Advances in Engineering Software 74 (2014) 40–48

Contents lists available at ScienceDirect

Advances in Engineering Software

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

The GranOO workbench, a new tool for developing discrete element simulations, and its application to tribological problems

Damien André [a,*], Jean-luc Charles [a], Ivan Iordanoff [a], Jérôme Néauport [b]

[a] Arts & Métiers ParisTech, I2M-DuMAS-MPI, UMR 5295 CNRS, F-33405 Talence, France
[b] Commissariat á l'Énergie Atomique, Centre d'Études Scientifiques et Techniques d'Aquitaine, BP 2, 33114 Le Barp, France

ARTICLE INFO

Article history:
Received 14 January 2014
Received in revised form 17 April 2014
Accepted 17 April 2014
Available online 20 May 2014

Keywords:
Software
C++
Object-oriented
Mechanics
Discrete element method
Tribology
Fracture

ABSTRACT

Discrete models are based on the descriptions of the physical states (e.g., velocity, position, temperature, magnetic momenta and electric potential) of a large number of discrete elements that form the media under study. These models are not based on a continuous description of the media. Thus, the models are particularly well adapted to describe the evolution of media driven by discontinuous phenomena such as multi-fracturation followed by debris flow as occurs in wear studies.

Recently, the use of discrete models has been widened to face problems of complex rheological behaviors and/or multi-physical behaviors. Multi-physical problems involves complex mathematical formulations because of the combination of different families of differential equations when a continuous approach is chosen. These formulas are often much simpler to express in discrete models, in which each particle has a physical state and the evolution of that state is due to local physical interactions among particles. Since the year 2000, this method has been widely applied to the study of tribological problems including wear (Fillot et al., 2007) [1], the thermo-mechanical behavior of a contact (Richard et al., 2008) [2] and subsurface damage due to surface polishing (Iordanoff et al., 2008) [3]. Recent works have shown how this method can be used to obtain quantitative results (André et al., 2012) [4]. To assist and promote research in this area, a free platform *GranOO* has been developed under a C++ environment and is distributed under a free *GPL* license. The primary features of this platform are presented in this paper. In addition, a series of examples that illustrate the main steps to construct a reliable tribological numerical simulation are detailed. The details of this platform can be found at http://www.granoo.org.

Introduction

Tribological phenomena involve a wide range of scales, typically from the nanometer (at the surface scale) to the meter scale (at the mechanism scale). From the physical point of view, mechanics must be coupled with thermal, material and physico-chemical behavior to understand these phenomena. To study such complex problems, adapted numerical tools must be used to understand and predict the contact behavior. The finite element method is often used for these problems, presenting advantages such as broadly available commercial software that is easy to use. However, it is difficult for the finite element method to describe multi-fracturation followed by debris, as occurs in wear studies [3].

Molecular dynamic methods are increasingly applied to the study of tribological problems [5–7]. Free software exists in that field that can be used by a large number of scientists. However, the simulated time and space scales are often small compared to the scales of tribological phenomena. Over the past ten years, discrete element models have been shown to be interesting tools that can take contact into account at the right scale and solve multi-physical problems. Unfortunately, discrete element commercial software is often restricted to a single application and is difficult to deal with complex problems. The consequence is that tribological studies using discrete element models are limited by software difficulties. The *GranOO* workbench has been developed to offer the scientific community in general and the tribological community in particular a free and rather easy-to-use discrete element software. This paper briefly outlines the main aspects of the discrete element model developed in *GranOO* and the primary features of the *GranOO* software. The final section illustrates the methodology used to construct a reliable tribological DEM simulation by describing a series of examples.

* Corresponding author. Tel.: +33 (0) 5 56 84 53 91; fax: +33 (0) 5 56 84 53 66.
E-mail address: damien.andre@u-bordeaux.fr (D. André).

http://dx.doi.org/10.1016/j.advengsoft.2014.04.003
0965-9978/

The explicit discrete element model

Discrete element methods describe the media as an assembly of a great number of interacting elements. The simplest approach, the lattice method, involves in the connection of a set of nodes by a joint with a given stiffness. Schlangen and Garboczi [8], Van Mier et al. [9] and Cusatis et al. [10] use this method to study cracks in concrete structures. The nodes have no mass and this method is not adapted to dynamic problems. The cracks do not generate surfaces and the method is not adapted to wear problems. Another type of discrete element method consists of the connection of a set of particles by joints and/or contact laws. These discrete elements are spherical [11,12] or polyhedral [13,14]. The contact laws can be derived from regular laws [11,13,14,12] or non-regular laws [15].

To enable the discrete elements to move, Newton's law can be solved using an explicit or implicit algorithm. The explicit approach was preferred to solve dynamic problems at small time scales. So, the implicit resolution and the non-regular mechanic are not available in the *GranOO* project. *GranOO* deals with spherical particles, regular contact laws and different types of cohesive joints (e.g., springs and beams) to link the discrete elements that belong to the same media in a three dimensional model. The model involves the explicit integration algorithm shown as Algorithm 1, where:

- t is the current time and dt is the integration time step,
- $p(t), \dot{p}(t)$ and $\ddot{p}(t)$ are the linear position, velocity and acceleration of the discrete element, respectively,
- $q(t), \dot{q}(t)$ and $\ddot{q}(t)$ are the angular position, velocity and acceleration of the discrete element, respectively. The concept of quaternions is used to describe these quantities [16, Section 2.5].

Algorithm 1. Explicit dynamic resolution

Require: $\vec{p}(0)\ \dot{\vec{p}}(0)\ \ddot{\vec{p}}(0)\ q(0)\ \dot{q}(0)\ \ddot{q}(0)$
  $t \leftarrow 0$
  for all iteration n **do**
    for all discrete element i **do**
      $\vec{p}_i(t+dt) \leftarrow$ Explicit integration
      $\vec{f}_i(t+dt) \leftarrow$ Sum of force acting on i
      $\ddot{\vec{p}}_i(t+dt) \leftarrow$ Newton second law
      $\dot{\vec{p}}_i(t+dt) \leftarrow$ Explicit integration
      $q_i(t+dt) \leftarrow$ Explicit integration
      $\vec{\tau}_i(t+dt) \leftarrow$ Sum of torque acting on i
      $\dot{q}_i(t+dt) \leftarrow$ Angular momentum law
      $\dot{q}_i(t+dt) \leftarrow$ Explicit integration
    end for
    $t \leftarrow t + dt$
  end for

This algorithm is particularly well adapted to tribological problems, presenting the following advantages:

- dynamic effects can be taken into account,
- easy and fast contact detection (between spheres),
- quantitative simulation of material behavior [4] and
- effects at small time scale can be considered by the explicit algorithm.

This method has proven to be an efficient way to face tribological problems such as wear or thermo-mechanical contact behavior [1,2,17]. An implementation of this algorithm is given in the processing time loop in Listing 3. The default integration scheme used by *GranOO* is the *velocity Verlet* scheme [18]. However, user can easily switch with another schemes thanks to custom plug-ins (see Section 'The plug-ins and the input XML file'). In addition, to increase the performance of this sequential algorithm many parallelisation strategies can be used: force decomposition, particle decomposition or domain decomposition [19]. At this time, *GranOO* uses particle decomposition method associated to multithreading techniques. User can choose, for the most time-consuming processes, the number of threads to execute in parallel. It is done through the input file thanks to the keyword *ThreadNumber* (see Listing 3 and Section 'The plug-ins and the input XML file').

```
<GranOO Version="1.0">

  <ComputeProblem TotIteration="20000"/>
  <SampleFile File="cylinder.gdd" />

  <PreProcessing>
    <PlugIn Id="ConvertBondToBeam3D"
            YoungModulus="109e9"
            RadiusRatio="0.6386"/>

    <PlugIn Id="SetDensity3D" Value="3000"/>
    <PlugIn Id="ComputeOptimalTimeStep3D"/>
    <PlugIn Id="InitSensor"/>
  </PreProcessing>

  <Processing>
    <PlugIn Id="Check3D" />
    <PlugIn Id="ResetLoad3D" />
    <PlugIn Id="ApplyLoad3D" />
    <PlugIn Id="ApplyBondLoad3D" ThreadNumber="2"/>
    <PlugIn Id="IntegrateAccelerationLinear3D" />
    <PlugIn Id="IntegrateAccelerationAngular3D"/>
    <PlugIn Id="ApplyBoundaryCondition3D" />
    <PlugIn Id="SaveDomain3D" IterLoop="1000"/>
    <PlugIn Id="WriteSensorData3D"/>
  </Processing>

  <PostProcessing>
  </PostProcessing>

 <MathFunction Id="Load">
    <RampAndConstant Limit="10000" Constant="50e3"
                     VariableRef="Iteration"/>
 </MathFunction>

  <Load Id="Tension" DiscreteElement3DSet="Right">
    <TotalForce>
      <Vector3D X="Load" Y="0." Z="0."/>
    </TotalForce>
  </Load>

  <BoundaryCondition Id="Fix" DiscreteElement3DSet="Left">
    <Displacement>
      <Vector3D X="0." Y="0." Z="0."/>
    </Displacement>
  </BoundaryCondition>

</GranOO>
```

Listing 3. The input XML file for the tension test.

The discrete element method developed in *GranOO* is based on the following physical description:

- the discrete element stores the mass, volume, acceleration, velocity and position and is able to store other physical properties such as magnetic moment, electric potential or temperature,
- the joint stores the rheological behavior,
- the contact stores the interface behavior.

The next section will show how this physical description has been translated into the C++ platform *GranOO*.