Computers and Geosciences 123 (2019) 38–46

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

ICY: An interface between COMSOL multiphysics and discrete element code YADE for the modelling of porous media

Pouyan Pirnia*,[1], François Duhaime[2], Yannic Ethier[3], Jean-Sébastien Dubé[4]

Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, 1100 Notre-Dame Ouest, Montreal, Quebec, H3C 1K3, Canada

ARTICLE INFO

Keywords:
Porous media
Discrete-element method
Finite-element method
YADE
COMSOL
Internal erosion

ABSTRACT

The thermal, mechanical and hydrodynamic behaviour of porous media in geoscience applications is usually modelled through the finite-element (FEM) or finite-difference methods. These continuum models tend to perform poorly when modelling phenomena that are essentially dependent on behaviour at the particle scale or phenomena that are not accurately described by partial differential equations (PDE), such as internal erosion and filtration. The discrete nature of granular materials can be modelled through the discrete-element method (DEM). However, in some instances, DEM models would benefit from an interface with continuum models to solve coupled PDEs or to model phenomena that occur at a different scale. This paper introduces ICY, an interface between COMSOL Multiphysics, a commercial finite-element engine, and YADE, an open-source discrete-element code. The interface is centred on a JAVA class. It was verified using the simple example of a sphere falling in water according to Stokes' law. For this example, the drag force was calculated in COMSOL and body forces (gravity, buoyancy and drag) on the sphere were summed in YADE. The paper also presents an application example for the interface based on the modelling of internal erosion tests.

1. Introduction

Flow through porous media, like soil deposits or earth dams, has conventionally been analysed within a continuum framework. Continuum models have had particular success in capturing some important aspects of porous media behaviour, such as seepage and stress-strain behaviour. Nevertheless, some phenomena, such as internal erosion, derive from complex microstructural mechanisms at the particle scale that cannot currently be upscaled and described by macroscale partial differential equations (PDE). Since continuum models do not explicitly take into account the discrete nature of porous media, phenomena like internal erosion should be modelled at the particle scale (Guo and Zhao, 2014). At the same time, these phenomena often depend on macroscale parameters such as stress and pore pressure. A multiscale approach is thus needed.

The discrete-element method (DEM) is becoming increasingly common in the modelling of porous media (O'Sullivan, 2015). With DEM, the motion and interaction (contact forces) of a large number of small particles are computed. This approach considers explicitly each particle in a granular porous media and the contact forces between them. Hence, it can simulate finite displacements and rotations of particles (Cundall and Hart, 1992). Besides the capability of DEM to simulate complex phenomena in granular materials, the main advantage of DEM compared with other methods is the relative simplicity of governing equations and computational cycle.

The discrete element method has had great success in reproducing the mechanical response of dry granular material at both the particle and continuum scales (e.g., O'Sullivan et al. 2008). However, for field scale applications, such as earth dams, it is not feasible to model structures solely with DEM. The current practical limit on the number of particles in a model using personal computers is around 100 000 (O'Sullivan, 2015). For fine sand with a uniform diameter of 0.10 mm, this translates to a maximum model volume on the order of 70 mm^3 for hexagonal close packing. As a consequence, to be included in the

* Corresponding author.
E-mail addresses: seyed-pouyan.pirnia.1@ens.etsmtl.ca (P. Pirnia), francois.duhaime@etsmtl.ca (F. Duhaime), yannic.ethier@estmtl.ca (Y. Ethier), jean-sebastien.dube@etsmtl.ca (J.-S. Dubé).

[1] Pouyan Pirnia developed the theory, programmed the interface and the examples. He wrote the paper.
[2] François Duhaime had the original idea of combining DEM and FEM codes. He supervised the programming of the interface and the examples. He also revised and improved the manuscript.
[3] Yannic Ethier supervised the project and edited the manuscript.
[4] Jean-Sébastien Dubé supervised the project and edited the manuscript.

https://doi.org/10.1016/j.cageo.2018.11.002
Received 4 January 2018; Received in revised form 25 October 2018; Accepted 1 November 2018
Available online 02 November 2018
0098-3004/

modelling of large scale applications, DEM must be coupled with continuum models in a multiscale analysis where small scale DEM simulations are conducted for selected nodes in the model. The continuum model can also be used to calculate boundary conditions for the DEM simulations (e.g., hydraulic gradient and stress). This type of hybrid multiscale models remains in development and has not seen widespread use (Indraratna et al., 2015; Chareyre et al., 2012; Elmekati and El Shamy, 2010; Eberhardt et al., 2004).

A large number of DEM codes are currently available for applications involving granular media. The most common commercial DEM codes are PFC2D and PFC3D (Itasca, 2004). These codes have been used in a significant number of elemental studies where different external stress tensors are applied on soil elements to study their behaviour and to reproduce the macroscopic behaviour of soils (O'Sullivan, 2015; Ding et al., 2014). Although these commercial software packages allow some modification using an embedded scripting language named FISH, it is not as versatile as some open source codes. Python has recently been integrated directly into *PFC* 5.0. It allows models to be manipulated from Python scripts.

The molecular dynamic code LAMMPS (Plimpton, 1995) is one example of an open-source code that can be used for DEM simulations. It allows parallel computing on distributed memory machines using message-passing techniques (MPI). Parallel computing makes LAMMPS suitable for the simulation of large numbers of particles. However, increasing the number of processors does not reduce computation time linearly (O'Sullivan, 2015). LAMMPS and its derivative LIGGGHTS, have been used for some applications involving granular materials (e.g., Huang et al., 2014; Bym et al., 2013). LIGGGHTS stands for LAMMPS Improved for General Granular and Granular Heat Transfer Simulations.

YADE ("Yet Another Dynamical Engine") is a highly flexible and extensible open-source DEM package used in geotechnical engineering. The YADE framework permits changes, extensions and code reuse besides providing many low-level operations through plugins and libraries (Kozicki and Donzé, 2009). YADE has been developed for shared-memory parallel execution environment using OpenMP (Open Multi-Processing) increase the calculation speed on multiprocessor systems. YADE has been shown to require computation times that are similar to PFC3D (Jakob, 2012). A few methods can already be used with YADE to calculate hydrodynamic forces on particles: pore network flow and Lattice-Boltzmann method (Lominé et al., 2013; Chareyre et al., 2012).

There are already a few examples of DEM and FEM codes that can be interfaced. For instance, PFC3D has a Computational Fluid Dynamics (CFD) module that allows fluid-particle interactions to be modelled based on the volume-averaged coarse-grid approach (Furtney et al., 2013). Goniva et al. (2010) developed a CFD-DEM coupling to solve fluid-particle interactions using OpenFOAM, a finite-volume code, with LIGGGHTS. Zhao and Shan (2013) modified the OpenFOAM library to solve the locally averaged Navier–Stokes equation based on the coarse grid approximation method proposed by Tsuji et al. (1993). The interface between the open-source FEM platform Kratos and DEM engine DEMPack is another example. It has made possible the coupling of DEM with fluid dynamic, heat transfer and structural analyses in the same package (Santasusana Isach, 2013). Finally, Guo and Zhao (2014) have presented a framework to couple a large scale continuum model based on FEM with small scale DEM simulations conducted at each Gauss point of the large-scale FEM mesh. Open source codes Escript (Gross et al., 2007) for FEM and YADE for DEM were used.

Hybrid FEM-DEM models for soils often require computing resources that are not readily available. For instance, the Guo and Zhao (2014) model required approximately 4 h of computing time with 16 processors to solve a 2D problem involving only 240 elements. Also, existing interfaces are often programmed with specific applications in mind and often require extensive programming to develop new applications.

Recently, a growing trend has been observed toward the development of versatile multiphysics finite difference and finite element software packages. Multiphysics codes involve the coupled simulation of multiple phenomena. They can involve the resolution of sets of PDEs, for example the combined analysis of stresses and strains, heat transfer, seepage and solute transport in a porous media (e.g., Finsterle et al., 2014). They can also involve the combination of different model types, for example finite elements and molecular dynamics models (Keyes et al., 2013).

Some multiphysics software packages, such as COMSOL Multiphysics (COMSOL, 2016a), can be integrated in scripts or linked with other codes (Duhaime and Chapuis, 2014; Nardi et al., 2014). COMSOL and PHREEQC, a thermodynamic equilibrium code, were successfully coupled by Nardi et al. (2014) to create an interface named iCP (Interface COMSOL-PHREEQC). iCP is written in JAVA and uses the COMSOL JAVA-API and the IPhreeqc C++ dynamic library.

This paper presents ICY, a multipurpose interface that allows data to be exchanged between continuum models based on COMSOL Multiphysics (FEM) and particle scale model based on YADE (DEM). Details on the interface code are first given. The interface is then verified with the simple example of a particle falling in water according to Stokes' law. An application example involving the modelling of internal erosion tests in porous media is also presented. In this example, DEM is used to compute particle displacements, while hydrodynamic forces on particles are calculated based on Darcy's law with FEM. Other potential applications for the coupled model are finally presented.

Versatility is a key feature of ICY. The current interface allows virtually any partial differential equations (PDEs) and ordinary differential equations (ODEs) to be coupled with a discrete element simulation. The interface presented in this paper constitutes an important step toward the integration of multiscale modelling in porous media applications.

2. Methodology

2.1. COMSOL and YADE

COMSOL Multiphysics is a commercial finite element engine that can solve simultaneously a large range of preprogrammed partial differential equations (PDEs). COMSOL has two main interfaces for geoscience and geotechnical applications: the subsurface flow and geomechanics modules. The phenomena that can be modelled include fluid flow, seepage, chemical reactions, stress-strain behaviour and heat transfer. Custom partial differential equations (PDEs), ordinary differential equations (ODEs) and initial value problems can also be specified without programming, through a graphical user interface (GUI).

COMSOL models can be created through a graphical user interface. Each model is described by a model tree which includes a series of nodes that describe the model geometry, material properties, boundary conditions, PDE, solutions, etc. The information associated with these nodes can be accessed and modified by JAVA classes or MATLAB scripts.

The DEM code interfaced with COMSOL, YADE, is an open-source C++ framework with a Python script interface (Kozicki and Donzé, 2009; Šmilauer et al., 2015a). All computation parts (methods and algorithms) are programmed in C++ using object oriented models. This feature allows developers to add new algorithms and plug-ins. The Python interface is used to describe the model, to control the simulation and for post processing. YADE can be installed with Debian and Ubuntu Linux operating systems. With DEM, Newton's second law of motion (Force = mass × acceleration) is applied individually to each grain of a granular material (Cundall and Strack, 1979). Using the resultant forces on each particle, the velocity and position of each particle are calculated at the end of each time step. The changes in contact status (come into contact or lose contact) are automatically determined (O'Sullivan, 2014).

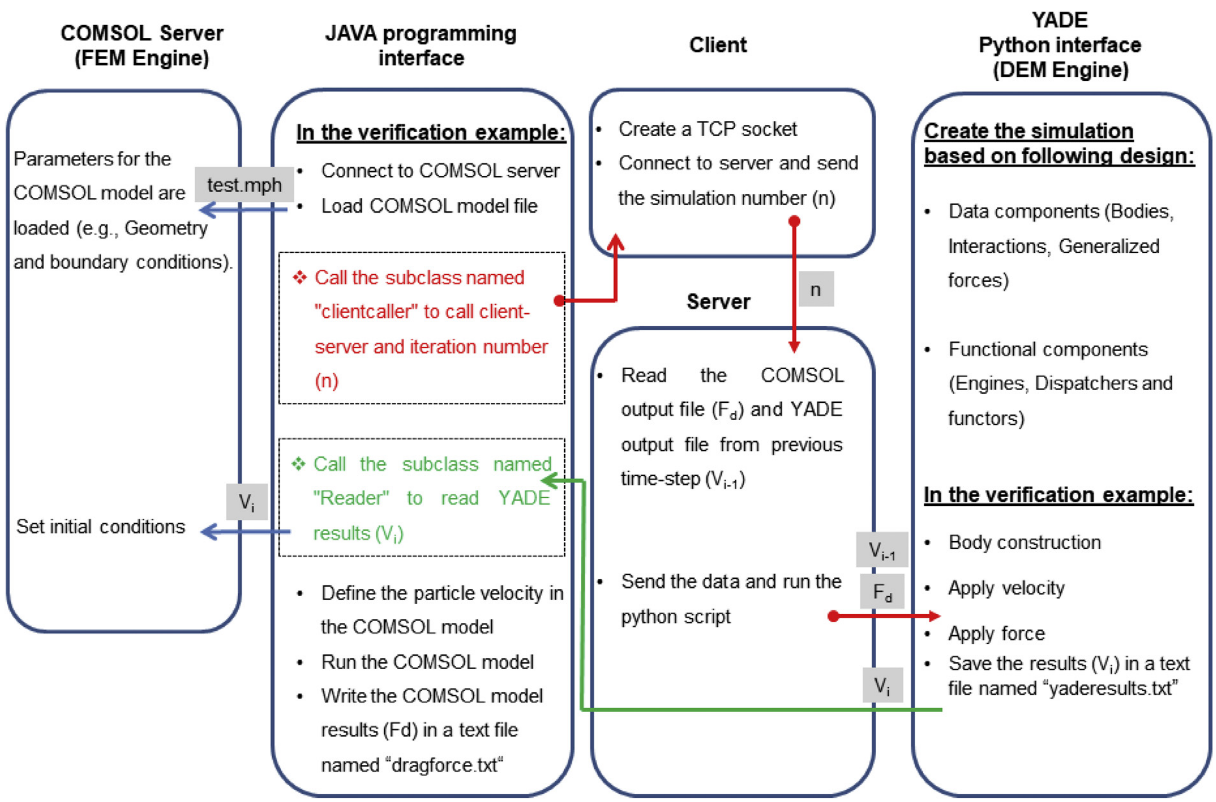

Fig. 1. Schematic view of the ICY algorithm in the context of the verification example.

2.2. *Coupling procedure*

The interface between COMSOL (FEM) and YADE (DEM) involves a partially coupled framework. The algorithm is presented in Fig. 1. The partially coupled approach solves the continuum and discrete element equations separately for each time step (Goodarzi et al., 2015). Results are exchanged between the two models at the end of each time step.

A JAVA class was used to control COMSOL. The JAVA interface was chosen because of its speed, the capability of being combined with Python code (the programming language used with YADE), and the fact that it does not require the MATLAB LiveLink module for COMSOL (COMSOL, 2016b). Through the JAVA class, command lines are sent to COMSOL to change the initial conditions for each time step, to set the parameter values received from YADE, and to run the simulation. The information sent to COMSOL varies for each application. Other JAVA classes are used in ICY. In the current interface, the algorithm includes two subclasses which are controlled by the main class.

The first subclass, named Clientcaller, was written to connect the interface to YADE and to supply the initial values of the required parameters and variables for YADE's interface via the client-server. As can be seen in Fig. 1, the algorithm features a client-server between the JAVA class controlling COMSOL and the Python script controlling YADE. The server for the client-server connection should not be confused with the COMSOL server.

The client-server acts as an agent between the two interfaces. It saves computing time by allowing the YADE interface to run independently of the JAVA interface and COMSOL. Additionally, statistical functions (e.g., mean, median) can be applied on the YADE input data from COMSOL and on the YADE data from the previous time step directly, on the server. Hence, the Python interface script remains intact. The client and server tasks are as follow:

I. The client receives simulation information (e.g., iteration number) from the JAVA interface via a terminal in Linux.
II. The client creates a TCP socket and sends the information to the server.
III. The COMSOL and YADE model results (e.g., drag force and pressure, particle velocity from previous time step) are formatted by the client script as command line arguments to be sent to the Python script that controls the current YADE time step.

The second subclass, called Reader, was programmed to read and organise results files produced by YADE for COMSOL simulations. It reads the YADE result file including the porous media mechanical response (e.g., particle velocity, porosity or permeability values) and sends them to the main class. The main class then assigns these parameters in the COMSOL model. Eventually, the main class runs the COMSOL model and save the results in predefined files. More details about the two subclasses will be presented for the internal erosion example (Fig. 6).

3. Verification

The simple example of a sphere falling in a water column was chosen to demonstrate that data can be successfully exchanged between COMSOL and YADE with ICY. YADE was used to apply Newton's second law on the sphere (discrete element), while COMSOL was used to solve the fluid flow (velocity, pressure and density) around the sphere using the Navier-Stokes equations (continuum). The steady-state incompressible Navier-Stokes equations (neglect inertial term; Stokes flow) with no-slip boundary conditions on the sphere were solved in the COMSOL model.

When a particle falls into a fluid, it accelerates until it reaches a constant (terminal) velocity. In this example, the particle acceleration and terminal velocity were calculated using two methods. First, they were calculated using Stokes' law. Secondly, they were calculated by applying Newton's second law in YADE with a drag force calculated in COMSOL.

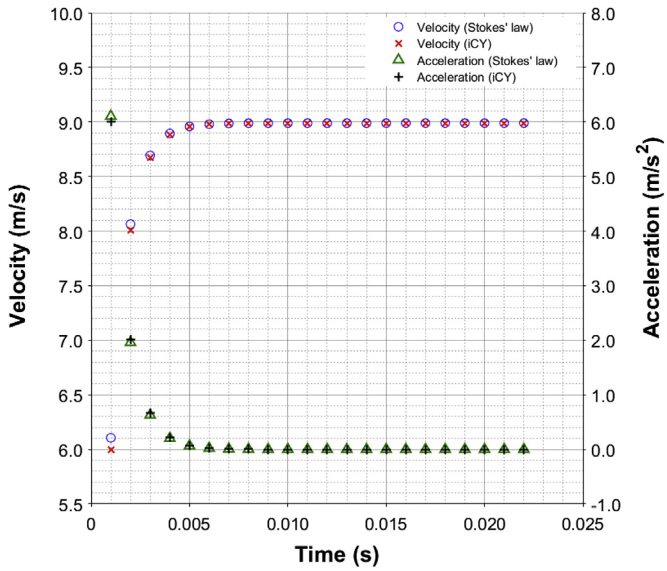

Fig. 2. Comparison of terminal velocity and accelerations through Stokes' law and FEM-DEM model.

3.1. Fall velocity and Stokes' equation

Stokes' law expresses the settling velocity of a sphere falling through a viscous fluid. Stokes derived the forces acting on a sphere sinking in a viscous liquid under the effect of gravity. The drag force F_D is expressed as:

$$F_D = 3\pi\mu V D_S \tag{1}$$

where μ is the kinematic viscosity of fluid, V is the sphere velocity and D_S is the sphere diameter.

Combining Eq. (1) with other forces imposed on the sphere falling in a quiescent and viscous fluid gives the acceleration of the sphere and its terminal velocity. Three forces act on the sphere when it is dropped into a column of liquid: buoyancy (F_B), viscous drag (F_D) and weight (F_w) respectively. Buoyancy and viscous drag are directed upward. Weight is directed downward. The sphere acceleration (a) and velocity can be calculated by combining these forces:

$$F_B = \left(\frac{\pi}{6} D_S^3\right)\gamma_w \tag{2}$$

$$F_w = \left(\frac{\pi}{6} D_S^3\right)\gamma_s \tag{3}$$

$$a = \frac{F_w - F_B - F_D}{m} \tag{4}$$

$$V_{i+1} = V_i + a\,\Delta t \tag{5}$$

where:

- m is the particle mass,
- γ_w is the specific weight of water,
- γ_s is the specific weight of the sphere,
- Δt is the time step interval between time t_i and t_{i+1},
- V_i is the particle velocity at t_i,
- V_{i+1} is velocity of the particle in t_{i+1}.

F_B and F_W remain constant while the sphere is falling. Drag force is the only time-dependent force. Since $V = 0$ at the beginning, drag force is initially equal to zero and acceleration is maximum. With time, the drag force increases until it reaches a constant value. From then onward, the particle falls at a constant rate called the terminal velocity.

3.2. YADE model

The falling particle model has two main parts. On the YADE side, a sphere with diameter 0.1 mm and zero initial velocity was created. The weight and buoyancy were constant during the simulation. The drag force was calculated in COMSOL. During each time step, COMSOL solved the Navier-Stokes equations based on the velocity received from the previous time step in the YADE model. Then the calculated drag force was sent to YADE by the JAVA interface and client-server to compute the acceleration and the velocity for the next time step. The time step was set to 0.001 s in YADE. The COMSOL model is solved in a steady-state condition. Therefore, the time step in YADE is the ICY or global time step. Note that for other applications, different time steps can be used for ICY (global time step), YADE and COMSOL. This will be the case in the application example. Particles density and gravity acceleration implemented in the YADE model were 2500 kg/m3 and 9.806 m^2/s, respectively.

3.3. COMSOL model

A particle with the same diameter as in the YADE model was created. The particle velocity calculated in YADE was applied as a boundary condition on fluid velocity (inlet) in COMSOL by the JAVA interface. Drag force was calculated in COMSOL and sent back to YADE by the client-server for the next time step. In COMSOL, part of the model tree was defined through the GUI (geometry, materials, fluid properties, boundary conditions, and mesh). In the geometry node, a sphere with a radius of 0.05 mm was created in a box representing the mathematical domain (width, depth and length of 40 mm). These dimensions were sufficiently large to prevent wall effects. Laminar and incompressible flow with a reference pressure of 1 atm was applied at the outlet. The mesh size near the sphere was shown to have a large influence on drag force accuracy. After trying a wide range of element sizes, a maximum element size of 0.0078 mm was chosen with an element growth rate (size ratio for two contiguous elements) of 1.5. The results do not change with a smaller mesh size.

3.4. Verification results

Results from Stokes' law and the FEM-DEM model are compared in Fig. 2. The velocity values for the COMSOL-YADE model are almost equal to those from Stokes' law. According to Stokes' law, it takes 0.007 s for the particle acceleration to decrease to almost zero. After 0.022 s, the particle moves with a velocity of 0.008989 m/s. The FEM-DEM results are almost identical. After 0.022 s, the particle velocity is 0.008990 m/s, a difference of 0.0027%. This shows that the drag force and particle velocity are correctly exchanged between COMSOL and YADE.

4. Application example

In this section, the simulation of an internal erosion test first presented by Tomlinson and Vaid (2000) is used as an application example for ICY. This permeameter test was chosen because it involves two layers of monodisperse spherical glass beads and the specimen is relatively small. Thus it can easily be simulated in YADE. In this example, YADE was used to solve the motion and to calculate the contact forces and torques for a large number of particles by means of the DEM, while fluid flow was solved with COMSOL based on the FEM.

4.1. Apparatus, testing materials and procedure

The permeameter used by Tomlinson and Vaid (2000) is presented schematically in Fig. 3. The specimen was composed of two layers of glass beads: a finer layer on top and a coarser layer at the bottom. The bottom and top layers had thicknesses of 3.7 and 1.9 cm respectively.

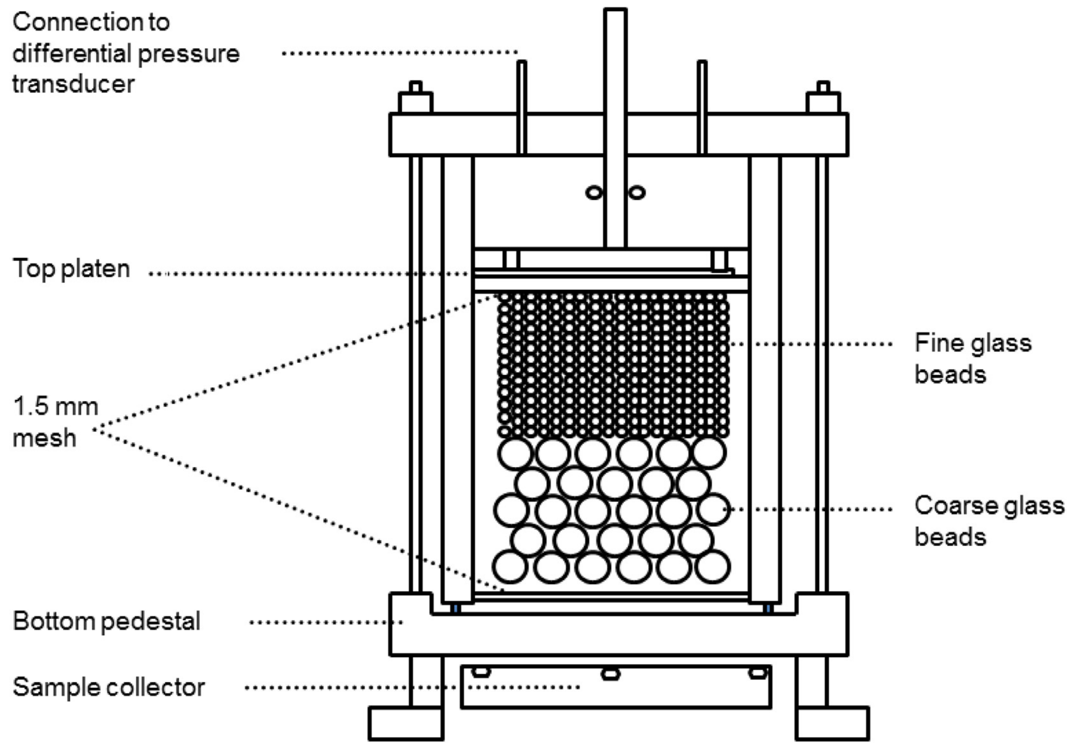

Fig. 3. Schematic diagram of laboratory permeameter (adapted from Tomlinson and Vaid, 2000).

Spherical glass beads with a uniform surface texture were used to exclude the influence of particle shape on the internal erosion results (Tomlinson and Vaid, 2000). The minimum round fraction was 70% and the glass density was 2500 kg/m^3. A test with 3-mm glass beads in the coarse layer and 0.346-mm glass beads in the fine layer (grain-size ratio of 8.7) under a confining pressure of 100 kPa was selected to be reproduced by the coupled model.

The cylindrical permeameter has an inside diameter and a height of 10 cm. The base pedestal has 5-mm holes to allow water and the glass beads to reach the sample collector. The specimen bottom is covered with a mesh with 1.5-mm openings. This mesh can retain the glass beads from the bottom layer inside the permeameter, but it allows water and the finer glass bead to flow out of the specimen.

At the beginning of the test, the confining stress applied by the top platen was gradually increased to 100 kPa over the course of several minutes. The specimen was then left under this stress for one hour. Thereafter, the desired hydraulic gradient was applied to the specimen through a 5 mm hole in the top platen. A small hydraulic head difference of 2 cm was first applied to initiate flow in the specimen. Finally, the upstream hydraulic head was increased rapidly from 2 to 23 cm in 1 min.

4.2. Fluid-DEM coupling theory

For the permeameter test, the same body forces as in the verification example (weight, buoyancy and drag force) were applied to the discrete spheres. Drag force, the only variable force, was directed downward.

There are two main approaches for the computation of hydrodynamic forces. The first approach is the sub-particle scale method. In this method, the Navier-Stokes equations are solved at the pore scale with an appropriate computational fluid dynamics (CFD) method, for example the Lattice-Boltzmann method (LBM) (Lominé et al., 2013). This approach requires computational resources that are not readily available.

The second approach, the coarse-grid method, is less computationally intensive. It was proposed by Tsuji et al. (1993). In this method, the fluid cell embraces several particles. Fluid flow derives from average pressures and velocities in several pores in each cell.

The principal difference between the sub-particle and coarse-grid methods is how the porous media topology is represented. The microscale grain arrangement is not considered explicitly for coarse-grid methods. The frictional losses are calculated based on Darcy's law and macroscale permeability values. With sub-particle methods, frictional losses are calculated by solving the Navier-Stokes equations at the microscale. The grain arrangement from the DEM simulation is considered explicitly.

Goodarzi et al. (2015) developed a coarse-grid framework to model fluid-soil interaction. The fluid was modelled as a continuum on a Eulerian mesh. The equivalent drag force was calculated from the Ergun equation (Ergun, 1952). It was then applied to particles at the microscopic scale in the DEM simulation. In this paper, a coarse-grid method based on Darcy's law was applied to model the Tomlinson and Vaid (2000) experiment.

DEM and coarse-grid calculations of drag force with FEM are the two main components in the DEM-FEM model. These components have a feedback on each other. In the DEM model, the movement of particles is influenced by the drag force calculated with the coarse-grid method. The particle displacements have in return an influence on the permeability and the hydraulic gradient that are calculated with the coarse-grid method. The hydraulic gradients are assessed in COMSOL for the whole permeameter by solving a water conservation equation based on Darcy's law:

$$\nabla.\,(K\nabla h) = 0 \tag{6}$$

where K is the hydraulic conductivity (m/s) and h is the hydraulic head (m).

Based on Darcy's law, the flow rate in a porous media is related to the hydraulic head difference and the porous media hydraulic conductivity or permeability:

$$v = K\frac{\Delta h}{L} \tag{7}$$

where:

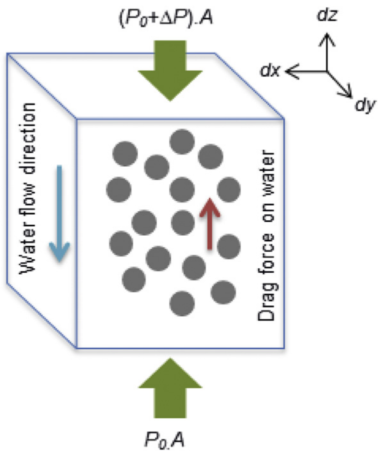

Fig. 4. Effective forces on water in a volume of porous media.

- v is the Darcy velocity (m/s),
- L is the flow path length (m), and
- Δh is hydraulic head change (m) over length L.

The influence of drag force on the fluid results in a force acting in the direction opposite to fluid movement. From force equilibrium considerations (Fig. 4), the drag force on particles (F_D) can be derived based on Darcy's law:

$$F_D = \Delta U.\ A \quad (8)$$

$$F_D = \Delta U.\ dx.\ dy \quad (9)$$

where $\Delta U = \Delta h.\gamma_w$ is the difference between the real pressure differential (ΔP) and the hydrostatic pressure differential ($dz.\gamma_w$). The hydraulic head (h) is the sum of the pressure head (P/γ_w) and the elevation head (z).

The total drag force can be applied on each particle proportionally to their volume or surface (Zeghal and El Shamy, 2004). If it is applied proportionally to their volume, the drag force on each particle is given by:

$$\mathrm{F_{DPi}} = \frac{\mathrm{F_D}}{(1-\mathrm{n}).\ \mathrm{V_T}}.\ \mathrm{V_{Pi}} \quad (10)$$

where:

- F_{DPi} is drag force on particle i,
- n is porosity,
- V_T is total volume of box, and
- V_{Pi} is the volume of particle i.

If the volume definition (dx dy dz) and Eq. (9) are substituted in Eq. (10), the drag force on each particle can be defined as:

$$F_{DPi} = \frac{\Delta U}{(1-n)dz}.\ V_{Pi} \quad (11)$$

4.3. Model implementation

The DEM specimen is presented in Fig. 5a. Compared to the test setup, the domain has a smaller horizontal section (1 cm × 1 cm) to reduce the total number of particles. The real height of the coarse-grained layer was used (3.7 cm). To further reduce the number of particles, the fine-grained layer thickness was halved. To compensate for the smaller number of fine particles, the eroded particles gathered in the bottom container were moved to the top of the fine-grained layer at the end of each global time step in the coupled COMSOL-YADE simulation (0.5 s).

A 2D mesh with 1.5 mm holes was produced by Gmsh, a finite element mesh generator (Geuzaine and Remacle, 2009). The mesh was located at the bottom of the coarse-grained layer (Fig. 5a). It allowed the fine particles to reach the container below the mesh.

The DEM specimen contains 160 coarse particles and 25000 fine particles. The DEM specimen was compacted by a wall generated at the top of the fine particles layer. The wall moved downwards at a constant velocity (0.01 m/s). The compaction was stopped when porosity of the layer reached 0.5. After settlement and compaction of the fine-grained layer, a small portion of finer beads (0.17 g), approximately 3000 particles, fell through the specimen. These particles were removed from the container before subjecting the specimen to the hydraulic gradient. This initial segregation was also reported by Tomlinson and Vaid (2000) in the experimental tests. This mass was removed from the container as well. At the end of this stage, the specimen is ready to be submitted to the hydraulic gradient.

The YADE time step was determined based on the P-wave velocity in the spheres as calculated by the PWaveTimeStep function (Šmilauer et al., 2015b). The P-wave velocity is a function of the particles' density and Young's modulus (E). FrictPhys interactions were used for the contact model in YADE. This contact model is based on the classical linear elastic-plastic law of Cundall and Strack (1979).

To have a longer time-step, the density scaling technique presented by O'Sullivan (2015) was used in this study. The particles' density was multiplied by 100. This increases particle weight by a factor of 100. The buoyancy (Eq. (2)) and drag force Eq. (12) were also multiplied by 100 to maintain the same proportions between forces.

The Young's modulus and Poisson ratio of the particles were set to 0.01 GPa and 0.3 respectively. A small Young's modulus value was assigned to decrease the P-wave velocity and to increase the maximum stable time step as a result. The damping coefficient and friction angle (φ) were found to be the most influential parameters in this simulation. The damping coefficient dissipates kinetic energy at the particle contacts. A wide range of damping coefficients and friction angles were tested for the YADE model (see application results and discussion for a comparison). The coupled model shows results that are similar to the experimental results with a damping coefficient of 0.4 and a friction angle of 17.19°.

According to Tomlinson and Vaid (2000), confining pressure has a negligible effect on the stability of finer beads, especially for the particle size ratio of 8.7 used in this example. A similar observation made with the numerical model during preliminary tests. Therefore, the 100 kPa confining pressure was not taken into account in the numerical model. Particles density was set to 2500 kg/m^3 in the YADE model. Fluid density and viscosity were set to 1000 kg/m^3 and 0.001 Pa s in the COMSOL model, respectively.

The COMSOL component of the coupled model was used to calculate the hydraulic gradient and the drag force. It consists in a 1-D domain representing the real thicknesses of the two layers (3.7 and 1.9 cm, Fig. 5). Based on Eq. (11), the drag force on each particle depends on the hydraulic gradient, porosity and particle volume. The coarse-grained layer was divided into 5 sections to calculate 5 average drag forces for each time step (Fig. 5a). The highest number of cells that could be used was 5 because of the filter layer thickness (3.7 cm) and the average particle size (0.167 cm). When dividing the filter layer in 5 cells, the thickness of each cell is 0.74 cm. This results in a ratio between cell and average particle size of 4.5. According to O'Sullivan (2014), cell dimensions should be 5 to 10 times larger than the average particle size. The model results with two cells are also compared to those with five cells to verify the sensitivity of the model with respect to the number of cells. The delay might be due to a smoothing effect of the pressure gradient that results in a smoothing of drag forces. Therefore, the number of cells is an important parameter that needs to be chosen carefully.

The hydraulic conductivity for the 5 filter sections were defined as parameters that were modified based on the YADE results as explained in the next section. The hydraulic conductivity values were assigned to the center of the five cells in the COMSOL model. The hydraulic

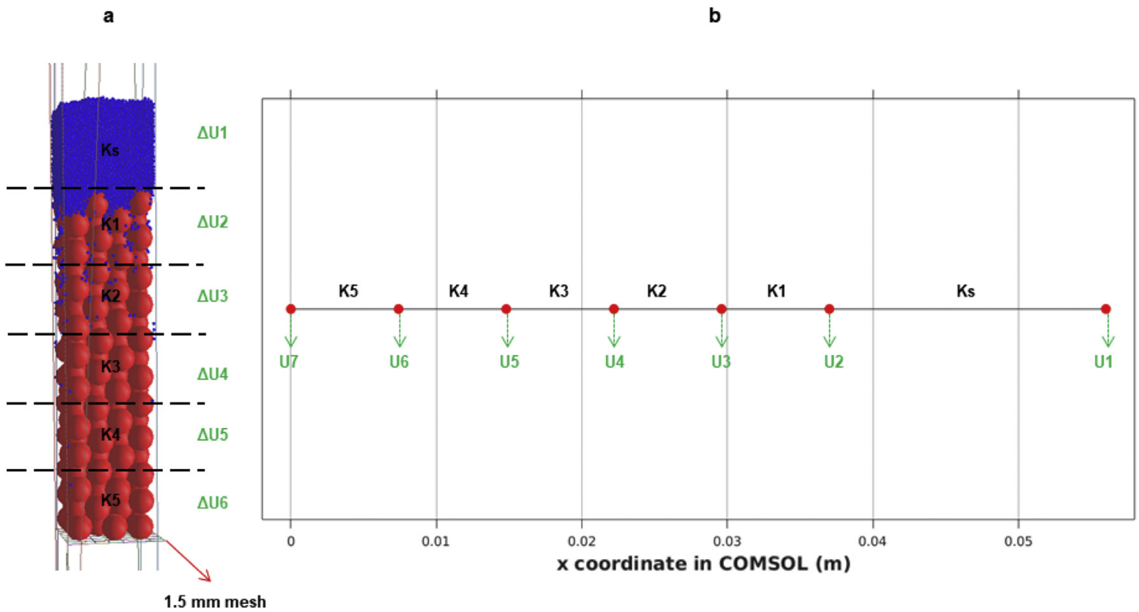

Fig. 5. Model implementation in COMSOL and YADE. The x coordinate in COMSOL (b) represents the vertical axis in YADE (a).

conductivity (K) in the water conservation equation (Eq. (6)) was defined as a linear interpolation of the K values for the five sections. The permeability of the fine-grained layer was set to 0.00134 m/s. It was evaluated with the Kozeny-Carman equation (Chapuis and Aubertin, 2003) considering the porosity of the layer of fine particles as 0.37. The hydraulic head at the top of the fine-grained layer was set to 23 cm as in the experiment.

4.4. Calculation sequence

Fig. 6 illustrates the sequences of calculation used for this simulation. For each global time step, the YADE simulation was first conducted. The hydraulic conductivity of each layer $(K_1, K_2, K_3, K_4, K_5)$ was predicted based on the new particle distribution and the Kozeny-Carman equation. The porosity and specific surface (total grain surface divided by total grain mass) of each layer were calculated in YADE.

The drag force equation programmed in YADE's Python interface requires the average pressure differential (e.g., $\Delta U_2 = U_2 - U_3$) in each cell. The pressure values at the cells' top and bottom boundaries (U_1, U_2, U_3, U_4, U_5, U_6, U_7) were calculated in the COMSOL model based on the previously mentioned hydraulic conductivity values (Fig. 5b). After every global time step in COMSOL, pressures at the subdomains' boundaries are saved in a text file. This file is read by the client-server at the beginning of global time step in YADE to supply 5 average pressure differentials to the Python interface. Based on the pressure gradients, the applied drag forces are updated at the beginning of each new global time steps in YADE. It should be apparent that the COMSOL model is solved as a steady-state (stationary problem).

5. Application results and discussion

According to Tomlinson and Vaid (2000), the base layer was all eroded in 45 s after the application of the hydraulic head difference of 23 cm for the modelled experiment (Fig. 7). The mass of eroded particles reported by Tomlinson and Vaid (2000) was 190 g. This corresponds to 2.41 g for a numerical specimen with a 1 × 1 cm section. The coupled model (five fluid cells) with time-steps of 0.5 s, friction angle 17.2° and damping 0.4 in YADE resulted in the complete erosion (2.41 g) of the base layer in 48 s (circle markers). As shown in Fig. 7, the coupled model results are dependent on friction angle and damping coefficients in the YADE model. The results also indicate that the number of fluid cells for the coarse-grid approach can influence erosion. A model with the same parameters, but two fluid cells in YADE, reached the same total erosion 10 s later (plus markers). It could stem from a smoothing effect of the pressure gradient resulting in a smoothing of drag forces. Therefore, number of cells is an effective parameter that needs to be chosen regarding the case study.

The same test was also simulated exclusively with YADE with a constant drag force corresponding to the initial hydraulic gradient in the coarse-grained layer. In this case, erosion was stopped after 7 s with 0.39 g of eroded particles in the container.

A comparison of the FEM-DEM results with the experimental results confirms that using Darcy's law and a continuum model to calculate pressure gradients and drag force for the modelling of internal erosion in granular material can give realistic results. The main part of drag force calculation is done in YADE with a negligible computational cost.

Modelling the same test but under a constant average drag force in YADE reveals the necessity of using a multiscale approach in the modelling of internal erosion. Piping was also stopped after a few seconds under larger but still constant hydraulic head difference (up to 100 cm). The main reason is that finer particles are trapped gradually in the coarse-grained layer. This clogs the coarse-grained layer and eventually stops erosion. In reality, the migration of finer particles to empty spaces in the coarse-grained layer gradually raises the pore pressure and the drag force, thus limiting clogging. This process is considered in the FEM-DEM computational cycle.

6. Conclusions

This paper introduced ICY, an interface between COMSOL, a FEM engine and YADE, a DEM code. The interface is based on a series of JAVA classes. The interface was verified with the simple example of a sphere falling in water according to Stokes' law. In this test, the particle motion was simulated using YADE. Drag force on the particle was calculated by solving the Navier-Stokes equations in COMSOL. Comparison between simulation and analytical results showed that the framework could accurately replicate the results obtained from Stokes' law. The coupled model was then applied to reproduce a laboratory erosion test with drag force calculated with a coarse-grid method. The numerical results were in good agreement with the experimental results. The coarse-grid method can be substituted for pore-scale approaches with a higher computational cost such as LBM and pore

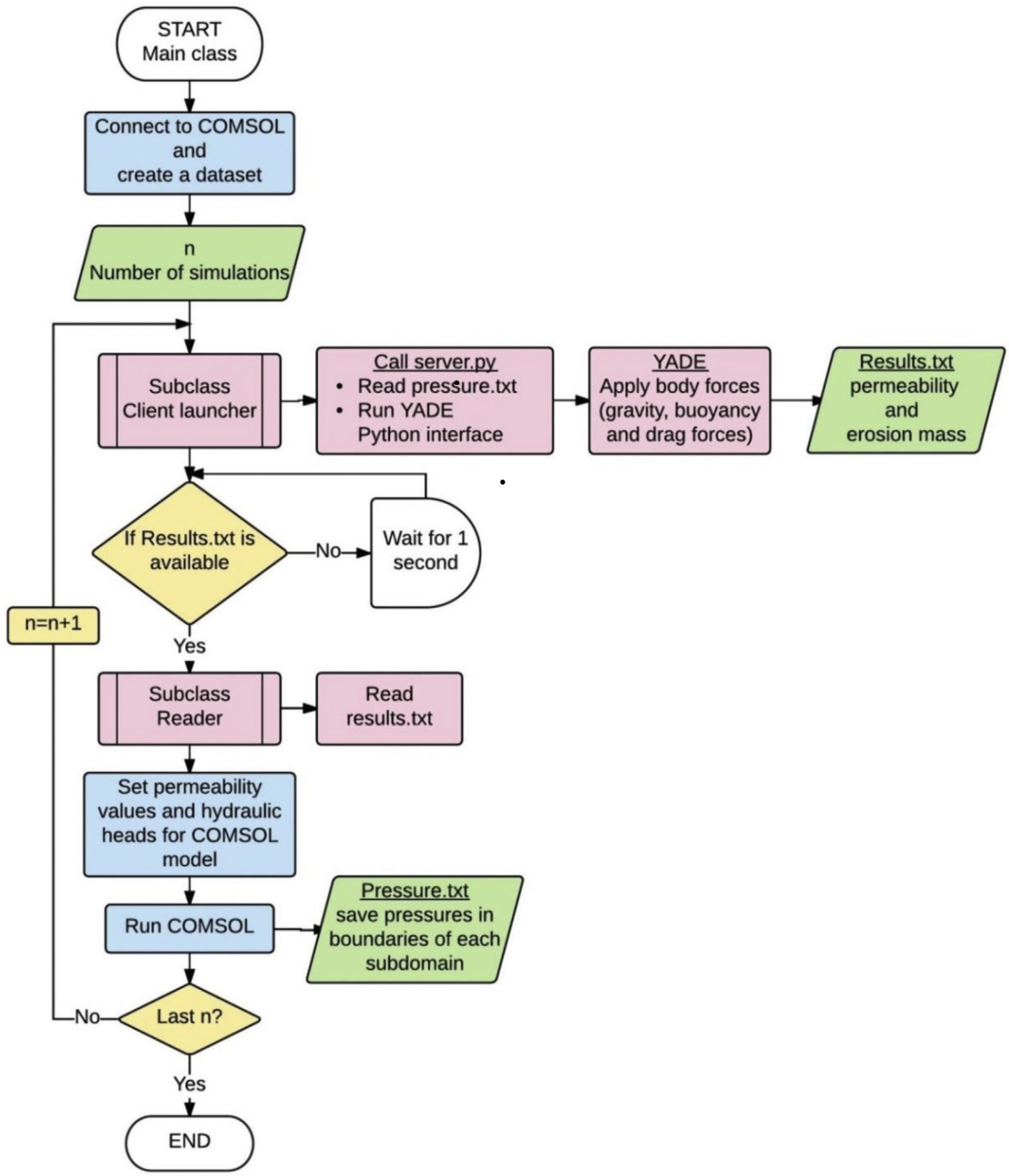

Fig. 6. Calculation sequence in FEM-DEM simulation of internal erosion.

network flow methods in case studies with very large number of particles.

Regardless of accuracy of FEM-DEM results, the main objective of this study is developing a versatile interface between DEM and FEM models. A great number of applications in geoscience could benefit from multiscale models that consider both the particle and continuum scales. Multiscale FEM-DEM models could be used for instance in the study of mineral industry applications (e.g., granular material segregation and sedimentation), geotechnical applications (e.g., internal erosion and fluidized bed) and energy extraction (e.g., sand production problem).

The coupled model might be used to simulate fluid-particles interaction for large scale applications in soil mechanics and geosciences in future. A multiscale scheme based on ICY is already under development to simulate internal erosion tests of a large permeameter.

Computer code availability

Name of code: ICY. **Developer:** Pouyan Pirnia and François Duhaime. Contact address: Laboratory for Geotechnical and Geoenvironmental Engineering (LG2), École de technologie supérieure, 1100 Notre-Dame Ouest, Montreal, Quebec, H3C 1K3, Canada. **Telephone number:** +1–5143968959. **E-mail:** pouyan.pirnia@gmail.com. **Year first available:** 2018. **Hardware required:** recommended 3 GHz or more, 8 cores. **Software required:** YADE, COMSOL Multiphysics, JAVA integrated development environment (IDE). **Program language:** Java, Python. **Program size:** 28 MB. **Supplemental file:** ICY instruction guide. **The source code and supplemental file are available at:** https://github.com/pouyanpirnia/ICY-2018.

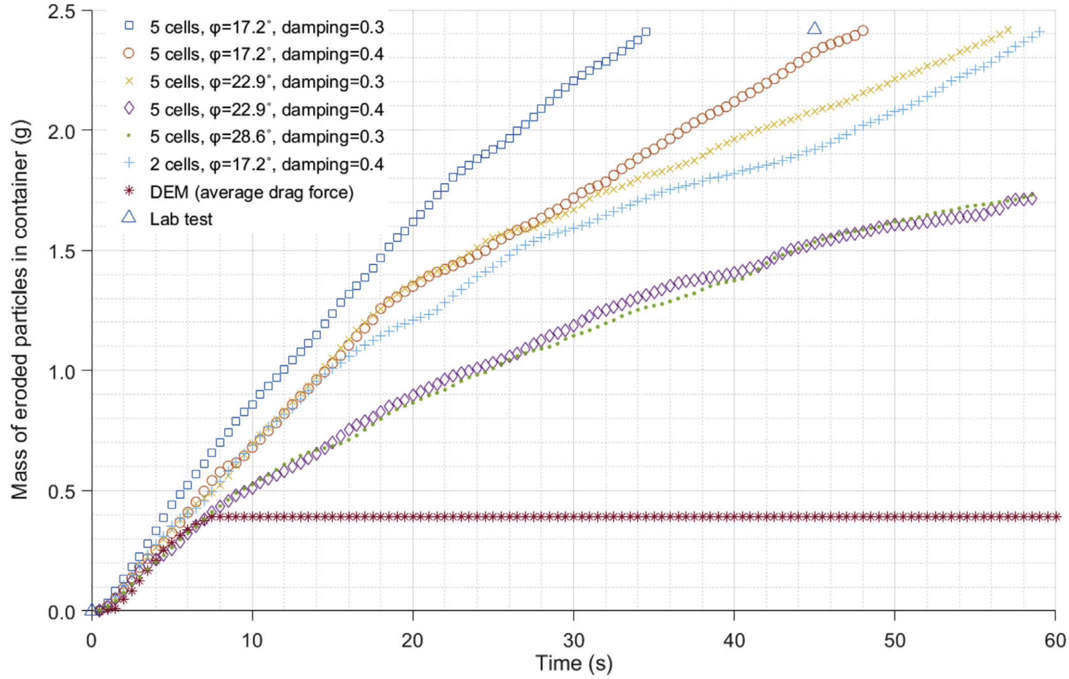

Fig. 7. Eroded mass for the experimental test, FEM-DEM model with different parameters and DEM under constant drag force.

Acknowledgements

The authors gratefully acknowledge the support of Hydro-Québec and NSERC for this project.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cageo.2018.11.002.

References

Bym, T., Marketos, G., Burland, J., O'sullivan, C., 2013. Use of a two-dimensional discrete-element line-sink model to gain insight into tunnelling-induced deformations. Geotechnique 63, 791.

Chapuis, R.P., Aubertin, M., 2003. On the use of the Kozeny Carman equation to predict the hydraulic conductivity of soils. Can. Geotech. J. 40, 616–628.

Chareyre, B., Cortis, A., Catalano, E., Barthélemy, E., 2012. Pore-scale modeling of viscous flow and induced forces in dense sphere packings. Transport Porous Media 94, 595–615.

COMSOL, A.B., 2016a. COMSOL Multiphysics – Reference Manual, Version 5.2a. COMSOL AB, Stockholm, Sweden.

COMSOL, A.B., 2016b. LiveLink for MATLAB User's Guide, Version 5.2a. COMSOL AB, Stockholm, Sweden.

Cundall, P.A., Hart, R.D., 1992. Numerical modelling of discontinua. Eng. Comput. 9, 101–113.

Cundall, P.A., Strack, O.D., 1979. A discrete numerical model for granular assemblies. Geotechnique 29, 47–65.

Ding, X., Zhang, L., Zhu, H., Zhang, Q., 2014. Effect of model scale and particle size distribution on PFC3D simulation results. Rock Mech. Rock Eng. 47, 2139–2156.

Duhaime, F., Chapuis, R.P., 2014. A coupled analysis of cavity and pore volume changes for pulse tests conducted in soft clay deposits. Int. J. Numer. Anal. Methods GeoMech. 38, 903–924.

Eberhardt, E., Stead, D., Coggan, J., 2004. Numerical analysis of initiation and progressive failure in natural rock slopes—the 1991 Randa rockslide. Int. J. Rock Mech. Min. Sci. 41, 69–87.

Elmekati, A., El Shamy, U., 2010. A practical co-simulation approach for multiscale analysis of geotechnical systems. Comput. Geotech. 37, 494–503.

Ergun, S., 1952. Fluid flow through packed columns. Chem. Eng. Prog. 48, 89–94.

Finsterle, S., Sonnenthal, E.L., Spycher, N., 2014. Advances in subsurface modeling using the TOUGH suite of simulators. Comput. Geosci. 65, 2–12.

Furtney, J., Zhang, F., Han, Y., 2013. Review of methods and applications for incorporating fluid flow in the discrete element method. In: Proceedings of the 3rd International FLAC/DEM Symposium, Hangzhou, China.

Geuzaine, C., Remacle, J.F., 2009. Gmsh: a 3-D finite element mesh generator with built-in pre-and post-processing facilities. Int. J. Numer. Methods Eng. 79, 1309–1331.

Goniva, C., Kloss, C., Hager, A., Pirker, S., 2010. An open source CFD-DEM perspective. In: Proceedings of OpenFOAM Workshop, Göteborg, pp. 1–10.

Goodarzi, M., Kwok, C.Y., Tham, L.G., 2015. A continuum-discrete model using Darcy's law: formulation and verification. Int. J. Numer. Anal. Methods GeoMech. 39, 327–342.

Gross, L., Bourgouin, L., Hale, A.J., Mühlhaus, H.-B., 2007. Interface modeling in incompressible media using level sets in Escript. Phys. Earth Planet. In. 163, 23–34.

Guo, N., Zhao, J., 2014. A coupled FEM/DEM approach for hierarchical multiscale modelling of granular media. Int. J. Numer. Methods Eng. 99, 789–818.

Huang, X., Hanley, K.J., O'Sullivan, C., Kwok, F.C., 2014. Effect of sample size on the response of DEM samples with a realistic grading. Particuology 15, 107–115.

Indraratna, B., Ngo, N.T., Rujikiatkamjorn, C., Sloan, S.W., 2015. Coupled discrete element–finite difference method for analysing the load-deformation behaviour of a single stone column in soft soil. Comput. Geotech. 63, 267–278.

Itasca, P.D., 2004. PFC3D (particle flow code in 3 dimensions) manual. In: ICG, Minneapolis, Minnesota.

YADE/Christian Jakob, 2012. Comparisons with PFC3D. [ONLINE] Available at: https://yade-dem.org/wiki/Comparisons_with_PFC3D, Accessed date: 24 May 2018.

Keyes, D.E., McInnes, L.C., Woodward, C., Gropp, W., Myra, E., Pernice, M., Bell, J., Brown, J., Clo, A., Connors, J., 2013. Multiphysics simulations Challenges and opportunities. Int. J. High Perform. Comput. Appl. 27, 4–83.

Kozicki, J., Donzé, F., 2009. Yade-open dem: an open-source software using a discrete element method to simulate granular material. Eng. Comput. 26, 786–805.

Lominé, F., Scholtès, L., Sibille, L., Poullain, P., 2013. Modeling of fluid–solid interaction in granular media with coupled lattice Boltzmann/discrete element methods: application to piping erosion. Int. J. Numer. Anal. Methods GeoMech. 37, 577–596.

Nardi, A., Idiart, A., Trinchero, P., de Vries, L.M., Molinero, J., 2014. Interface Comsol-PHREEQC (iCP), an efficient numerical framework for the solution of coupled multiphysics and geochemistry. Comput. Geosci. 69, 10–21.

O'Sullivan, C., 2014. Particulate Discrete Element Modelling: a Geomechanics Perspective. CRC Press.

O'Sullivan, C., 2015. Advancing Geomechanics Using DEM, International Symposium on Geomechanics from Micro to Macro, IS-Cambridge 2014, September 1, 2014 - September 3, 2014. Taylor and Francis - Balkema, Cambridge, United kingdom, pp. 21–32.

O'Sullivan, C., Cui, L., O'Neill, S.C., 2008. Discrete element analysis of the response of granular materials during cyclic loading. Soils Found. 48, 511–530.

Plimpton, S., 1995. Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. 117, 1–19.

Santasusana Isach, M., 2013. Kratos Dem, a Paralel Code for Concrete Testing Simulations Using the Discrete Element Method. Universitat Politècnica de Catalunya.

Šmilauer, V., et al., 2015a. Yade Documentation, second ed. The Yade Projecthttps://doi.org/10.5281/zenodo.34073. http://yade-dem.org/doc/.

Šmilauer, V., et al., 2015b. Reference manual. Yade Documentation, second ed. The Yade Projecthttps://doi.org/10.5281/zenodo.34045. http://yade-dem.org/doc/.

Tomlinson, S.S., Vaid, Y., 2000. Seepage forces and confining pressure effects on piping erosion. Can. Geotech. J. 37, 1–13.

Tsuji, Y., Kawaguchi, T., Tanaka, T., 1993. Discrete particle simulation of two-dimensional fluidized bed. Powder Technol. 77, 79–87.

Zeghal, M., El Shamy, U., 2004. A continuum-discrete hydromechanical analysis of granular deposit liquefaction. Int. J. Numer. Anal. Methods GeoMech. 28, 1361–1383.

Zhao, J., Shan, T., 2013. Coupled CFD–DEM simulation of fluid–particle interaction in geomechanics. Powder Technol. 239, 248–258.