19th International Conference
ENGINEERING MECHANICS 2013
Svratka, Czech Republic, May 13 – 16, 2013

FEM – DEM COUPLING AND MuPIF FRAMEWORK

J. Stránský [1]

Abstract: *MuPIF is a multi-physics integration tools that faciliates the implementation of multi-physics and multi-level simulations assembled from independently developed applications (Patzák et al. , 2012). Such independently developed applications might be a finite element code and a discrete element code. In this contribution, the design of integration of discrete element method (DEM) into existing continuous based (e.g. FEM) MuPIF components is presented.*

Keywords: ***FEM, DEM, multi-physics, Python***

1. Introduction

Numerical simulations are nowadays an indispensable part of engineering and science development. Usually the simulation is performed by a computer program, which is focused on narrower or wider class of problems (such as solid mechanics, fluid dynamics, heat analysis etc.). If a combination of two classes of problems is required (coupling of mechanical and heat analysis for instance), often it is possible to find a code allowing such approach. However, in some cases, there exists no such program allowing desired problems combination. For instance, it is possible to couple mechanical and heat analysis within the chosen code, but we would like to use a special material model for mechanical analysis, which is not implemented.

One possible approach to solve such situation would be to write a new or extend existing program implementing requested features. Another possible approach would be to use existing independently developed codes, each one focused on specific class of problems, and "glue" them together. The latter approach is a motivation of MuPIF (multi-physics integration framework) tool development. So far, only continuum based methods (finite element method, finite difference method etc.) were considered in the implementation.

The discrete element method (DEM) is (together with the finite element method - FEM) one of the leading methods for numerical solution of solid mechanics problems. It describes material as a set of rigid particles interacting with each other by discrete forces. These forces are caused by mutual displacement and rotation of particles. Usually such system is solved in terms of explicit dynamics.

The summary of existing MuPIF features is introduced in section 2 and the design of new DEM specific implementation is the topic of section 3.

[1] Ing. Jan Stránský, Czech Technical University in Prague, Faculty of Civil Engineering, Thákurova 7, 166 29 Praha 6, tel. +420 224 35 54 17, e-mail jan.stransky@fsv.cvut.cz

2. MuPIF

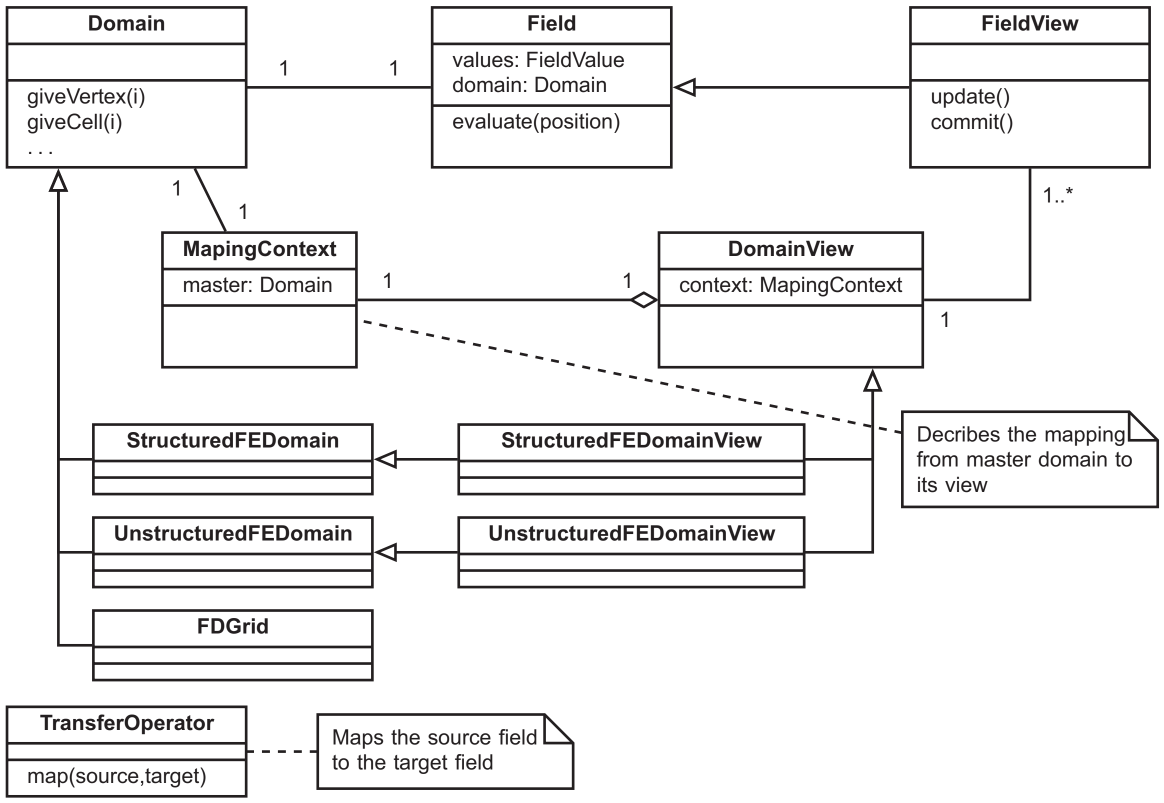

Fig. 1: Design of MuPIF framework according to Patzák et al. (2012).

MuPIF (multi-physics integration framework) serves to facilitate combination of independently developed programs. It provides high-level abstract data exchange interface between individual codes (components). The MuPIF itself is written in Python, modern dynamic interpreted object oriented language. The object oriented structure together with inheritance and polymorphism concepts allow easy adjustment of default implementation to specific situations, while preserving unified interface design and the possibility to reuse newly created code in other contexts or with different program(s). Such context (e.g. aforementioned mechanical and heat coupling) may be coded as a relatively independent application, in MuPIF called "agent". Final application may be composed of several such agents (see below for specific example).

The unified interface is the key concept of MuPIF design, therefore each component needs to implement their own application interface, which plays a role of connecting link between the component and MuPIF itself. As a consequence, one component (program) may be replaced with any other program providing required interface, so the user can choose specific program according to specific situation - solver speed, material model library etc.

The overall MuPIF design is shown in figure 1. The computational domain is represented by abstract class `Domain` describing geometry of solved problem and also providing services for spatial search etc. Derived class `DomainView` represents a subset of `Domain`, e.g. boundary, material region etc. `MappingContext` class serves for communication between master `Domain` and corresponding `DomainView`. Any field (solution displacement field or applied body forces field for instance) is represented by `Field` class. Derived `FieldView` class mediates mapping between `Field` defined on master `Domain` and corresponding `DomainView`.

The mutual data exchange between individual components (programs) is performed as an exchange between `Domain-Field` pairs. Typically, one application (provider) delivers source `Field`, defined on corresponding `Domain`, while another application (receiver) requires or accepts source field. Each `Domain` may have different discretizations (or even different type of discretization), thus for each problem pair there could be different mapping algorithm. To preserve unified interface, `TransferOperator` with its `map` function is used or overloaded according to specific requirements.

3. Design of the new DEM implementation into MuPIF

There exist several types of FEM – DEM coupling, e.g. surface coupling, see Munjiza (2004); Oñate & Rojek (2004); Villard et al. (2009), volume coupling, see Rousseau at al. (2009); Xu et al. (2002) or multiscale coupling, see Rojek & Oñate (2007); Wellmann & Wriggers (2012). For each type, different approaches are used, therefore this section is divided into corresponding subsections. In case of UML diagram, current design is represented with white background, while newly proposed designs are highlighted by gray background.

3.1. DEM in general

In this subsection, general features of DEM implementation and differences with the FEM implementation will be discussed. FEM represents the computational domain with finite elements. Usually, the elements have a geometric shape (triangle, tetrahedron, box . . .) defined by vertices.

Contrary to FEM, DEM represents domain by discrete elements (rigid particles), generally of arbitrary shape. It is also usual in DEM to represent boundaries with triangular particles, which can be rigid or flexible. The triangular particles may be standalone (behaving in the same way as DEM particles) or defined by vertices and connectivity table, behaving more like FEM mesh (at least from geometry point of view). See figure 2 for illustration.

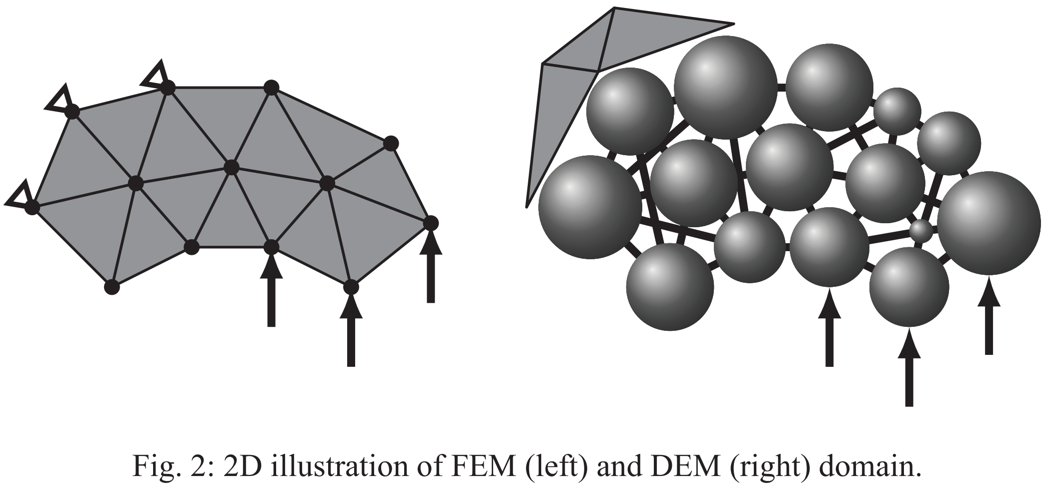

Fig. 2: 2D illustration of FEM (left) and DEM (right) domain.

The first difference is `Vertex` and `Cell` classes (from the name, the primary purpose is FEM-like methods). In FEM, `Cell` represents a finite elements defined by several `Vertex` instances. The spatial identifier (bounding box) of the cell is computed as an outer envelope of `Cell`'s vertices.

Within the context of DEM particles, `Vertex` represents the center of the particle, while `Cell` represents the particle itself (a DEM `Cell` has exactly one `Vertex`). Bounding box of such cell is therefore defined by the particle itself.

The FEM-like triangular mesh (see above) is more naturally represented by vertices and triangular cells, as it would be in the case of FEM mesh.

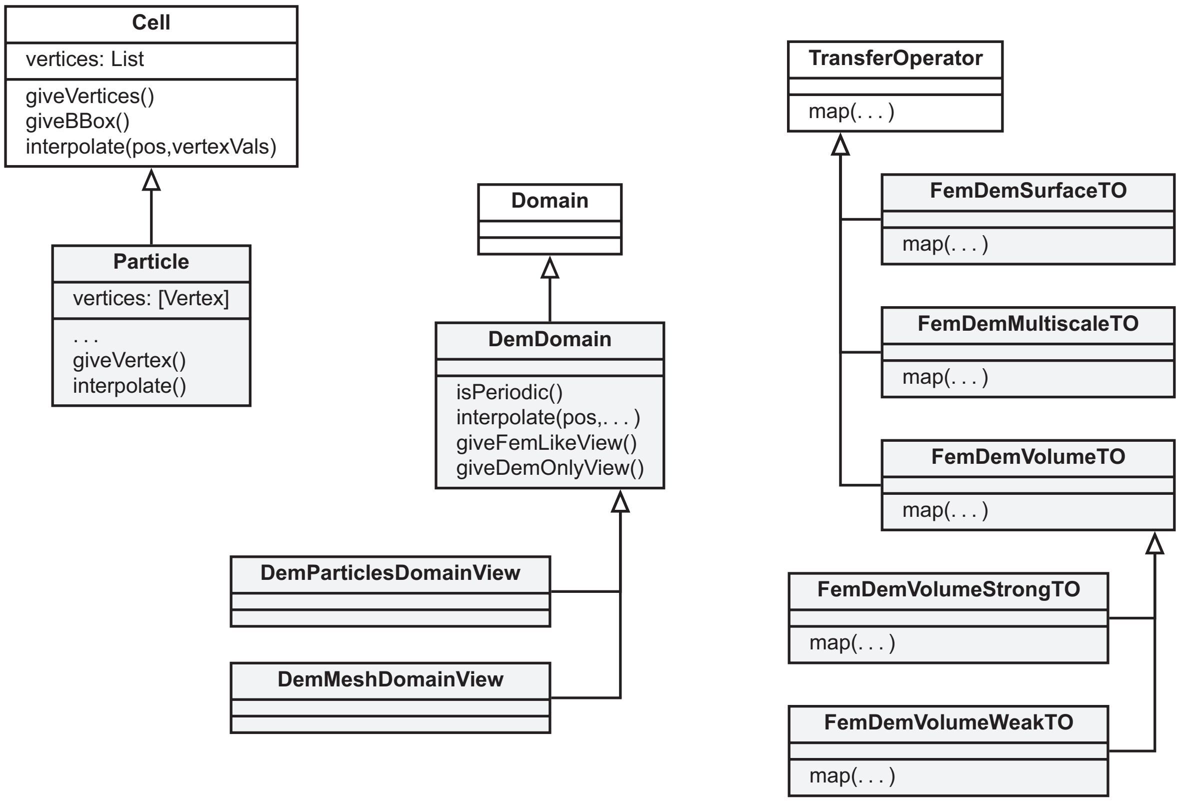

Fig. 3: New DEM related classes.

3.2. Surface coupling

Surface coupling is the easiest and most straightforward FEM – DEM coupling strategy. It considers FEM and DEM computational domains as strictly separated. As it is usual in DEM codes to represent boundaries by triangular particles, the natural approach is to use copy of FEM boundary (see `DomainView` in previous section) inside DEM code. As a consequence, the communication between FEM and DEM code would be only within two `DomainView` instances, on FEM part representing the boundary and on DEM part being composed of triangular particles (representing the same boundary, but from "opposite" part).

In DEM simulation, particles and boundary (represented by triangular particles) may overlap, causing repulsive forces. Such force are transfered from DEM to FEM code and play the role of load (natural, Neumann) boundary conditions. The displacement is computed and transfered back to DEM code, thus playing the role of displacement (essential, Dirichlet) boundary condition.

For the purpose of surface coupling itself, new class `FemDemSurfaceTO`, derived from `TransferOperator`, will be implemented. It "hides" all the implementation, i.e. it provides the functionality described above in its `map` function. Furthermore, a new MuPIF agent specialized on surface FEM – DEM coupling will be implemented, providing simple interface for this task. As was already mentioned, both the FEM and the DEM codes may be replaced by any other program (assuming that the new program already has got proper MuPIF interface).

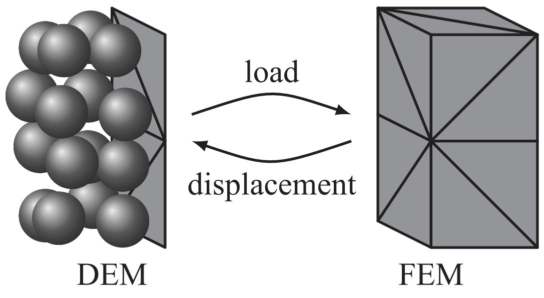

Fig. 4: Illustration of surface coupling.

3.3. Volume coupling

Volume coupling is another possible combination of FEM and DEM simulations. It is similar to the surface coupling, but considers certain overlap between both domains. The overlapping parts may have coinciding "vertices", but in general case the overlapping domains may be independent on each other.

Another variety of subtypes of volume coupling is produced by the enforcement of compatibility conditions, resulting in strong or weak coupling, see Xu et al. (2002) for more details. In the case of strong coupling, the overlapping particles are fixed to the elements, while in the other case (weak coupling), the compatibility and/or equilibrium conditions are satisfied in weak sense. MuPIF is designed such that similar strong/weak coupling may be easily implemented, see Patzák et al. (2012).

From the implementation point of view, the methodology is the same as in the case of surface coupling, i.e. subclassing `TransferOperator` (possibly covering all aforementioned volume coupling types) and creating independent agent for FEM – DEM volume coupling.

See figure 5 for illustration.

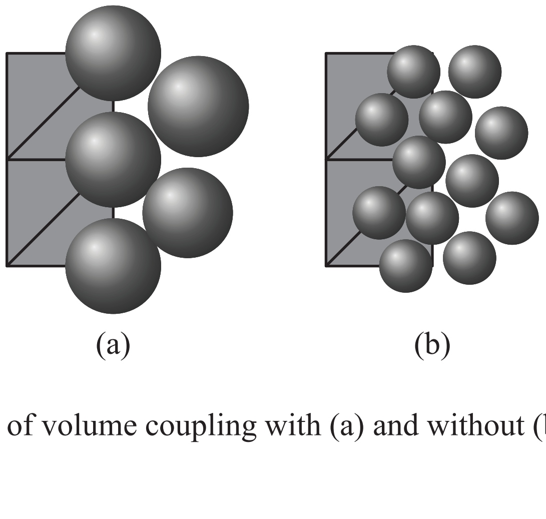

Fig. 5: 2D illustration of volume coupling with (a) and without (b) coinciding geometry.

3.4. Multiscale coupling

The last discussed FEM – DEM coupling strategy is multiscale coupling. It uses different solution strategies on different scales. On higher (macro) scale, it uses FEM as a numerical solution tool, while using DEM for lower (micro) scale.

As an example of such approach, consider a sample of sand. In reality, it is composed of individual grains, therefore DEM could be the right modeling approach. However, because of very high computational costs of DEM, the sand is considered as continuum from macroscopic point of view and FEM is used for macroscopic description. To preserve particular nature of the sample, stress – strain law in each integration point is determined not from predefined formulas, but rather from microscale simulations performed on smaller sand samples solved by DEM, see figure 6.

From the implementation point of view, design of such approach is already implemented by Patzák et al. (2012), so the DEM code itself only needs to support periodic boundary conditions and stress and stiffness estimation. Again, as a final implementation step, new agent for FEM – DEM multiscale coupling will be created.

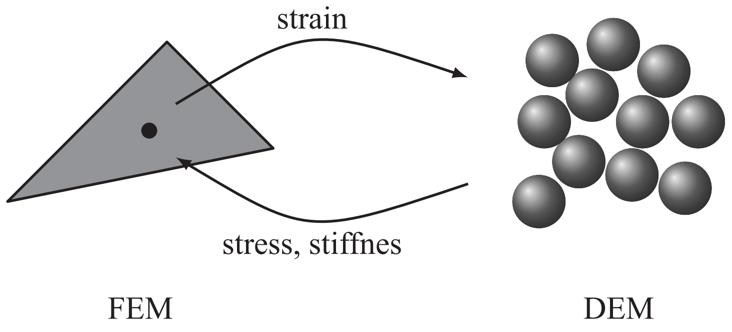

Fig. 6: 2D illustration of multiscale coupling.

4. Conclusion and future work

A new design of integration of discrete-based methods (specifically the discrete element method) into MuPIF framework was proposed. It covers all major DEM–FEM coupling strategies (surface, volume and multiscale), providing for each strategy its "agent" (simple one–purpose application) and related derived classes. The design is focused on independence on particular program and only features of the discrete element method itself were considered.

Despite the effort of the author, the implementation and results obtained with the proposed methods have not been completed before this paper deadline. However, they will be presented during the conference and on web pages of related projects MuPIF (`mech.fsv.cvut.cz/mupif`), OOFEM (`oofem.org`) and YADE (`yade-dem.org`) as soon as they are finished.

5. Acknowledgment

The financial support of project SGS13/034/OHK1/1T/11 is gratefully acknowledged.

6. References

Munjiza, A. (2004) *The combined finite-discrete element method*. Wiley, Chichester.

Oñate, E. & Rojek, J. (2004) Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Computer Methods in Applied Mechanics and Engineering*, vol. 139, 3087–3128.

Patzák, B., Rypl, D. & Kruis, J. (2012) MuPIF – A distributed multi-physics integration tool. *Advances in Engineering Software*, Available online 3 November 2012, http://www.sciencedirect.com/science/article/pii/S0965997812001329.

Rojek, J. & Oñate, E. (2007) Multiscale analysis using a coupled discrete/finite element model. *Interaction and Multiscale Mechanics*, vol. 1, no. 1, 1–31.

Rousseau, J., Frangin, E., Marin, P. & Daudeville, L. (2009) Multidomain finite and discrete elements method for impact analysis of a concrete structure. *Engineering Structures*, vol. 31, 2735–2743.

Villard, P., Chevalier, B. & Le Hello, B. & Combe, G. (2009) Coupling between finite and discrete element methods for the modelling of earth structures reinforced by geosynthetic. *Computers and Geotechnics*, vol. 36, 709–717.

Wellmann, C. & Wriggers, P. (2012) A two-scale model of granular materials. *Computer Methods in Applied Mechanics and Engineering*, vol. 205–208, 46–58.

Xu, M., Gracie, R. & Belytschko, T. (2002) Multiscale modeling with extended bridging domain method. In: *Bridging the Scales in Science and Engineering* (J. Fish ed.). Oxford University Press.