

# Towards a Micromechanical Understanding of Landslides—Aiming at a Combination of Finite and Discrete Elements with Minimal Number of Degrees of Freedom

Jan Mueller, Akira Kyotani, Hans-Georg Matuttis

Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Chofu, Japan  
Email: jan@matuttis.mce.uec.ac.jp

**How to cite this paper:** Mueller, J., Kyotani, A. and Matuttis, H.-G. (2020) Towards a Micromechanical Understanding of Landslides—Aiming at a Combination of Finite and Discrete Elements with Minimal Number of Degrees of Freedom. *Journal of Applied Mathematics and Physics*, 8, 1779-1798. <https://doi.org/10.4236/jamp.2020.89134>

**Received:** July 24, 2020

**Accepted:** September 11, 2020

**Published:** September 14, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

In this paper, we propose a combination of discrete elements for the soil and finite elements for the fluid flow field inside the pore space to simulate the triggering of landslides. We give the details for the implementation of third order finite elements (“P<sub>2</sub> with bubble”) together with polygonal discrete elements, which allows the formulation with a minimal number of degrees of freedom to save computer time and memory. We verify the implementation with several standard problems from computational fluid dynamics, as well as the decay of a granular step in a fluid as test case for complex flow.

## Keywords

Finite Element Method, Discrete Element Method, Multi-Phase Flow

---

## 1. Micromechanics of Landslides

Landslides due to strong rain occur in rather confined areas, while the regions with the same soil and rain conditions are much larger. This means there must be additional triggers. But on the micromechanical level, the mechanisms on the grain scale, the flow inside the pore space and its restructuring are not yet understood. Sano [1] proposed a hydrodynamic scenario (how flow through neighboring cavities sweeps a whole channel free to form a larger duct which allows vehement flow), but as a cause of landslides it does not seem very likely: The capacity of soil to preserve spaces which can be washed out without interference by granular dynamics is rather limited. A combined mechanism where the weight of the water induces shear bands in the soil which can then be washed out by the

flow looks much more plausible. Another possible effect which loosens up dense granular matter is “Reynolds dilatancy”, where external stresses (in the case of landslides, that may be the additional weight of water due to strong rains) loosen up the soil homogeneously. Any scenario has to obey the laws of physics for particles and grains, and must be verified accordingly. In this paper, we outline a reliable simulation method for particles in fluids in two dimensions. For three-dimensional simulations, the two-dimensional situation should be understood first. A verification by direct micromechanical simulation is preferable to experiments here, as it allows much better control over the initial state and enables us to examine the exact physical processes occurring anywhere at any time within the geometry.

## 2. A Combined Simulation Method of Particles and Fluids

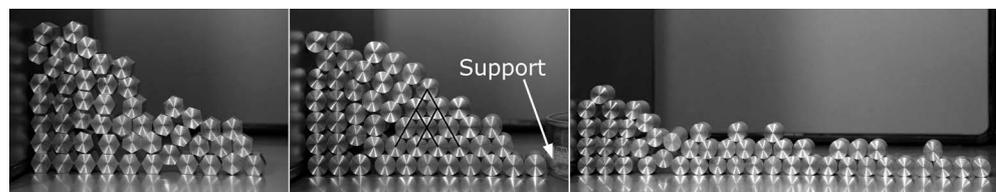
As far as the choice between Eulerian (grid-based) and Lagrangian (particle-based) representations is concerned, for the micromechanical understanding of landslides, we chose a particle-based formulation for the soil, and finite elements for the fluid. A list of all symbols with explanations is given in **Table 1**.

### 2.1. Choice of the Discrete Element Method

For the discrete element method (DEM), often round particles are used, which cannot form heaps (see **Figure 1**) without unphysical twists (switching off the rotation or introducing an unphysically high coefficient of rolling friction). Also monodisperse particles, especially with negligible elongation, cannot model large granular slopes correctly, as the particles will allow crystal-like ordering which allows slipping along the “crystal axes” with minimal mechanical resistance. Usually, the most convenient choice to obtain physical behaviour is a distribution of polygons with varying number of corners which are inscribed into ellipses with a statistical variation of the length of the half axes.

### 2.2. Choice of the Fluid Simulation Method

We refrain from simulating the fluid with a particles-based method, because there are problems with the spatial resolution and shot noise would be introduced even in equilibrium flow. For the complicated pore space between polygonal particles, exact discretization is possible via a trigonal grid, so finite elements are the natural



**Figure 1.** The polygonal particles (left) form stable heaps with fluctuations around a “straight” slope. Round particles need support on the sides to form heaps (center) and form a “rounded” slope nevertheless, or they may simply slide along the crystal axes (black lines) and roll away (right) when they have reached the smooth floor.

**Table 1.** List of symbols.

Variable	Description	Variable	Description
$A$	centroid of DEM particle contact overlap area	$res_i^{(-)}$	residual at mesh node $i$ , ( $p$ ) for pressure, ( $u$ ), ( $v$ ) for respective velocities
$A, \dots, J$	nodes of a finite element (index)	$S$	centroid of DEM particle contact overlap area $A$
$A_e$	area of a finite element	$T$	transpose operator
$a_g$	gravitational acceleration	$t$	time
$b$	distance from circle center to wall	$\hat{t}$	tangential direction unit vector
$C_1, C_2$	DEM particle center of mass	$U$	velocity vector
$d$	diameter	$U_0$	maximum inflow velocity
$dl$	line integral	$u$	horizontal velocity
$e$	finite element (index)	$u_{BC}, v_{BC}$	velocity boundary conditions
$entry_{\dots}$	entries for the Jacobian $J$	$u_{BCTOP}, v_{BCTOP}$	
$F_{da}$	DEM damping force	$v$	vertical velocity
$F_{drag}$	fluid force acting on DEM particles	$x$	horizontal position
$F_{el}$	DEM elastic force	$Y$	Young's modulus
$g_A^{(-)}, \dots, g_J^{(-)}$	coefficients for the element functions, ( $p$ ) for pressure, ( $u$ ), ( $v$ ) for respective velocities	$Y$	vertical position
$g_i^{(-)}, g_j^{(-)}$	coefficient for the element function at node $i$ or $j$ , ( $p$ ) for pressure, ( $u$ ), ( $v$ ) for respective velocities	$\Gamma$	DEM particle boundary
$\mathbf{g}, \mathbf{g}_i, \mathbf{g}_{i-1}$	solution vector (entirety of all $g_i^{(-)}, g_j^{(-)}$ ), for current $i$ or next $i+1$ Newton-Raphson iteration	$\Delta_g$	step vector for one Newton-Raphson iteration
$i, j$	mesh node (index), indexing variables	$\Delta A$	change of $A$ over $\tau$
$J$	Jacobian of the Newton-Raphson, method	$\delta_{\varepsilon}$	Kronecker delta
$k$	aspect ratio $t/b$	$\varepsilon, \zeta$	Cartesian direction (index)
$L_{\dots}$	coefficients of the barycentric coordinate functions	$\eta$	dynamic viscosity
$l_{char}$	characteristic length for DEM particle contact	$\theta_A, \dots, \theta_J$	third order FEM shape functions
$I_1, I_2$	vector $SC_{\dots}$	$i$	Newton-Raphson iteration step (index)
$m_1, m_2$	DEM particle mass	$\lambda$	wall correction factor
$m_{red}$	DEM particle contact reduced mass	$\mu$	friction coefficient
$\hat{n}$	normal direction unit vector	$\nu$	kinematic viscosity
$n - X$	value from $X$ timesteps before (index)	$\rho$	density
$P_{\dots}$	polynomial order of a finite element	$\rho_F, \eta_F$	fluid properties (density, dynamic viscosity)
$p$	normalized pressure	$\rho_p, d_p$	particle properties (density, diameter)
$p_{actual}$	pressure	$\tau$	timestep length
$q$	quadrature point (index)	$\phi_A, \dots, \phi_F$	second order FEM shape functions
$r$	radius	$\phi_i, \phi_j$	second order (or bubble) FEM shape function at node $i$ or $j$
$res$	residual vector for the Newton-Raphson method	$\phi_{p2a}$	second order finite element function with third order bubble
		$\psi_A, \dots, \psi_C$	first order FEM shape functions
		$\psi_i, \psi_j$	first order FEM shape function at node $i$ or $j$
		$\psi_{p1}$	first order finite element function
		$\omega_q$	quadrature weight of $q$
		$\omega_T$	trial function

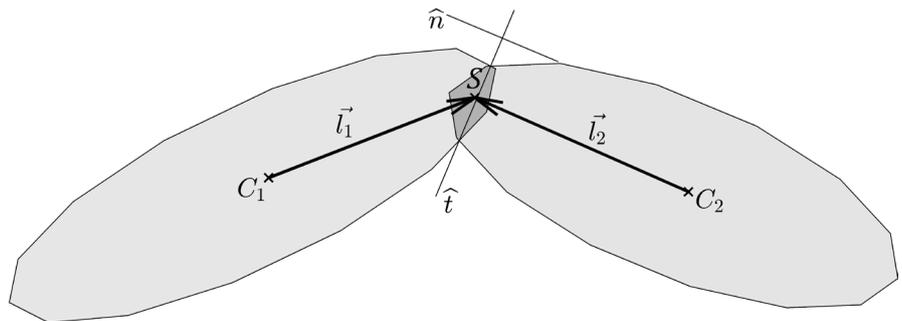
choice for modeling the fluid (Quadrilateral elements would have to be distorted so they offer no advantages in this case). The arbitrary connections of the pore space also require the use of unstructured grids (arbitrary connectivity between elements), though in practice we will generate the grid in the pore space with a constrained Delaunay triangulation, so the connections are limited. Desirable is a formulation with the minimum necessary degrees of freedom, as it is not possible to multiply lattice points near particle boundaries, which would also multiply the computational effort and the memory requirements. Other aspects of the simulation can also be determined as an optimal choice of minimal computational cost, geometrical convenience and the implementation of the actual physical situation.

### 3. Interaction between Discrete-Element Particles

The particles in the discrete element method (DEM) interact via a “hard-particle, soft-contact” interaction (see **Figure 2**): The polygonal particle outlines are moved as rigid particles. The forces act on the centroid  $S$  of the overlap polygon, forces in normal direction are elastic and damping forces, Coulomb friction acts in tangential direction. The magnitude of the elastic force is computed from the overlap area  $A$  and the Young’s modulus  $Y$  (units [N/m] in two dimensions)

$$F_{el} = Y \cdot \frac{A}{l_{char}}, \tag{1}$$

with the characteristic length  $l_{char} = 4|l_1||l_2|/(|l_1|+|l_2|)$  from the vectors  $l_1, l_2$  connecting the particles’ centers of mass  $C_1, C_2$  with  $S$ . This choice of the characteristic length  $l_{char}$  normalizes the sound velocity of a space-filling rectangular packing to the continuum sound velocity with the same Young’s modulus



**Figure 2.** The interaction of particles (centers of mass  $C_1$  and  $C_2$ ) is separated into tangential direction  $\hat{t}$  (given by the intersection of the particle outlines) and normal direction  $\hat{n}$ . The normal force is composed of the elastic force (proportional to the area  $A$  of the overlap polygon in dark shading) and the Young’s modulus  $Y$  as well as the damping force (proportional to the change of the overlap area). The numerical exact implementation of Coulomb friction (proportional to the normal force and the friction coefficient  $\mu$  acts in the tangential direction  $\hat{t}$ . The vectors  $l_1, l_2$  between  $C_1, C_2$  and the centroid  $S$  of the overlap polygon are used to compute the inverse characteristic length  $l_{char}$  as well as the torques on the particles.

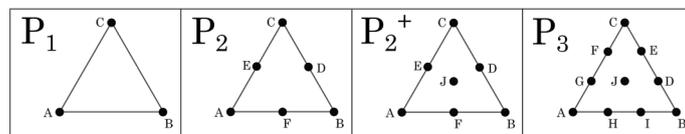
and density. The damping force is chosen in analogy to the linear oscillator

$$F_{da} = \sqrt{m_{red}} \cdot Y \cdot \frac{\Delta A}{\tau} \cdot \frac{1}{l_{char}} \tag{2}$$

proportional to  $\Delta A$ , the area change for an integration timestep  $\tau$ , with the reduced mass of the colliding particles  $m_{red} = m_1 m_2 / (m_1 + m_2)$ . Additionally, there is a cutoff so that  $|F_{da}| < |F_{el}|$ . The solid friction is computed according to Kregel and Matuttis [2], where the static many body friction problem is dealt with “numerically exact” as unilateral constraint problem. The forces between the polygonal DEM-particles are surface forces and are therefore not central, so any torques acting on  $C_1, C_2$  are computed with force arms  $l_1, l_2$  and the sum of the forces (More details about interaction laws can be found in Matuttis and Chen [3]).

### 4. FEM with Cubic Bubble

An unstructured triangular grid for the fluid simulation still leaves choices for the actual finite elements used. Gresho and Sani [4] present a multitude of options with different order polynomials used as shape functions for the pressure and velocity representation, some of which are shown in Figure 3. As stated in Chapter 1, our fluid phase should obey the equations of motion for Newtonian fluids as closely as possible, which requires the use of the Navier-Stokes equation (NSE). This narrows down the choice of elements by the Ladyzhenskaya-Babuška-Brezzi (LBB) stability condition: It stipulates that the order of elements for the velocities must be one order higher than for the pressures, as the NSE also contains the velocities in one higher order than the pressures [5]. Accordingly, the Taylor-Hood element ( $P_2P_1$ ) with second order shape functions  $P_2$  for the velocities and first order shape functions  $P_1$  for the pressures had been used in previous simulations [6], but caused issues in case of high particle densities combined with higher flow velocities [7]. Second order elements (polynomials) model a flow field with constant curvature, while for complex flow between granular particles in close proximity ( $<$ particle diameter) which move at different or even opposing velocities, the curvature of the flow field can vary. To take this into account, at least third order elements are necessary (see Figure 4). Instead increasing the number of elements between particles has not turned out to be practicable: Using e.g. two elements with different constant curvature side by side does not provide a more plausible flow field or increases the stability of the simulation [7]. Using 4 (or 5) elements would increase the number of mesh points



**Figure 3.** Different triangular finite elements, defined by their polynomial order  $P$ . The  $P_2^+$  element is a combination of the second order  $P_2$  element and the bubble function (at node  $J$ ) of the third order  $P_3$  element.

in two dimensions by a factor of 16 (or 25), which means an increase of the computational effort by over one order of magnitude, which is prohibitive. For all of these reasons, in this work, we will be exploring the possibilities of the  $P_2^+P_1$  element, which adds a third order bubble to the second order shape functions (Figure 3) and thus provides a linear variation in curvature without the need for a full third order  $P_3$  element, while still being LBB stable [4]. While the  $P_3$  looks attractive in theory, for our purposes it has no advantages over the  $P_2^+$  element, as we are mostly concerned adapting to the curvature of the flow field. On the contrary, having to deal with 10 instead of 7 nodes per element would increase the computational effort and memory consumption by over 40 percent without leading to foreseeable advantages.

### 4.1. Shape Functions

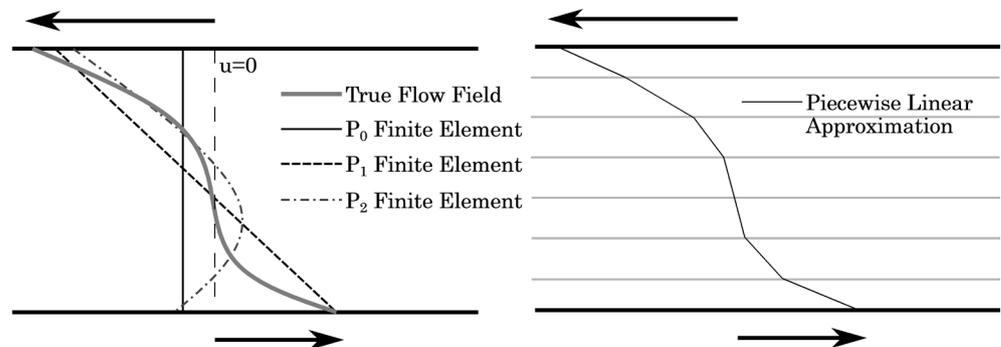
The shape functions are handled in a barycentric coordinate system (local to each element) and must be derived from the Cartesian coordinates of the elements in the grid. Treating every single element separately, we apply a coordinate transformation to the Cartesian coordinates  $[x_1, y_1][x_2, y_2][x_3, y_3]$  of the corners of the triangular element to get the coefficients of its barycentric coordinate functions. This works easiest via left division (MATLAB's backslash-operator)

$$\begin{bmatrix} L_{A0} & L_{B0} & L_{C0} \\ L_{Ax} & L_{Bx} & L_{Cx} \\ L_{Ay} & L_{By} & L_{Cy} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \setminus \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

with the resulting coordinate functions also acting as the first order (affine) shape functions  $\psi_{...}$  so that we have

$$\begin{aligned} \psi_A &= L_{A0} + L_{Ax} \cdot x + L_{Ay} \cdot y \\ \psi_B &= L_{B0} + L_{Bx} \cdot x + L_{By} \cdot y \\ \psi_C &= L_{C0} + L_{Cx} \cdot x + L_{Cy} \cdot y \end{aligned} \tag{4}$$

forming the basis of the first order ( $P_1$ ) finite element. Combined with scaling coefficients  $g_A^{(p)}, \dots, g_C^{(p)}$ , they act as the element function



**Figure 4.** Complex flow with varying curvature between moving particles (upper and lower horizontal thick lines with arrows indicating the direction of motion). The varying curvature cannot be modeled with first and pure second order finite elements (left). A piecewise linear approximation (right) would need at least six to seven elements to properly represent the flow field.

$$\psi_{P1} = g_A^{(p)}\psi_A + g_B^{(p)}\psi_B + g_C^{(p)}\psi_C, \tag{5}$$

which represents the pressure  $p$  field over our element. Combinations of the affine shape functions  $\psi_{...}$  from Equation (4) form the second order (quadratic) shape functions  $\phi_{...}$ . There are six in total with

$$\begin{aligned} \phi_A &= 2 \cdot \psi_A^2 - \psi_A, \\ \phi_D &= 4 \cdot \psi_B \cdot \psi_C \end{aligned} \tag{6}$$

and their respective permutations, forming the basis of the  $P_2$  finite element. To obtain the  $P_2^+$  element, we expand the  $P_2$  basis with the product of all  $\psi_{...}$  from Equation (4)

$$\theta_J = 27 \cdot \psi_A \cdot \psi_B \cdot \psi_C, \tag{7}$$

the third order (cubic) bubble shape function  $\theta_J$ .  $\theta_J$  in combination with all the  $\phi_{...}$  functions from Equation(6) and the coefficients  $g_A^{(u)}, \dots, g_F^{(u)}, g_J^{(u)}$ , acts as the element function

$$\phi_{P2+}^{(u)} = g_A^{(u)}\phi_A + g_B^{(u)}\phi_B + g_C^{(u)}\phi_C + g_D^{(u)}\phi_D + g_E^{(u)}\phi_E + g_F^{(u)}\phi_F + g_J^{(u)}\theta_J. \tag{8}$$

To treat velocity, a vector quantity, as scalar values, we are splitting it into its Cartesian components, horizontal velocity  $u$  and vertical velocity  $v$ . Accordingly, Equation (8) represents only the horizontal velocity field and an analogous equation  $\phi_{P2+}^{(v)}$  for the vertical velocity field exists as well. In other words, a shape function can be seen as some sort of unit vector for the field amplitude at its associated node on a single finite element.

### 4.2. Flow Equations

The coefficients  $g_{...}^{(u,v,p)}$  from Equations (5) and (8) scale the individual shape functions of an element in a way that allows the full element function  $\phi_{P2+}^{(u)}$ ,  $\phi_{P2+}^{(v)}$  or  $\psi_{P1}$  respectively to approximate the actual flow behaviour as closely as possible. The entirety of all  $g_{...}^{(u,v,p)}$  is the unknown (solution vector) we need to solve our flow equations for. To calculate the flow velocities  $u$  and  $v$  we use the incompressible stationary Navier-Stokes equation for  $x$  and  $y$  directions

$$u \frac{du}{dx} + v \frac{du}{dy} = -\frac{dp}{dx} + \nu \left( \frac{d^2u}{dx^2} + \frac{d^2u}{dy^2} \right) \tag{9}$$

and

$$u \frac{dv}{dx} + v \frac{dv}{dy} = -\frac{dp}{dy} + \nu \left( \frac{d^2v}{dx^2} + \frac{d^2v}{dy^2} \right) + a_g. \tag{10}$$

With the kinematic viscosity  $\nu = \eta/\rho$  and the normalised pressure  $p = p_{actual}/\rho$  obtained by normalizing the dynamic viscosity  $\eta$  and the actual pressure  $p_{actual}$  with the density  $\rho$ . While rock has a sound velocity which is a multiple of that of water, the sound has to travel through the contacts between sand grains so there is much less material than in bulk rock. Thus, the sound velocity of the granular phase is considerably below that of the fluid and the fluid

can therefore be treated as incompressible. The gravitational acceleration  $a_g$  is also an essential term in the vertical ( $y$  direction) Navier-Stokes equation, as the fluid simulation will later run with suspended particles, it is  $a_g$  which actually leads to the buoyancy force of the fluid acting on the particles. The pressure is calculated from the continuity equation

$$\frac{du}{dx} + \frac{dv}{dy} = 0. \tag{11}$$

Equations (9)-(11) can now be adapted into their weak form by introducing the arbitrary trial function  $\omega_r$  [8] so we get

$$\iint \left[ \frac{du}{dx} \frac{d\omega_r}{dx} + \frac{dv}{dy} \frac{d\omega_r}{dy} + \frac{1}{\nu} \left( u \frac{du}{dx} + v \frac{dv}{dy} + \frac{dp}{dx} \right) \omega_r \right] = 0 \tag{12}$$

for the horizontal velocity,

$$\iint \left[ \frac{dv}{dx} \frac{d\omega_r}{dx} + \frac{dv}{dy} \frac{d\omega_r}{dy} + \frac{1}{\nu} \left( u \frac{dv}{dx} + v \frac{dv}{dy} + \frac{dp}{dy} \right) \omega_r + \frac{1}{\nu} a_g \cdot \omega_r \right] = 0 \tag{13}$$

for the vertical velocity and

$$\iint \left[ \left( \frac{du}{dx} + \frac{dv}{dy} \right) \omega_r \right] = 0 \tag{14}$$

for the pressure. Instead of enforcing a spurious “=0” in the Equations (12)-(14), it is much more meaningful in a numerical implementation (*i.e.* with rounding and discretization errors) to demand that the right-hand side (the “residual” *res*) may be finite, but should be as small as possible. This transforms the problem from an equality problem (which may be unsolvable) to a much more manageable optimization problem. Then we rewrite the continuum field equations into systems of coupled discrete equations at each node  $i$  and expand the trial function  $\omega_r$  in terms of test functions with unknown prefactors. Moreover, in the Galerkin-FEM method, as test functions the shape functions  $\phi_i$  from Equations (6) and (7) are used for velocity. This means  $\phi_i$  can take the form of any of the six second order shape functions  $\phi_A, \dots, \phi_F$  or of the third order bubble function  $\theta_j$  depending on the position of node  $i$  in the triangle. In the same way the shape functions  $\psi_i$  from Equation (4) are used as test functions for the pressure.

As a rule, a shape function only exists on a single element and is associated to only one node of that element. As a node is usually connected to multiple elements, the combination of all associated shape functions (one for each connected element) is the basis function (in the one-dimensional case this would e.g. be the infamous hat-function) [9]. In our case, the basis function of a node on the edge of a triangle ( $D, E, F$  see **Figure 3**) always consists of two shape functions as the node always connects two elements (except on the boundary). A node on a vertex ( $A, B, C$ ) may connect an arbitrary number of elements (as we have an unstructured grid) and a “bubble”-node ( $J$ ) exists only on a single element.

For the Galerkin method, the same family of functions is used for the basis functions as well as for the test functions. As our further calculations are performed element wise, we can instead use the single shape functions  $\phi_i$  and  $\psi_i$  and obtain the identical results for the summation of the residuals on each node.

Finally, to approximate the flow field and minimize the residual, an unknown prefactor precedes each test function. When discretizing Equations (12)-(14) for all nodes  $i$ , these coefficients must be determined, as they hold the information about the flow state at each individual node. In fact, these are the coefficients  $g_i^{(\dots)}$  from Equations (5) and (8), where (...) stands for the flow variables ( $u$ ), ( $v$ ) and ( $p$ ) at every node  $i$ . This results in  $\omega_T = g_i^{(u)} \cdot \phi_i$  respectively  $\omega_T = g_i^{(v)} \cdot \phi_i$  in Equations (12) and (13) and  $\omega_T = g_i^{(p)} \cdot \psi_i$  in Equation (14) [8]. So by integrating Equations (12)-(14) on all nodes and numerically solving for all  $g_i^{(\dots)}$  we obtain our FEM solution.

As the computational effort for the exact spatial quadrature of Equations (12 - 14) is prohibitive, we resort to approximation via Gauss quadrature. From the many possible approaches, we opted for the six-point quadrature formula by Gockenbach [10] with a degree of precision ( $deg_p = 4$ ), as any methods with a higher degree of precision produced the same results but with considerably increased CPU-time cost [11]. Using Equation (12) to calculate the equilibrium over a single mesh node  $i$  and rewriting it as a sum over all quadrature points  $q$  (with their quadrature weights  $\omega_q$ ) of all connected mesh elements  $e$  (with their area  $A_e$ ), gives as residual

$$res_i^{(u)} = \sum_{e,q} A_e \cdot \omega_q \left[ \frac{du}{dx} \Big|_{q(e)} \frac{d(g_i^{(u)} \cdot \phi_i(q(e)))}{dx} + \frac{du}{dy} \Big|_{q(e)} \frac{d(g_i^{(u)} \cdot \phi_i(q(e)))}{dy} + \frac{1}{v} \left( u(q(e)) \frac{du}{dx} \Big|_{q(e)} + v(q(e)) \frac{du}{dy} \Big|_{q(e)} + \frac{dp}{dx} \Big|_{q(e)} \right) (g_i^{(u)} \cdot \phi_i(q(e))) \right] \tag{15}$$

for the horizontal velocity on that node. The notation  $\Big|_{q(e)}$  indicates that a given derivative is to be evaluated on the position of the current quadrature point  $q$  on the current element  $e$ . Time evolution is performed via the backward-difference formula of second order (BDF2) formulation provided by Gear [12] (see also Gresho & Sani [4]), so an additional term

$$+ \frac{1}{v} \left( \left( \frac{1 + 2 \frac{\tau}{\tau_{n-1}}}{1 + \frac{\tau}{\tau_{n-1}}} \right) u(q(e)) - \left( 1 + \frac{\tau}{\tau_{n-1}} \right) u_{n-1}(q(e)) + \frac{\left( \frac{\tau}{\tau_{n-1}} \right)^2}{1 + \frac{\tau}{\tau_{n-1}}} u_{n-2}(q(e)) \right) \cdot \frac{\tau_{n-1}}{\tau} (g_i^{(u)} \cdot \phi_i(q(e))) \tag{16}$$

is added in the sum of Equation(15). The length of the current timestep is denoted by  $\tau$ , the index  $n - X$  indicates values from previous timesteps. There are certain considerations for the use of the BDF2-solver: While it is A-stable (stiffly stable), so that it can ignore spurious fluctuations from timestep to time-

step (which is an advantage with permanently changing grid geometries), it is also L-stable, which means that it will not damp out instabilities in the flow. This means that BDF2 is not expected to (and, in our experience, does indeed not) delay or suppress the development of vortices in e.g. the flow behind fixed bluff bodies when the flow speed is increased from Stokes flow to Reynolds numbers beyond 10.

The residual of the vertical velocity is calculated analogously to Equations (15) and (16), however the gravity term

$$+\frac{1}{\nu} \cdot a_g \cdot g_i^{(v)} \cdot \phi_i(q(e)) \tag{17}$$

must be added inside the summation. The pressure residual obtained from Equation (14) is accordingly

$$res_i^{(p)} = \sum_{e,q} A_e \cdot \omega_q \left[ \left( \left. \frac{du}{dx} \right|_{q(e)} + \left. \frac{dv}{dy} \right|_{q(e)} \right) \left( g_i^{(p)} \cdot \psi_i(q(e)) \right) \right]. \tag{18}$$

In our incompressible Navier-Stokes equation, the pressures act as constraint forces for the constraint of incompressibility. In our choice of FEM, the pressures are realized as Lagrange parameters for the BDF2 formulation [4]. They can vary non-smoothly between timesteps without additional relaxation or time evolution, as there is no time evolution equation for the pressures which would require a smooth variation. Any nodes on boundaries with prescribed values use the difference between the intended and actual flow parameters as their residual.

### 4.3. Iteration within a Single Timestep

Within a single timestep, Newton-Raphson iteration is used to minimize the residuals. Deriving the residual Equations (15)-(18) for each flow variable gives us the entries for the Jacobian of the Newton-Raphson method [13]. As each equation can be derived by every flow variable ( $u, v, p$ ) except for Equation (18) where the pressure derivative is trivial, this results in a total of eight different equations for Jacobian entries. Within a single element (with surface  $A_e$ ), these equations link the flow states  $u, v$  and  $p$  of two nodes  $i$  and  $j$  together. They again can be evaluated via the same Gockenbach quadrature method as described above (We omit these equations in this section for brevity and list them in **Appendix A** of this article). All these entries are combined into the (sparse) Jacobian  $J$  via MATLAB's sparse command. As we have largely different scales in the velocities and the size of the finite elements, we use a rescaling of the largest line and column elements of the Jacobian as preconditioning [6].

Using the Jacobian and the residual vector  $res$  containing the residuals of all flow states at all nodes, the Newton-Raphson step vector  $\Delta_g$  for all of these flow states can be calculated as

$$\Delta_g = J \setminus res. \tag{19}$$

By applying the Newton-Raphson step to the solution vector  $g$

$$\mathbf{g}_{t+1} = \mathbf{g}_t - \Delta \mathbf{g}, \tag{20}$$

we complete a full Newton-Raphson iteration  $t$ . Depending on the size of the residual, we now continue with another iteration or advance to the next time-step.

As our simulations feature moving granular particles, the geometry will also vary over time. The mesh has to change slightly with every timestep to accommodate the position change of the particles and requires interpolation of the old flow values onto the nodes of the new mesh. Details regarding interpolation of a finite element containing a bubble function are described by Mueller *et al.* [11]. An overview over the grid generation by regular remeshing via constraint Delaunay triangulation at larger intervals and grid relaxation from timestep to timestep can be found in Ng *et al.* [14].

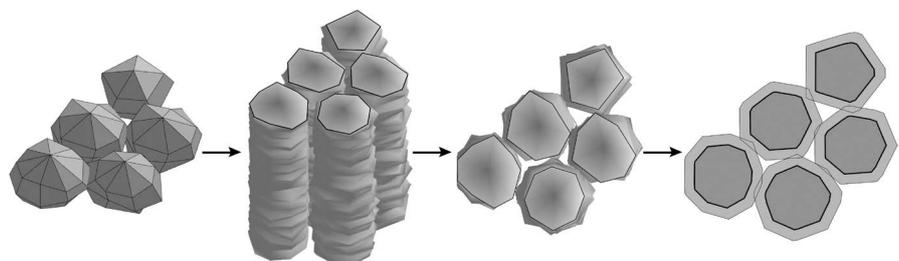
### 5. Coupling between Fluids and Particles

The simplifying step from three-dimensional to two-dimensional particles means that the DEM-particles are effectively rods in three dimensions. When we assume that these rods would have a rough surface as shown in Figure 5, we end up with a connected fluid space, which can be mimicked in two dimensions by using a larger “shadow” for the particle-particle interaction and a smaller particle “core” which is seen as boundary by the fluid. The width of the core can be adapted to simulate different porosities.

To compute the particle-flow interactions based on hydrodynamic principles, we have to apply no-slip boundary conditions at the particle surfaces. Other approaches, such as immersed boundary conditions [15] violate the no-slip condition and will cause unphysical drag forces between particle and flow. In our approach, the actual force of the fluid on the particle is computed as

$$F_{drag} = \int_{\Gamma} \left\{ -p\delta_{\varepsilon\zeta} + \eta \left( (\nabla \mathbf{U}) + (\nabla \mathbf{U})^T \right) \right\} \cdot \hat{\mathbf{n}} dl, \tag{21}$$

where the values for the normalized pressure  $p$  and velocity vector  $\mathbf{U}$  are evaluated on the particle boundary  $\Gamma$  from the values computed with the FEM. Further  $\delta_{\varepsilon\zeta}$  is the Kronecker delta function dependent on the Cartesian directions  $\varepsilon$  and  $\zeta$ ,  $\hat{\mathbf{n}}$  is the normal to  $\Gamma$  and T is the transpose operator.



**Figure 5.** From left to right: Physical three-dimensional particles, three-dimensional rough rods, projection and “averaging” along the z-axis and discrimination into a core region (bold line, dark shading) for the particle-fluid interaction and shadow region (light shading) for the particle-particle interaction so the fluid space in the simulation is connected.

The time evolution via the BDF2 predictor-corrector during a single timestep is the following: At the beginning of a new timestep, the new positions (which determine the new particle outlines and therefore the boundaries of the fluid) are computed in the predictor. The grid is then deformed according to the new particle positions and the flow velocities are interpolated onto the new grid. The possibility of interpolating from one timestep to the next with slightly changing meshes is owed to the stability of the (implicit) BDF integrator: For explicit integrators, the size of the timestep would probably have to be reduced considerably. The reaction forces from the fluid are computed from the configuration obtained from the predictor and the velocities (of the particles and the fluid) at the end of the timestep are determined with the corrector.

## 6. Verification

To verify the physical accuracy of our simulation code, especially in regard to fluid-solid interaction, we selected three test scenarios. These allow us to confirm that our code is not only providing the accuracy which is expected in the field of computational fluid dynamics, but also that it is capable of handling the specific demands posed by the simulation of granular media and that the continuous change of the mesh does not affect the stability or accuracy.

### 6.1. Wall Correction Factor

We recreated the simulation geometry from Richou *et al.* [16] to compare our results of the wall correction factor for a fixed two-dimensional circle (respectively a circular cylinder) between two parallel walls. As our simulation is intended to handle polygonal particles, it is impracticable to aim for an implementation of perfectly circular shapes here. Neither curved element outlines nor extremely small discretization would serve any purpose when our main goal is to verify whether the simulation reproduces reasonably well parameters with finite input accuracy. Thus we decided on a regular dodecagonal particle shape as described in the simulations by Ng [6] as a rough approximation for the cylinder. In the simulations by Richou *et al.*, the main parameters are the Reynolds number

$$Re = \frac{\rho_F \cdot r \cdot U_0}{\eta_F} \quad (22)$$

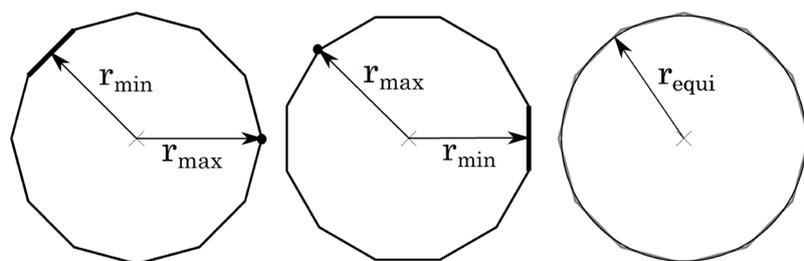
with fluid density  $\rho_F$ , dynamic viscosity  $\eta_F$  and maximum inflow velocity  $U_0$  and the aspect ratio

$$k = r/b \quad (23)$$

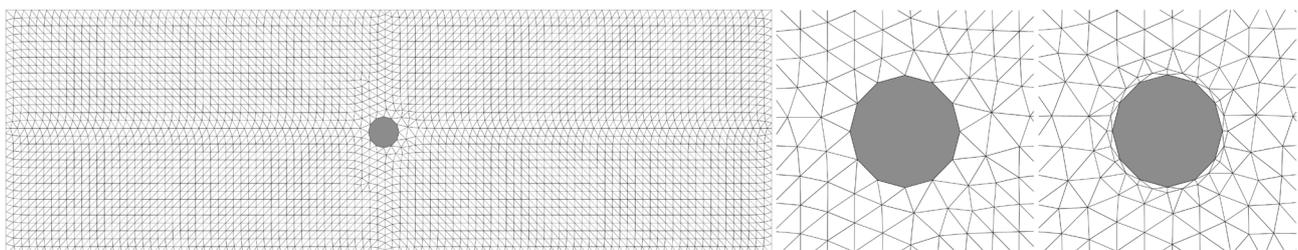
between cylinder size and wall distance  $b$ , which both depend heavily on the cylinder radius  $r$ . As can be seen in **Figure 6**, the radius of a regular dodecagon has visible variation which will lead to different results for Equations (22) and (23) for different particle orientations and effective radii. Accordingly we run our simulations with  $r_{\max} = 15.000 \text{ mm}$ ,  $r_{\min} = r_{\max} \cdot \cos(\pi/12) = 14.489 \text{ mm}$  as well

as with the Sauter radius  $r_{equi} = \sqrt{A/\pi} = 14.658$  mm the radius of a circle with equivalent area  $A = 3 \cdot r_{max}^2$ . In their simulations, Richou *et al.* used a Reynolds number of  $Re = 2 \times 10^{-4}$ . We set the parameters for our fluid to a density of  $\rho_F = 1000$  kg/m<sup>3</sup>, a dynamic viscosity of  $\eta_F = 500$  Pa·s and adjusted the maximum inflow velocity  $U_0$  according to the radius to reproduce the same Reynolds number. The distance from the particle center to the in- and outflow boundaries is 375 mm. As in the work of Richou *et al.*, we use a parabolic velocity profile to define both in- and outflow boundary conditions. The aspect ratio was selected as  $k = 0.125$  with  $b$  again adjusted to match. Further, the dodecagon can face the flow either with a corner “corner on”, or with a side “side on” (see **Figure 6**). As buoyancy was not computed in Richou *et al.* and the particle will not be moving (only the drag forces acting on it will be calculated), gravity was turned off for these simulations.

We have simulated all cases at two different mesh resolutions, first with a thin layer ( $0.1 \cdot r_{max}$ ) of mesh elements around the particle and second with mesh elements of roughly the particle edge length (see **Figure 7**). As **Table 2** shows, the results are in good agreement with the data by Richou *et al.* Deviations due to the various reference radii are to be expected as we do not have a circular particle and they are the strongest factor of influence. Effects due to the orientation are minor, although when a side of the dodecagon is facing the flow, the wall correction factor (and thus the fluid force) is slightly increased. An important finding is that the Mesh resolution, while having an effect, is by far not as pronounced as the impact of different reference radii. So even our coarse mesh (particle size  $\approx 20$  elements) produces reliable results and should allow significantly faster calculations than immersed solid methods where the particle size



**Figure 6.** Different orientations of a dodecagonal particle: “corner on” (left) and “side on” (center), with the different radii  $r_{min}$  and  $r_{max}$  of the dodecagon with the Sauter radius  $r_{equi}$  (right) for comparison.



**Figure 7.** From left to right: Mesh of the full simulation domain, with coarse and fine mesh (see **Table 1.**) around the particle.

**Table 2.** Wall correction factor  $\lambda$  of a dodecagon for different orientations, reference lengths  $r$  and mesh resolutions. Relative error is calculated based on the value given by Richou *et al.*  $\lambda_{ref} = 10.6500$  [16].

$\lambda$	Orientation	$r$	$b$ [mm]	$U_0$ [mm/s]	Mesh	Rel. Error [%]
10.094	corner on	$r_{max}$	120.000	6.667	fine	5.22
10.114	side on	$r_{max}$	120.000	6.667	fine	5.03
10.660	corner on	$r_{min}$	115.911	6.902	fine	0.09
10.694	side on	$r_{min}$	115.911	6.902	fine	0.41
10.474	corner on	$r_{equi}$	117.265	6.822	fine	1.66
10.505	side on	$r_{equi}$	117.265	6.822	fine	1.36
10.168	corner on	$r_{max}$	120.000	6.667	coarse	4.53
10.218	side on	$r_{max}$	120.000	6.667	coarse	4.06
10.693	corner on	$r_{min}$	115.911	6.902	coarse	0.40
10.782	side on	$r_{min}$	115.911	6.902	coarse	0.73
10.557	corner on	$r_{equi}$	117.265	6.822	coarse	0.87
10.593	side on	$r_{equi}$	117.265	6.822	coarse	0.53

often is  $\geq 100$  elements [15]. This means that for our granular simulations in fluids, the simulation results will not be distorted by a large grid size. The particle shape (which in general will follow a polydisperse size and shape distribution) will be the dominant effect. These results prove the physicality of the microscopic model we use for the fluid-solid interaction as the data by Richou *et al.* had been experimentally verified by Taneda [17].

### 6.2. Cavity Flow

For further verification, we recreated the simulation geometry from Kawahara [18] to simulate the flow inside a 1 m by 1 m cavity. The parameters for the fluid density  $\rho_F = 1000 \text{ kg/m}^3$  and dynamic viscosity  $\eta_F = 10 \text{ Pa}\cdot\text{s}$  were adopted from Kawahara. The boundary condition on the top (wall) of the fluid domain was set to a horizontal velocity of  $u_{BCTOP} = 1 \text{ m/s}$ ,  $v_{BCTOP} = 0 \text{ m/s}$  (parallel to the boundary), the remaining three sides were set to a no slip  $u_{BC} = 0 \text{ m/s}$ ,  $v_{BC} = 0 \text{ m/s}$  boundary condition.

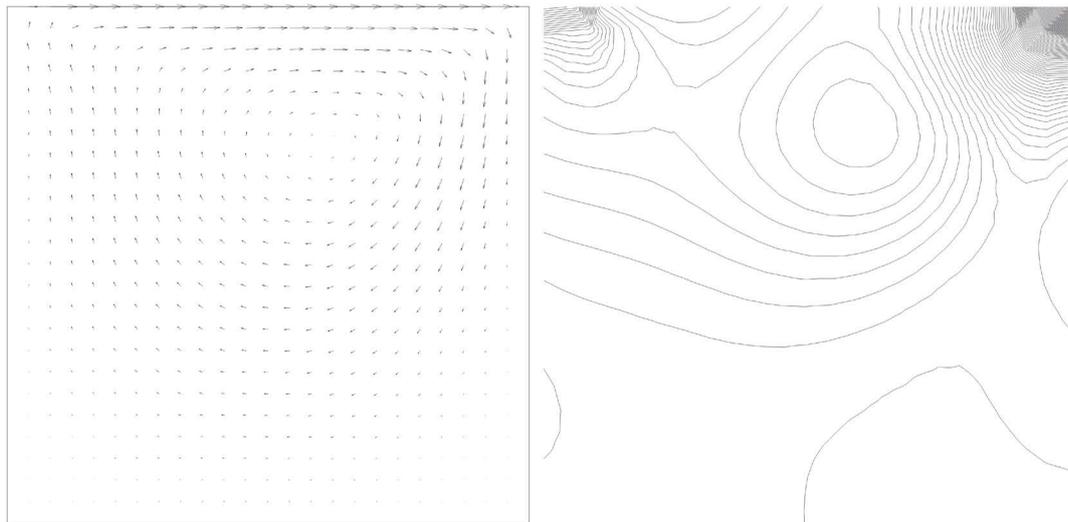
Our results for flow field and pressure distribution shown in **Figure 8** match the results given in [18] for  $P_2b-P_1$  (different notation to  $P_2^+P_1$ ) to the point, where they are virtually indistinguishable. This is an indicator that our simulation code is not only qualitatively correct, but also quantitatively reliable and able produce results which are expected in the field of computational fluid dynamics.

### 6.3. Granular Step

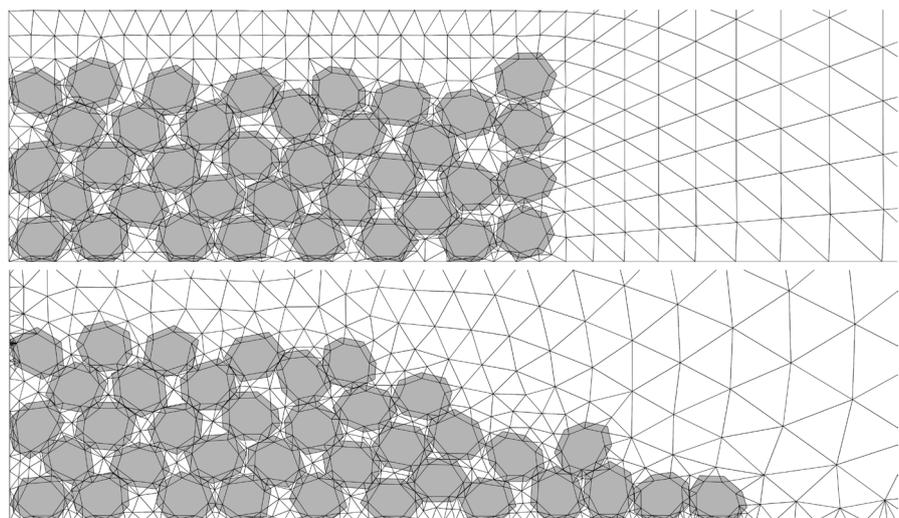
To prove the capabilities of our simulation code concerning the challenges posed by granular media immersed in fluids, we created a simulation of a collapsing granular step inspired by the experiments performed by Rondon *et al.* [19]. The findings by Rondon indicated a two-dimensional behaviour of the collapse

process, which should allow for a valid recreation with our simulation code. The fluid with density  $\rho_F = 1000 \text{ kg/m}^3$  and dynamic viscosity  $\eta_F = 0.023 \text{ Pa}\cdot\text{s}$  is contained in a rectangular box 70 cm wide and 15 cm tall. A rectangular section of 4 cm by 2 cm in the bottom left contains 37 particles as the granular step. The particles are polygonal with six to eight edges, have a density of  $\rho_p = 2500 \text{ kg/m}^3$  and an average “shadow” diameter of  $d_{p,\text{shadow}} \approx 4 \text{ mm}$ . The “core” diameter seen by the FEM is  $d_{p,\text{core}} = 0.75d_{p,\text{shadow}} \approx 3 \text{ mm}$ . To form the granular step, the rightmost column of particles is fixed in place while the remainder of the step is filled by particles falling in via a dry DEM simulation. When the particles have settled, the FEM part of the simulation is added and the movement of the rightmost particles is unlocked.

As seen in **Figure 9**, the simulation is able to compute the collapse without



**Figure 8.** Flow field (left) and pressure (right) inside the cavity. These results match the ones generated by Kawahara [18].



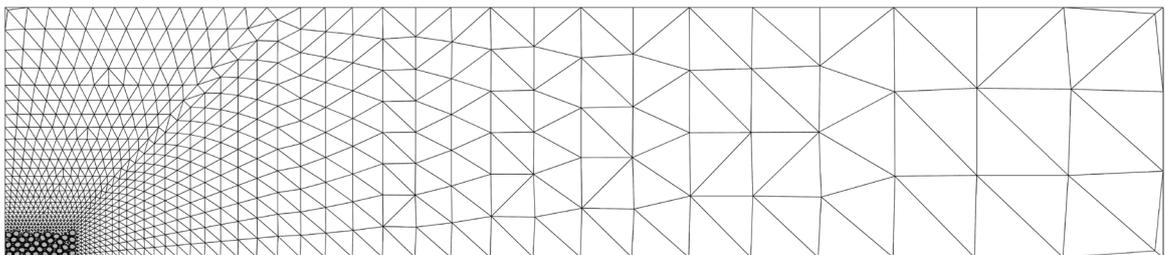
**Figure 9.** Granular step in the initial configuration at  $t = 0 \text{ s}$  (top) and fully collapsed at  $t = 0.35 \text{ s}$  (bottom) when the particles have practically stopped moving.

issue, while the mesh size in the vicinity of the particles is for the most part roughly the same as the particle edge size ( $\approx 1.5$  mm). To avoid additional computational effort and unnecessary degrees of freedom, the mesh is coarsened in the region without particles as seen in **Figure 10**. This grading of the mesh will not have any adverse effects: On one hand, it will retain the original flow field (which would be modified by “cutting off” the fluid domain close to the particles) while on the other hand, the discretization error introduced by the larger elements is compensated by the vanishing of the flow velocity towards the boundaries (Reference runs for a single sinking particle have even shown that the use of a graded mesh gives more accurate results with respect to the symmetry of the trajectory, compared to a fine mesh, because the rounding error due to the conditioning of the Jacobian is reduced [20]).

The particles in our current simulation are still about one order of magnitude bigger than the ones used by Rondon *et al.*, which means we are dealing with a simpler geometry. However this is ideal as a test scenario without sacrificing realism. Now that our simulation code has proven its capabilities the next step is to gradually reduce the particle size (thereby increasing the particle count) to create a physically meaningful two-dimensional equivalent of the experiments by Rondon *et al.* and investigate the effect of the packing density on the decay process and the geometry of the end configuration (similar to the simulations by Ng [21]).

## 7. Summary and Conclusions

We have presented a simulation of DEM-particles in Fluids where the particles are coupled to the fluid with the full physical interaction (no slip) from computational fluid dynamics. The accuracy of the time evolution was verified via the wall correction factor. For the parameter range under investigation, no instabilities develop, even when mesh elements bordering the particles are about  $1/7^{\text{th}}$  of the particle size. This is a result of the added third order bubble, which stabilizes the simulation for coarser meshes while at the same time guaranteeing a smoother velocity field even with large elements. As a result, less degrees of freedom



**Figure 10.** Full simulation domain for the granular step. Top and right wall are placed far away from the particles to avoid influences from their boundary conditions. The mesh coarsens towards these empty boundaries to reduce computation times. The not ideal shape of triangles in the right upper and lower corner is an effect of the mesh relaxation and the number of prescribed grid points near the right wall. Nevertheless, there are no negative numerical ramifications for the solution from the triangle shape, as the velocity values at the boundary are set and the neighboring computed values are few and of small magnitude.

are needed than those necessary for lower order elements, and the stability is improved compared to  $P_2P_1$ -elements [7]. In that respect, our simulation is “optimal”, as it will not be possible to reduce the degrees of freedom for the FEM further without jeopardizing the stability. As the computational effort is proportional to the number of degrees of freedom, the speed (for a given number of granular particles in a given volume region) will also be optimal.

Neither the accuracy nor the stability is negatively affected by the movement of the particles or the resulting subtle distortions of the mesh in every timestep; the particles come to a physical rest with a physical angle of repose and without any residual noise amplitude. With the current code, we can now move towards larger system sizes.

There remain some “do’s and don’ts” concerning the combination of remeshing, graded mesh and finite elements with bubble functions but a discussion of this would lead too far here and will be addressed in a further publication.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Sano, O. (2011) Flow-Induced Waterway in a Heterogeneous Granular Material. *Computer Physics Communications*, **182**, 1870-1874. <https://doi.org/10.1016/j.cpc.2010.12.001>
- [2] Kregel, D. and Matuttis, H.-G. (2018) Implementation of Static Friction for Many-Body Problems in Two Dimensions. *Journal of the Physical Society of Japan*, **87**, Article ID: 124402. <https://doi.org/10.7566/JPSJ.87.124402>
- [3] Matuttis, H.-G. and Chen, J. (2014) Understanding the Discrete Element Method: Simulation of Non-Spherical Particles for Granular and Multi-Body Systems. John Wiley and Sons, Hoboken. <https://doi.org/10.1002/9781118567210>
- [4] Gresho, P.M. and Sani, R.L. (2000) Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow. John Wiley and Sons, Hoboken.
- [5] Boffi, D., Brezzi, F. and Fortin, M. (2013) Mixed Finite Element Methods and Applications. Springer, Berlin. <https://doi.org/10.1007/978-3-642-36519-5>
- [6] Ng, S.H. (2015) Two-Phase Dynamics of Granular Particles in a Newtonian Fluid. Ph.D. Thesis, The University of Electro-Communications, Tokyo.
- [7] Morikawa, F. (2015) Simulation of Particles in Fluids Using Finite Elements. Master Thesis, the University of Electro-Communications, Tokyo. (In Japanese)
- [8] Kwon, Y.W. and Bang, H. (2000) The Finite Element Method Using MATLAB. Second Edition, CRC Press, Boca Raton.
- [9] Danaila, I., Joly, P., Kaber, S.M. and Postel, M. (2007) An Introduction to Scientific Computing, Twelve Computational Projects Solved with MATLAB. Springer, Berlin. <https://doi.org/10.1007/978-0-387-49159-2>
- [10] Gockenbach, M.S. (2006) Understanding and Implementing the Finite Element Method. Society for Industrial and Applied Mathematics, Philadelphia. <https://doi.org/10.1137/1.9780898717846>

- [11] Mueller, J. and Matuttis, H.-G. (2019) Expanding P2P1 FEM Elements with Third Order Bubbles for Simulating Many Particles in Fluids. *Proceedings of the Japan Society of Fluid Mechanics Annual Meeting 2019*, Tokyo, 14 September 2019, Article ID: 122.
- [12] Gear, C.W. (1971) *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall, Upper Saddle River.
- [13] Burkardt, J. (2013) Finite Element Treatment of the Navier Stokes Equations, Part IV, Solving the Discretized Navier Stokes Equations. Technical Report, Florida State University, Tallahassee.
- [14] Ng, S.H. and Matuttis, H.-G. (2011) Adaptive Mesh Generation for Two-Dimensional Simulation of Polygonal Particles in Fluid. *Theoretical and Applied Mechanics Japan*, **59**, 323-333.
- [15] Kajishima, T. (2019) Immersed Boundary/Solid Method for the Numerical Simulation of Particle-Laden Flows. *Fluid Dynamics Research*, **51**, Article ID: 051401. <https://doi.org/10.1088/1873-7005/ab27e7>
- [16] Richou, A.B., Ambari, A. and Naciri, J.K. (2004) Drag Force on a Circular Cylinder Midway between Two Parallel Plates at Very Low Reynolds Numbers Part 1: Poiseuille Flow (Numerical). *Chemical Engineering Science*, **59**, 3215-3222. <https://doi.org/10.1016/j.ces.2003.10.031>
- [17] Taneda, S. (1963) Experimental Investigation of the Wall Effect on a Cylindrical Obstacle Moving in a Viscous Fluid at Low Reynolds Numbers. *Journal of the Physical Society of Japan*, **19**, 1024-1030. <https://doi.org/10.1143/JPSJ.19.1024>
- [18] Kawahara, M. (1999) Chapter 4 Finite Element Method. In: Yasuhara, M. and Dai-guuji, H., Eds., *Numerical Fluid Dynamics, Basics and Applications*, University of Tokyo Press, Tokyo, 83-103. (In Japanese)
- [19] Rondon, L., Pouliquen, O. and Aussillous, P. (2011) Granular Collapse in a Fluid: Role of the Initial Volume Fraction. *Physics of Fluids*, **23**, Article ID: 073301. <https://doi.org/10.1063/1.3594200>
- [20] Mueller, J., Izumi, I. and Matuttis, H.-G. (2018) Graded FEM-Meshes for Improved Accuracy and Performance. *Proceedings of Dai-Sanjuunikai Suuchiryuutairikigaku Symposium, 32nd Numerical Fluid Dynamics Symposium of the Japan Society of Fluid Mechanics*, Tokyo, 13 December 2018, Article ID: E10-2.
- [21] Ng, S.H. and Matuttis, H.-G. (2013) Polygonal Particles in Fluids. *AIP Conference Proceedings*, **1542**, 1138. <https://doi.org/10.1063/1.4812137>

### Appendix A—Entries for the Jacobian

On each node  $i$  of every  $P_2^+P_1$  finite element  $e$ , the flow Equations (15)-(18) are derived by every flow state  $u_j, v_j = g_j^{(u,v)} \cdot \phi_j$  or  $p_j = g_j^{(p)} \cdot \psi_j$  at every node  $j$  of the same element [13]. By calculating the derivative only at each quadrature point  $q$  and summing them up with respect to the element area  $A_e$  and the quadrature weights  $\omega_q$ , we are able to integrate (via Gauss quadrature [10]) over the element and obtain the following equations for the Jacobian entries:

Horizontal velocity (Equation (15)) derived by horizontal velocity  $u$

$$\begin{aligned} &entry_{u(j),u(i)} \\ &= \sum_q A_e \cdot \omega_q \left[ \frac{d(g_j^{(u)} \cdot \phi_j(q))}{dx} \cdot \frac{d(g_i^{(u)} \cdot \phi_i(q))}{dx} + \frac{d(g_j^{(u)} \cdot \phi_j(q))}{dy} \cdot \frac{d(g_i^{(u)} \cdot \phi_i(q))}{dy} \right. \\ &\quad \left. + \frac{1}{v} \left( g_j^{(u)} \cdot \phi_j(q) \cdot \frac{du}{dx} \Big|_q + u(q) \frac{d(g_j^{(u)} \cdot \phi_j(q))}{dx} + v(q) \frac{d(g_j^{(u)} \cdot \phi_j(q))}{dy} \right) (g_i^{(u)} \cdot \phi_i(q)) \right] \end{aligned} \tag{A1}$$

derived by vertical velocity  $v$

$$entry_{v(j),u(i)} = \sum_q A_e \cdot \omega_q \left( \frac{1}{v} \cdot g_j^{(v)} \cdot \phi_j(q) \cdot \frac{du}{dy} \Big|_q \cdot g_i^{(u)} \cdot \phi_i(q) \right) \tag{A2}$$

and derived by pressure  $p$

$$entry_{p(j),u(i)} = \sum_q A_e \cdot \omega_q \left( \frac{1}{v} \frac{d(g_j^{(p)} \cdot \psi_j(q))}{dx} \cdot g_i^{(u)} \cdot \phi_i(q) \right). \tag{A3}$$

Vertical velocity (vertical analogous to Equation (15)) derived by horizontal velocity  $u$

$$entry_{u(j),v(i)} = \sum_q A_e \cdot \omega_q \left( \frac{1}{v} \cdot g_j^{(u)} \cdot \phi_j(q) \cdot \frac{du}{dx} \Big|_q \cdot g_i^{(v)} \cdot \phi_i(q) \right), \tag{A4}$$

derived by vertical velocity  $v$

$$\begin{aligned} &entry_{v(j),v(i)} \\ &= \sum_q A_e \cdot \omega_q \left[ \frac{d(g_j^{(v)} \cdot \phi_j(q))}{dx} \cdot \frac{d(g_i^{(v)} \cdot \phi_i(q))}{dx} + \frac{d(g_j^{(v)} \cdot \phi_j(q))}{dy} \cdot \frac{d(g_i^{(v)} \cdot \phi_i(q))}{dy} \right. \\ &\quad \left. + \frac{1}{v} \left( g_j^{(v)} \cdot \phi_j(q) \cdot \frac{dv}{dy} \Big|_q + u(q) \frac{d(g_j^{(v)} \cdot \phi_j(q))}{dx} + v(q) \frac{d(g_j^{(v)} \cdot \phi_j(q))}{dy} \right) (g_i^{(v)} \cdot \phi_i(q)) \right] \end{aligned} \tag{A5}$$

and derived by pressure  $p$

$$entry_{p(j),v(i)} = \sum_q A_e \cdot \omega_q \left( \frac{1}{v} \frac{d(g_j^{(p)} \cdot \psi_j(q))}{dy} \cdot g_i^{(v)} \cdot \phi_i(q) \right). \tag{A6}$$

Note that the gravity term of Equation (17) vanishes as it is not dependent on

any flow state. Pressure (Equation (18)) derived by horizontal velocity  $u$

$$entry_{u(j),p(i)} = \sum_q A_e \cdot \omega_q \left( \frac{d(g_j^{(u)} \cdot \phi_j(q))}{dx} \cdot g_i^{(p)} \cdot \psi_i(q) \right) \tag{A7}$$

and derived by vertical velocity  $v$

$$entry_{v(j),p(i)} = \sum_q A_e \cdot \omega_q \left( \frac{d(g_j^{(v)} \cdot \phi_j(q))}{dy} \cdot g_i^{(p)} \cdot \psi_i(q) \right). \tag{A8}$$

Equation (18) does not have any pressure dependencies of its own, so it vanishes when derived by pressure  $p$ . Note that any of the above equations involving pressure is only evaluated for nodes  $i, j$  which are part of the  $P_1$  element.

Time evolution via BDF2 again requires an additional term which is Equation (16) derived by  $u$  for Equation (A1) or the vertical analogous of Equation (16) derived by  $v$  for Equation (A5)

$$+\frac{1}{v} \left( \left( \frac{1 + 2 \frac{\tau}{\tau_{n-1}}}{\tau \cdot \left( 1 + \frac{\tau}{\tau_{n-1}} \right)} \right) g_j \cdot \phi_j(q) \right) g_i \cdot \phi_j(q). \tag{A9}$$

As the BDF2 formulation is only dependent on  $u$  or  $v$  respectively, the component vanishes for all other derivatives.