



HAL
open science

LMGC90

Frédéric Dubois, Rémy Mozul

► **To cite this version:**

Frédéric Dubois, Rémy Mozul. LMGC90. 11e colloque national en calcul des structures, CSMA, May 2013, Giens, France. hal-01717115

HAL Id: hal-01717115

<https://hal.science/hal-01717115>

Submitted on 25 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

LMGC90

F. Dubois^{1,2}, R. Mozul^{1,2}

¹ LMG, Université Montpellier 2, France, {frederic.dubois,remy.mozul}@univ-montp2.fr

² Laboratoire MIST, IRSN - CNRS - Université de Montpellier 2

Résumé — This paper gives a short overview of LMG90 platform in terms of architecture and main features. Some examples illustrate its possibilities.

Mots clés — contact, dynamics, multiple physics, DEM, FEM, OpenSource, Fortran, Python.

1 Introduction

LMGC90 is an open platform dedicated to the modeling of large collections of interacting objects (2D/3D). It aims at modeling objects of any shape with various mechanical behavior and to take into account interaction laws as complex as necessary. Furthermore multi-physics couplings (thermal effects, fluids, *etc*) are progressively taken into account.

LMGC90 is designed as a research software which offers to developers the possibility to add new physical models (behavior law, interaction law, *etc*), numerical models (finite element, natural element, *etc*), technical features (contact detection, visualization, parallelism, *etc*) and numerical strategies (time integration, numerical solver, *etc*). Furthermore end-users are able, through the supervisor, to interact with the internal database and to use/modify these information during the simulation.

As all research software, versatility of LMG90 software has been obtained designing a dedicated architecture, however it needs to evolve permanently with the new scientific requirements. In the present paper we refer to the architecture of the second version.

2 Architecture

As summarized on FIGURE 1, LMG90 is divided into independent parts :

- **Core** which is dedicated to the simulation itself. The **Core** of LMG90 software is written in Fortran9x, which language is widely known as computationally efficient. **Core** provides a Fortran9x public interface allowing to drive a simulation and access to the database.
- a supervising part, which relies on the **Core** public API. The common usage is performed through the **ChiPy** Python module which is build on a C API provided by a Fortran200x wrapping of the **Core** public API. An advanced-user can directly write a standalone program calling the Core public API or can build new functionality which can be added to the standard **ChiPy** module. A **Sandbox** mechanism is proposed to manage these developments (compiling, versioning, *etc*).
- **Bindings** which embed full external libraries. A generic API has been defined for sets of key features on the **Core** side while a specific implementation, to fulfill the desired functionality based on an existing library, is done on the **Bindings** side. Existing bindings concern linear algebra (use of LAPACK or MUMPS), bulk behavior (Matlib developed by Laurent Stainier), contact detection algorithms, contact solvers (SICONOS/Numerics) and finite element library (PELICANS).
- **User module** which permits to define, through Fortran9x functions, some very specific possibilities like user material.

2.1 Core

Core relies on various Fortran9x modules. A module gathers type definitions and methods working on data of these types. Moreover modules also hold data of these defined types in order to limit side effects.

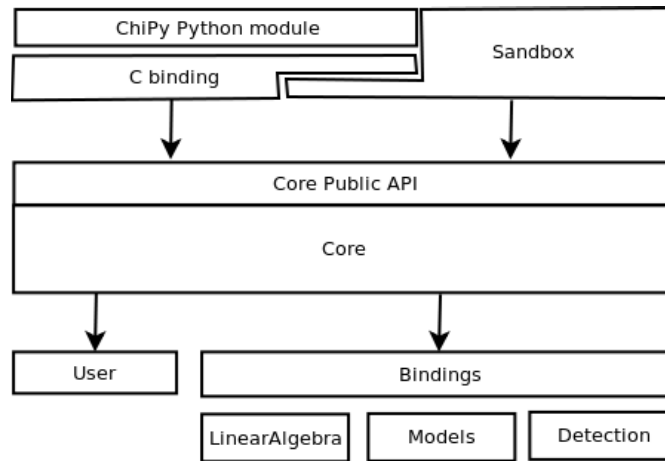


Fig. 1 – LMGC90 modules

Usually the data and types are private, and a set of public functions are available to use the module. Thus a structure close to an object (in the Oriented Object Programming sense) is obtained but without polymorphism or inheritance.

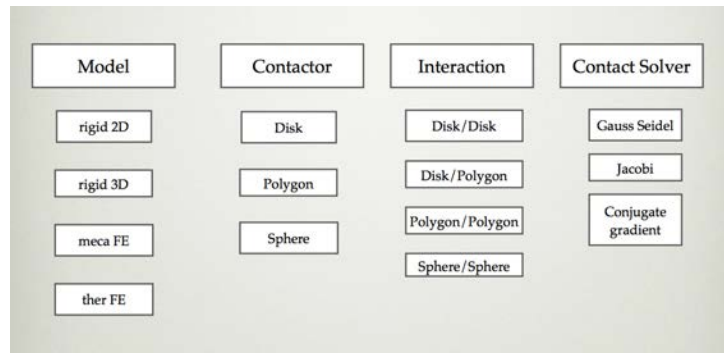


Fig. 2 – Current architecture of LMGC90 software

The general architecture of the *Core* and its on-going evolution is described in [14]. Currently the software is roughly organized around four main components as described on figure 2 :

- **Model** : the modeling of the different bodies to simulate. It relies on a suitable spatial discretization. The model also holds the corresponding linear system, which depends on the numerical strategy used (time integration for example), to compute the evolution of its state due to external loads.
- **Contactor** : LMGC90’s naming of the geometric description of contact locus. Each **Contactor** belongs to only one **Model**, but one **Model** can own several **Contactors** allowing to define cluster of particles for example.
- **Interaction** : define the contact detection algorithms for pairs of **Contactors** of a given type. Each detection generates a list of potential contacts describing the contact locus, the local frame, mappings with objects degrees of freedom and the contact law to be used.
- **Contact Solver** : the different algorithms available to solve the contact problem.

The different modules representing these components do not formally exist since they are too generic to be implemented in an efficient way with only one module. For each component a set of modules is used instead. For example it exists various **Model** modules : 2D rigid, 3D rigid and mechanical, thermal or porous finite element. In the same way there are more than 20 **Contactor** modules and 28 **Interaction** modules. There are currently 4 **Contact Solver** modules implemented in LMGC90.

The kernel of the software is the **Contact Solver** part. A contact solver works on a list of contacts. Each contact, thanks to its contactors, is related to supporting models. Thus, when the contact solver modify

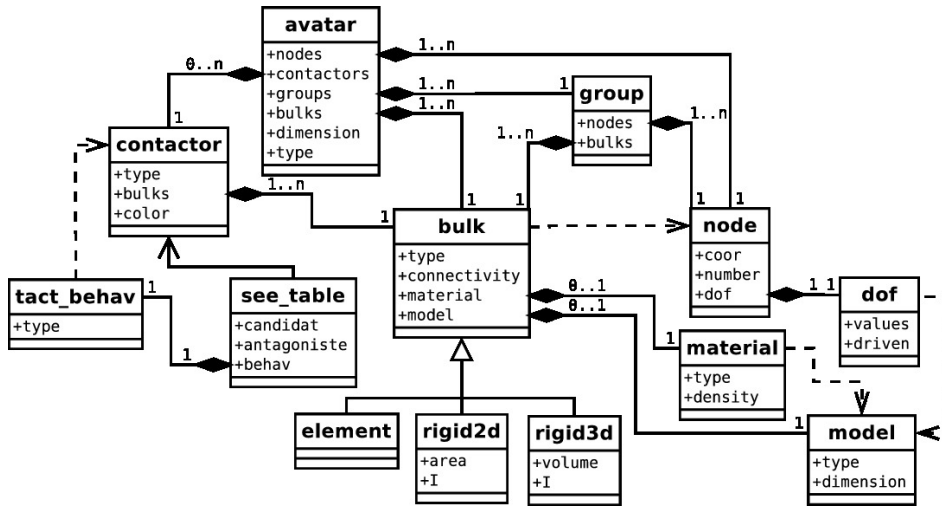


Fig. 3 – Class diagram of the pre-processor

the interactions, the state of the supporting models is updated.

2.2 Generating a Python interface

Plugging Fortran code into Python is usually seen as an awkward problem, despite `f2py`¹ which aims to automatically generate a python interface to Fortran source code. Unfortunately `f2py` needs to determine which compiler and compiler version is available which raises technical issues. Furthermore it is impossible to add docstrings (Python documentation) to the generated module. Therefore we have decided to use a new feature of the Fortran 200x norm which allows to automatically bind a Fortran routine in C through the use of the `ISO_C_BINDING` module. Adding a header file, which describes the prototypes of the routines, allows to use `SWIG`², an automatic code conversion tool [3]. Combining this approach with the use of `doxygen` permits to generate the docstrings.

3 Main features

LMGC90 offers a framework to describe 2D/3D multi-body/multi-contact mechanical systems.

LMGC90 is driven by Python scripts ; no graphical user interface are provided yet. The various features are described in the following.

3.1 Pre-processing

A Python module provides functionality to define the various aspects of the considered system. A specific python set of classes was designed for this purpose, see figure 3. Therefore each object, independently of its model, of the system is described in the same way.

Considering large collection of objects raises issue since automatic and parametric tools are mandatory (to generate shapes and compute initial position respecting some constraints for example) and handling details are necessary (setting some parameter in a given zone) ; in such situation scripting is powerful.

We have developed functionality allowing to build geometry of granular material, masonry structure or rock mass considering either rigid or deformable objects. A rigid object geometry is either defined using built-in primitives or using a mesh.

The geometry of a deformable object is defined using a mesh which can be generated using built-in functions or loaded from an external file.

1. <http://www.scipy.org/F2py>

2. <http://www.swig.org/>

Once geometry is defined the data necessary for each numerical model (an avatar) are : material properties, numerical modeling options, boundary conditions, *etc.* More details are given in [7].

3.2 Modeling

- **Bulk behavior** of objects (refers to model in previous section) may be considered as rigid or deformable (linear or non linear) by means of finite element. Behavior parameters can be set using fields and may depend on external fields.
Various physics are available : mechanics, heat transfer, saturated porous media, *etc.* Coupled behaviors can be considered. Coupling with external models are possible (*e.g.* fluid dynamics). In case of multiple discretizations of the same object or physics coupling at different scales collocation/projection methods are available.
- **Contactors**, which define the potential locus of contacts may be described by simple convex primitive (disk, polygon, sphere, polyhedron, *etc.*), cluster of primitives or meshes.
- **Contact detection** is performed for almost any combination of **Contactors**. Detection is performed in two or three stages : rough, broad and local. Rough detection aims at giving a list of neighbor objects using their bounding box or sphere. Broad detection aims at cleaning the neighbor list in obvious case ; it is necessary when considering polyhedron. Fine detection computes contact locus, frame, *etc.*
When considering rigid bodies with convex polyhedron contactor an algorithm based on common plane is used.
- **Interaction laws** describe the contact behavior. A large set of laws is available, *e.g.* frictional contact, cohesion (capillarity, damage, brittle, *etc.*), wire, rod. Heat transfer at contact may be taken into account.

Once models are loaded in LMGC90 various possibilities are offered to check the validity of the model (introspection, visualization) and to correct it.

3.3 Simulating

LMGC90 dynamics multi-contact modeling strategy is mainly based on the **Non Smooth Contact Dynamics** method : non smooth dynamics framework, implicit time integration, implicit contact solvers (NLGS, PCG, *etc.*). See [11] for details on this method. Python supervisor is very useful to finely drive such strategy.

Thanks to the modular architecture other strategies are or may also be implemented (molecular dynamics strategy, explicit time integration, quasi-static time evolution, *etc.*).

Applications with a large number of interacting bodies are reachable through parallel computing (Shared or distributed memory).

A non stationary thermal or diffusion solver is available.

During computation it is possible to check the relevance of the analysis through some global indicators (convergence norm, quality of interaction laws computation, *etc.*).

3.4 Post-processing

LMGC90 generates two categories of post-processing files of its simulation results :

- xml-vtk visualization files. These files describe the state of objects with various physical nodal fields, contact networks, Gauss points with their values, *etc.* Visualization may be performed with any tool supporting xml-vtk format as Paraview, Mayavi or Visit.
- multi column text files. These files contain the result of analyses performed on all or part of the studied samples as contact orientation distribution, total dissipated energy, *etc.* They are suitable for any kind of plotting tool (xmgrace, Matplotlib, gnuplot, *etc.*).

Through the Python interface its also possible to access to various data which can be analyzed using built-in Python functions.

3.5 An open platform

LMGC90 development relies on collaborative programming tool (subversion & trac)³. The sources are provided under CECILL license (*i.e.* GPL).

The LMG90 software is known to successfully run on Linux and MacOs operating systems. The pre-requisites are widely used packages found within the classic package managers (dpkg, apt-get, yum, port). The compilation process relies on a CMake⁴ build; all necessary steps (including downloading with subversion) for a basic installation are gathered in a simple shell script available on the website. Access to advanced features like parallel computing requires a cmake tuning "by hand".

Such an open platform enables coupling capabilities with external software (Finite Element code : Code_Aster, Pelicans) and libraries (Physical Models : MatLib, Linear Algebra : Lapack, Contact detection : Rapid, *etc.*). Many of these coupling possibilities are enlighten in [15]. The LMG90 developer team is part of the Saladyn project (funded by ANR)⁵ and the Degrip project (funded by Oseo/FEDER).

4 Examples of Application

4.1 Granular Material from rheology to structure

A typical use of LMG90 concerns the study of the rheology of granular materials. As an example FIGURE 4 shows two types of results concerning the deformation of a sample made of 40,000 polyhedrons (8 faces). On the left a detail of the packing is given ; the grey scale corresponds to the coordination number. On the right the contact network is represented ; line thickness is proportional to the force. The strong network is in grey and the weak in red.

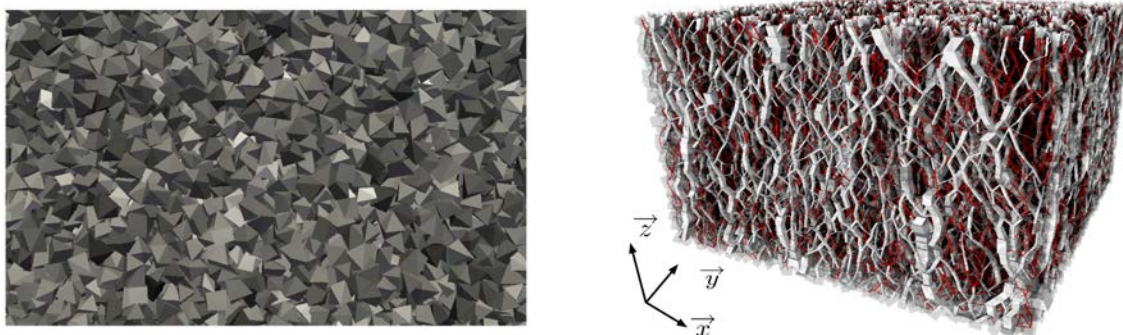


Fig. 4 – Tri-axial compression of a polyhedral sample [13]

Basic features help with studying influence of shapes, interaction laws and bulk physics on the behavior of a collection of objects.

4.2 Masonry structures

A typical example is the stability study of ancient monument as Nîmes arena. On FIGURE 5 block geometry details of the and computed pressure into the joints due to dead load are given. This example was built using the Roman's conception rules. This kind of modeling allows assessing the stability and the safety of masonry structures under static loads or dynamical natural risk (seism [17], landslide, *etc*) taking into account the influence of the design pattern and the joint behavior.

3. https://subver.lmgc.univ-montp2.fr/trac_LMG90v2

4. <http://www.cmake.org>

5. <http://saladyn.gforge.inria.fr>

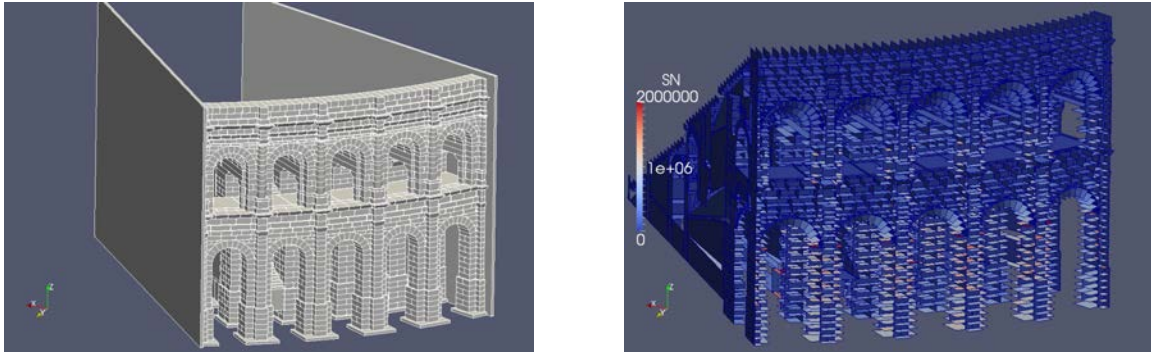


Fig. 5 – Modeling of Nîmes arena. Left the geometry. Right the pressure between blocks once at rest

See presentations [18] and [19] for more details.

4.3 Modeling Fracture

LMGC90 allows to model fracture in various media. FIGURE 6 shows (left) fracture of an heterogeneous media under horizontal traction load [8] and (right) the study of stability of a rock mass under self weight load [12]. Using a Frictional Cohesive Zone Model, fracture can be modeled, at microscopic or meso-

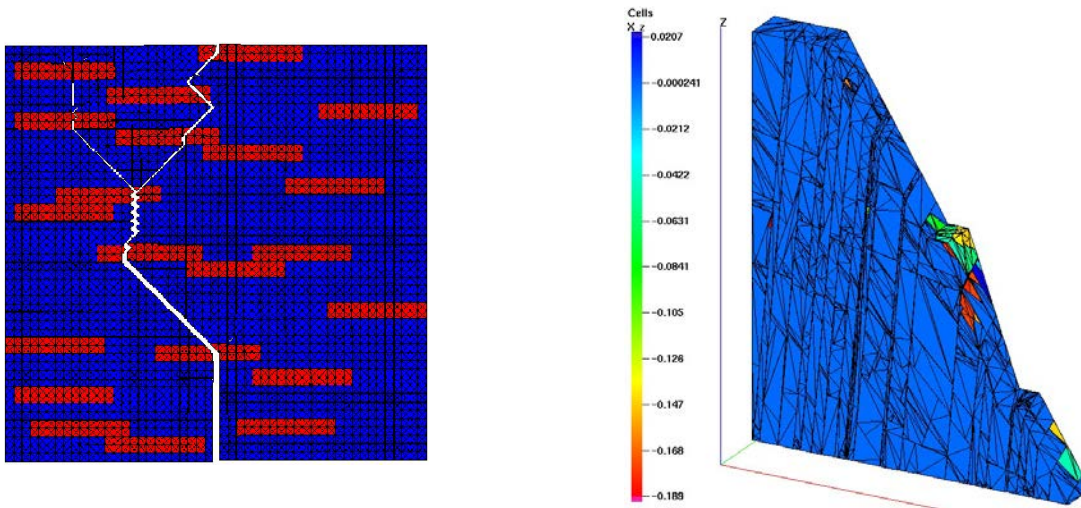


Fig. 6 – fracture modeling examples

scopic scale, from initiation to post-failure.

Considering rock mass as a fractured media, avalanches on natural or mining slopes, stability tunnels, *etc* can be studied.

See presentations [20] and [21] for more details.

4.4 Multiple physics couplings

LMGC90 allows to consider various physics at different scales such as thermal coupling [11], fluid particle interaction [9, 10], electrical conductivity [5]. FIGURE 6 shows (left) Sedimentation of thin particles in gas and (right) an immersed avalanche of a loose granular assembly in a small closed box.

See presentations [20] and [21] for more details.

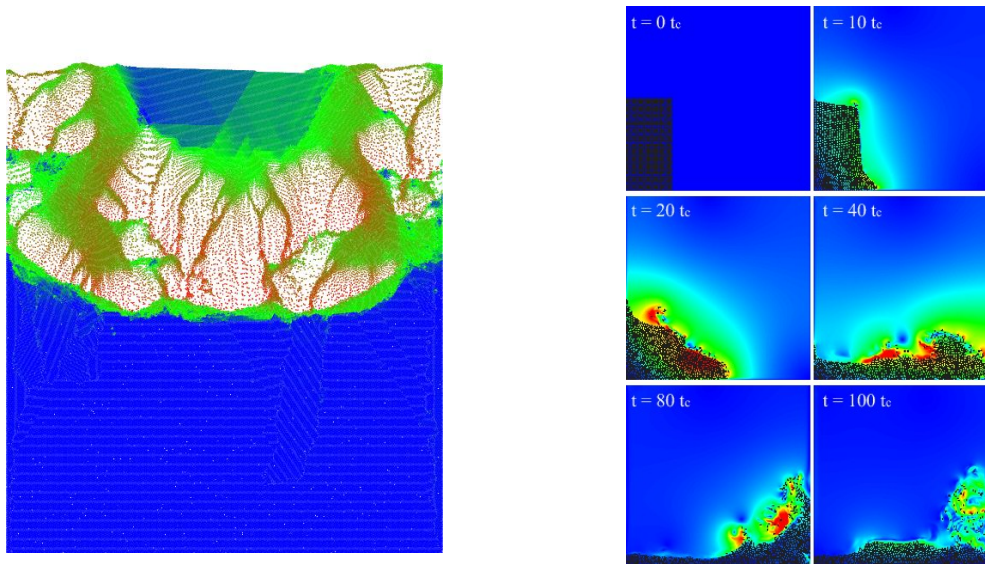


Fig. 7 – coupling modeling examples

5 Conclusion

Started more than 10 years ago, the LMGC90 project is becoming more and more active from a research or application point of view.

The nowadays challenges are multi-physics and multi-representation either to perform the switch of an object models (rigid - deformable) or to reduce-refine a part of a structure made of several objects.

The price to pay is a constant evolution of the overall software architecture and an improvement of the core technology.

Fortunately the Open-Source world provides a bunch of impressive software libraries that can be used.

Acknowledgements :

The authors are grateful to all the LMGC90's contributors and specially : M. Jean, M. Renouf, A. Martin, P. Taforel, V. Visseq, D. Ambard, W. Saber-Cherif.

This work is supported by SALADYN ANR project <http://saladyn.gforge.inria.fr> of the program COSINUS ANR-08-COSI-014-01 and OSEO-FEDER (Degrip Project)

Références

- [1] M. Jean, J.J. Moreau. *Unilaterality and dry friction in the dynamic of rigid body collections*, Proc. Contact Mechanics Int. Symp., Edt A. Curnier, 31-48, 1992.
- [2] M. Jean. *The non-smooth contact dynamic method*, Computer methods in applied mechanics and engineering, 177, No 3-4, 235-257, 1999.
- [3] D. M. Beazley. *Automated scientific software scripting with SWIG*, Future Generation Computer Systems, 19(5) :599–609, 2003.
- [4] F. Dubois, M. Jean. *LMGC90 : une plateforme de développement dédiée à la modélisation des problèmes d'interaction*, Proc. 6ème Colloque National en Calcul des Structures, Giens, volume 1, 111-118, 2003.
- [5] M. Renouf, N. Fillot. *Coupling electrical and mechanical effects in discrete element simulations*, IJNME, 74(2), 238-254, 2008
- [6] A. Markus, M. Baudin, *FLIBS - A collection of Fortran modules*, <http://flibs.sourceforge.net/>, 2010
- [7] A. Martin, M. Bagnéris, F. Dubois, R. Mozul. *Conception d'un outil adapté à la mise en données des systèmes discrets*, Proc. 10ème Colloque National en Calcul des Structures, Giens, 2011.

- [8] F. Perales, F. Dubois, Y. Monerie, B. Piar, L. Stainier. *A NonSmooth Contact Dynamics-based multi-domain solver. Code coupling (Xper) and application to fracture*. European Journal of Computational Mechanics 19, 389-417, 2010.
- [9] A. Martin. *Ecoulement confiné d'un matériau granulaire en interaction avec un gaz, application à la relocation du combustible nucléaire*, Thèse de doctorat de l'Université de Montpellier 2, 2010.
- [10] V. Topin, F. Dubois, Y. Monerie, F. Perales, A. Wachs. *Micro-rheology of dense particulate flows : Application to immersed avalanches*. Journal of Non-Newtonian Fluid Mechanics 166 (1-2), 63-72, 2011.
- [11] F. Radjai, F. Dubois *Discrete Numerical Modeling of Granular Materials*, ISTE Ltd and John Wiley & Sons Inc, 2011.
- [12] A. Rafiee, M. Vinches, F. Dubois. *The Non-Smooth Contact Dynamics method applied to the mechanical simulation of a jointed rock mass*, ENOC Congress, 2011
- [13] E. Azema, F. Dubois, F. Radjai. *On the quasi-static behavior of granular material composed with irregular polyhedral particles : effects of grains angularity*, to be published.
- [14] F. Dubois, M. Jean, M. Renouf, R. Mozul, A. Martin, M. Bagneris. *LMGC90*, Proc. 10ème Colloque National en Calcul des Structures, Giens, 2011.
- [15] V. Acary, M. Brémond, H.P. Cao, F. Dubois, S. Mazet, R. Mozul. *The Salodyn toolbox*, Proc. 11ème Colloque National en Calcul des Structures, Giens, 2013.
- [16] H. Tran. *Combined application of structural geology, the mechanics of discrete media and the analysis of in situ stresses and displacements for the modelling behaviour of fractured rock masses*. Thèse de doctorat de l'Université de Montpellier 2, 2013.
- [17] P. Taforel. *Apport de la Méthode des Eléments Discrets à la Modélisation des Maçonneries en Contexte Sismique : Vers une Nouvelle Approche Numérique de la Vulnérabilité Sismique*, Thèse de doctorat de l'Université de Montpellier 2, 2012.
- [18] P. Taforel, F. Dubois. *Modélisation des maçonneries à joints de mortier par une approche discrète avec des blocs déformables*, Proc. 11ème Colloque National en Calcul des Structures, Giens, 2013.
- [19] A. Martin, V. Visseq, P. Taforel, F. Dubois. *Décomposition de domaine pour le calcul des maçonneries*, Proc. 11ème Colloque National en Calcul des Structures, Giens, 2013.
- [20] W. Saber Cherif, F. Dubois, A. Chrysochoos, L. Cangémi, N. Brusselle. *Modélisation hydro-mécanique de propagation de fissures dans un milieu poreux déformable : Application au stockage géologique de CO₂*, Proc. 11ème Colloque National en Calcul des Structures, Giens, 2013.
- [21] F. Dubois, Y. Monerie, F. Perales, V. Topin. *Xper et ApoloGRIFF : Des plateformes numériques pour l'étude des interactions multiphysiques entre corps*, Proc. 11ème Colloque National en Calcul des Structures, Giens, 2013.