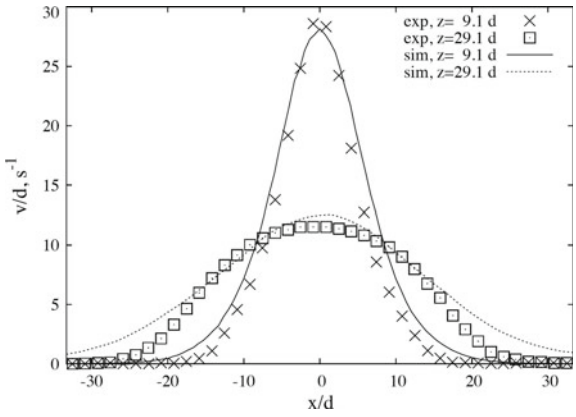


Fig. 17 Comparison of downward velocity: simulation and experiment

In total it is obvious that the discrete element method can predict flow states in a granular material accurately based on the introduced interface laws between the particles. However due to the large number of particles involved in one discretization it will be necessary to use parallel computing architectures. This is the only possibility to obtain reasonable results within engineering time frames.

DEM Using Parallel Solvers

Many engineering problems have to be modelled by using a large number of particles. These are e.g. milling applications or outflow problems in silos as discussed in the last section. Since many particles interact which each other high computing power

is needed to model a particle flow application in an accurate manner. This leads to computer codes that include algorithms which can make the transition from serial to parallel computing.

Within such models millions of spherical or arbitrary geometry particles have to be generated in order to describe a typical in granular flow problem accurately. The main application areas being simulations in geomechanics, structural, oil and gas engineering and other processes involving granular materials and particulate flows. When using parallel computing chunks of particles have to be distributed to the different processors. This distribution not only affects the computational part but also the set up of the problem. Within the latter task scaling of mesh generators necessarily requires mesh splitting before multiple processors assignment and domain decomposition techniques. During the run time—due to large movements of particles a load balancing procedure has to be executed in order to distribute the computational load equally to all processors.

Solver DEMFLOW

The mesh-free particle code DEMFLOW, developed at our institute over many years, is a parallel and fully open source framework for the simulation of the flow of granular materials, fluid and particulate flows. This particle solver was initially designed for cluster computations and was further developed to run efficiently on massively parallel high performance computer (HPC) architectures allowing large scale computations of engineering problems. The focused application areas of DEMFLOW are particle and multiphase flows and fluid-particle interaction. The DEM-module is completely developed in-house.

A common feature shared by mesh-free particle methods which also include meshless methods like smoothed particle hydrodynamics (SPH), see Gómez-Gesteira et al. (2012a); Gomez-Gesteira et al. (2012b), is that they all represent computationally very intensive simulation techniques. Due to their explicit nature, the corresponding solvers require very small time steps. In classical discrete element methods the restriction is due to the displacement-driven nature of the force computations and in the SPH due to the CFL condition. As a result, a large number of time steps need to be performed in order to simulate a time period of practical interest. Moreover, to discretize real engineering problems in a 3D framework of a particle method, a set of at least several millions particles is required for a sufficiently detailed reconstruction of a complex system. In general, the simulation of systems with more than one million particles can only be carried out in a reasonable way if a massively parallelized strategy is used. This is also necessary in order to handle the aspect of memory requirement, especially in really large scale problems.

A widely used strategy for the parallelization task in computational mechanics is the domain decomposition approach in conjunction with the use of the Message Passing Interface (MPI) library for inter-domain or inter-process communication. In the framework of this strategy, the computational domain is geometrically split

into sub-domains according to the number of specified processes (tasks, cores). Each generated sub-domain is then assigned to a core, including the respective particles. Both the communication and the data exchange between the cores are realized by the network interconnect and controlled by the instructions of the MPI library. Communication between sub-domains is necessary since the force computation or kernel evaluation of a particle close to a boundary of its sub-domain requires information about the particles positioned in the next boundary layer of the corresponding adjacent sub-domain.

A crucial point in the domain decomposition strategy, which demands great attention when applied to particle methods, is the aspect of the rescaling load-balance of the processes. As particles can cover large distances by traversing several sub-domains of the computational domain this occurs especially in highly dynamic systems, like in rotary mixing drums (for an illustration, see Fig. 20). Many dynamic systems are often characterized by a heterogeneous particle distribution over the entire domain. As a result, some sub-domains include only a small number of particles and while others have a large number of particles. In the former case, the cores are underused and complete their jobs faster than the cores in the latter case. Thus, they are idle while waiting for the rest of the processes to complete. In general, unequal workload leads to a strong decrease in the efficiency of the calculation with respect to the number of used cores. Consequently, the development of powerful algorithms for a continuous adaptation of the domain decomposition in terms of a homogeneous workload across all processes is inevitable for efficient computations of large scale engineering problems. This critical point represents a great challenge, especially in case of irregular 3D computational domains with a highly dynamic change of the particle distribution.

A second essential aspect that needs to be accounted for in order to design a high performance simulation tool in the framework of particle methods is directly linked to the organization and management of the particle data in the memory. The procedure for the force or kernel evaluation represents the most computationally intensive part in a simulation time step. The corresponding routine is carried out for each particle and over the neighbors of each individual particle. A common approach is to create and maintain an array index list (Verlet list) which stores the neighbors of each particle. The Verlet list is generally built by the use of the Linked-Cell method, see Allen and Tildesley (1987). By using this list, the evaluations are only performed in the respective neighborhood of a discrete element, thus avoiding unnecessary computations and minimizing numerical costs of a simulation, see also Section "Search Algorithm". At the beginning of a simulation, the order of the particle data sequence assembled in the memory corresponds usually to the real particle neighborhood connectivity, which is represented by the indices in the neighbor list. With time, the initial arrangement of the particles will be scattered due to large relative particle motions. As the neighborhood connectivity changes, the particle data in the memory can be considered as dispersed with respect to the accesses to the neighbor list. This leads to a significant amount of cache misses, because the particles are now more or less randomly distributed. Thus, a currently requested data of a neighbor is increasingly less likely to be present in the loaded cache line compared to the

beginning of the simulation run. However, in case of a cache miss, fetching the data of a particle from the main memory is several magnitudes slower than retrieving the data from one of the caches (L1-, L2- or L3-cache). Consequently, very efficient particle reordering algorithms and data management strategies are essential to maximize cache hits and to sustain the performance of a code.

There are a number of particle solvers in the literature that use the domain decomposition approach in combination with MPI for parallelization. Some of those advanced tools based on mesh-free particle methods are briefly mentioned below. A sophisticated SPH fluid solver, called SPH-Flow, is presented in Maruzewski et al. (2009). One of the applications of this code was the simulation of the impact of a snooker ball onto the free surface of water. The authors report a parallel efficiency of 58% when solving the problem with 124 million fluid particles on 2048 cores, 72% for 7.5 million particles with 128 cores and 87% for 3.5 million particles with 64 cores. The computations for the performance analysis were carried out on a Blue Gene/L high performance computer. Gadget-2 is another advanced open source particle code reported in the literature. This SPH code extends the work of Springel (2005) and is designed to perform parallel large scale cosmologic simulations. A modified version of Gadget-2 regarding its application to hydrodynamic problems is presented in Ulrich and Rung (2006). In this work, the authors report that the solver shows a super-linear speedup behavior when applying it up to 256 cores. As a scalability test, they considered a fixed-size problem of a sloshing application with 40 million fluid particles. The computations for the performance test were carried out on the HLRN2 high performance computer. Motivated by the libraries that are provided by the well-known frameworks of PETSc and Trilinos for linear algebra, Sbalzarini et al. (2006) have designed and developed a MPI based middleware for particle methods (among others, for DEM, MD and SPH), which consists of highly efficient parallel libraries. The collection of open source libraries associated with this project is known as the Parallel Particle-Mesh (PPM) Library. The overall aim of the PPM middleware is to provide a very flexible framework with which a programmer can develop highly efficient particle-based software for HPC architectures in a short time. For instance, Walther and Sbalzarini (2009) developed a DEM tool in the setting of PPM and demonstrated a parallel efficiency of 40% on 192 cores of a Linux cluster by performing simulations of 3D sand avalanches with up to 122 million particles. The efficiency of a SPH code implemented by adapting the PPM libraries is presented in Sbalzarini et al. (2006) for a vortex ring simulation. The paper shows that this particle solver reaches an efficiency of 91% on 128 cores of a Cray XT4 HPC architecture.

The software DEMFLOW is developed on the basis of PPM libraries. In fact, the optimization of a parallel particle software regarding its load-balancing capabilities can yield a far more efficient tool as optimizing the respective calculation algorithms assigned to the cores. The collection of libraries that the PPM middleware provides in this context are very efficient and complex algorithms that can be included relatively easily in DEMFLOW. The corresponding PPM libraries are highly optimized and tested on HPC architectures with fairly encouraging results, see Sbalzarini et al. (2006) and Walther and Sbalzarini (2009). Those libraries include, among others,

a range of adaptive domain decomposition approaches, assignment algorithms for sub-domains onto cores according to their individual performances, dynamic load-balancing and heuristic criteria for particle domain updating. The integration of the parallel PPM libraries in DEMFLOW in terms of load-balancing will significantly contribute to the further development of the particle code regarding its parallel efficiency on HPC architectures.

The best way to minimize the amount of cache misses in particle based computations is to reorder the particle data in the memory according to the particle locality. However, in 3D, it is quite hard to sort scattered particle data for optimal cache use; one has to also consider that the neighborhood of each particle changes with time for relative particle motions. To deal with this problem, Springel (2005) uses (in his astrophysics SPH-software Gadget-2) an approach based on the Peano-Hilbert curve method. By applying this method, one can create a space-filling curve that maps the considered 3D domain onto a 1D curve. In Gadget-2, this curve is employed both to realize the domain decomposition and to reorder the particle data in the memory. In the latter case, the algorithm maps the particle positions onto the Peano-Hilbert curve and sorts the particles based on this mapping. Finally, a particle arrangement in memory which conforms to a large extent to the real neighborhood of the particles is obtained. In Springel (2005), it is reported that computational tests with Peano-Hilbert ordering show a performance gain of the code Gadget-2 by a factor of two when compared with the performances achieved by neglecting data sorting. This approach was implemented in DEMFLOW.

Numerical Test: Medium Number of Cores

Both major factors of a particle solver—adaptive load-balancing and dynamic particle data reordering—have severe impact on the parallel performance of a code.

As the algorithms which should cover these two aspects are not yet developed in a sophisticated manner and implemented into the DEMFLOW solver, the computation domain of the system chosen to study its performance is only slightly affected by a traversing particle flow. THe next example was selected to discuss the scalability of the code on a parallel computer. For that particles were used to roughly model a fluid.

The considered system consists of two relatively light cubes that plunge into a compact particle domain which is fully discretized with fluid particles, as illustrated in Fig. 18. Here, load-balancing and data reordering are of less importance. The performance tests were run on a Linux-Cluster. The cluster is equipped with 16 nodes where most of the nodes contain either two Intel Xeon E5-2642-v2 CPUs (2×6 cores) or two Intel Xeon E5-2670 CPUs (2×8 cores). The speedups and efficiency of the software are computationally evaluated on the basis of four discretization levels of the system: the original problem consists of 5 million particles, the half-sized model of 2.5 million particles, the double-sized model of 10 million particles and the quadrupled-sized model of 20 million particles. For each discretization level, a

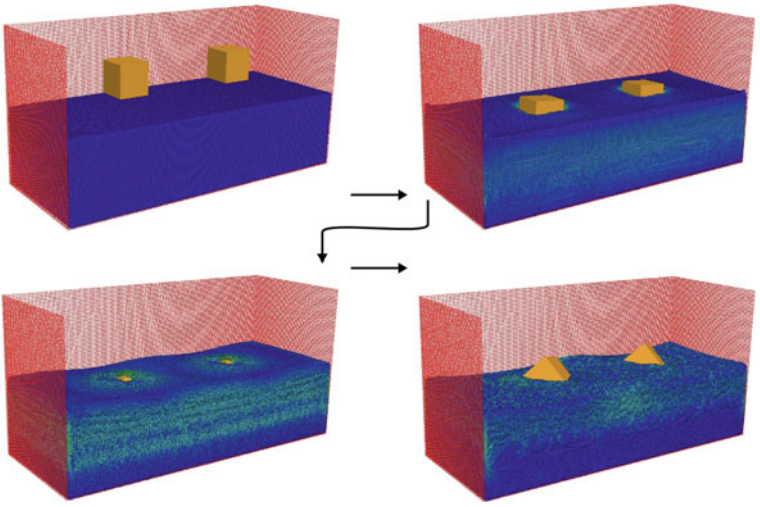


Fig. 18 Two light solid cubes plunging into water. Simulation results at different configurations in time

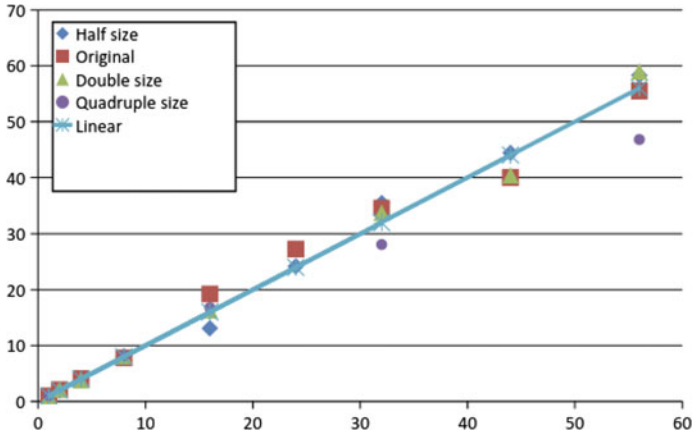


Fig. 19 Speed-ups of DEMFLOW for different problem sizes

fixed-size problem is examined. The obtained speedups and efficiencies of the test computations are displayed in Fig. 19, respectively. A linear speed-up can be observed for the original, half-sized and double-sized system. For the original problem, the study even shows slight super-linear behavior of the program for a smaller number of cores. Furthermore, a speedup of 47 is obtained for the largest system when 56 cores are used for computation. The latter case results in an efficiency of 84%, whereas the other three cases achieve efficiencies in the range of 90–120%. It can be concluded that the results obtained in this study are successful and promising as they show a very good speedup of the solver up to 56 cores.

Numerical Test: Large Number of Cores

DEMFLOW has been adapted to run on a high performance computer with large core numbers in order to validate its efficiency on HPC architectures. For this task a system which implies a sophisticated strategy both for load-balancing with adaptive domain decomposition and for particle-data reorganization was considered.

It can be shown that the optimization of a parallel particle software regarding its load-balancing capabilities can yield a far more efficient tool as optimizing the respective calculation algorithms assigned to the cores. The collection of libraries that the PPM middleware provides in this context very efficient and complex algorithms.

Hence discrete element methods have become a more and more powerful tool, especially for the treatment of granular materials and in process engineering. The method can nowadays be applied to problems that 10–1000 million particles for an accurate model. These processes are run on high performance parallel computers or GPU systems. The problem of the interchanging contact conditions due to large particle motions presents a challenge for the development of algorithms that scale well for large numbers of processors.

Here we will focus on an application that uses the discrete element method to model the mixing process of different particles in a drum. In this application 25 million of particles are used. The particles are mixed by rigid blades that rotate about the middle axis. The blades move up and down to perform the mixing. Here many small time steps have to be executed to follow the complex motion of the particles in a mixer. Such problems necessitate the use of parallel computing machines with many cores. The problem is depicted in Fig. 20a which also shows the allocation of parts of the particles to a specific core of the parallel computer.

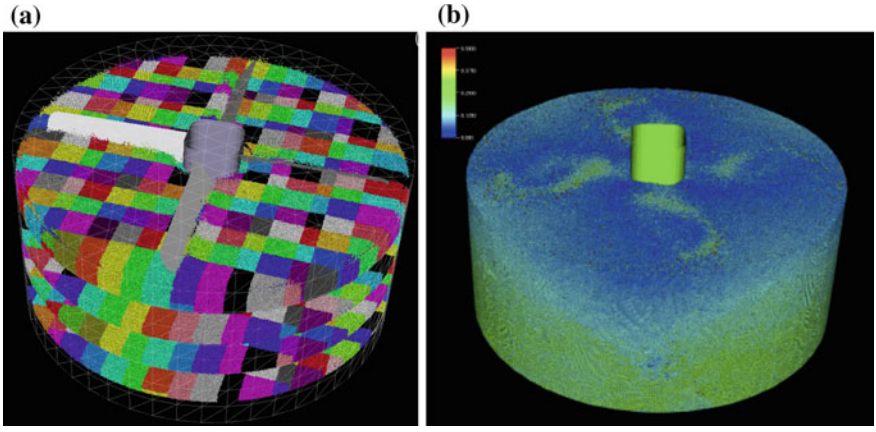


Fig. 20 **a** Set-up of the particles in the mixer. The color code shows the allocation to a core in the parallel computer. **b** Velocity distribution of the particles during mixing

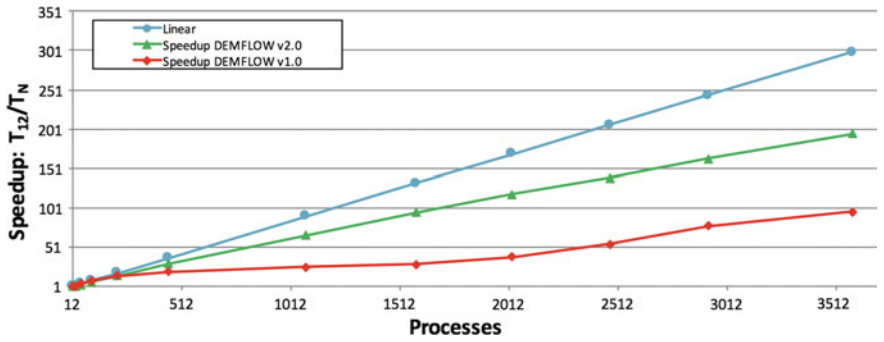


Fig. 21 Speed-up for different algorithmic treatments

In Fig. 20b the velocity distribution during the mixing process is depicted. One can easily observe the complex movement of the particles that change locations and by that also come into contact with other particles thus leaving the location of the initial allocation and thus have to be moved to another core. This usually requires a reallocation and slows down the computation. In order to get a good performance sophisticated algorithms have to be used that on one hand have to be fast and on the other limit data transfer. By that a speed-up that is linear, even for a large number of processors, can be achieved, see Fig. 21.

The blue curve shows the theoretical limit for a speed-up investigation up to 3500 processors. The red curve shows the performance for one algorithm while the more sophisticated algorithm and related data handling amounts to the green curve that produces an extremely good linear speed-up over all number of processors that reaches 66% of the theoretical limit.

Conclusion

The discrete element method is a very good tool for the prediction of granular flows and for the determination of macrsocopic material parameters of granular materials. It is a time consuming method since on one side millions of particles have to be used for adequate models and on the other side very small time steps have to be used to resolve the complex interaction laws between particles. This means that discrete element technologies have to be solved on computers with parallel architecture.

It was shown that DEMPACK exhibits a near linear scalability for small scale problems but also for large scale problems with rapidly changing interconnections of the particles due to large movements.

References

Alder, B. J., & Wainwright, T. E. (1957). Phase transition for a hard sphere system. *Journal of Chemical Physics*, *27*(5), 1208–1209.

Allen, M. P., & Tildesley, D. J. (1987). *Computer simulation of liquids*. New York: Oxford University Press.

Avci, B., & Wriggers, P. (2012). A dem-fem coupling approach for the direct numerical simulation of 3d particulate flows. *Journal of Applied Mechanics*, *79*, 01901.

Brilliantov N. V., Albers N., Spahn F., & Pöschel T. (2007). Collision dynamics of granular particles with adhesion. *Physical Review E, 76*(5, Part 1).

Brilliantov N. V., Spahn F., Hertzsch J. M., & Pöschel T. (1996). Model for collisions in granular gases. *Physical Review E, 53*(5, Part B), 5382–5392.

Choi, J., Kudrolli, A., & Bazant, M. Z. (2005). Velocity profile of granular flows inside silos and hoppers. *Journal of Physics: Condensed Matter*, *17*, 2533–2548.

Choi, J., Kudrolli, A., Rosales, R. R., & Bazant, M. Z. (2004). Diffusion and mixing in gravity-driven dense granular flows. *Physical Review Letters*, *92*, 174301.

Cundall, P. A., & Strack, O. D. L. (1979). Discrete numerical model for granular assemblies. *Geotechnique*, *29*(1), 47–65.

Dhia, H. B., & Rateau, G. (2005). The arlequin method as a flexible engineering design tool. *International Journal of Numerical Methods in Engineering*, *62*, 1442–1462.

Dominik, C., & Tielens, A. G. G. M. (1995). Resistance to rolling in the adhesive contact of 2 elastic spheres. *Philosophical Magazine A—Physics of Condensed Matter Structure Defects and Mechanical Properties*, *72*(3), 783–803.

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, *181*(3), 375–389.

Gómez-Gesteira, M., Crespo, A. J., Rogers, B. D., Dalrymple, R. A., Dominguez, J. M., & Barreiro, A. (2012a). Sphysics-development of a free-surface fluid solver-part 2: Efficiency and test cases. *Computers & Geosciences*, *48*, 300–307.

Gomez-Gesteira, M., Rogers, B. D., Crespo, A. J., Dalrymple, R. A., Narayanaswamy, M., & Dominguez, J. M. (2012b). Sphysics-development of a free-surface fluid solver-part 1: Theory and formulations. *Computers & Geosciences*, *48*, 289–299.

Hertz, H. (1882). Über die Berührung fester elastischer Körper. *Journal für die reine und angewandte Mathematik*, *92*, 156–171.

Ishibashi, I., Perry, C., & Agarwal, T. K. (1994). Experimental determinations of contact friction for spherical glass particles. *Soils and Foundations*, *34*, 79–84.

Iwashita, K., & Oda, M. (1998). Rolling resistance at contacts in simulation of shear band development by dem. *Journal of Engineering Mechanics-ASCE*, *124*(3), 285–292.

Johnson, A. A., & Tezduyar, T. E. (1997). 3d simulation of fluid-particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering*, *145*(3–4), 301–321.

Johnson, K. L., Kendall, K., & Roberts, A. D. (1971). Surface energy and contact of elastic solids. *Proceedings of the Royal Society of London Series A-Mathematical and Physical Sciences*, *324*(1558), 301–313.

Kruggel-Emden, H., Sturm, M., Wirtz, S., & Scherer, V. (2008). Selection of an appropriate time integration scheme for the discrete element method (dem). *Computers & Chemical Engineering*, *32*(10), 2263–2279.

Kuhn, M., & Bagi, K. (2004). Alternative definition of particle rolling in a granular assembly. *Journal of Engineering Mechanics-ASCE*, *130*(7), 826–835.

Loskofsky, C., Song, F., & Newby, B. Z. (2006). Underwater adhesion measurements using the JKR technique. *Journal of Adhesion*, *82*(7), 713–730.

Luding, S. (2004). Micro-macro transition for anisotropic, frictional granular packings. *International Journal of Solids and Structures*, *41*(21), 5821–5836.

Maruzewski, P., Le Touze, D., & Oger, G. (2009). Sph high-performance computing simulations of rigid solids impacting the free-surface of water. *Journal of Hydraulic Research*, *47*, 126–134.

Maugis, D. (1992). Adhesion of spheres—The jkr-dmt transition using a dugdale model. *Journal of Colloid and Interface Science*, *150*(1), 243–269.

Pöschel, T., & Schwager, T. (2005). *Computational Granular Dynamics*. Springer.

Quentrec, B., & Brot, C. (1973). New method for searching for neighbors in molecular dynamics computations. *Journal of Computational Physics*, *13*(3), 430–432.

Sbalzarini, I. F., Walther, J. H., Bergdorf, M., Hieber, S. E., Kotsalis, E. M., & Koumoutsakos, P. (2006). PPM—A highly efficient parallel particle-mesh library for the simulation of continuum systems. *Journal of Computational Physics*, *215*, 566–588.

Springel, V. (2005). The cosmological simulation code gadget-2. *Monthly Notices of the Royal Astronomical Society*, *364*, 1105–1134.

Ulrich, C., & Rung, T. (2006). Validation and application of a massively-parallel hydrodynamic SPH simulation code. *Proceedings of NuTTS '09*.

Verlet, L. (1967). Computer experiments on classical fluids. i. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*, *159*(1), 98–103.

Walther, J. H., & Sbalzarini, I. F. (2009). Large-scale parallel discrete element simulations of granular flow. *Engineering Computations*, *26*(6, Sp. Iss. SI), 688–697; *4th International Conference on Discrete Element Methods* (2007). Australia, Brisbane.

Wellmann, C. (2011) *A Two-Scale Model of Granular Materials Using a Coupled Discrete-Finite Element Approach*. Dissertation, B11/1, Institute for Continuum Mechanics, Leibniz University Hannover.

Wellmann, C., Lillie, C., & Wriggers, P. (2008). A contact detection algorithm for superellipsoids based on the common-normal concept. *Engineering Computations*, *25*(5–6), 432–442.

Wellmann, C., & Wriggers, P. (2012). A two-scale model of granular materials. *Computer Methods in Applied Mechanics and Engineering*, *205–208*, 46–58.

Wriggers, P. (1987). On consistent tangent matrices for frictional contact problems. In G. Pande & J. Middleton (Eds.), *Proceedings of NUMETA '87*. M. Nijhoff Publishers, Dordrecht.

Wriggers, P. (2006). *Computational Contact Mechanics* (2nd ed.). Berlin Heidelberg: Springer.

Wriggers, P., Van, T. V., & Stein, E. (1990). Finite-element-formulation of large deformation impact-contact-problems with friction. *Computers and Structures*, *37*, 319–333.

Zhu, H. P., Zhou, Z. Y., Yang, R. Y., & Yu, A. B. (2007). Discrete particle simulation of particulate systems: Theoretical developments. *Chemical Engineering Science*, *62*(13), 3378–3396.

Zohdi, T. I. (2007). *Introduction to the modeling and simulation of particulate flows*. SIAM.