Proceedings of ALGORITMY 2024
pp. 225–234

DEM SIMULATIONS OF SETTLING OF SPHERICAL PARTICLES USING A SOFT CONTACT MODEL AND ADAPTIVE TIME STEPPING

PAVEL STRACHOTA*

Abstract. We present a simple and flexible Discrete Element Method (DEM) model for simulating the dynamics of spherical particle systems. The aim is to utilize commonly available ODE integrators that are usually inappropriate for DEM, in particular the Runge-Kutta-Merson and Dormand-Prince solvers with adaptive time stepping. This is achieved by using a novel soft contact model with repulsive and frictional forces smoothly varying in time, which allows the time step adaptivity algorithms to work properly. The model parameters are calibrated so that a realistic random close packing can be obtained from simulations of particle settling at the bottom of a container. A reference minimal implementation in MATLAB and a complete implementation in C with OpenMP parallelization are introduced and their computational performance is assessed.

Key words. adaptive time stepping, discrete element method, spherical particles, soft contact model, OpenMP parallelization

AMS subject classifications. 70F35, 70F40, 65L06, 65Y05

1. Introduction. Discrete Element Method (DEM) [12, 20] is a class of methods for simulating dynamics of complex systems of discrete bodies subject to mutual collisions and interactions. Our motivation for using DEM was the need of creating a porous bed made of settled spherical particles in order to perform simulations of water freezing and thawing inside the porous structure at pore scale [17, 24]. For this purpose, a realistic geometry and volume of the pores needs to be obtained, which corresponds to random close packing [21] of the spheres. This requirement precludes the use of collision averaging techniques such as the multiphase particle-in-cell method [1, 13, 14] including its recent improvements [2, 6, 22].

In this paper, we present a simple and flexible DEM model able to utilize common ODE integrators with readily available implementations, in particular the Runge-Kutta-Merson [7] and Dormand-Prince [5] solvers with adaptive time stepping. This class of solvers is known to be inappropriate for DEM [10, 12, Chap. 2]. However, we propose a soft contact model with repulsive and frictional forces smoothly varying in time, which allows the time step adaptivity algorithms to work properly. A reference minimal implementation in MATLAB and a complete implementation in C with OpenMP parallelization [8] are introduced. The model parameters are calibrated and parallel performance is investigated.

2. Particle dynamics model. The studied system involves n identical spherical particles with unit mass and radius r. For each $i \in \{1, 2, \dots, n\}$, denote the position of the center of the i-th particle at the given time $t \geq 0$ by $\boldsymbol{x}_i\,(t)$, its velocity by $\boldsymbol{v}_i\,(t)$, and the particle's angular velocity by $\boldsymbol{\omega}_i\,(t)$. Note that for brevity, the time dependence will be dropped in the rest of the text. The particles are located in a convex container bounded by m planar walls with outward pointing unit normal vectors $\boldsymbol{n}_k$ for $k \in \{1, 2, \dots, m\}$.

[*] Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, (pavel.strachota@fjfi.cvut.cz).

Repulsive force. Consider two particles repelled from each other by a force acting along the connecting line of the particle centers. Let its magnitude depend solely on the distance (gap) l between the particle surfaces. As the particles approach each other, their kinetic energy transforms into potential energy, which is released again as the particles separate. Any such repulsive force strong enough to prevent particles from running through one another can therefore be used to approximate perfectly elastic collisions where the total energy and linear momentum are conserved. In addition, to use adaptive time stepping in the numerical solution, this force needs to be smooth enough with respect to the positions of the particles. Finally, it must become negligibly small as soon as the particle surfaces get far away from each other.

To satisfy the above requirements, we propose the magnitude F of the repulsive force in the form

$$F\left[l\right] = F_0 \exp\left(-\kappa l\right) \tag{2.1}$$

where F_0 is the repulsion at contact and κ is the stiffness exponent. This is very different from the usual soft contact models [9, 15] which only assume nonzero forces once the particle surfaces start to penetrate each other (i.e., $l < 0$).

Next, we allow for collisions described by the coefficient of restitution $e \in [0,1]$ [11], where $e = 0$ means perfectly inelastic collisions and $e = 1$ means perfectly elastic collisions. Generally, only a fraction (equal to e^2) of potential energy accumulated during the approach of the surfaces is recovered to kinetic energy as the surfaces separate. To model this, the mutual repulsive force during separation (unloading) phase is reduced by the factor e^2 as follows:

If $d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is the mutual distance of the centers of the particles i, j , its time derivative evaluates to the projection of the mutual velocity into the direction from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$, i.e.,

$$\dot{d}_{i,j} = \frac{1}{d_{i,j}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) \cdot \left(\boldsymbol{v}_i - \boldsymbol{v}_j\right). \tag{2.2}$$

The sign of (2.2) determines whether the particles are approaching each other (loading phase) or separating from each other (unloading phase). Then the repulsive force acting on the i-th particle imposed by collision with the j-th particle is given by

$$\boldsymbol{F}_{\mathrm{rep},i,j} = F\left[d_{i,j} - 2r\right] \rho \left[-\dot{d}_{i,j}\right] \frac{1}{d_{i,j}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right), \tag{2.3}$$

where

$$\rho\left[v\right] = e^2 + \frac{1}{2}\left(1 - e^2\right)\left(1 + \tanh\left(\psi v\right)\right) \tag{2.4}$$

is a rebound function ensuring a smooth transition from the full force to the reduced force upon particle separation. Essentially, $\rho\left[v\right] \approx e^2$ for $v < 0$ and $\rho\left[v\right] \approx 1$ for $v > 0$. ψ is the dissipation "focusing" coefficient. The model given by equations (2.3)–(2.4) is a smooth nonlinear analog of the widely used Walton-Braun elastoplastic model [9, 12, 15, 20, 23].

Similarly, the repulsive force acting on the i-th particle imposed by collision with the k-th wall at the distance $d_{i,k}$ from the particle center is given by

$$\boldsymbol{F}_{\mathrm{rep},i,\boldsymbol{n}_k} = -F\left[d_{i,k} - r\right] \rho \left[\boldsymbol{v}_i \cdot \boldsymbol{n}_k\right] \boldsymbol{n}_k. \tag{2.5}$$

Frictional force. We employ the Coulomb friction model, i.e., the frictional force is independent of the mutual tangential velocity of the surfaces. The magnitude of the frictional force is assumed to be proportional to the magnitude of the repulsive force [15, Sect. 4.1]. In accordance to this, consider the force acting on the i-th particle imposed by friction against the j-th particle in the form

$$\boldsymbol{F}_{\mathrm{fr},i,j} = -\varphi F\left[d_{i,j} - 2r\right] \Sigma\left[\left\|\boldsymbol{t}_{i,j}\right\|\right] \frac{\boldsymbol{t}_{i,j}}{\left\|\boldsymbol{t}_{i,j}\right\|}, \tag{2.6}$$

where φ is the friction coefficient and the mutual tangential velocity of the particle surfaces is calculated as

$$\boldsymbol{t}_{i,j} = \left(\boldsymbol{v}_i - \boldsymbol{v}_j\right) - \frac{\dot{d}_{i,j}}{d_{i,j}}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) - \frac{r}{d_{i,j}}\left(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j\right) \times \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right). \tag{2.7}$$

The frictional force needs to be regularized for tangential velocities near zero, which is achieved by using a continuously differentiable factor (a "limiter" inspired by [19])

$$\Sigma\left[v\right] = \begin{cases} \frac{3v^2}{v_{\min}^2} - \frac{2v^3}{v_{\min}^3} & v < v_0, \\ 1 & v \geq v_0, \end{cases} \tag{2.8}$$

where $v_{\min}$ is the minimum velocity magnitude for which the friction reaches its full strength.

Similarly, for friction between the i-th particle and the k-th planar wall, we have

$$\boldsymbol{F}_{\mathrm{fr},i,\boldsymbol{n}_k} = -\varphi F\left[d_{i,k} - r\right] \Sigma\left[\left\|\boldsymbol{t}_{i,\boldsymbol{n}_k}\right\|\right] \frac{\boldsymbol{t}_{i,\boldsymbol{n}_k}}{\left\|\boldsymbol{t}_{i,\boldsymbol{n}_k}\right\|}, \tag{2.9}$$

$$\boldsymbol{t}_{i,\boldsymbol{n}_k} = \boldsymbol{v}_i - \left(\boldsymbol{v}_i \cdot \boldsymbol{n}_k\right)\boldsymbol{n}_k + r\boldsymbol{\omega}_i \times \boldsymbol{n}_k. \tag{2.10}$$

Frictional torque. Besides linear acceleration, the frictional forces impose torque on the spheres. The torque acting on the i-th particle imposed by friction against the j-th particle is

$$\boldsymbol{\tau}_{i,j} = -\frac{r}{d_{i,j}}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) \times \boldsymbol{F}_{\mathrm{fr},i,j} \tag{2.11}$$

and the torque imposed by friction against the k-th planar wall is

$$\boldsymbol{\tau}_{i,\boldsymbol{n}_k} = r\boldsymbol{n}_k \times \boldsymbol{F}_{\mathrm{fr},i,\boldsymbol{n}_k}. \tag{2.12}$$

Equations of particle dynamics. In total, the equations of motion of the i-th particle read

$$\dot{\boldsymbol{x}}_i = \boldsymbol{v}_i, \tag{2.13}$$

$$\dot{\boldsymbol{v}}_i = \sum_{\substack{j=1 \\ j\neq i}}^{n}\left(\boldsymbol{F}_{\mathrm{rep},i,j} + \boldsymbol{F}_{\mathrm{fr},i,j}\right) + \sum_{k=1}^{m}\left(\boldsymbol{F}_{\mathrm{rep},i,\boldsymbol{n}_k} + \boldsymbol{F}_{\mathrm{fr},i,\boldsymbol{n}_k}\right) + \boldsymbol{g}, \tag{2.14}$$

$$I\dot{\boldsymbol{\omega}}_i = \sum_{\substack{j=1 \\ j\neq i}}^{n}\boldsymbol{\tau}_{i,j} + \sum_{k=1}^{m}\boldsymbol{\tau}_{i,\boldsymbol{n}_k}, \tag{2.15}$$

where $\boldsymbol{g} = \left(0,0,-g\right)^{\mathrm{T}}\ \mathrm{m\,s^{-2}}$ is the gravitational acceleration and $I = \frac{2}{5}r^2$ is the moment of inertia of a homogeneous sphere with unit mass. The evolution of the system of spherical particles is governed by the system of ordinary differential equations (ODE) (2.13)–(2.15) for $i = \{1,2,\ldots,n\}$ together with the initial conditions $\boldsymbol{x}_i\left(0\right) = \boldsymbol{x}_{i,\mathrm{ini}}$, $\boldsymbol{v}_i\left(0\right) = \boldsymbol{v}_{i,\mathrm{ini}}$, $\boldsymbol{\omega}_i\left(0\right) = \boldsymbol{\omega}_{i,\mathrm{ini}}$.

3. Numerical solution. In the following, we discuss and compare three variants of the model:

1. The *basic* variant only considers the repulsive forces, which is equivalent to assuming $\varphi = 0$ in (2.6) and (2.9).
2. The *friction* variant considers repulsion and friction, but not the rotation of the spheres. It is equivalent to setting $I \to +\infty$ in (2.15) together with $\boldsymbol{\omega}_{i,\mathrm{ini}} = \mathbf{0}$.
3. The complete model referred to as *friction+rotation*.

The implementation has been carried out in MATLAB (the basic variant only) and in C (all variants). For the solution of the ODE system (2.13), the MATLAB built in ODE solvers with adaptive time stepping have been employed. For a very small number of particles (tens of particles), stiff solvers with implicit time integration such as `ode15s` are suitable. Otherwise, implicit methods are too demanding and explicit solvers based on higher order Runge-Kutta methods [5] such as `ode45` (Dormand-Prince) should be used instead. In that case, a user-defined restriction on the maximum time step size may be necessary in order to prevent collision events being missed. In the C variant of the code, the Runge-Kutta-Merson (RKM) [7] solver implementation proposed in our earlier work [18] has been used, leveraging OpenMP [8] for multi-threaded parallelization. In the evaluation of the right hand side of (2.13)–(2.15), the parallel `for` loop over $i \in \{1, 2, \ldots, n\}$ is employed.

With any of the above options, the time step is automatically adjusted to resolve the collision events with sufficient accuracy. Note that smoothness of the stiffness coefficient (2.1), the rebound function (2.4), and the limiter (2.8) is a key prerequisite for the time step adaptivity algorithms to work properly. Due to the particle interaction exponentially decaying with distance, the system (2.13)–(2.15) can be solved as is. However, to speed up the computation, interactions of surfaces beyond a maximum interaction distance $l_{\max}$ can be skipped, provided that $F\left[l_{\max}\right]$ is sufficiently small so as not disturb time step adaptivity.

4. Simulations. In this section, the properties of the proposed DEM model will be demonstrated and the settings of the model parameters will be justified.

Single sphere bouncing calibration. In Fig. 4.1, the results of a bounce test with a single particle and different values of the coefficient of restitution are shown. The particle is dropped from the height of $h_0 = 16$ m with zero initial velocity. The final time of the simulation is $T = 10$ s and the remaining parameters are the same as in Tab. 4.2. The x_3 coordinate represents the height of the center of the spherical particle, so the radius of the sphere is irrelevant.

It can be observed that the simulation led to very accurate results. For $e = \sqrt{2}/2$, the kinetic energy of the sphere and hence the elevation of the highest point reached after each rebound is halved. The choice of the numerical solver (`ode15s` or `ode45` in MATLAB, RKM in C) plays a negligible role as long as reasonable accuracy settings are applied (`odeset('RelTol',1e-10,'AbsTol',1e-8)`). The stiffness exponent κ in (2.1), however, must be chosen carefully. If κ is small, the contact becomes softer and the particle position drops below zero. This leads to a slightly delayed rebound, as shown in Tab. 4.1. If κ is large, the simulation approximates hard contact accurately, but some numerical issues arise in simulations with many spheres (see below). Based on the above facts, the value $\kappa = 150$ has been chosen for the subsequent simulations. For this setting, choosing F_0 in (2.1) within the range between 1 and 100 produces almost identical results.

Table 4.1: Time t_{rebound} until a single sphere released from the elevation of $h_0 = 16\,\text{m}$ rebounds back to the maximum height. Comparison of simulations using different values of the stiffness exponent κ with the analytical solution for a hard contact $t_{\text{rebound}} = (1+e)\sqrt{2h_0/g}$.

		$e = 1$			$e = \sqrt{2}/2$	
κ	Analytical	Simulated	Error [%]	Analytical	Simulated	Error [%]
30		3.6409	+0.79		3.108	+0.84
150	3.6122	3.6184	+0.17	3.0832	3.0883	+0.17
500		3.6134	+0.03		3.0858	+0.08

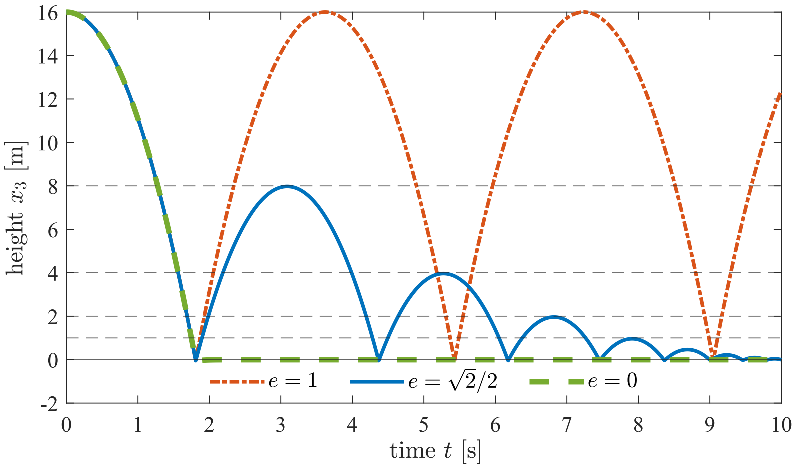


Figure 4.1: Trajectory of a single bouncing sphere dropped from the initial height of 16 m, with three different values of the coefficient of restititution e.

Simulations of spheres settling. In Fig. 4.2, the simulations of 200 falling spheres used to form a fixed porous bed for the simulations of freezing and thawing in [17] are shown, using all three variants of the solver (see Sect. 3). The parameters for the simulation are in Tab. 4.2. The initial configuration of the spheres (Fig. 4.2a) is organized in a randomly perturbed grid pattern to allow for easy alignment of the spheres as they stack at the bottom. Fig's 4.2b–4.2d display the final (almost) steady state, which was attained prior to the final time. No particle jittering or similar numerical instability effects occurred.

In Fig. 4.3, we assess the resulting configuration with respect to the volume fraction ε_{s} occupied by the spheres in the bottom part of the container. For the basic model without friction (Fig. 4.2b), the spheres gradually slide to a partially organized and partially random pattern. As a result, ε_{s} stabilizes at a value over 0.7, while the maximum theoretical value of ε_{s} for spheres arranged in a regular lattice is around 0.74 [21]. On the other hand, ε_{s} should be around 0.64 for a closed *random* packing. This can be achieved by incorporating friction and rotation of the spheres (Fig. 4.2d shows the result for $\varphi = 0.2$ and Fig. 4.3 includes the results for two values of φ). Friction without rotation produces unrealistic results (Fig. 4.2c).

Table 4.2: Parameters for the simulations of the settling of spheres

Parameter	Value	Meaning
r	0.1 m	radius of the spherical particles
n	200	particle count
e	0.4	coefficient of restitution
φ	0.2	friction coefficient
g	9.81 ms^{-2}	gravitational acceleration
T	8 s	final time
F_0	10 N	repulsion at contact
κ	150	stiffness exponent
ψ	10	dissipation focusing coefficient
$v_{\min}$	0.01 ms^{-1}	friction regularization threshold
$l_{\max}$	0.1 m	maximum interaction distance of surfaces
Δt_{ini}	0.1 s	initial time step
δ	0.1	error tolerance in the R-K-Merson solver [18]

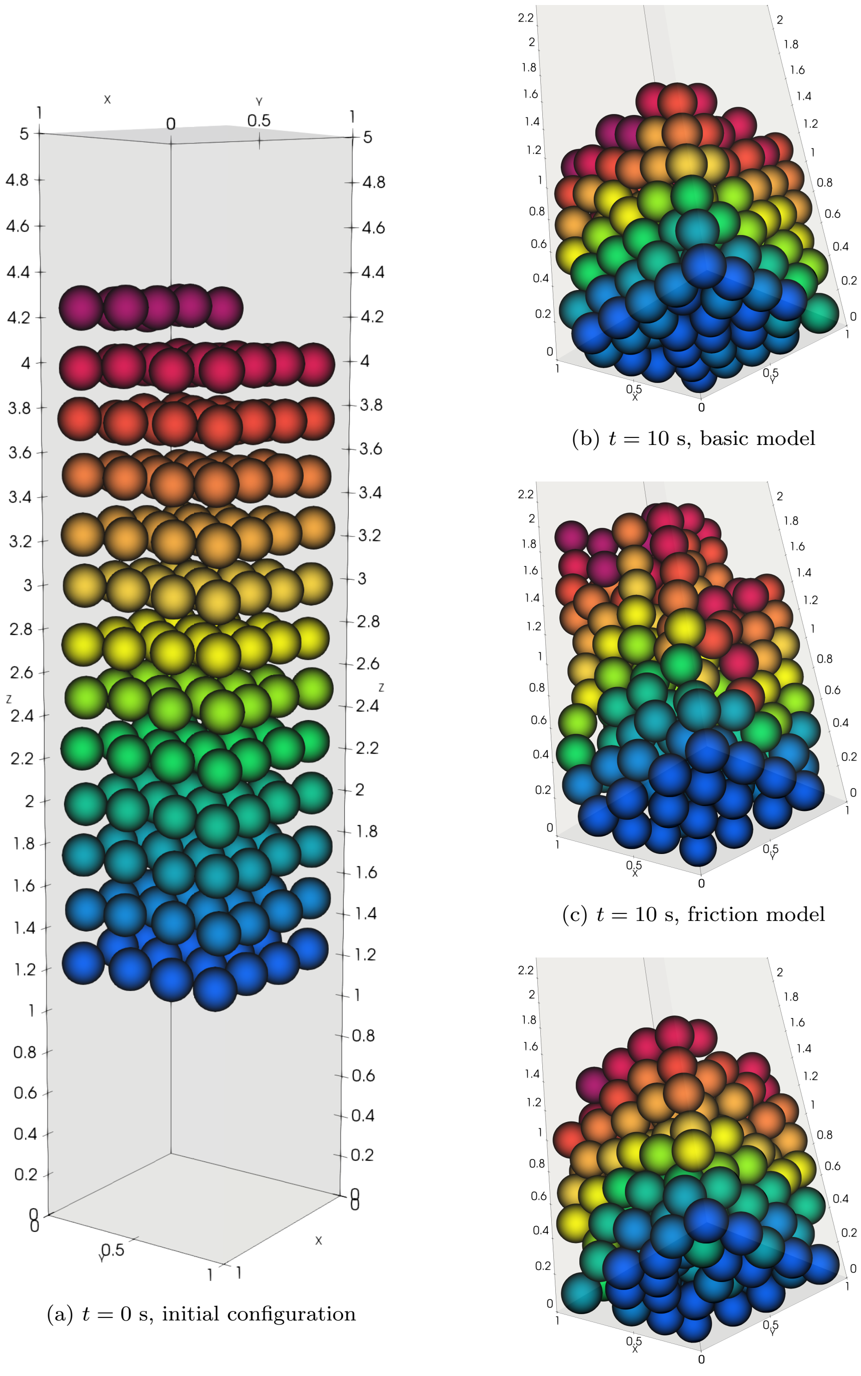
As for the effect of different choices of κ in this simulation, small values around $\kappa = 30$ result in mutual penetration of the spheres and ε_{s} being too large. Large values around $\kappa = 500$ lead to slight but unrealistic vibrations that prevent the spheres from settling down completely even without friction, causing ε_{s} to be too small.

Computational costs. All simulations were performed on a system with AMD EPYC 7543 32-Core Processor, 3200 MHz DDR4 memory, running CentOS 7.8. MATLAB R2023b and gcc 11.3 were used for the C and MATLAB variants, respectively. In MATLAB, the simulation with 200 spheres using the basic model took over 2 hours with 37399 right hand side evaluations (using `ode45`), which illustrates the serious performance limitations of this platform. In contrast, the same (parallel) simulation implemented in C took only around 2 seconds with 72190 r.h.s. evaluations (using RKM). Adaptive time stepping for the different model variants is illustrated in Fig. 4.4. Parallel efficiency of the C implementation of the full model with friction and rotation and parameters according to Tab. 4.2 is shown in Fig. 4.5

5. Conclusion.

We demonstrated DEM simulations of spherical particle dynamics based on a novel soft contact model suitable for commonly available ODE solvers with adaptive time stepping. The model parameters were first calibrated for a single bouncing particle and subsequently for obtaining realistic random close packing at the bottom of a container. The proposed algorithms have a notably simple implementation which is publicly available (see below). At the moment, no advanced contact search algorithms [16] for performance optimization were employed. However, parallel implementation allows to perform simulations with hundreds of particles within a couple of minutes on a multi-core CPU. Generalization of the algorithm to unequal sphere sizes and masses is easily possible. Generalization to non-spherical shapes can be achieved by including rolling resistance [3] or by using a multi-sphere model [4].

Data availability. The public GitHub repository
`https://github.com/radixsorth/PorousFreezeThaw`
provides the following materials under MIT License:



(a) $t = 0$ s, initial configuration

(b) $t = 10$ s, basic model

(c) $t = 10$ s, friction model

(d) $t = 10$ s, friction+rotation

Figure 4.2: The initial and final state of the simulation of 200 falling spheres, creating the porous bed used for simulations of water freezing and thawing in [17]. The dimensions of the rectangular base of the container are 1 m × 1 m. Simulation parameters are in Tab. 4.2. The steady state was reached prior to the final time T.

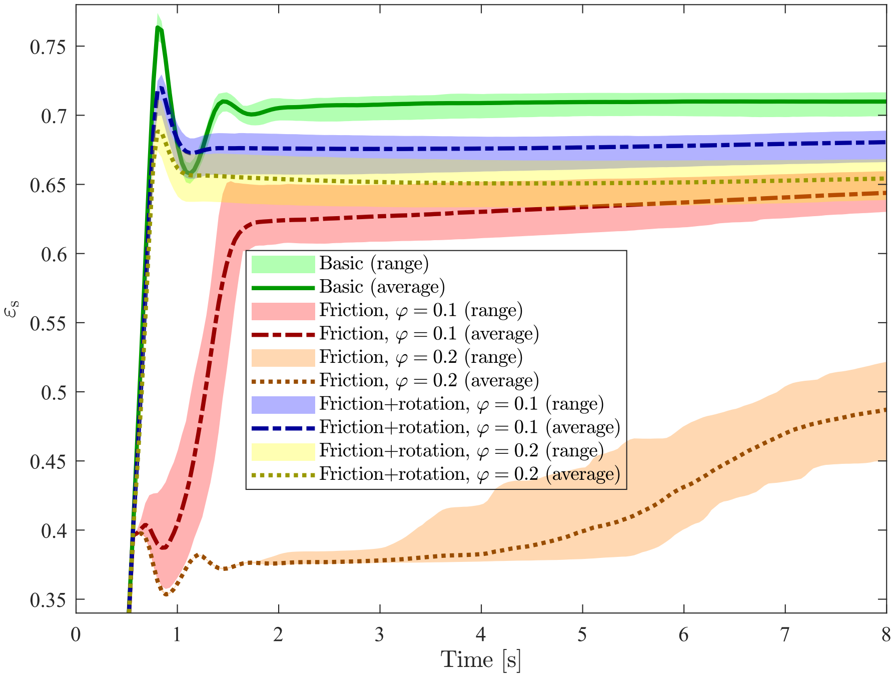


Figure 4.3: Evolution of the solids volume fraction ε_s in a cubic subdomain $[0,1]^3$ at the bottom of the vessel. Comparison of three model variants and two values of the friction coefficient φ. In each case, 10 simulations with different randomly perturbed initial conditions were performed. The ranges of the obtained values of ε_s and the averages are shown. Simulation parameters are given in Tab. 4.2 and the initial condition can be seen in Fig. 4.2a.

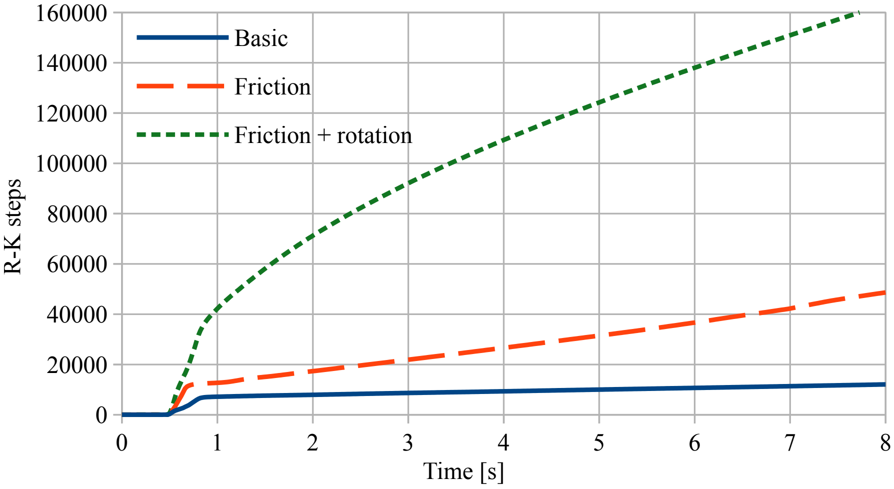


Figure 4.4: Number of successful time steps of the RKM solver as a function of the physical time. Different behavior of the time step adaptivity is observed for the three variants of the DEM simulation. Simulation parameters are given in Tab. 4.2.

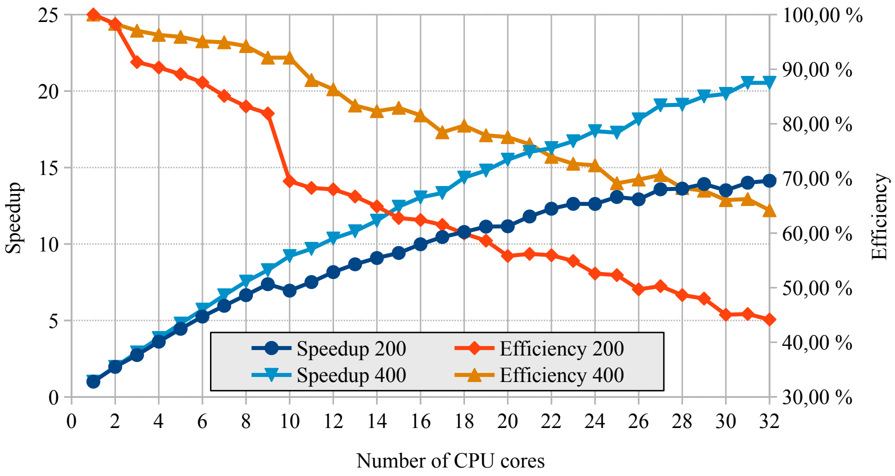


Figure 4.5: Speedup and parallel efficiency for OpenMP-parallelized simulations with 200 and 400 falling spheres. The wall time of the reference simulation on 1 CPU core was **4 min 22 s** for 200 spheres and **35 min** for 400 spheres. All values are obtained as averages from 3 consecutive runs.

- source code of the DEM spherical particle dynamics simulator
- 3D visualizations of the particle collision simulation results

In addition, it also includes the source code of the freezing and thawing simulator and files associated to the results presented in [17].

Acknowledgments. This work is part of the project *Multiphase flow, transport, and structural changes related to water freezing and thawing in the subsurface*, No. GA21-09093S of the Czech Science Foundation. Partial support by the grant *Modeling, prediction, and control of processes in nature, industry, and medicine powered by high performance computing*, No. SGS23/188/OHK4/3T/14 of the Grant Agency of the Czech Technical University in Prague, is gratefully acknowledged.

References.

[1] M. J. Andrews and P. J. O'Rourke. The multiphase particle-in-cell (MP-PIC) method for dense particulate flows. *International Journal of Multiphase Flow*, 22(2):379–402, 1996.

[2] M. Beneš, P. Eichler, J. Hrdlička, J. Klinkovský, M. Kolář, T. Smejkal, P. Skopec, J. Solovský, P. Strachota, and A. Žák. Experimental validation of multiphase particle-in-cell simulations of fluidization in a bubbling fluidized bed combustor. *Powder Technol.*, 416:118204, 2023.

[3] M. A. Benmebarek and M. M. Rad. Effect of rolling resistance model parameters on 3D DEM modeling of coarse sand direct shear test. *Materials*, 16:2077, 2023.

[4] N. Berry, Y. Zhang, and S. Haeri. Contact models for the multi-sphere discrete element method. *Powder Technol.*, 416:118209, 2023.

[5] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations.* Wiley, Chichester, 2nd edition, 2008.

[6] U. Caliskan and S. Miskovic. A chimera approach for MP-PIC simulations of dense particulate flow using large parcel size relative to the computational cell size. *Chemical Engineering Journal Advances*, 5:100054, 2020.

[7] J. Christiansen. Numerical solution of ordinary simultaneous differential equations of the 1st order using a method for automatic step change. *Numer. Math.*, 14:317–324, 1970.

[8] L. Dagum and R. Menon. Openmp: An industry standard API for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.

[9] A. Di Renzo and F. P. Di Maio. Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes. *Chemical Engineering Science*, 59:525–541, 2004.

[10] H. Kruggel-Emden, M. Sturm, S. Wirtz, and V. Scherer. Selection of an appropriate time integration scheme for the discrete element method (DEM). *Comput. Chem. Eng.*, 32:2263–2279, 2008.

[11] G. Kuwabara and K. Kono. Restitution coefficient in a collision between two spheres. *Jpn. J. Appl. Phys.*, 26(8):1230–1233, 1987.

[12] H.-G. Matuttis and J. Chen. *Understanding the Discrete Element Method: Simulation of Non-Spherical Particles for Granular and Multi-body Systems.* Wiley, 2014.

[13] P. J. O'Rourke and D. M. Snider. Inclusion of collisional return-to-isotropy in the MP-PIC method. *Chemical Engineering Science*, 80:39–54, 2012.

[14] P. J. O'Rourke and D. M. Snider. An improved collision damping time for MP-PIC calculations of dense particle flows with applications to polydisperse sedimenting beds and colliding particle jets. *Chemical Engineering Science*, 65(22):6014–6028, 2010.

[15] J. Rojek. *Contact Modeling for Solids and Particles*, chapter Contact Modeling in the Discrete Element Method, pages 177–228. Springer, 2018.

[16] H. Sigurgeirsson, A. Stuart, and W.-L. Wan. Algorithms for particle-field simulations with collisions. *J. Comput. Phys.*, 172:766–807, 2001.

[17] P. Strachota. Three-dimensional numerical simulation of water freezing and thawing in a container filled with glass beads. *arXiv*, (arXiv:2401.01672):1–12, 2024.

[18] P. Strachota and M. Beneš. A hybrid parallel numerical algorithm for three-dimensional phase field modeling of crystal growth. In A. Handlovičová and D. Ševčovič, editors, *ALGORITMY 2016, 20th Conference on Scientific Computing, Vysoké Tatry - Podbanské, Slovakia, March 14 - 18, 2016. Proceedings of contributed papers*, pages 23–32. Comenius University, Bratislava, 2016.

[19] P. Strachota, A. Wodecki, and M. Beneš. Focusing the latent heat release in 3D phase field simulations of dendritic crystal growth. *Modelling Simul. Mater. Sci. Eng.*, 29:065009, 2021.

[20] C. Thornton. *Granular Dynamics, Contact Mechanics and Particle System Simulations: A DEM study.* Sprin, 2015.

[21] S. Torquato, T. M. Truskett, and P. G. Debenedetti. Is random close packing of spheres well defined? *Phys. Rev. Lett.*, 84(10):2064–2067, March 2000.

[22] V. Verma and J. T. Padding. A novel approach to MP-PIC: Continuum particle model for dense particle flows in fluidized beds. *Chem. Eng. Sci.: X*, 6:100053:1–13, 2020.

[23] O. R. Walton and R. L. Braun. Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks. *J. Rheol.*, 30:949–980, 1986.

[24] A. Žák, M. Beneš, and T. H. Illangasekare. Pore-scale model of freezing inception in a porous medium. *Comput. Methods Appl. Mech. Eng.*, 414:116166, 2023.