20th International Conference
ENGINEERING MECHANICS 2014
Svratka, Czech Republic, 12 – 15 May 2014

COMBINATION OF FEM AND DEM WITH APPLICATION TO RAILWAY BALLAST-SLEEPER INTERACTION

J. Stránský*

Abstract: *Combination of two numerical methods - continuum based finite element method (FEM) and discrete based discrete element method (DEM) is presented in this contribution. The method is applicable to explicit dynamic problems and specifically will be applied to modeling of interaction between railway ballast (modeled by polyhedral discrete elements) and railway sleeper (modeled with standard finite elements). The problem is addressed from both theoretical and implementation point of view.*

Keywords: DEF, FEM, Surface coupling, Explicit dynamics, Multi-method.

1. Introduction

Numerical simulations are an indispensable part of current engineering and science development. For different engineering areas there are different numerical methods used. In solid phase mechanics, the leading methods are the finite element method (FEM) and the discrete element method (DEM). FEM is rigorously derived from the continuum theory and is being used for the description of deformable continuous bodies, while DEM describes particulate materials, usually modeled by perfectly rigid particles and their interactions determined from fictitious overlaps of these rigid particles.

Often an engineering problem can be modeled using only one of the aforementioned methods. A railway sleeper would be simulated by FEM, an assembly of ballast particles by DEM. One possible approach how to model an interaction between sleeper and ballast would be to split the problem into two domains (the sleeper part modeled by FEM and the ballast part modeled by DEM) and appropriately couple them. This approach is described in this contribution.

There are countless software programs for both FEM and DEM. Some of them are commercial (usually) without possibility to change the code and adjust the behavior to our requirements (combination with another software for instance). However, there exist programs with open source code, which the user can modify, possibly for coupling with other programs. In the present article, coupling of FEM code OOFEM (Patzák and Bittnar, 2001) and DEM code YADE (Šmilauer et al., 2010) is presented. Both programs have the core written in C++ (providing efficient execution of time consuming routines), user interface written in Python (modern dynamic object oriented scripting language, providing easy to use scripting while preserving the C++ efficiency) and extensible object oriented architecture allowing independent implementation of new features - new material model or new particle shapes for instance.

Theoretical aspects of the combination are described in section 2 and the implementation in section 3.

2. Theory

2.1. Discrete element method

In the discrete element method, the particles are represented as a set of perfectly rigid particles. In this contribution, the particles have a shape of convex polyhedrons (Eliáš, 2013). The convex polyhedrons are defined as an intersection of several half-spaces. Each half-space is defined by oriented plane, see Fig. 1.

* Ing. Jan Stránský: Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7/2077; 166 29, Prague; CZ, jan.stransky@fsv.cvut.cz

DEM solves numerically equations of motion, which are according to the perfect rigidity assumption idealized as a mass point with 6 degrees of freedom (3 displacements and 3 rotations). Forces and moments in equations of motion can be prescribed (gravity for instance) or are computed according to constitutive laws from mutual displacements and rotations of individual particles. In the current work, the normal force between two particles is directly proportional to the volume of intersection of two particles or is zero if there is no intersection. The point of action is placed to the center of mass of intersecting polyhedron and its direction is determined as a normal to the plane approximating in the sense of least squares "visible" edges of the intersection (edges of the intersection belonging to both intersecting particles). Also the shear force is incrementally computed from mutual displacement and rotation of both particles. The algorithm is described by Eliáš (2013) in detail and illustrated in Fig. 1.

Analogical algorithm may be applied to the case, when one of the particles is a planar triangle. For interaction evaluation, the triangle is “extruded” to form a wedge, represented as a polyhedron, and the algorithm described above is applied. After the contact force acting on the triangular particle is known (its direction, magnitude and point of action), it can be approximated with the help of FEM-like linear approximation to triangle vertices. This concept is used in the next section.

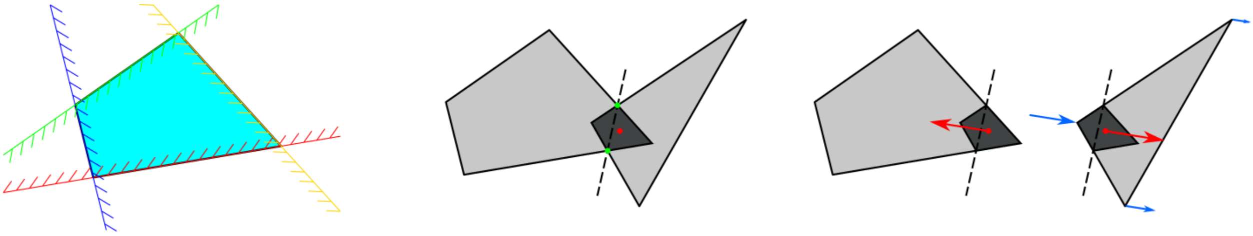

Fig. 1: 2D illustration: polyhedron as an intersection of half-spaces (left), intersection of two polyhedrons (middle) and contact force (right) applied to the centroid of intersection (red) and its interpolation to vertices (blue).

2.2. Combination with finite element method

The method according to Stránský (2013) for combination of DEM and FEM is used. As was already mentioned, each domain of the problem is solved separately. Ballast particles are solved by polyhedral DEM, while the sleeper is solved by FEM with linear tetrahedrons. The problem is solved as explicit dynamics.

Firstly, the surface of tetrahedral FEM mesh is extracted, forming a set of triangles. These triangles are then copied into DEM solution. In this work they are called “FEM particles”.

In each time step of the problem, DEM domain is solved at first. In this stage, the FEM particles are not allowed to move playing a role of fixed boundary. The results of DEM part of the solution are new positions of polyhedral particles and also forces applied to triangular FEM particles. These forces are approximated to triangle vertices (corresponding to nodes of FEM mesh) and the resulting vertex force is computed as a sum of contribution of all triangles sharing the vertex.

These vertex forces are then passed to the FEM domain as external load (nodal forces). New time step of the FEM domain is solved according to these forces, resulting in new nodal positions. Values of new nodal positions are passed back to DEM domain. The position and shape of FEM particles are updated according to this information.

Then a new time step is solved following the same algorithm.

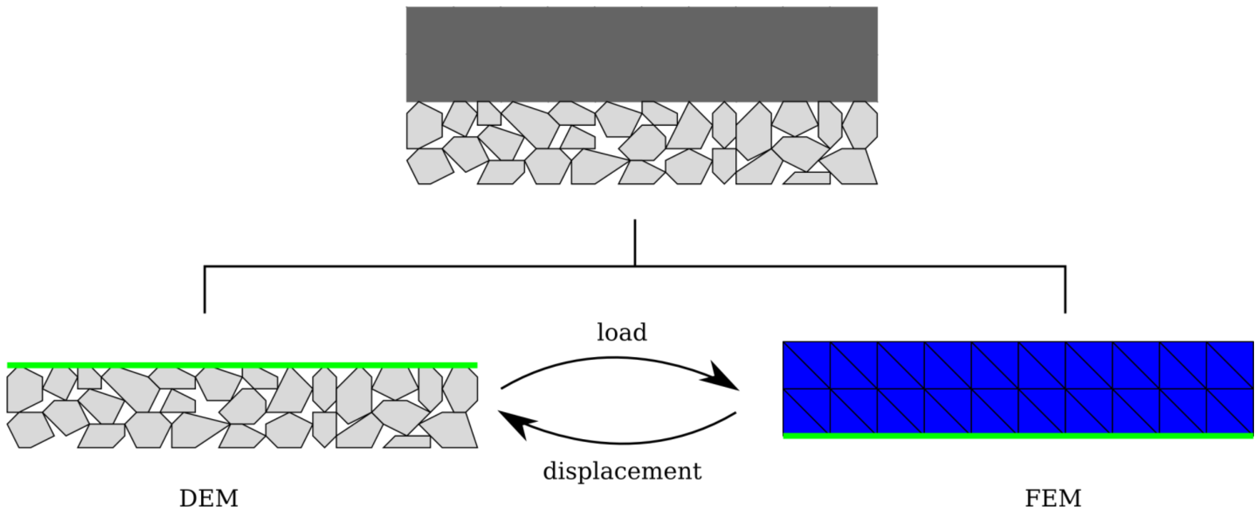

Fig. 2: Illustration of DEM and FEM combination.

3. Implementation

The implementation of polyhedral particles in YADE software, including contact detection, contact forces evaluation etc., is described in Eliáš (2013). The described implementation also involves polyhedron-triangle interactions, therefore no new implementation is needed.

Both chosen programs OOFEM and YADE work based on input files (YADE uses Python controlling scripts and OOFEM has its own format of text input files). For each domain, the input file is created at first, containing information about geometry, discretization, boundary and initial conditions, materials etc. OOFEM input file contains information only about finite element mesh, YADE input file only about polyhedral particles.

Both programs use Python scripting language as a user interface. Firstly, from one master controlling Python script both FEM and DEM domains are initialized according to prearranged input files. To the DEM domain (containing only polyhedrons so far), the FEM particles (extracted as a boundary of FEM domain) are copied as triangular particles.

From the master script it is possible to access and dynamically modify variables in both programs.

The master script in the beginning:

- initializes OOFEM, which reads its input file;
- initializes YADE, which reads its input file;
- gets information from OOFEM about FEM boundary, passes this information to YADE, which creates copies of these triangles.

The master script then follows algorithm described in section 2.2:

- YADE computes one time step;
- DEM contact forces are transferred to nodal forces and are passed to OOFEM;
- OOFEM computes one time step and the displacements are passed to YADE;
- YADE updates positions of FEM particles
- proceeds to the next time step

The implementation and the source code are intended to be open source with the possibility of free download. Also integration into a wider framework, specifically MuPIF (Multi-Physics Integration Framework) by Patzák et al. (2013) would be a natural direction of further development. In such case, also other programs than OOFEM and YADE might be used in the same context.

4. Summary

A method for combination of FEM and DEM applicable for explicit dynamic problems was presented in this contribution.

The application of the method will be modeling of interaction of railway ballast and sleeper interaction.

Despite the effort of the author, the implementation, documentation, code publication and examples have not been fully completed before this paper deadline. However, they will be presented during the conference and on web pages of related projects as soon as they are finished.

A future work on the topic may address more efficient implementation or using the method on other application topics.

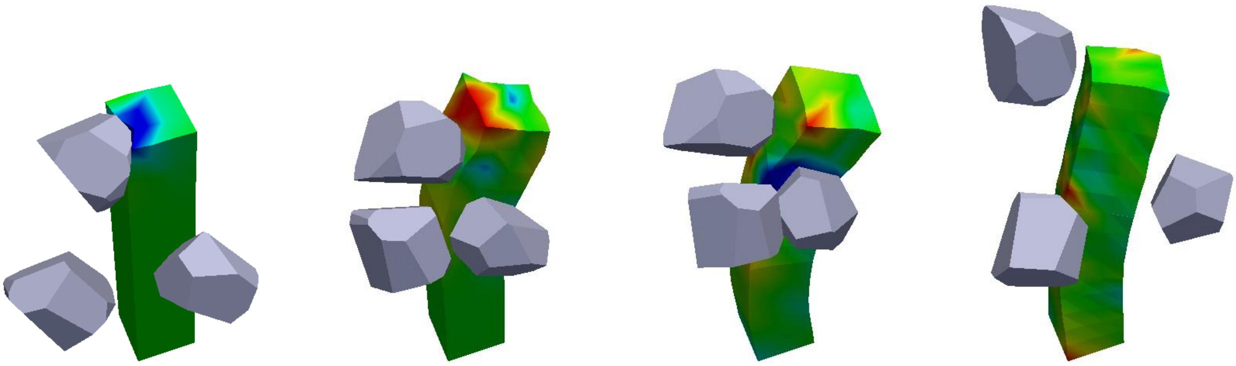

Fig. 3: Example of simple simulation.

Acknowledgement

Financial support of the Czech Technical University in Prague under project SGS no. OHK1-089/14 is gratefully acknowledged.

References

http://mech.fsv.cvut.cz/mupif

http://www.oofem.org

http://yade-dem.org

Eliáš, J. (2013) DEM simulation of railway ballast using polyhedral elemental shape, in: Particle-Based Methods III. Fundamentals and Applications (M. Bischoff et al. eds.), CIMNE, Barcelona, Spain, pp. 247-256.

Patzák, B., Bittnar, Z. (2001) Design of object oriented finite element code. Advances in Engineering Software, 32, pp. 759-767.

Patzák, B., Rypl, D., Kruis, J. (2012) MuPIF - A distributed multi-physics integration tool. Advances in Engineering Software, 60-61, pp. 89-97.

Stránský, J. (2013) Open source DEM-FEM coupling, In: Particle-Based Methods III. Fundamentals and Applications (M. Bischoff et al. eds.), CIMNE, Barcelona, Spain, pp. 46-57.

Šmilauer, V., Catalano, E., Chareyre, B., Dorofeenko, S., Duriez, J., Gladky, A., Kozicki, J., Modenese, C, Scholtès, L., Sibille, L., Stránský, J., Thoeni, K. (2010) Yade Documentation (V. Šmilauer ed.), The Yade Project, 1st ed. http://yade-dem.org/doc/.