Graded FEM-Meshes for Improved Accuracy and Performance

○ Jan Mueller, The University of Electro Communications, 1-5-1 Chofugaoka Chofu-shi Tokyo,
Email: jan@matuttis.mce.uec.ac.jp
Ichirou Izumi, The University of Electro Communications, 1-5-1 Chofugaoka Chofu-shi Tokyo,
E-mail: IchiroIzumi@matuttis.mce.uec.ac.jp
Hans-Georg Matuttis, The University of Electro Communications, 1-5-1 Chofugaoka Chofu-shi Tokyo,
E-mail: hg@mce.uec.ac.jp

In computer simulations, a higher spatial or temporal resolution is usually associated with higher accuracy. While it is well known that higher resolution comes at the cost of more numerical operations and therefore longer computation times, it is often overlooked that this increase in numerical operations also incurs more rounding errors. In this research, we simulate granular particles (modeled by a polygonal discrete element method - DEM) in an incompressible Newtonian fluid (modeled by finite element method - FEM). The simulations with graded FEM-meshes towards the particle free regions provide both better performance as well as higher accuracy compared to a fine, constant mesh size.

1. Introduction

Granular materials in a fluid medium give rise to many phenomena like sedimentation, settling or liquefaction. While these phenomena can be observed in nature and are easily reproduced in laboratory experiments, the microscopic interactions between fluid and granular particles are still not fully understood. By modelling the granular particles via discrete element method (DEM) and combining it with a finite element method (FEM) for the fluid part, our MATLAB simulation can deal with these interactions at a microscopic level.

For the time being, we limit ourselves to two-dimensional simulations. First of all, for 3D granular systems with a symmetry (either rotational or translational), the simulation of 2D cross sections yielded physically meaningful results in the past. Furthermore, for steadfast progress, an understanding of the two-dimensional situation is a prerequisite for three-dimensional simulations in the field of granular materials. We have witnessed how several groups, starting right away with writing three-dimensional simulation codes, either discontinued their efforts very soon or completed only partially useful codes with limited applicability.

2. The Need for a Graded Mesh

For many systems of particles in fluids, the boundaries are far away from the actual region of interest. Accordingly, a sizeable part of the domain is filled with fluid, like in the experiments by Rondon et al. on the collapse of a granular step (1) (as sketched in Figure 1). A numerical simulation of such geometries with conventional uniform spatial discretization would require a considerable part of the CPU time for the solution of the fluid equations. So not the number of granular particles in the region of interest becomes the limiting factor, but the computational effort spent on the buffer spaces towards the boundaries where nothing relevant happens at all. As another negative side effect, redundant equations for a fluid flow of vanishing magnitude may become the dominant source for the numerical error of the simulation (see Figure 2). While the growth of these errors can be influenced by the choice of the discretization elements, one should stay close to the "sweet spot" between high discretization- and rounding error to achieve optimal performance and accuracy.

When one tries to recreate an actual experimental setup as seen in Figure 1 for a simulation, the original dimensions and boundary conditions should match, even when the experiments have not been designed with numerical recreation in mind. Scaling down the domain would change the behaviour of the system or inhibit particle movement, limiting the comparability.

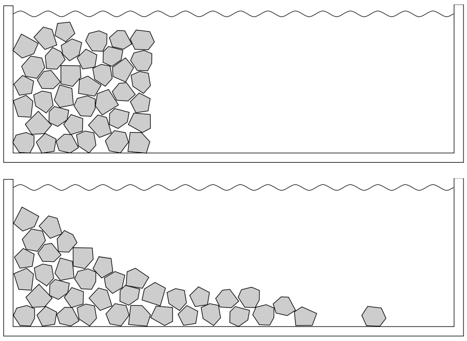

Fig. 1: Sketch of the two-dimensional analogon of a collapsing granular heap in fluid, similar to the experiments conducted by Rondon et al. Particle size is exaggerated.

Flow velocities and their gradients towards remote, fixed boundaries will be negligibly small. Consequentially, even if a coarser grid is used in these areas, the effect on the simulation accuracy will be negligible as well. This motivates the implementation of a graded mesh that coarsens towards the boundaries, to improve performance and allow the simulation of a larger number of particles in the actual region of interest.

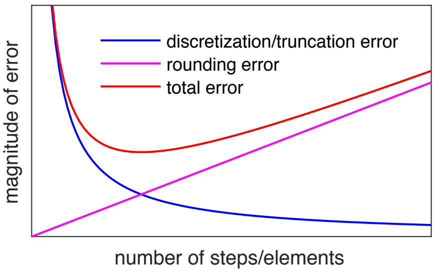

Fig. 2: Qualitative scaling of the error for a numerical simulation. Computational effort scales with the number of time steps and/or mesh elements

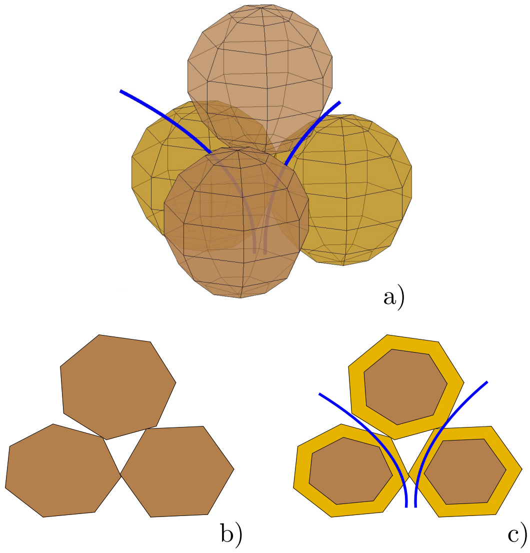

Fig. 3: Three-dimensional particles with connected pore space and fluid flow (blue lines)(a), secluded pore space between two-dimensional particles (b) and connected pore space between our two-dimensional DEM particles with interaction via "shadows" and fluid boundaries given by the "core" so fluid is allowed to flow (c).

3. Discretization of the Fluid Domain

Our FEM code (2) is tuned for coupled use with our DEM simulation. Consequently, it is based on approaches which may be unfamiliar to users of general purpose CFD software. In the following sections we will outline the characteristics of our approach.

The focus of our simulations is often the behaviour of granular slopes in fluids. While spherical particles are easy to model in DEM simulations, they cannot form heaps: Any slope would disintegrate, as the bottom particles would simply roll away, driven by the weight of the particles above. For spherical or other curved outlines, there is additionally the problem of modelling the adjacent fluid boundary without computational overhead on the one hand and with sufficient accuracy on the other hand. Therefore, the particles in our DEM-Simulation are modeled as convex irregular polygons, which also allows the most flexible approach for a vast range of physical materials.

Physically, in three dimensions, the (fluid filled) pore space between the particles is interconnected, allowing fluid to flow (see Figure 3a). But as the simulation is two-dimensional, each pore space in the granular bulk would be isolated (see Figure 3b). Therefore, the particle boundaries for the particle-particle interaction are not simultaneously treated as as the boundaries of the fluid. Instead, a smaller "core" outline is constructed inside each particle. The fluid only interacts with the cores of the particles, while the DEM continues to use the larger "shadow". In this way, we ensure the pore spaces always remain connected and the flow is not blocked (see Figure 3c). Outer wall boundaries have identical outlines for FEM and DEM.

As particles move, the mesh points on their outline change position. The flow in the shifted geometry can be obtained via interpolation, which is readily available from the space filling FEM discretization. In contrast, for finite difference methods, not even oblique edges could be implemented as boundaries in a mathematically exact way.

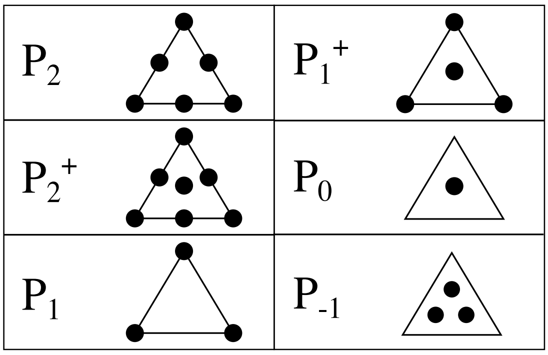

Fig. 4: Types of triangular finite elements. The abbreviation for the polynomial order of the base functions is given on the left and the location of the nodes inside the element on the right.

The pore space between the polygonal particles can only be discretised exactly with unstructured triangular grids. Any discretization with rectangles would lead to zero order errors, so the resulting accuracy of the simulation would also be of order zero. Furthermore, the flexibility of an unstructured triangular grid makes the integration into a graded mesh approach much easier as well.

With the mesh structure set, the actual finite element functions must be chosen. Gresho and Sani (3) list a wide variety of triangular finite elements. Figure 4 shows the basic element functions depending on the order of the polynomials used as base functions. To obtain the force of the fluid on the particles, pressure nodes on the particle surfaces (i.e. the fluid boundaries) are necessary. Accordingly, this requires nodes on either the edges or corners of the FEM-triangle, where the pressure is defined. This excludes the use of P_0 or P_{-1} elements, so the minimum order is P_1 (first order / "linear" or better "affine"). The Ladyzhenskaya-Babuska-Brezzi (LBB) condition requires that the polynomials for the velocities are one order higher than for the pressures (due to the order in the governing equations of the flow) to avoid stability problems. The minimal choice then is the Taylor-Hood element (P_2P_1), combining P_2 (second order / quadratic) velocity base functions with first order pressure base functions in the simplest of ways, consequently allowing somewhat straightforward discretization of the Navier-Stokes- and continuity equations and implementation into the code.

4. Solver and Fluid-Particle-Interaction

As phenomena like liquefaction are transient processes, our simulation needs to be time dependent. Time integration is performed via the implicit backward differentiation formula of second order (BDF2) for both FEM- and DEM parts. BDF2 is A-stable, which allows the choice of a large time step, as well as L-Stable, so perturbations are not damped out numerically (4). As an implicit solver, BDF2 is also not affected by the von-Neumann stability criterion, so the time step is not limited by mesh size (5).

BDF2 is implemented as a predictor-corrector scheme, where at the beginning of a time step, an initial solution (prediction) is calculated and later corrected. The nonlinear equations for the corrector after Gresho and Sani (6)

$$\begin{bmatrix} \frac{1+2\Delta t_n/\Delta t_{n-1}}{\Delta t_n(1+\Delta t_n)/\Delta t_{n-1})} M + K + N(u_{n+1}) & C \\ C^T & 0 \end{bmatrix} \begin{pmatrix} u_{n+1} \\ P_{n+1} \end{pmatrix} = \begin{pmatrix} M \left[\frac{1+\delta t_n/\Delta t_{n-1}}{\Delta t_n} u_n - \frac{(\delta t_n/\Delta t_{n-1})^2}{\delta t_n(1+\delta t_n/\Delta t_{n-1})} \right] + f_{n+1} \\ g_{n+1} \end{pmatrix} \tag{1}$$

are solved via Newton-Raphson iteration. The structure of this equation corresponds to the index-1-formulation of constraint equations (7)

$$\begin{bmatrix} M(q) & G^T(q) \\ G(q) & 0 \end{bmatrix} \begin{pmatrix} u' \\ \lambda \end{pmatrix} = \begin{pmatrix} f(q,u)' \\ -g_{qq}(q)(u,u) \end{pmatrix} \tag{2}$$

with the pressures P_{n+1} as Lagrange parameters λ.

Accordingly, the predictor step is computed under the assumption of constant pressures and forces. The velocities at the particle surfaces are taken directly from the DEM solution as these appear to the FEM as boundary conditions. On the circumference of the particles, the fluid velocity is set to the velocity of the corresponding material points due to linear and angular motion of the particle. Then forces and pressures are computed based on the values of the kinematic variables outputted from the predictor. Based on the deviation of the forces from the previous timestep, the correction is applied. The advantage of the formulation in Equation 1 is that pressures are computed as Lagrange parameters. This allows for a non-smooth variation of the pressures between time steps, which would be devestating for a time integrator.

The fluid forces F^D excerted on the boundary Γ of the DEM-Particles follow from the integration of the fluid pressure and velocity gradient (friction) over the particle surface (8)

$$F^D = \int_\Gamma \Big\{ -p\delta_{ij} + \eta_f \left((\nabla u) + (\nabla u)^T \right) \Big\} \hat{n}\, dl \tag{3}$$

with p as the pressure, δ_{ij} as the Kronecker delta, η_f as the dynamic Viscosity, ∇u as the velocity gradient, and $\hat{n}$ as the normal vector along the boundary Γ pointing into the fluid domain. F^D, together with DEM-internal forces (e.g. gravity, contact forces), gives the total force on a particle.

When the particles are moving, the mesh points on the particle boundaries also move. In regular intervalls, a mesh relaxation step and a new Delaunay triangulation are performed to avoid the formation of bad mesh elements. Each time the flow field on the new points is interpolated from the function values at the points of the previous time step. The computational cost for remeshing is negligible compared to the fluid solver routines.

5. Implementing the Graded Mesh

A graded mesh needs to coarsen towards regions and boundaries with negligible flow velocities in such a way that the discretization error even for large cells remains negligible. In the simplest case of a circular particle inside a rectangular domain, the particle would first be discretized as a regular polygon with the usual (fine) mesh resolution. Then, concentric circles are fitted around this polygon with increasing radii according to

$$r_{i+1} = r_i + \Delta l \frac{r_i}{r_0} \tag{4}$$

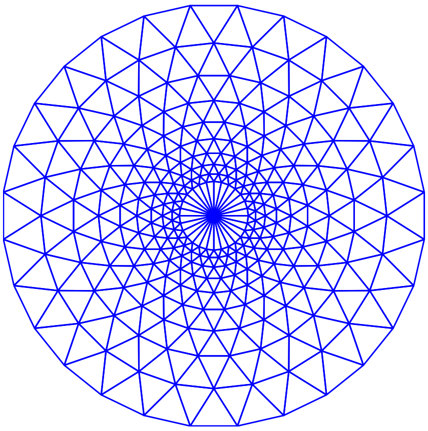

Fig. 5: Graded mesh around a circular particle. The particle fills the innermost circle so the triangles towards the center are deleted in the computational grid.

with r_i being the current radius, r_0 the radius of the circular particle, and Δl the mesh size on the particle surface. Each mesh node on the particle surface (regular polygon) is then projected outwards onto the other circles and shifted by $1/2 \cdot (r_{i+1} - r_i)$ for every second circle to create a staggered structure. Finally, the constrained Delaunay algorithm from MATLAB is used to connect the nodes into triangles and to generate the mesh seen in Figure 5. This method allows the triangular mesh cells to keep nearly equilateral shape, ensuring good mesh quality.

For use with rectangular domains, the circular mesh is truncated at the boundaries. Nodes close to the boundaries are projected onto the domain outline, and any degenerate triangle with an aspect ratio outside $0.5 > \epsilon > 2$ is removed by fusing them with neighbouring elements (see Figure 6).

Afterwards, an algorithm is applied to relax the triangles into a more equilateral shape. All edges are treated as springs (loaded towards equilateral shape) which deform towards a configuration with smaller tension for each spring in each triangle (for details see (9)). Imbalances in the aspect ratio can be reduced further by a subsequent Delaunay triangulation which reconfigures edges towards nearer grid points.

The method can be used in a similar way for agglomerates of multiple particles. Here a circular outline is fitted around the particle bulk and the graded mesh is constructed on the outside of this circle. On the inside, the fine (body-fitted) mesh is used. Instead of a full circle, circular sectors can be used if they allow a better fit to the geometry. Based on Equation 4, the automatic increase in scale σ for triangles in each circle is

$$\sigma = r_0 \frac{1 + \Delta l}{1} \tag{5}$$

resulting in values of $\sigma \approx 1.4$ for single particles and up to $\sigma = 1.2$ for particle groups.

6. Behaviour of a Single Particle

To verify our graded mesh approach, we compared the sinking of a single particle in a rectangular fluid domain both with a fine mesh and a graded mesh.

The geometry features a 3 × 11 mm fluid domain bounded by 0.5 mm wide immobile DEM-walls on each side. A circular DEM-particle of 1 mm diameter in the center of the fluid domain moves under the influence of

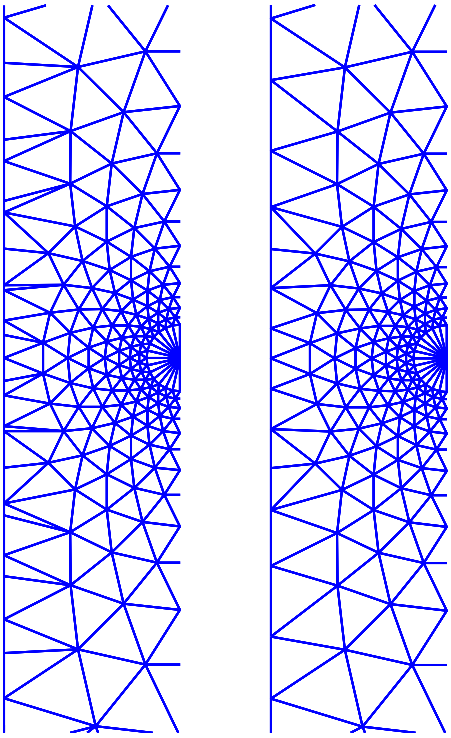

Fig. 6: Elimination of degenerate elements at the domain boundaries. Mesh before (left) and after (right) merging of elements.

gravity and fluid forces. The particle has a density of $\rho_p = 2650\,\mathrm{kg/m^3}$ (soil) and the fluid is treated as water ($\rho_f = 1000\,\mathrm{kg/m^3}$, $\eta_f = 1 \cdot 10^{-3}\,\mathrm{Pa{\cdot}s}$) with the dynamic viscosity η. The flow field and particle are initialized with velocities set to zero.

Equidistant (fine) meshing results in 8914 nodes. The graded mesh reduces the number of nodes by $\approx 88\,\%$ to only 1034 nodes, without any change of the point resolution at the particle surface (see Figure 7). A timespan of $1 \cdot 10^{-3}\,\mathrm{s}$ (1 ms) real time is simulated on a standard PC (Mac Mini 2012, Core i7 2.3 GHz, 4 GB DDR3 1600 MHz) and computation times and particle behaviour are written out for evaluation.

The performance increase due to the graded mesh is immediately apparent as the computational time is reduced by $\approx 82\,\%$ as shown in Table 1.

Mesh Type	Computation Time
fine (old) mesh	$7.4378 \cdot 10^3\,\mathrm{s}$
graded (new) mesh	$1.3170 \cdot 10^3\,\mathrm{s}$

Tab. 1: Computation time for a single particle in a rectangular fluid domain with different mesh types.

A look at the velocity of the sinking particle in y-direction (vertical) shows no relevant deviation between mesh types (see Figure 8), proving that use of the graded mesh does not affect simulation accuracy in any negative way. The sinking velocity increases due to gravity until it nears a maximum value, when gravitational force and fluid forces approach equilibrium.

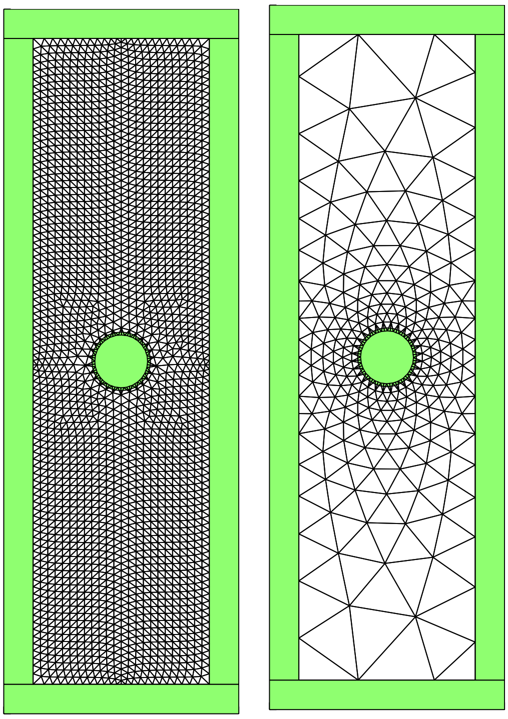

Fig. 7: Simulation geometry with fine (left) and graded mesh (right).

The velocity in x-direction (horizontal) however shows a clear deviation between mesh types (see Figure 9). As the geometry is perfectly symmetric along the vertical, initial velocities for the flow field and particle are zero, and there are no forces applied in x-direction, the particle movement should not have any horizontal component at all. Examining the order of magnitude for the horizontal velocity component (Figure 9) shows that it is negligibly small ($10^{-6}\,\mathrm{m/s}$) compared to the vertical one (Figure 9, $10^{-3}\,\mathrm{m/s}$). That the error is finite indicates numerical rounding errors mentioned in Section 2.. However, the deviation from pure vertical movement for the graded mesh is reduced to about $\approx 1/5$, compared to the fine one. This shows that the graded mesh approach brings the simulation closer to the "sweet spot" for minimal numerical error shown in Figure 2 while reducing the computational effort.

Deviations in the horizontal velocities are also reflected in the particle trajectory (see Figure 10). The y-movement (sinking) is basically independent of the mesh, while the drift in x-direction for the equidistant (fine) mesh is about 10 times greater than for the graded one.

A look at the vertical acceleration in Figure 11 shows very similar particle behaviour for both mesh types where the acceleration is quickly reduced due do fluid forces. However, the graph for the graded mesh shows a slight wiggle around the mean acceleration curve. This wiggle is due to slight discretization errors that are not dampened out due to L-stability of the solver. But as BDF2 is an implicit solver, the inaccuracies are not exaggerated and are averaged out to form smooth velocity and position curves.

Finally the acceleration in horizontal direction (which should be zero) shows the same wiggle as well (see Fig-

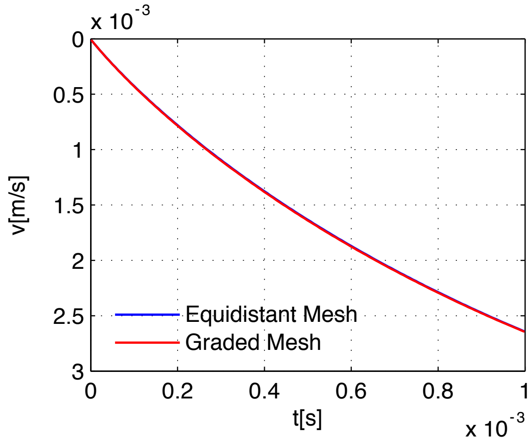

Fig. 8: Vertical velocity of the sinking particle for the equidistant (blue) and graded (red) mesh.

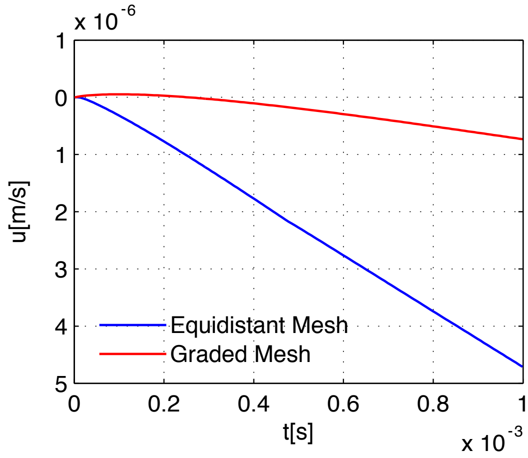

Fig. 9: Horizontal velocity of the sinking particle for the equidistant (blue) and graded (red) mesh.

ure 12). As the density of the sinking particle is rather close to that of the surrounding fluid, the relative error in the horizontal acceleration oscillates around numerically zero and is rather large for both meshes. For the physically measurable velocity in Figure 9, one sees that these fluctuations are averaged out. As no gravitational acceleration acts in the x-direction, the wiggle is now visible even for the fine mesh, although not as pronounced. However the variation for the graded mesh stays nearly centerd around $0\,\mathrm{m/s^2}$, while the mean value for the fine mesh drifts off to negative values due to rounding errors as stated above.

7. Multi Particle Configuration

The graded mesh was developed with many-particle applications in mind. In particular for sedimentation, there are ususally spacious free flow regions. For the geometry of the experiment by Rondon et al.(1), we have joined a body-fixed fine mesh surrounding $n \times n$ particles with a sectorial cutout of the previous graded mesh constructed via concentric circles (see Figure 13). Settings for materials are unchanged from the simulation in Section 6. and the simulation is run for 0.16 s. Figure 14 shows how the particles have sunken to the bottom of the fluid domain and are reordering themselves to form

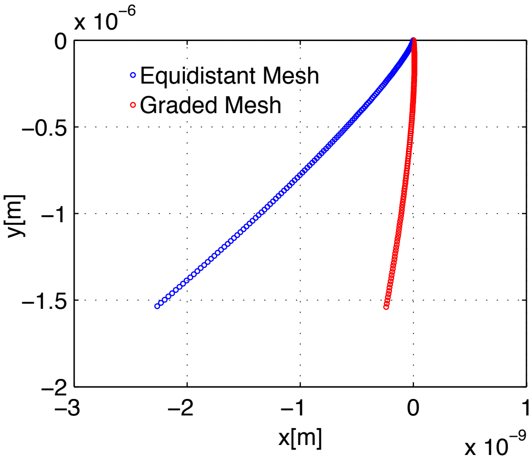

Fig. 10: Trajectory of the sinking particle for the equidistant (blue) and graded (red) mesh.

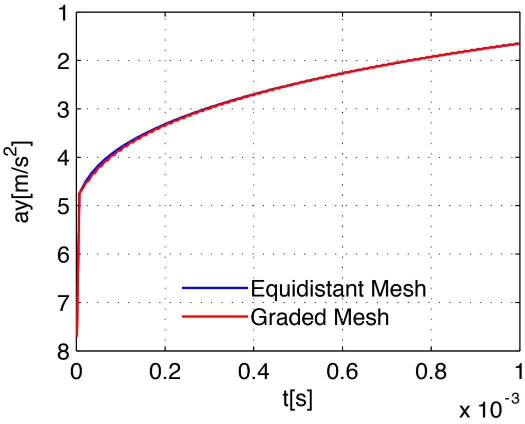

Fig. 11: Vertical acceleration of the sinking particle due to gravity for the equidistant (blue) and graded (red) mesh.

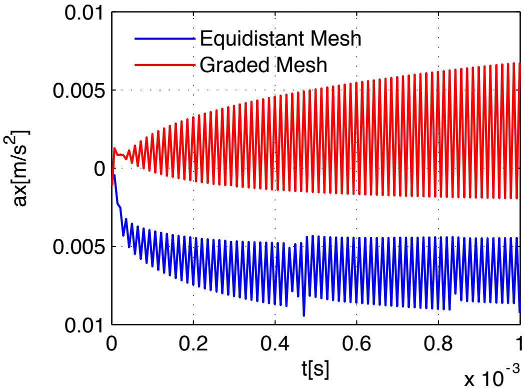

Fig. 12: Horizontal acceleration of the sinking particle for the equidistant (blue) and graded (red) mesh.

a slope.

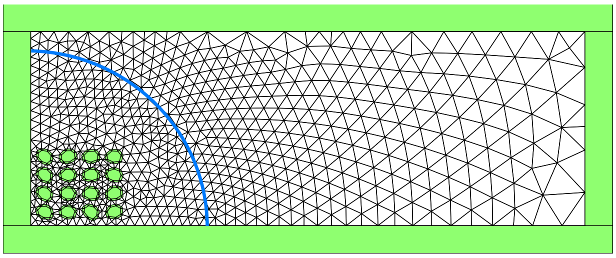

Fig. 13: Initial multi particle geometry. The body fitted mesh around the particles fills a circular sector and transitions into the graded mesh along the blue line. Variations in element size are a result of the mesh relaxation.

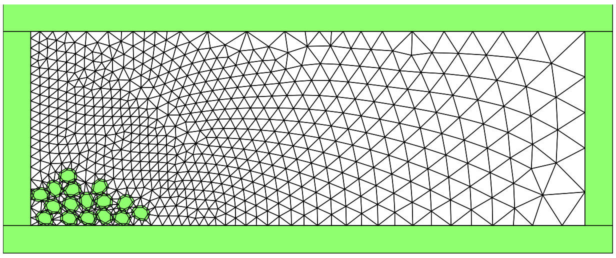

Fig. 14: Multi particle geometry after 0.16 s. Most particles have completed sedimentation.

8. Conclusion and Outlook

In Section 6. we have shown for a single particle that using graded mesh can significantly reduce computation times without negatively impacting accuracy. Furthermore, due to the resulting smaller number of nodes we reduce rounding errors and - although slightly increasing discretization errors - are able to minimize the overall numerical error, allowing for more physically accurate results.

We have also shown in Section 7., that the graded mesh can be applied to multi-particle geometries as well and simulations containing an even larger number of particles are currently built to better recreate physical experiments.

References

(1) L. Rondon, O. Pouliquen and P. Aussillous, "Granular Collapse in a Fluid: Role of the Initial Volume Fraction", Physics of Fluids, 23 (2011)

(2) Shi Han N., "Two Phase Dynamics of Granular Particles in a Newtonian Fluid", The University of Electro Communications, PhD thesis (2015)

(3) P. M. Gresho and R. L. Sani, "Incompressible Flow and the Finite Element Method - Volume Two", (1998), pp. 552

(4) E. Hairer and G. Wanner, "Solving Ordinary Differential Equations II", Springer, 2. rev. ed. (1996), pp. 42 & pp. 241

(5) Shi Han N., "Two Phase Dynamics of Granular Particles in a Newtonian Fluid", The University of Electro Communications, PhD thesis (2015) pp. 53

(6) P. M. Gresho and R. L. Sani, "Incompressible Flow and the Finite Element Method - Volume Two", (1998), p. 806

(7) E. Hairer and G. Wanner, "Solving Ordinary Differential Equations II", Springer, 2. rev. ed. (1996) p. 469

(8) Shi Han N., "Two Phase Dynamics of Granular Particles in a Newtonian Fluid", The University of Electro Communications, PhD thesis (2015) p. 85

(9) Shi Han N., "Two Phase Dynamics of Granular Particles in a Newtonian Fluid", The University of Electro Communications, (2015) pp. 66