



Application of the Discrete Element Method to Finite Inelastic Strain in Multi-Materials

Robin Gibaud

► To cite this version:

Robin Gibaud. Application of the Discrete Element Method to Finite Inelastic Strain in Multi-Materials. Mechanics of materials [physics.class-ph]. Université Grenoble Alpes, 2017. English. NNT : 2017GREAI086 . tel-01761756

HAL Id: tel-01761756

<https://theses.hal.science/tel-01761756>

Submitted on 9 Apr 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.

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : 2MGE : Matériaux, Mécanique, Génie civil,
Electrochimie

Arrêté ministériel : 25 mai 2016

Présentée par

Robin GIBAUD

Thèse dirigée par **Luc SALVO**, Professeur, Grenoble INP

préparée au sein du **Laboratoire Science et Ingénierie des
Matériaux et Procédés**
dans l'**École Doctorale I-MEP2 - Ingénierie - Matériaux,
Mécanique, Environnement, Energétique, Procédés,
Production**

Application de la méthode des éléments discrets aux déformations finies inélastiques dans les multi-matériaux

Application of the Discrete Element Method to Finite Inelastic Strain in Multi-Materials

Thèse soutenue publiquement le **28 novembre 2017**,
devant le jury composé de :

Monsieur Luc SALVO

Professeur des Universités, Grenoble INP, Directeur de thèse

Monsieur Samuel FOREST

Directeur de Recherche, CNRS, Rapporteur

Monsieur André KATTERFELD

Professeur, Otto-von-Guericke-Universität Magdeburg, Rapporteur

Madame Dominique POQUILLON

Professeur, INP-ENSIACET, Président

Monsieur Bruno CHAREYRE

Maître de Conférences, Grenoble INP, Examineur

Monsieur Pierre LHUISSIER

Chargé de Recherche, CNRS, Co-directeur de thèse



“Je sers la science et c’est ma joie.”
Basile, *discipulus simplex*

À Mohammed Colin-Bois, et tous les autres.

General Abstract

Forming of multiphase materials involves complex mechanisms linked with the rheology, morphology and topology of the phases. From a numerical point of view, modeling such phenomena by solving the partial differential equation (PDE) system accounting for the continuous behavior of the phases can be challenging. The description of the motion and the interaction of numerous discontinuities, associated with the phases, can be conceptually delicate and computationally costly. In this PhD, the discrete element method (DEM) is used to phenomenologically model finite inelastic strain in multi-materials. This framework, natively suited for discrete phenomena, allows a flexible handling of morphological and topological changes.

Ad hoc attractive-repulsive interaction laws are designed between fictitious particles, collectively rearranging to model irreversible strain in continuous media. The numerical behavior of a packing of particles can be tuned to mimic key features of isochoric perfect viscoplasticity: flow stress, strain rate sensitivity, volume conservation. The results for compression tests of simple bi-material configurations, simulated with the DEM, are compared to the finite element method (FEM) and show good agreement. The model is extended to cope with tensile loads. A method for the detection of contact and self-contact events of physical objects is proposed, based on a local approximation of the free surfaces.

The potential of the methodology is tested on complex mesostructures obtained by X-ray tomography. The high temperature compression of a dense metallic composite is modeled. The co-deformation can be observed at the length scale of the phases. Two cases of “porous” materials are considered. Firstly, the simulation of the compression and the tension of aluminum alloys with pores is investigated. These pores stem from the casting of the material, their closure and re-opening is modeled, including the potential coalescence occurring at large strain. Secondly, the compression of a metallic foam, with low relative density, is modeled. The compression up to densification involves numerous interactions between the arms.

Résumé global

La mise en forme de matériaux multiphasés comprend des mécanismes complexes en lien avec la rhéologie, la morphologie et la topologie des phases. Du point de vue numérique, la modélisation de ces phénomènes en résolvant les équations aux dérivées partielles (EDP) décrivant le comportement continu des phases n'est pas trivial. En effet, de nombreuses discontinuités associées aux phases se déplacent et peuvent interagir. Ces phénomènes peuvent être conceptuellement délicats à intégrer au modèle continu et coûteux en termes de calcul. Dans cette thèse, la méthode des éléments discrets (DEM) est utilisée pour modéliser phénoménologiquement les déformations finies inélastiques dans les multi-matériaux.

Des lois d'interactions attractive-répulsive sont imposées à des particules fictives, dont les ré-arrangements collectifs modélisent les déformations irréversibles de milieux continus. Le comportement numérique des empilements de particules est choisi pour reproduire des traits caractéristiques de la viscoplasticité parfaite et isochore: contrainte d'écoulement, sensibilité à la vitesse de déformation, conservation du volume. Les résultats d'essais de compression de bi-matériaux simples, simulés avec la DEM, sont comparés à la méthode des éléments finis (FEM) et sont en bon accord. Le modèle est entendu pour pouvoir supporter des sollicitations de traction. Une méthode de détection de contacts et d'auto-contacts d'objets physiques est proposée, basée sur l'approximation locale des surfaces libres.

Les capacités de la méthodologie sont testées sur des mésostructures complexes, obtenues par tomographie aux rayons X. La compression à chaud d'un composite métallique dense est modélisée. La co-déformation peut être observée à l'échelle spatiale des phases. Deux cas de matériaux "poreux" sont considérés. Premièrement la simulation de la compression puis traction d'alliages d'aluminium présentant des pores de solidifications: leur fermeture et ré-ouverture mécanique est modélisée, y compris leur coalescence à grande déformation. Deuxièmement la simulation de la compression de mousse métallique de faible densité: la compression jusqu'à densification provoque de nombreuses interactions entre les bras de matière.

Contents

1	General Introduction	11
1.1	Introduction	11
1.2	Outline	12
I	Mechanical and Numerical Context	15
2	Experimental Background	19
2.1	Metallic Composites	19
2.2	Amorphous/Crystalline Composites	22
2.3	Design of a Model Material	30
2.4	X-Ray Tomography	33
2.5	PhD Objective: Observed Physical Phenomena	35
3	Simulation Background	37
3.1	Limits and Contributions of Modeling Approaches	37
3.2	Description of the Studied Phenomena	47
3.3	PhD Objective: Requirements for a Modeling Tool	51
II	Review of Modeling Strategies	53
4	Key Modeling Choices	57
5	Lecture Grid: Lagrangian Methods	61
5.1	Overview	62
5.2	Continuous Constitutive Law	64
5.3	Discrete Constitutive Law	70
6	Comparison of Selected Methods	75
6.1	FEM, EFG, SPH, DEM and lattice model	75
6.2	Ongoing Challenges	77
III	Modeling Approach	79
7	PhD Objective: Research Question and Strategy	83
8	The Discrete Element Method	85
8.1	Neighbor Detection	86
8.2	Interaction Forces	87

8.3	Time Integration	88
8.4	Boundary Conditions	92
8.5	Parallel Computing	94
8.6	Control Metrics	95
9	Principle of the Developed Method	99
9.1	Finite Inelastic Transformation	101
9.2	Boundary Conditions	104
9.3	Discretization of Continua	105
9.4	Contact Event Detection	106
9.5	Self-Contact Event Detection	107
10	Chosen Numerical Tools	111
10.1	DEM Tools	112
10.2	FEM Tools	113
IV	Compression of Dense Bi-Material	115
11	Single Material	119
11.1	Interaction Law Choice	119
11.2	Calibration Procedure	120
11.3	Verification	124
11.4	Macroscopic Behavior	127
12	Bi-Material Test Cases	131
12.1	FEM Reference	131
12.2	Parallel Configuration	132
12.3	Series Configuration	133
12.4	Spherical Inclusion Configuration	134
13	Complex Multi-Material Mesostructure	139
13.1	Computation on a Full Sample	139
13.2	<i>In Situ</i> Configurations	141
V	Tension-Compression of “Porous” Material	147
14	Dense Material	151
14.1	Interaction Law Choice	151
14.2	Macroscopic Behavior	152
14.3	Influence of the Number of Particles	158
15	Self-Contact Detection	161
15.1	Threshold Choice	161
15.2	Compression-Tension of a Spherical Pore	166
16	Complex “Porous” Mesostructures	169
16.1	Casting Pores in Aluminum Alloy	169
16.2	Aluminum Open Cell Foam	173

General Conclusion	189
17 General Conclusion	191
17.1 General Objective	191
17.2 Modeling Strategy	192
17.3 Results and Applications	193
17.4 Limits and Specificities	196
17.5 Potential of the Model	198
 Appendices	 203
A Local Field	205
B DEM Implementation	209
B.1 Implementation Choices and Issues	209
B.2 Sources: Interaction Law <i>BILIN</i>	210
B.3 Sources: Interaction Law <i>TRILIN</i> , Contact and Self-Contact	212
B.4 Typical Input Script	222
C FEM Script	227
D IJMS Article	231
Notation	259
Glossary	261
Bibliography	265

Chapter 1

General Introduction

1.1 Introduction

This PhD is focused on the exploratory development of a numerical method to model the mechanical behavior of metallic alloys under large strain. Most of the efforts were dedicated to numerical issues arising in the design of the model. However, the inspiration of the work stems from practical and experimental issues. The study is aimed at the understanding of the deformation at the scale of the mesostructure of architected materials.

Historically, the development of materials for structural applications allowed the fulfillment of increasingly demanding requirements. The limits for both service and manufacturing requirements have been pushed further by the introduction of novel materials and the improvement of existing ones. Since the 1960s, and the rise of *composite* materials [84, p.321], an active axis of development is the association of distinct phases. The design of composites typically allows compromises between mechanical properties, and up to some extent contrives contradictory behaviors.

In the *architected* materials, the development is focused on the control of the morphology, the distribution and the topology of the phases. Indeed, such materials assemble several monolithic materials, or materials and empty space, to meet challenging functional requirements. More specifically, metallic architected materials display promising properties for structural applications. Complex and controlled architectures can be elaborated using for example casting, powder technology or additive manufacturing.

The study of large plastic and viscoplastic strains in architected materials are of major interest, both from an engineering and scientific point of view. Indeed, such finite inelastic strains can be observed at different stages of the life cycle of a product: during the forming processes (e.g. hot forming) or during service (e.g. shock absorption). The prediction of the mechanical behavior is thus useful in engineering contexts. In addition, the macroscopic behavior is driven by phenomena occurring at the scale of the mesostructure of the constitutive phases, under active scientific investigation:

- The motion and the interaction of the interfaces between phases;
- Topological changes like pore closure or phase fragmentation.

In the objective of describing and understanding these physical phenomena, simulation and experiment are complementary and intimately interlaced tools. Grassroots reference in most scientific queries, the experimental approach is a decisive tool to identify dominant processes and choose a conceptual description of a phenomenon. In the case of finite strain

of multi-materials, the deformation mechanisms of the phases can be temporally tracked in three dimensions by techniques as X-ray tomography.

However, the observation of physical systems implies a high level of complexity and heavy limitations on the control of the experimental configuration. The careful design of experimental setups can partially decouple phenomena and isolate the effects of distinct parameters. Thus, a judicious choice of materials can help to decorrelate the effects of morphology or rheology in the deformation mechanisms of multi-materials.

Built and developed in parallel to the experimental route, numerical modeling is a valuable tool to arbitrarily and independently study distinct mechanisms, allowing a fine control on parameter and configuration repeatability. Nevertheless, phenomena involving numerous and massively interacting interfaces are still challenging for simulation tools.

This PhD, starting from a concrete material science example, is thus focused on the development of a modeling method tailored to the description of finite strains in metallic multi-materials. Taking a step aside from more mainstream strategies, first of which the FEM, the proposed model is phenomenological: continuous media are numerically discretized using sets of interacting particles. The re-arrangement of these particles are expected to mimic typical traits of the deformation of the continuum. The developed framework, based on the DEM, is trusted to allow flexible handling of interface interactions and topological events. Its application to the simulation of continuous media is less straightforward, thus concentrating the development efforts.

Although the DEM is now a well established tool for the simulation of elastic and brittle behaviors of continuous media, the design of the proposed model was exploratory. To our knowledge, no anterior DEM algorithm was applicable to inelastic strain in incompressible material. The unconventional and phenomenological characteristics of the approach triggered a specific emphasis on the delimitation of the modeling scope.

1.2 Outline

The manuscript is organized in five main parts, each of which is opened by an illustrated page of selected highlights:

- Part I presents an overview of the general context of the PhD, both from an experimental and a modeling point of view.

The chosen reference experiment is described along with a brief review of the material properties and the experimental techniques. The developed simulation method being rather unconventional, the limits of the modeling approach are discussed. The phenomena of interest and their conceptual description are presented.

- Part II reviews some potential numerical modeling strategies.

A specific effort to propose a comparative lecture grid of numerical methods aims to better locate the developed approach. Selected methods are compared, from the algorithmic and conceptual point of view, with a focus on Lagrangian kinematics.

- Part III focuses on the research question.

Based on the idealization of the studied phenomena (Part I) and the potential resolution strategies (Part II), the principle of the developed methodology is presented, along with the generic framework of the discrete element method (DEM) and the chosen numerical tools.

- Part IV details the effective development of the method for the simulation of the compression of dense multi-materials with viscoplastic behavior.

The calibration and setup of the model is first presented for a single material. Bi-phased test cases are compared to finite element method (FEM) simulation and the mesostructure from a full 3D sample is modeled.

- Part V extends the method to compressive and tensile loading on “porous” materials with low strain rate sensitivity. The self-contact events, i.e. when the pores close, are taken into account.

The behavior of a dense sample is first investigated. The self-contact detection algorithm is then tuned and tested on a simple geometry. Complex mesostructures of close and open cell materials are used to illustrate potential uses of the model.

Along with the highlights proposed at the beginning of each part, the reader may get a picture of the global approach with the three series of “PhD Objective”:

- Section 2.5 presents the observed physical phenomena of interest, triggering the investigation.
- Section 3.3 lists requirements for a potential numerical tool to study such phenomena.
- Chapter 7 states the research question, in close link with the chosen method.

The application of the model to complex mesostructures, obtained by X-ray tomography, is discussed:

- Chapter 13 for the compression of a dense composite with spheroidal inclusions.
- Section 16.1 for the mechanical closure and re-opening of casting pores.
- Section 16.2 for the compression of a foam with low relative density.

The proposed appendices include:

- Appendix A is a first approach to the estimation of the local stress field.
- Appendix B briefly discusses some practical implementation issues and provides the key source codes for the DEM simulations.
- Appendix C provides an example of the FEM scripts used as numerical reference.
- Appendix D is the manuscript of an article, presenting the main results of Part IV. The article was submitted to the International Journal of Mechanical Sciences, minor revisions were requested and we currently wait for the final decision on this amended version.

Part I

Mechanical and Numerical Context

2	Experimental Background	19
2.1	Metallic Composites	19
2.2	Amorphous/Crystalline Composites	22
2.2.1	Amorphous Metallic Alloys	22
2.2.2	Amorphous/Crystalline Elaboration	26
2.3	Design of a Model Material	30
2.4	X-Ray Tomography	33
2.5	PhD Objective: Observed Physical Phenomena	35
3	Simulation Background	37
3.1	Limits and Contributions of Modeling Approaches	37
3.1.1	Limits in Model Design	39
3.1.2	Resolution Strategies	40
3.1.3	Numerical Resolution	41
3.1.3.1	Halley's Comet	42
3.1.3.2	Model design and Computing Power	43
3.1.3.3	Computerized Numerical Approach	45
3.2	Description of the Studied Phenomena	47
3.3	PhD Objective: Requirements for a Modeling Tool	51

The general introduction of the manuscript briefly stated the engineering and scientific interest of the forming of composite metallic materials.

Part I aims to illustrate the general context of the PhD, and introduce the focus of interest. The part is divided into two chapters:

- Chapter 2 presents the experimental background, progressively focusing on a designed model material and the physical phenomena of interest. The numerous and complex material science issues are not thoroughly detailed.
- Chapter 3 puts specific emphasis on the choices intrinsic to the design of a model. Some guidelines regarding the conceptual idealization of the studied phenomena are proposed with a “scope statement” for a model.

Highlights - Part I

Mechanical and Numerical Context

- The physical phenomena of interest are mechanisms driving finite inelastic strain in architected metallic materials, at the scale of the constitutive phases.

The effects of morphology and rheology can be partially decorrelated by judicious choice of the tested materials. Experimental observations and predictive simulations are seen as complementary tools, built in close collaboration.

- A metallic composite was previously designed to focus on rheological effects, with an interesting dependency of the rheological contrast on the temperature and the strain rate.

The composite (spheroidal amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ inclusions in a crystalline copper matrix) is elaborated by co-extrusion of powders. *In situ* X-ray tomography hot compression, with co-deformation of the phases, are used as experimental reference.

- A numerical tool to study the deformation mechanisms is sought for.

The handling of numerous interface interactions and topological events (e.g. pore closure, neck creation and phase decohesion and fragmentation) is necessary.

- The chosen modeling strategy is phenomenological. In addition, its elementary variables are mathematically chaotic.

Sensible metrics of interest must be defined, along with a delimited credible modeling scope. The available computing power, a quantitative limiting factor on the accessible metrics, thus influences the qualitative modeling choices.

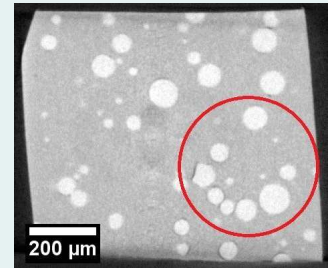


Figure 2.15a

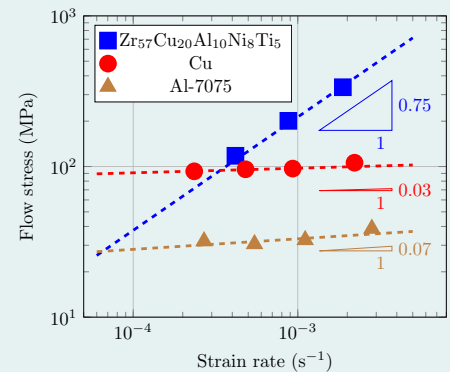


Figure 2.14

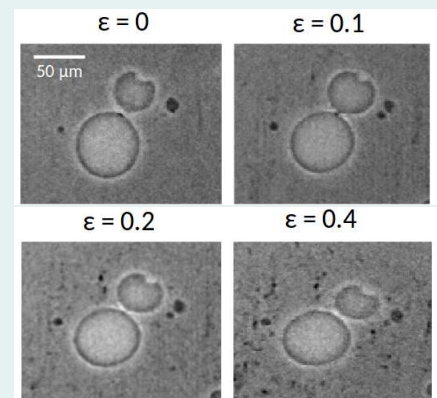


Figure 2.19

Chapter 2

Experimental Background

This chapter presents some experimental context to the mainly numerical work of this PhD. Starting from general consideration regarding composites, the chapter progressively focuses on a specific experimental setup. This brief overview will not dive into the complex details of the underlying material science issues. It is divided into five sections:

- Section 2.1 introduces general concepts about metallic composites.
- Section 2.2 concerns the specific case of amorphous/crystalline composites, along with some general consideration regarding amorphous alloys.
- Section 2.3 describes the design process of a *model material* aimed at the study of rheological effects.
- Section 2.4 introduces the uses of X-ray tomography.
- Section 2.5 sums up the physical phenomena of interest, to be studied experimentally and numerically.

2.1 Metallic Composites

Among existing materials, metallic alloys are providing interesting compromises for mechanical structural products. To fulfill new requirements, material development rely on the design of altogether new monolithic materials, or the association of distinct phases in composites. The description of the mechanical behavior of metallic composites involves many parameters. To better understand finite transformations in such materials, these parameters may be separately studied.

In an industrial context, engineers rely on material science to design products and processes, diagnosis symptoms and justify and optimize their choices [32, p.6]. Predictions and anticipations of the behavior can apply to any step of the life cycle, from elaboration to service and finally end of life. Materials must be designed and chosen to fulfill mechanical, economic or reglementary scope statements. Such scope of statements typically involve compromises between contradictory requirements.

In order to visually compare potential choices, maps of the properties of the materials can be drawn. For example on Figure 2.1 classes of materials are compared in the space (density,elastic modulus). In this example, to compare the performance of very diverse materials, isocontours of a comparison criterion can illustrate the compromise between stiffness and weight requirements. “Holes” are defined as regions of the map that are not

covered by existing materials. The “holes” lying in regions with interesting comparison criteria are potential axes of development for the design of new materials.

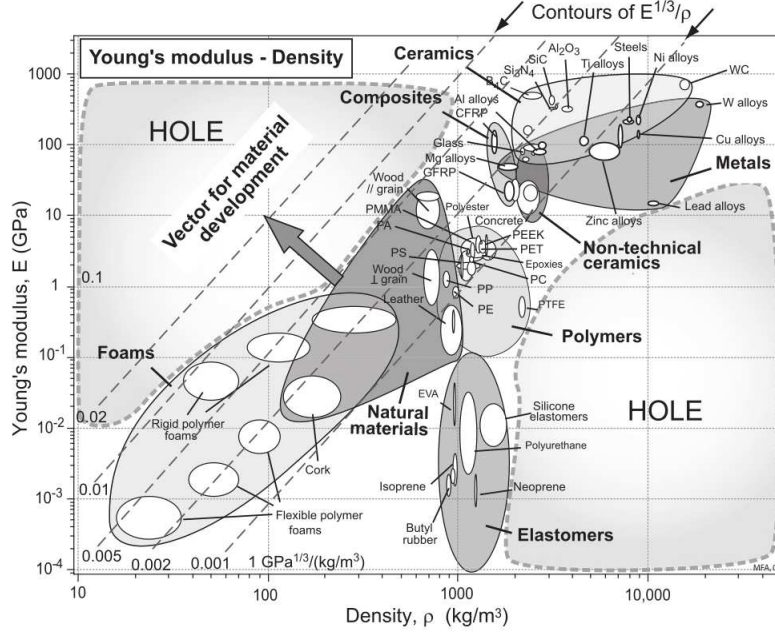


Figure 2.1: Comparison of classes of materials in the space (density, elastic modulus). Definition of “holes” as domains not covered by existing materials in this space. Definition of comparison criteria as isocontours of $\frac{\text{elastic modulus}^{1/3}}{\text{density}}$. Illustration from [12, p.5].

Among existing materials, our study will focus on metallic alloys, massively used in structural products for their interesting mechanical properties, among which their strength to density ratio, their toughness and their behavior at high temperature.

A first option to further explore property charts, and find better compromises, is the tuning of existing monolithic materials and the development of new ones. The composition, elaboration process and microstructure of materials have been heavily studied and improved in history. An example of development of a new monolithic material in the last decades is the elaboration of amorphous metallic alloys (see Section 2.2.1).

In parallel to the developments of new monolithic materials, composite materials – multiphase materials – can be designed. The association of distinct phases allow complementary or even contradictory properties in a single material. In composites, the overall behavior does not necessarily follow a rule of mixture but can follow a trajectory in material space extending current possibilities [12, p.6]. In addition, composites can also fulfill multi-functional requirements.

The distinction, based on length scale, between composite materials and new monolithic materials can be somewhat blurry. The development of nanoscale hybrid materials involves atomic or molecular effects and potentially generates properties that are qualitatively unseen in the initial materials. We focus here on mesoscale composites, where phases are associated at the scale of the micrometer. Among the numerous potential classes of materials, we focus exclusively on composites associating metallic phases (see Figure 2.2).

The challenge of composite design is the tuning – often constrained by practical issues – for each phase of numerous design variables:

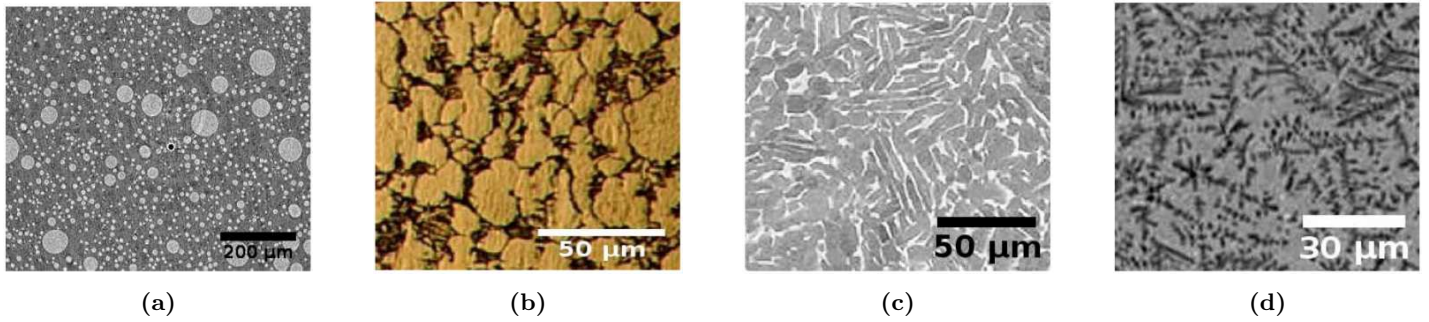


Figure 2.2: Examples of mesostructures of composites associating two metallic phases. (a) Al 18 % Cu matrix with size controllable Al_2Cu inclusions [229]. (b) TA6V equiaxial morphology [139, p.762]. (c) TA6V lamellar morphology [56, p.106]. (d) Crystalline dendrites in an amorphous matrix, $\text{Mg}_{71}\text{Zn}_{28}\text{Ca}_1$ [129, p.303].

- Morphology;
- Topology;
- Volume fraction;
- Relative and absolute size;
- Relative and absolute mechanical behavior.

In composite design, the potentially coupled effects of these parameters on the mechanical properties must be understood [12, p.5]. The prediction of the behavior of the composite is necessary for all the steps of the life cycle of the products, first of which service requirements of the final product and elaboration process from raw composites.

Numerous parameters and phenomena drive the deformation mechanisms of metallic composites. As a step toward an independent study of distinct phenomena, an appropriate selection of *model materials* may allow a partial decorrelation. For example, the comparison of composites with identical compositions but distinct mesostructures allows an emphasis on morphological effect. On Figure 2.2a, the size of the spherical inclusions is controllable at fixed composition. In addition to the size, distinct morphologies of the mesostructure can be compared for an identical composition. For example, an equiaxial morphology (Figure 2.2b) and a lamellar morphology (Figure 2.2c).

The effect of rheological effects can be more independently studied using a composite whose phases have a differential sensitivity to testing configuration. The example of crystalline/amorphous metallic composites is typical (Figure 2.2d): the flow stress of the amorphous phases is strongly dependent on both the strain rate and the temperature, while the crystalline phase is less influenced.

Our work is oriented toward the understanding of the role of the rheological contrast, i.e. the ratio of flow stress between the phases. We focus on the finite transformation context of hot forming, coping with potentially large strains, rotations and displacements. At the laboratory level, the global approach is to couple experimental and numerical approaches. A model material, with tunable rheological contrast, was designed associating amorphous and crystalline metallic phases.

2.2 Amorphous/Crystalline Composites

2.2.1 Amorphous Metallic Alloys

Amorphous metallic alloys are an example of development of a new class of monolithic materials. In amorphous metals, the atoms are not arranged on a regular periodic lattice, as it is the case in classical crystallized alloys. This atomic structure leads to radically distinct macroscopic properties, typically a high yield strength and a large elastic domain at room temperature; a large homogeneous plastic domain at high temperature.

At the atomic scale, amorphous alloys do not display long-range order, as it is the case in classical crystalline alloys, where atoms are regularly arranged on a lattice. Although local cluster arrangements can be found (Figure 2.3b), no long-range periodic pattern can be identified.

A historical route – elaboration processes will not be described here – to obtain such atomic structure in a metallic alloy, is fast quenching from liquid state. Below a critical cooling rate (Figure 2.3a), the crystallization kinetic is too slow, the disordered liquid is “frozen”: from the large time scale of atomic mobility, stems a solid-like state.

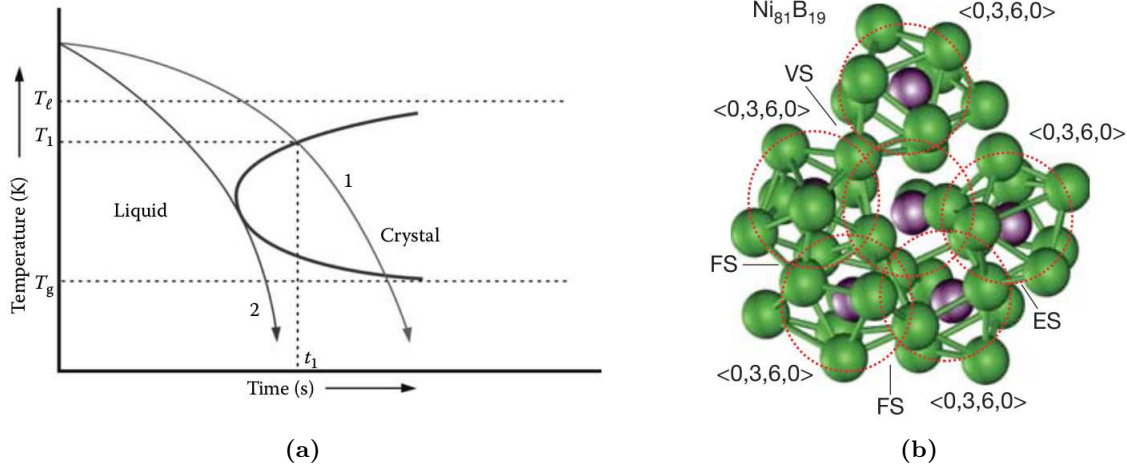


Figure 2.3: Liquid state synthesis of amorphous metallic alloys and typical atomic structure. (a) Schematic time-temperature-transformation diagram for solidification. Two routes from liquid to solid state. Route 1 leads to a crystallized structure. Route 2: critical cooling rate, leading to an amorphous solidification. Illustration from [226, p.52]. (b) Amorphous solids are characterized by the absence of long-range order. The atoms are locally organized – icosahedral clusters are circled in red in this example – displaying only short range order. Illustration from [207, p.422].

The amorphous solid state is only metastable. Two canonical temperatures are used in the literature to roughly quantify the major transitions in the behavior:

- The crystallization temperature (T_x), above which the amorphous structure crystallizes;
- The glass-transition temperature (T_g), above which notable macroscopic viscous flow can be observed.

These transitions are dynamic processes, thermally activated, and the definition of the threshold temperatures is thus not univocal. The temperature window $T_x - T_g$, referred to

as the supercooled liquid region [226, p.20], is an indicator of the ability of the amorphous alloy to be hot formed. The higher the temperature, the lower the flow stress and the shorter the crystallization time: a compromise must be found for forming application.

At macroscopic scale, the transition from amorphous to crystalline state – even only partial – leads to a drastic change in mechanical properties. The sought for properties, e.g. the plastic forming ability at high temperature and the large elastic region at room temperature, are lost. An estimation of the crystallization time may be measured by mechanical testing (Figure 2.4a). The kinetics and thermal activation – atomic motion being a thermally activated phenomenon – of crystallization does not lead to a unique crystallization temperature (T_x) but to a set of crystallizing conditions. For the amorphous alloy used in this work, the crystallization time roughly drops from 10^4 to 10^3 s with an increase of the temperature of 40°C (Figure 2.4b).

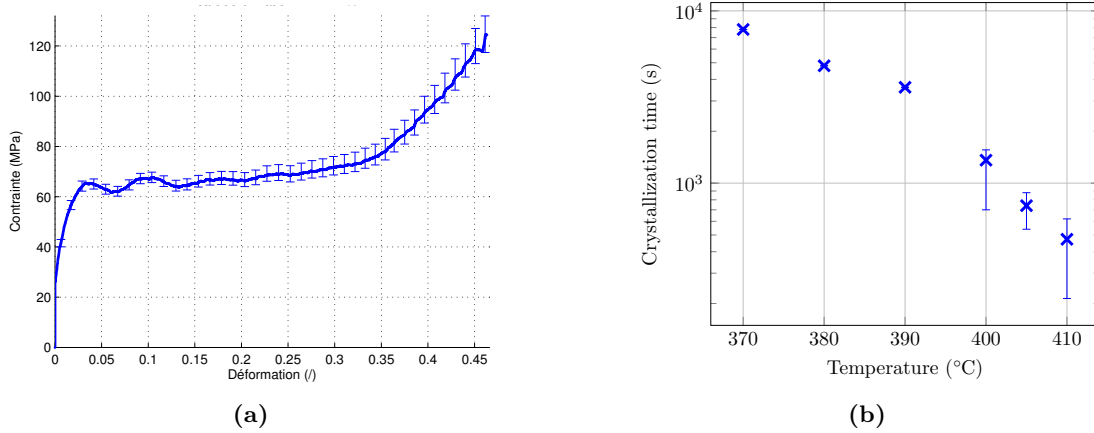


Figure 2.4: Macroscopic effects of crystallization on amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$. (a) Stress-strain plot, uniaxial compression: temperature 405°C , strain rate $4.42 \cdot 10^{-4} \text{ s}^{-1}$. Drastic increase of the flow stress at a strain of 0.3. Crystallization time around 700 s. From [85, p.4]. (b) Estimation of the crystallization time with respect to temperature.

The absence of atomic long-range ordering implies that typical lattice faults – for example the dislocations in crystals, whose motion are one of the mechanisms of inelastic strain (Figure 2.5a) – do not exist in amorphous alloys. Deformation mechanisms are thus drastically distinct for an identical chemical composition, justifying the sudden behavior shift in Figure 2.4a. Under shear stress, the atoms collectively reorganize within shear transformation zone (STZ). STZ are ephemeral transition events between equilibrium states (Figure 2.5b).

Whereas clearly distinct mechanisms are complementary to describe deformation regimes in crystalline metals – see for example copper in Figure 2.6a – similar STZ mechanisms lead to different flow regimes in amorphous alloys (Figure 2.6b).

At room temperature, amorphous alloys typically display high yield stress and a large elastic domain, but often little plasticity and a brittle behavior [104]. In this regime – *elastic* and *inhomogeneous deformation* regions in Figure 2.6b – no lattice faults accommodate stress *via* inelastic strain and the atomic mobility is low.

At higher temperature, closer to or above the so-called glass transition (T_g) the atoms are mobile enough for a macroscopic homogeneous strain of the material. The glass transition temperature is not intrinsic to the chemical composition and is influenced by the thermomechanical history of the material [226, p.19]. In addition, as for crystallization

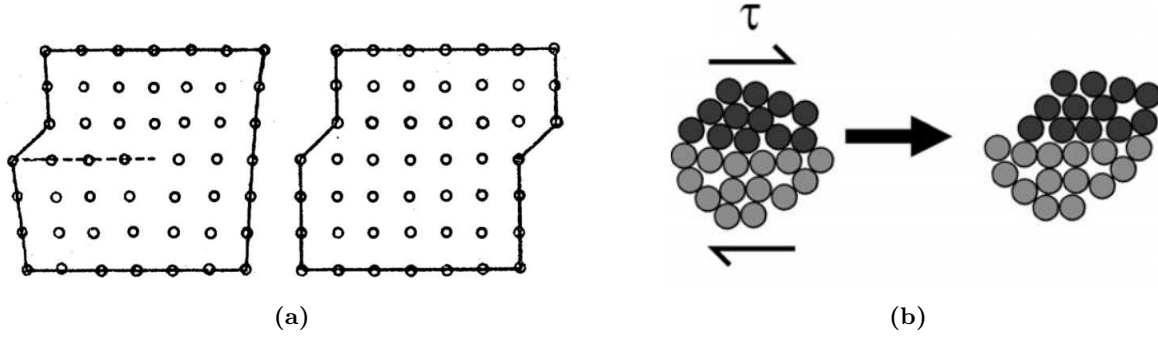


Figure 2.5: Two-dimensional schematics of atomic scale deformation mechanisms in solid metallic alloys. Macroscopic permanent strain stem from irreversible relative motion of atoms. (a) Crystalline idealized long-range order. A typical deformation mechanism: dislocation motion (physical lattice fault) in a crystal. Illustration from [228, p.369]. (b) Amorphous atomic structure. Mechanisms of shear transformation zone (STZ), first proposed by Argon [11]. Collective rearrangement (dynamic event, not a structural defect) of local clusters of dozens of atoms, from a low energy configuration to another. Illustration from [204, p.4068].

temperature, the definition of glass transition temperature is not univocal¹ and depends on the studied time scales.

In the *homogeneous deformation* region (Figure 2.6b), the behavior is typically viscoplastic with a strong sensitivity of the flow stress to both temperature and strain rate (Figure 2.7). At high temperature and moderated stresses, the behavior tends toward a Newtonian flow, where flow stress is proportional to strain rate. The specific atomic structure of the material and the lack of grains and crystalline order allow an almost arbitrary large plastic domain: no defects and discontinuities limit the deformation.

The viscoplastic behavior of amorphous alloys is classically approximated using the Norton law, whose tensorial form is [53, p.4]:

$$\dot{\bar{\epsilon}} = \frac{3}{2} \cdot \left(\frac{\sigma_{eq}}{K} \right)^{(1/M)} \cdot \frac{\text{dev}(\bar{\sigma})}{\sigma_{eq}} \quad (2.1)$$

with $\bar{\sigma}$ the flow stress tensor, $\dot{\bar{\epsilon}}$ the strain rate tensor, K the stress level and M the strain rate sensitivity. The equivalent Mises stress σ_{eq} is a scalar defined from the second invariant of the stress tensor $\bar{\sigma}$:

$$\sigma_{eq} = \sqrt{\frac{3}{2} \text{dev}(\bar{\sigma}) : \text{dev}(\bar{\sigma})} \quad (2.2)$$

In the two previous equations, $\text{dev}(\bar{\sigma})$ denotes the deviatoric part of the stress tensor:

$$\text{dev}(\bar{\sigma}) = \bar{\sigma} - \frac{1}{3} \cdot \text{trace}(\bar{\sigma}) \cdot \bar{I} \quad (2.3)$$

Where \bar{I} is the identity tensor. In the uniaxial case, the tensorial Norton law (Equation 2.1) can be simplified to the following scalar relation [126, p.106]:

$$\sigma = K |\dot{\epsilon}|^M \cdot \text{sign}(\dot{\epsilon}) \quad (2.4)$$

¹ It can be arbitrarily defined as the temperature at which the viscosity of the material is smaller than $10^6 \text{ MPa} \cdot \text{s}$ [226, p.17].

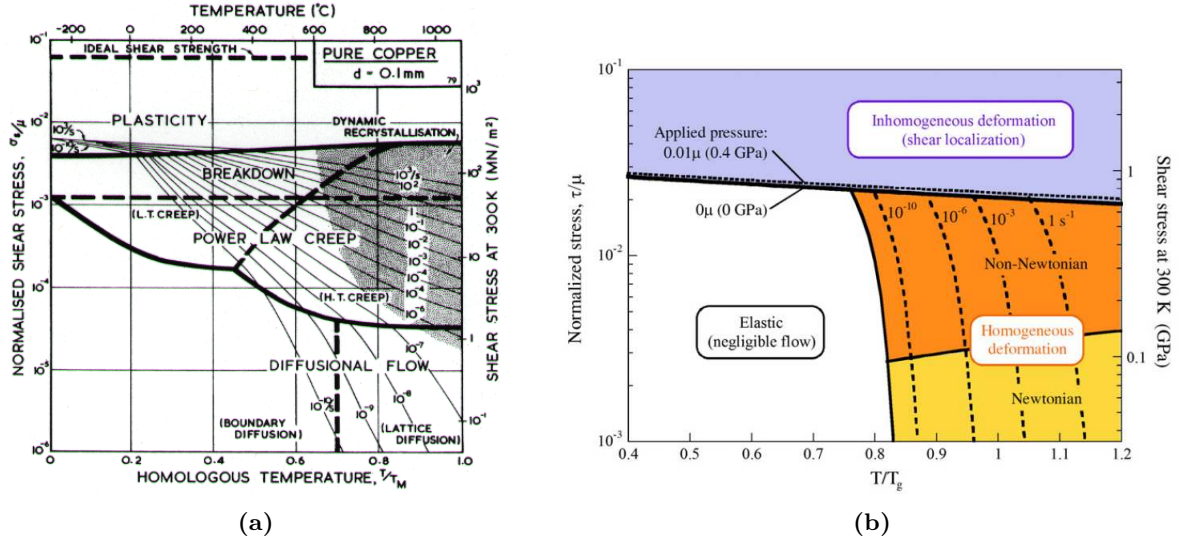


Figure 2.6: Deformation maps and macroscopic behavior: influence of the temperature and stress configuration. (a) Dominant deformation mechanisms in pure copper (normalized axis for isomechanical group of f.c.c. metals). Map from [80, Fig. 4.7]. (b) Generic amorphous metallic alloy behavior (absolute stress values given for $\text{Zr}_{41.2}\text{Ti}_{13.8}\text{Cu}_{12.5}\text{Ni}_{10}\text{Be}_{22.5}$). Dominant flow mechanism for the whole map: shear transformation zone (STZ). Map from [204, p.488].

Taking the example of $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ to illustrate typical hot temperature forming of amorphous alloys, the flow stress varies by a factor 10 when the temperature rises from 380 to 410 °C (Figure 2.7b). As a comparison, in the same temperature range, the flow stress of crystalline copper only varies by a few percent. In the studied temperature and strain rate ranges, the strain rate sensitivity ranges from 0.3 to 1, which is notably higher than the typical creep value for crystalline alloys [117]. The behavior is roughly Newtonian² above 405 °C and below $5 \cdot 10^{-4} \text{ s}^{-1}$. These strain rate sensitivities are comparatively high, with respect to typical values observed for the creep behavior of crystalline alloys ($M \approx 0.2$) [117].

² The flow stress is proportional to the strain rate, thus $M = 1$ in Equation 11.1.

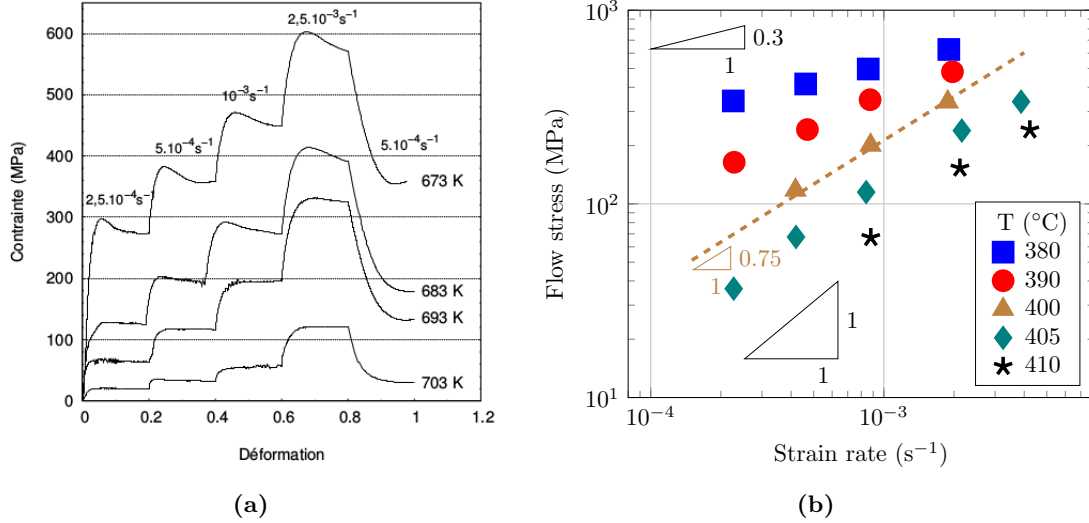


Figure 2.7: Macroscopic viscoplastic behavior of Zr based amorphous alloys, high temperature uniaxial compression tests. (a) Typical stress-strain plot of a strain rate jump: Zr_{52.5}Cu₂₇Al₁₀Ni₈Ti_{2.5}, between 400 – 430 °C and 2.5.10⁻⁴ – 2.5.10⁻³ s⁻¹. Plot from [182, p.64]. (b) Flow stress map of amorphous Zr₅₇Cu₂₀Al₁₀Ni₈Ti₅ between 380 – 410 °C and 2.10⁻⁴ – 4.10⁻³ s⁻¹. Norton law approximation at 400 °C: $M \approx 0.75$.

2.2.2 Amorphous/Crystalline Elaboration

The use of amorphous alloys as structural materials is hindered by their brittle behavior at room temperature. Hence the development of metallic composites, associating amorphous and crystalline phases. Elaboration techniques include liquid and solid state processing. In liquid state, or in situ, processing, little freedom remains on the choice of composition of the phases. Ex situ elaboration, from solid state, allows more flexibility for mechanical and experimental requirements, a major constraint being the crystallization behavior of the amorphous phase.

At room temperature, the typically brittle behavior of amorphous alloys – the shear localization region in the map shown Figure 2.6b – can be a major drawback to their use as structural materials. A work-around in material design is the composite strategy (Figure 2.8): the propagation of cracks in a brittle amorphous alloy may be hindered by an association with a ductile phase (Figure 2.8c), increasing the material toughness [104] (Figure 2.8a). The choice of a crystalline alloy as the ductile phase potentially allows a correct elastic modulus compatibility and chemical compatibility at the interfaces.

In addition to their potential structural use, amorphous/crystalline composites are promising as model materials to study the effect of rheological contrast between the phases in a composite. Indeed, the temperature and strain rate sensitivity of the flow stress in the amorphous alloy allows to selectively tune the flow stress in the phases. A wide range of rheological configurations can thus be tested with a single material. To study equivalently diverse configurations, numerous crystalline composites would need to be elaborated and compared. The interpretation of the results is delicate when the data from distinct composites are compared, as no straightforward method allows the direct control of all the parameters of influence. Using a single composite minimizes the uncertainties.

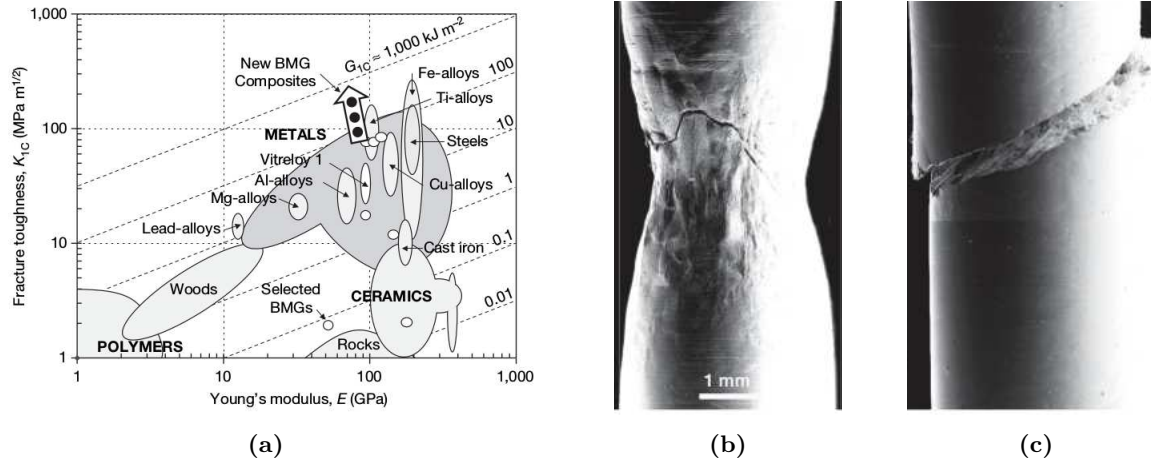


Figure 2.8: Ductility in amorphous crystalline composite. Illustrations from [104, p.1086,1088]. (a) Comparison of materials in the space (elastic modulus, mode I fracture toughness). Isocontours of critical energy for crack propagation. Tensile test at room temperature: (b) Ductile rupture of $\text{Zr}_{39.6}\text{Ti}_{33.9}\text{Nb}_{7.6}\text{Cu}_{6.4}\text{Be}_{12.5}$ amorphous/crystalline composite. (c) Typical brittle rupture of an amorphous alloy.

Amorphous/crystalline metallic composites can be elaborated *in situ*, from an homogeneous liquid state [180], taking advantage of the kinetics of dendrite growth, to selectively grow crystalline dendrites and freeze an amorphous structure (Figure 2.9a). The process can be delicate to control due to the diffusion during the solidification. If an excessive migration occurs, the local chemical composition modification might hinder the amorphous solidification.

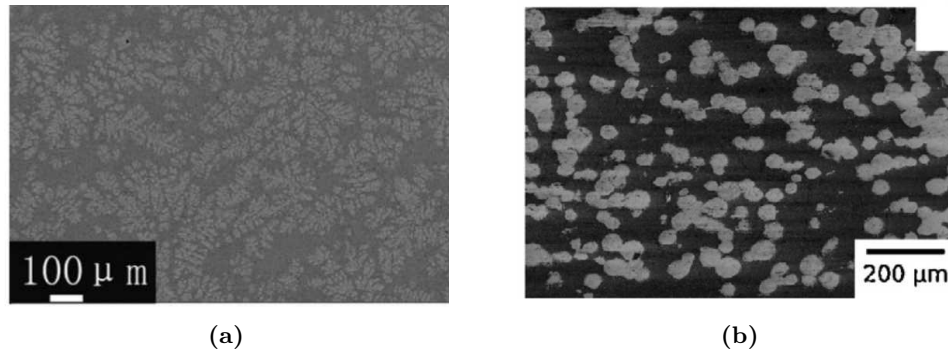


Figure 2.9: Examples of amorphous / crystalline metallic composites. (a) *In situ* elaboration from liquid state. Bridgman solidification of $\text{Zr}_{37.5}\text{Ti}_{32.2}\text{Nb}_{7.2}\text{Cu}_{6.1}\text{Be}_{17.0}$. Dendritic crystalline inclusions in an amorphous matrix. SEM image from [179, p.2]. (b) *Ex situ* elaboration by powder technology from solid state, using spark plasma sintering (SPS). Amorphous spherical inclusion of $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ in an aluminum (80 vol. %) matrix. SEM image from [169, p.113].

Our objective is to design a model material to study the effect of rheological contrast in metallic composites, taking advantage of the tunable rheological contrast in amor-

phous/crystalline composites. *In situ* elaborated composites are not well suited:

- Their dendritic mesostructure (Figure 2.9a) is too complex to focus the study on rheological issues.
- Their elaboration process is delicate to control and thus constrains the size of the phases and the overall samples.
- Little control is left on the relative properties of the phases.
- A similar chemical composition in the phases make X-ray three-dimensional imaging more challenging.
- Typical suitable alloys contain beryllium, whose toxicity is a supplementary experimental constraint.

In contrast, the *ex situ* elaboration displays interesting features for our purposes. As the phases are associated at solid state by thermomechanical processes, their choice is less strictly constrained. The mechanical and physical properties of the phases can thus be chosen a little more independently, to fulfill experimental requirements.

In the laboratory context of the PhD, a strong background of metallic alloy co-deformation and amorphous alloy elaboration led to the design of stratified composites since 2008 [182]. Plates of zirconium-based amorphous alloy and light crystalline alloy (magnesium and aluminum based) were co-pressed at high temperature (Figure 2.10). This simple geometry allowed a first approach to the study of the adhesion of the interfaces and the effects of temperature and strain rate on the co-deformation regime. The choice of the forming window is driven by compromises, taking into account crystallization of the amorphous phase at higher temperature and phase fragmentation at lower temperature.

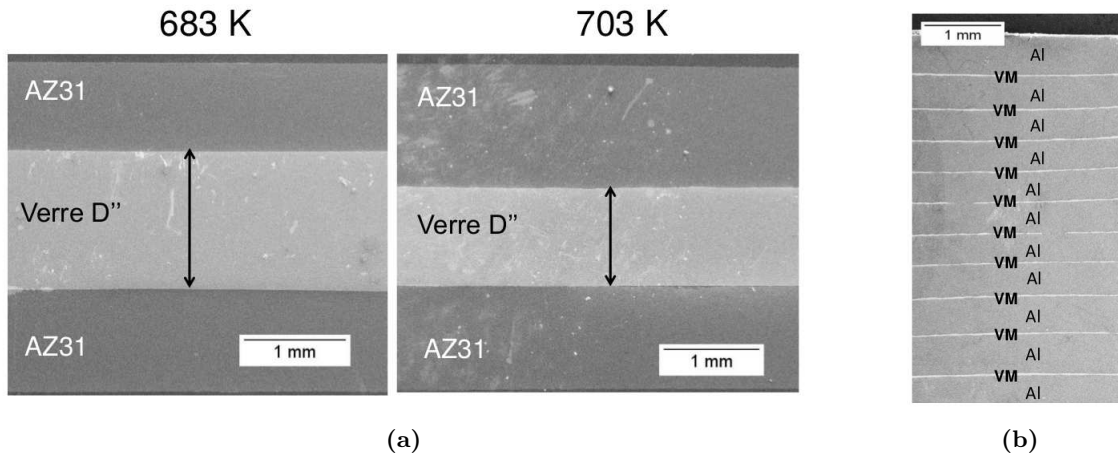


Figure 2.10: Stratified composites elaborated *ex situ* by hot co-pressing of amorphous $\text{Zr}_{52.5}\text{Cu}_{27}\text{Al}_{10}\text{Ni}_8\text{Ti}_{2.5}$ and crystalline light alloys. SEM images from [182, p.96,162]. (a) Three-layer with magnesium alloy (AZ31): {amorphous,crystalline,amorphous}. Effect of the co-pressing temperature on the relative thickness of the layers. (b) Multi-layer with aluminum alloy (Al-5056): {11 \times crystalline, 10 \times amorphous}.

Such stratified materials already exhibit numerous key properties, for example crossing stress-strain curves (Figure 2.14 on page 31), but are too anisotropic to study generic

mechanisms in composites. In order to find an intermediary complexity between *in situ* composites and the elementary stratified geometry, *ex situ* elaboration process can rely on powder technology. Traditional sintering time and temperature scales are unsuited for amorphous alloys, as they crystallize rapidly at high temperatures, triggering the use of less conventional procedures.

An example of powder thermomechanical processing is the spark plasma sintering (SPS). The SPS process is a field assisted sintering technology, where the sample is heated by Joule effect, allowing shorter sintering times. It has successfully been applied to the elaboration of both massive amorphous alloys [160, 161] and amorphous/aluminum composites [169, 73]. For these materials, investigation efforts were focused on the room temperature mechanical behavior, in the objective of studying service requirements in structural materials. Figure 2.11 displays typical stress-strain behavior at various volume fraction of such composites. Intermediary volume fractions provide a range of compromise between strength and ductility of the two phases.

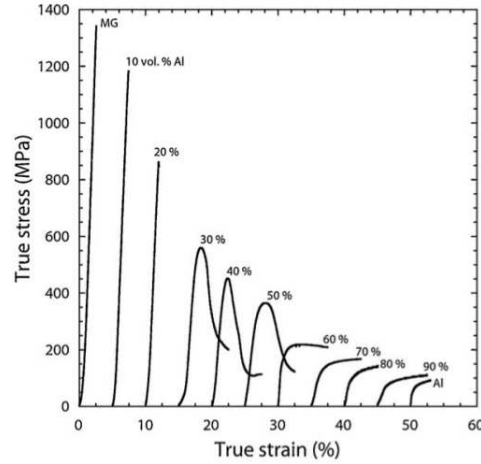


Figure 2.11: Stress-strain room temperature behavior of crystalline / amorphous metallic composite (see Figure 2.9b). Effect of the volume fraction, from pure amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ to pure crystalline aluminum [169, p.114].

Qualitatively, the volume fraction does influence the fracture surfaces [73, p.96]. At 10%vol of inclusions (Figure 2.12), the overall failure is driven by coalescence of multiple matrix/inclusions decohesion events (Figure 2.12a). Locally (Figure 2.12b), the dimpled surface of the crystalline matrix accounts for large plastic strain and the inclusions seems to display a relatively strong cohesion to the matrix.

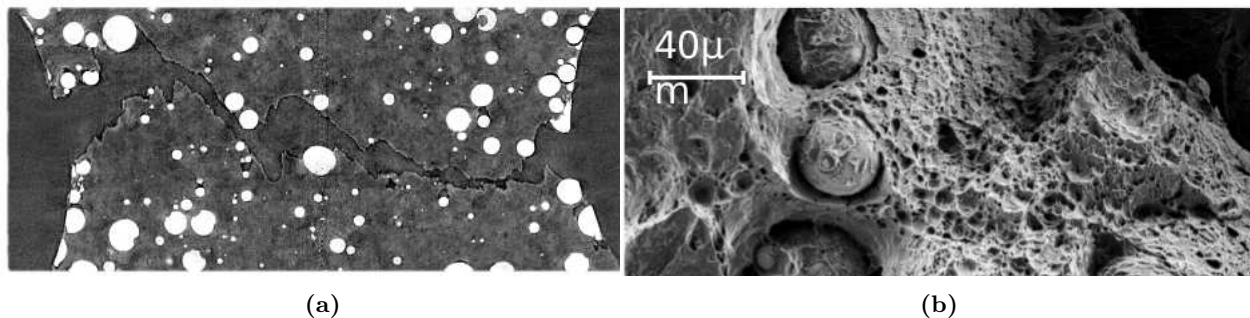


Figure 2.12: Rupture at room temperature of a metallic composite. Crystalline aluminum 1070 matrix and 10 %vol of base Zr alloy amorphous spheroidal inclusions. (a) Final coalescence of multiple events of decohesion between matrix and inclusions. Slice of a reconstructed tomography image [73, p.96]. (b) *Postmortem* SEM image [73, p.97].

2.3 Design of a Model Material

An amorphous/crystalline metallic composite is designed as a model material to study the effect of rheological contrast in high temperature forming of metallic composites. Experimental and mechanical concerns drive the selection of the phases and the elaboration process. Within a defined window of strain rate and temperature, the phases of the composite can be co-deformed with tuned relative rheology.

The global objective for the designed material is to exhibit a temperature and strain rate window in which co-deformation of the phases can be observed. The research environment of the laboratory – with strong emphasis on powder technology and co-deformation of materials – leads to the choice of high temperature co-extrusion of powders as elaboration process. The generic chosen geometry is a random dispersion of spheroidal inclusions of amorphous alloy in a crystalline matrix. This simple geometry aims to focus the study on the effects of rheological contrast, without excluding topological events. Two master internships [144, 85] focused on the elaboration of such model materials.

The choice of the amorphous phase was driven by practical considerations: $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ amorphous alloy is readily available as atomized powder (Figure 2.13a) and its bulk mechanical properties have been extensively studied: its supercooled liquid region is relatively large³, allowing a comfortable enough high temperature forming window before crystallization. In addition, it does not contain toxic components as beryllium. The crystalline phase was chosen with respect to the amorphous phase.

Aluminum alloys were the first chosen candidate as crystalline phase. Their moderate flow stress eases-off the extrusion process and their low X-ray absorption guarantee a good phase contrast with zirconium based alloys (see also Section 2.4). However, no satisfactory co-deformation conditions could be found with Al-6061 [144, p.22] nor Al-7075 [85, p.17]. The high temperature, or the low strain rate, needed to reach matching flow stresses between the phases (Figure 2.14) lead to the crystallization of the amorphous phase, before significant strain can be applied. Although aluminum is probably a good candidate for structural composites – due to its low density and comparable elastic modulus with zirconium based amorphous alloys – it is not well suited to build a model material for high temperature forming.

³ $T_x \approx 440^\circ\text{C}$ and $T_g \approx 380^\circ\text{C}$, the forming window is thus $T_x - T_g \approx 60^\circ\text{C}$ [169, p.113]. See also Figure 2.4b and Figure 2.7b for a qualitative illustration of the kinetics of thermal activation of both transitions.



Figure 2.13: SEM views of the initial state of the powders at identical scales. (a) Atomized amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$: spheroidal powder. SEM view from [73, p.41]. (b) Electrolytic crystalline copper: dendritic powder.

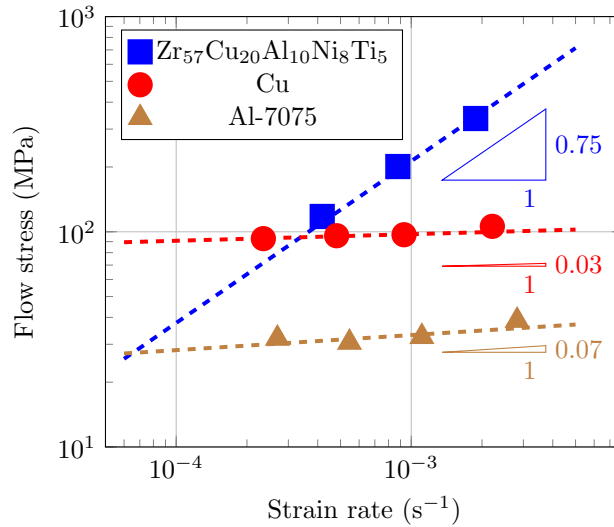


Figure 2.14: Flow stress at 400°C of amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ and crystalline pure copper and Al-7075. Norton law approximations and strain rate sensitivities.

Further developments led to the choice of pure copper as crystalline phase. Indeed, from the flow stress of the phases (Figure 2.14), co-deformation configurations may be reached for reasonable temperature and strain rate, for example around 400°C and $2.5 \cdot 10^{-4} \text{ s}^{-1}$. *Postmortem* observations were made of significant inelastic co-deformation of composites (Figure 2.15), in the window $390 - 405^\circ\text{C}$ and $2 \cdot 10^{-4} - 2 \cdot 10^{-3} \text{ s}^{-1}$ [85]. For lower temperature, or higher strain rate, the flow stress of the amorphous phase is too high and its deformation is thus negligible. On the contrary, for higher temperature or lower strain rate, the amorphous phase is soft but crystallizes before undergoing significant strain.

In the identified co-deformation configurations, the ratio of single phase flow stresses is in the range 0.25 – 4. The relative rheology of the phases of the composite can be tuned within this window, allowing the study of distinct contrast with a unique composite.

The elaboration procedure of the composite – starting from atomized $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ amorphous powder and electrolytic copper powders (Figure 2.13) – involves the following steps:

1. Powder mixing;
2. Room temperature compaction at 700 MPa;

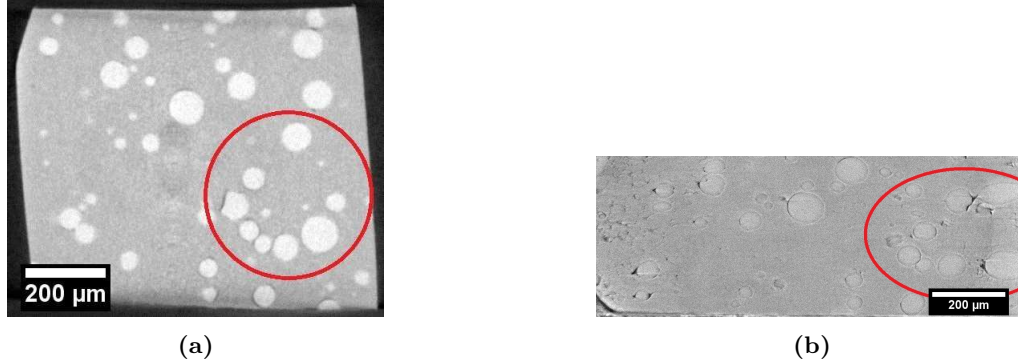


Figure 2.15: Model material composite: crystalline copper matrix, amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ spheroidal inclusions (15% volume fraction). Cross-sections from 3D tomography reconstructions (refer to Section 2.4. Images from [85, p.18]. (a) As elaborated state. View of a full millimetric sample prepared for compression tests. (b) *Post-mortem* co-deformed state of the same sample. Compression at 400°C and $2.5 \cdot 10^{-4} \text{ s}^{-1}$ (compression direction: vertical).

3. Insertion of the green body into a protective capsule (Figure 2.16a);
4. Co-extrusion at 380°C (Figure 2.16b).

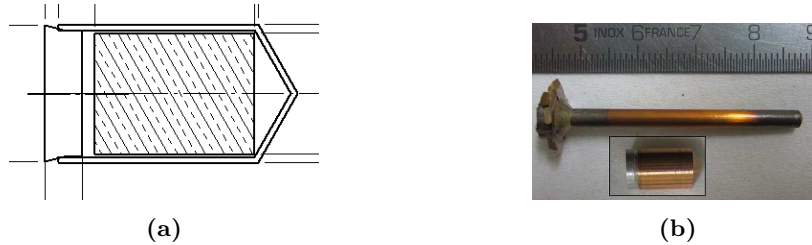


Figure 2.16: Encapsulated co-extrusion of compacted powders. (a) Schematic of the copper capsule containing the compacted powder (cross-hatched) during the extrusion. The capsule is closed by a steel lid. (b) Hot extruded and initial states. The composite is protected by the capsule and is not visible.

Extrusion is trusted to be a reliable process to limit the presence of residual porosity [73, p.36]. In the process, the total diameter is reduced from 7 to 3.1 mm (Figure 2.16b), the extrusion ratio [18, volume 2 p.118] is thus approximately 5. For low volume fractions of amorphous phase ($< 50\%$), the extrusion force is typically in the range $2 \cdot 10^3 - 3 \cdot 10^3 \text{ daN}$. The material stays approximately 10 min at 380°C , which roughly corresponds to one fourth of the crystallization time at this temperature (Figure 2.4a).

The procedure allows the elaboration of composites with up to 70% of volume fraction of amorphous phase. The composites with higher volume fraction were not cohesive enough to withstand the compression tests. In addition, a high volume fraction of amorphous phase increases the required extrusion force, sometimes reaching the maximal capacity ($5 \cdot 10^3 \text{ daN}$) of the setup [85, p.12].

2.4 X-Ray Tomography

Two-dimensional and destructive experimental techniques cannot track the temporal evolution of the investigated phenomena. The time and size scale and the studied phenomena justify the use of X-ray tomography for in situ experimental study.

From the experimental point of view, both two-dimensional and destructive techniques are unsuitable to track the temporal evolution of the morphology of a composite. Among existing three-dimensional and nondestructive imaging techniques, X-ray tomography matches well (Figure 2.17a) with the length scales of our designed model material: millimetric for a representative volume, micrometric for the phase morphology (Section 2.3).

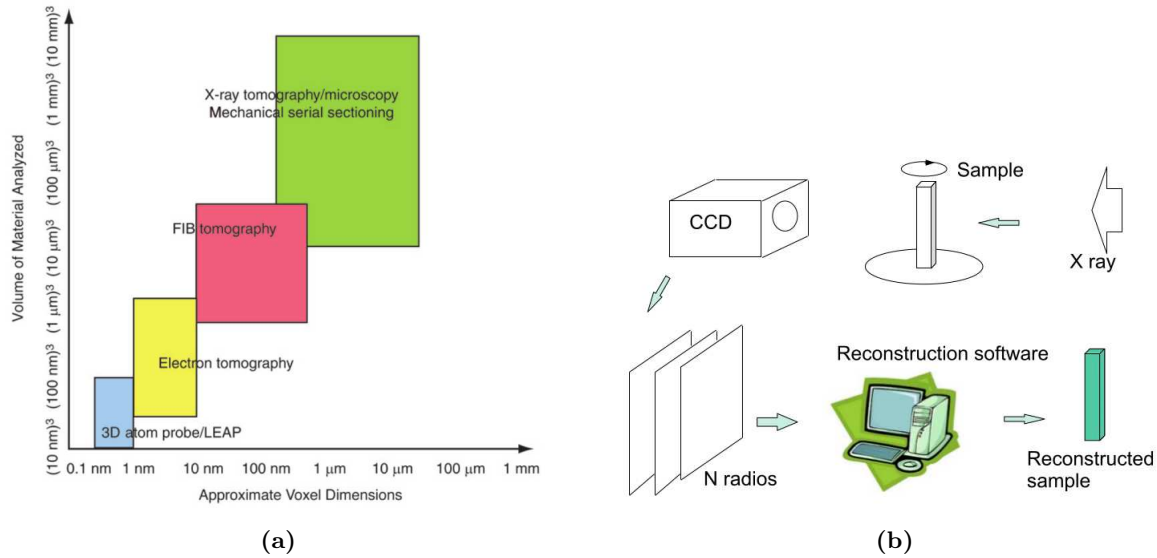


Figure 2.17: Overview of tomography principles and scope. (a) Size scales of three-dimensional imaging tools. Chart from [234, p.409]. (b) Main step of X-ray attenuation tomography, from 2D images to 3D volume reconstruction. Schematic from [38, p.291].

The global principle of tomography is to reconstruct a 3D representation of an object from various 2D observations, taken from distinct orientations (Figure 2.17b). A classical X-ray imaging technique is based on attenuation contrast⁴: the intensity of an X-ray beam is measured after crossing the sample. The relative attenuation of the beam in the distinct phases can induce a contrast on the 2D image, the projections. A series of projections are taken from distinct orientations to build a scan. Algorithmic procedures [115] allow to infer the 3D field of absorption from a scan: the reconstruction. This 3D image can be filtered and segmented to obtain a spatial distribution of distinct phases of multi-materials.

The non-destructive character of the method allows measurements on the same sample, before and after deformation. Several steps can thus be sequentially studied with interrupted tests [15]. If the scans can be made rapidly enough, a phenomenon can be observed without interruption: *in situ* [38, 128] experiments.

⁴Phase contrast, where the phase variation of the beam is measured instead of its intensity, will not be examined.

The choice of copper as constituent of the matrix leads to restrictions on the size of observable samples by X-ray tomography and to a more delicate solid-state elaboration of the model material, with issues regarding cohesiveness. Both copper and zirconium, the respective dominant components of the matrix and the inclusions, have a high level of absorption of X-rays (Figure 2.18), hence a high energy beam must go through the sample to image it. *Postmortem* measurements can be made on regular laboratory tomographs, providing a limited flux, but *in situ* imaging requires shorter scanning times. The *in situ* measurements were thus performed [35] at the European synchrotron radiation facility (ESRF). Using the chosen configuration (Table 2.1), for samples of diameter 0.5 mm, a full scan⁵ is performed in 7 s.

In counterpart, the set-up for *in situ* experiments at the ESRF imposes heavier security procedures, which are time consuming. With the current set-up, at least several minutes are elapsed between the introduction of the sample in the furnace and the actual beginning of the mechanical test, typically from 5 to 10 min. Such a constraint proves critical for our purpose.

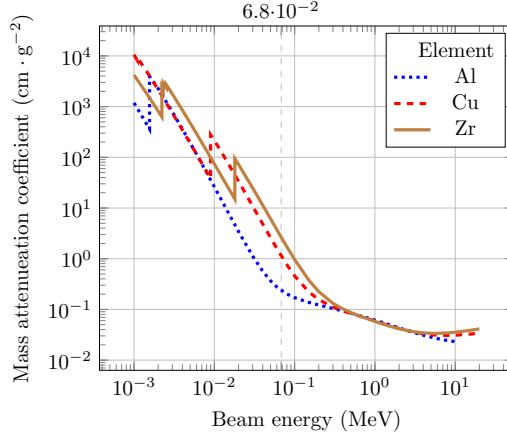


Figure 2.18: Energy absorption for distinct elemental media, depending on the energy of the photons. The energy range at beamline ID19 is $10^{-2} - 2.5 \cdot 10^{-1}$ MeV. *In situ* measurements were made at $6.8 \cdot 10^{-2}$ MeV. Data from [106].

⁵ This short imaging times minimizes the deformation during the scan. At a typical strain rate of $2.5 \cdot 10^{-4} \text{ s}^{-1}$, in 7 s, the variation of strain is lower than $2 \cdot 10^{-3}$. The geometrical variation is thus limited during the time between the beginning and the end of the scan.

Beam energy	$6.8 \cdot 10^{-2}$ MeV
Projections per scan	800
Field of view	$800 \times 960 \text{ pixel}^2$
Pixel size	$1.3 \text{ } \mu\text{m}$
Sample size	$0.5 \times 0.5 \times 0.5 \text{ mm}^3$
Time per scan	7 s
Data sampling	1 scan per minute
Compression velocity	$2 \cdot 10^{-1} - 5 \cdot 10^{-1} \text{ } \mu\text{m} \cdot \text{s}^{-1}$
Temperature	$400 \text{ } ^\circ\text{C}$

Table 2.1: X-ray tomography setup key parameters for *in situ* experiments at the ESRF, beamline ID19. Data from [35, p.6].

2.5 PhD Objective: Observed Physical Phenomena

The study focuses on the rheological parameter effects on quasistatic co-deformation of viscoplastic composites. The length scale of interest is the characteristic length of the phases. The phases are idealized as continuous media, following Norton law, but allowing potential topological changes.

The chosen model material is a composite with a pure crystalline copper matrix with 15 %vol of spheroidal amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ inclusions (see Section 2.3). The co-deformation at high temperature is experimentally studied *in situ* by X-ray tomography. Sub-millimetric samples are uniaxially compressed at $380 - 410^\circ\text{C}$ and $2 \cdot 10^{-4} - 4 \cdot 10^{-3} \text{ s}^{-1}$ typically above strains of 0.3.

The main phenomenon of interest in the frame of our study is the morphological evolution – at the micrometric mesoscale – of the phases in the composite (Figure 2.19). Phases are assumed to be continuous media, governed by perfect viscoplastic constitutive behavior – the Norton law – averaging in space all phenomena occurring at smaller length scales. In a finite transformation context, targeted strains are in the range $0.1 - 1$. Only inelastic strains are considered, the deformations are assumed to be isochoric in both phases. Dynamic effects are neglected, compression in the range $10^{-4} - 10^{-2} \text{ s}^{-1}$ are considered quasistatic.

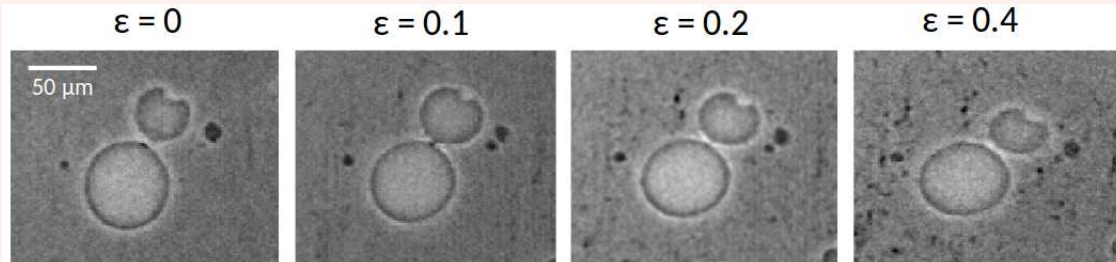


Figure 2.19: Main phenomenon of interest in the model material: co-deformation of the phases under compressive strain; interface interactions at the mesoscale (potential topological event). Tomography views from [35].

Topology can evolve (Section 3.2) during the deformation (Figures 2.19 and 2.20). Such changes can stem from the interaction of the interfaces between the phases, the decohesion of the phases or pore closure and opening.

The crystallization is a major limiting factor of the experiments. At 400°C , the amorphous phases start to crystallize in a little more than 10 min (Figure 2.4b). This crystallization induces a sudden increase of the flow stress. In the range $2.5 \cdot 10^{-4} - 5 \cdot 10^{-4} \text{ s}^{-1}$, strains of $0.3 - 0.6$ are thus upper bounds.

The crystallization time is cumulative on the whole life cycle of the amorphous alloy, including elaboration and installation in the experimental setup. The hot co-extrusion of the powders at 380°C last approximately 10 min (Section 2.3). Launching a test in the *in situ* apparatus at the ESRF lasts from 5 to 10 min, during which the sample is heated. Overall, it has been observed that the effective compression time available is a little above 500s, thus limiting the macroscopic strain before the behavior change. In

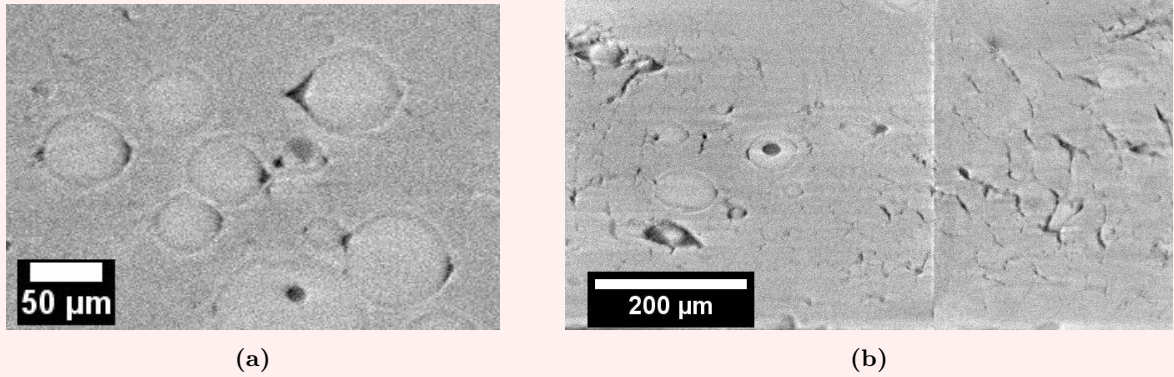


Figure 2.20: Typical defects appearing under compressive load in the model material. Additional topological events. (a) Decohesion at the interfaces of the phases. (b) Crystalline matrix decohesion. Tomography views from [85, p.18].

the tested configurations, the deformation of the inclusions were very limited after a macroscopic prescribed strain of $0.15 - 0.2$.

Chapter 3

Simulation Background

“Le chimiste invente des atomes, et puis les décompose en atomes plus petits qui gravitent comme des planètes autour de quelque soleil; belle machine pour penser plus avant; belle construction; idée. Mais s’il croit que c’est une chose, que c’est vrai, que l’objet est ainsi, il n’y a plus de penseur.”

Alain [6, §138]

In Chapter 2, the experimental background was presented, along with a specific experimental setup and the physical phenomena of interest. The global objective of this PhD is to propose a suitable numerical model and a phenomenological strategy was chosen.

The fact that the proposed model is loosely physically grounded triggers a specific interest to the process of model design in general. This chapter is thus an attempt to define the modelization needs and some general safeguards regarding the potential scope of use. It is divided into three sections:

- Section 3.1 attempts to locate our approach in a broader context, describing modeling issues in general and more specifically for computerized and discrete methods.
- Section 3.2 deals with some concepts to describe the observed phenomena of interest.
- Section 3.3 states the practical modelization objectives and some evaluation metrics.

3.1 Limits and Contributions of Modeling Approaches

Experimental and modeling tools can help to understand and predict physical phenomena. In both cases, the unknowns to be dealt with imply numerous choices that will deeply shape our description of the phenomena. In the objective of proposing a predictive model, the choice of the modeling strategy is also constrained by the foreseen resolution method, which is in turn influenced by the available computing power.

Two complementary and intimately intertwined approaches can be adopted to study a physical phenomenon: the observation through experimentation and the prediction through modeling. In most cases, models and experiments are closely built upon one another and have little independence. At least since Johannes Kepler [212, p.333], a physical phenomenon is considered rationally understood when it is described by a sequence of

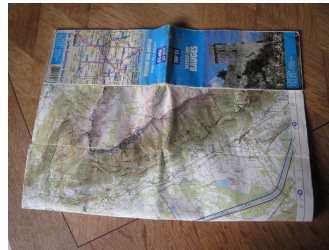
causal processes, with somewhat coherent logical rules and an acceptable agreement with observations made.

In this context, modeling is an approach where a phenomenon key features are idealized and represented either with a conceptual construction or an analogous system. Observation, far from being a purely passive and objective act, is necessarily an active interpretation of signs – a struggle to use what we know to describe what we see – conditioned by *a priori* concepts: we are literally blind to phenomena too alien to our preexisting thought structures. This assessment is common to linguistics¹, computing science², poetry³, mathematics⁴ and solid mechanics.

Experiments and simulations can be designed to be efficient tools to study a phenomenon, but models and observations must not be identified to the reality. Many decisive choices of what is most relevant, in a given objective, restrain our understanding to a necessarily partial and rough sketch. To illustrate the potential and limit of a model, a phone book can be considered as a typical example. Indeed, a phone book can be used to approximate the number of residents of an area or even interact with them (Figure 3.1a). However, the printed letters and digits cannot be identified to these residents. In a discrete element method (DEM) simulation of a granular material, the description of individual physical particles might be even cruder. In place of a phone book, a topographic map of the same area can provide common data, the name of the villages for example. This complementary model will be adapted to distinct modelization objectives.



(a)



(b)

Figure 3.1: Two examples of models of the department of *Savoie*. (a) A phone book. (b) A topographic map.

¹ “We dissect nature along lines laid down by our native languages. The categories and types that we isolate from the world of phenomena we do not find there because they stare every observer in the face; on the contrary, the world is presented in a kaleidoscopic flux of impressions which has to be organized by our minds—and this means largely by the linguistic systems in our minds. We cut nature up, organize it into concepts, and ascribe significances as we do, largely because we are parties to an agreement to organize it in this way—an agreement that holds throughout our speech community and is codified in the patterns of our language. The agreement is, of course, an implicit and unstated one, but its terms are absolutely obligatory; we cannot talk at all except by subscribing to the organization and classification of data which the agreement decrees. [...] it means that no individual is free to describe nature with absolute impartiality but is constrained to certain modes of interpretation even while he thinks himself most free.” [241, p.212–214].

² “There will always be things we wish to say in our programs that in all known languages can only be said poorly.” [168, §26].

³ “We live an epoch in which our inner lives are dominated by the discursive mind. This fraction of the mind divides, sections off, labels – it packages the world and wraps it up as ‘understood’. It is the machine in us that reduces the mysterious object which sways and undulates into simply ‘a tree’. Since this part of the mind has the upper hand in our inner formation, as we age, [...] we experience more and more generally, no longer perceiving ‘things’ directly, [...] but rather as signs in a catalogue already familiar to us. The ‘unknown’, thus narrowed and petrified, is turned into the ‘known’. A filter stands between the individual and life.” [186, p.5].

⁴ “Une chose m’avait déjà frappé [...]: c’est la grossièreté [...] du mode de raisonnement mathématique quand on le confronte avec les phénomènes de la vie, les phénomènes naturels.” [93].

3.1.1 Limits in Model Design

Model design must cope with inevitable unknowns: a model is intrinsically limited to be a partial and rough sketch. In addition, the ill-posed essence – their chaotic nature – of many physical phenomena sets a bound on what is possibly modeled, regardless of the simulation strategy used.

The prediction of physical phenomena, based on its prior understanding or aimed towards it, requires the choice and design of a model, its application and the interpretation of the obtained results. Regardless of the computing power available [22, p.17], all decisions in these tasks must accommodate with numerous unknowns. Even elementary metrics as the strain do not seem to be intrinsic, being described by multiple and sometimes contradictory theories [192, p.5].

In this rather blurry general framework, a dominant criterion in modern science [14] is to evaluate and quantify the reliability of the modeling tools and to assess measurable effects on reality as well as their agreement with our understanding. The delimitation of scopes of study allows the design of credible [203] and efficient models and experiments. The degree of confidence granted to a model must be evaluated for the foreseen application⁵ and allows their practical use, despite the uncertainties. Specific methodologies have been developed for numerical models, due to their complexity and their power, in order to assess their credibility [203] and to better delimitate scopes of validity [162].

When designing a model, the aim is thus to reproduce a physical phenomenon at a given scale, at least in a descriptive way and if possible in a predictive way. A model will be called *phenomenological* if it doesn't describe, explain or take into account the phenomena at scales coming immediately lower the studied scale. Stronger based models may be rough, but display more coherence with subscale phenomena.

Before any resolution is attempted, the nature and the physical phenomena must be evaluated. Chaotic and non-deterministic phenomena cannot be conceptually modeled in a *well-posed* fashion. A well-posed problem poses a unique solution which is stable, in the sense that infinitesimal changes to the initial conditions do not generate discontinuity jumps in the solution [97].

Ill-posed problems are frequent even in simple systems [171, p.9]. In classical mechanics, such an elementary system as collisions of balls moving on a straight line (Figure 3.2) proves to be ill-posed by essence, i.e. regardless of the modeling approach [82]. Indeed, the final velocities of the balls discontinuously depend on the initial distances.

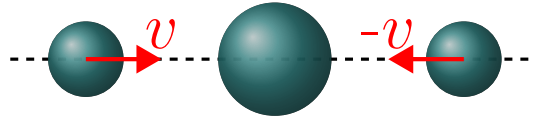


Figure 3.2: Example of an indeterminate problem in classical mechanics. At initial state, the central ball is stationary, the two others move toward it with equal velocity from an equal distance. The final velocities discontinuously depend on the initial distances [82, p.292].

Another canonical example of chaotic behavior is the three-body problem, the computation of the dynamics of a system of three punctual masses driven by gravitational force⁶,

⁵ “The behavior of a given material can be represented by a schematic model only in relation to the envisaged usage and the desired precision of the predictions.” [126, p.71].

which is ill-posed even in the hypothesis of absence of collisions [57]. As an interesting side note, preliminary hints to bound the error on the dynamics of a discrete system *representing a continuum* have been published [154, p.1533–1536], but the assumptions are far too restrictive to be used for our purpose.

Although it is illusory to attempt to compute exactly the chaotic variables of a phenomenon, the impossibility to study and model it is not implied. From a mathematical point of view, the problem needs to be regularized [171], formulated in a way that metrics of interest – for example statistical data on the chaotic variables – can be expressed in a well-posed fashion. From the engineer’s point of view, the progressive transition from ill- to well-posed problem is influenced by the smoothness, the stability and the uncertainties [22]: such criteria can be used to evaluate the difficulty of the modeling task.

3.1.2 Resolution Strategies

The existence of a model does not imply that it can practically be solved. The search for an analytical resolution is often illusory and work-around strategies have been heavily used, first of which analogous models, relying on the assumed similarity of two physical phenomena. Phenomenological models focus on key features of the studied phenomenon, without regards for incoherences with phenomena occurring at lower scales. Numerical discretization allows the subdivision of a global unsolvable problem, approximating the solution with local contributions.

A deep understanding of a phenomenon and the design of adequate models does not imply the ability to predict it: the designed model must also be solved.

Historically, the description of many mechanical problems by partial differential equation (PDE) and dynamic laws did not immediately allow the prediction of such phenomena with analytical resolution. Indeed, not only the efforts required for the resolution can be prohibiting, but every however slight modification of the studied system may require an altogether new strategy⁷. In fact, even without regards to mathematically unsolvable cases, the analytical solutions are often limited to canonical study cases [231, p.312], the general cases of arbitrarily complex geometries requiring crude global approximations.

The rise of numerical and discretized approaches opened the route to the efficient resolution – which does not necessarily lead to a deeper understanding⁸ – of large complex systems. They rely on the subdivision of a system in many easily solvable elementary problems [46, p.2]. This approach is described in Section 3.1.3.

A widely used work-around to solve complex problems is to rely on physical analogies between phenomena, potentially of distinct nature [18, vol.1, p.300]. The behavior of an easy to study, e.g. easy to measure, phenomenon is used as a model of another one.

To some extent, graphical calculus is an analogous resolution strategy: measurements on well chosen scaled drawing are the model results. Graphical calculus is based on the use of traditional drawing instruments and used to be a widespread tool for industrial applications, providing means to compute static⁹ or dynamic efforts [147, 81] in mechanical

⁶Poincaré proved the impossibility of solving the problem by first integrals. A purely mathematical solution up to n bodies, based on convergent series, exists but is without any practical interest to study the phenomenon [57, p.68–70]. The mathematical description of chaos changes nothing to its physical nature.

⁷“without proper regard for the individuality of the problem the task of computation will become hopeless” [46, p.12].

⁸The human intellectual capacities being limited, an adequate reduction of the number of manipulated variable is always necessary for the practical use of a model [238, p.251].

⁹The most canonical example arguably being the Cremona diagram for statics of trusses [81, p.37]. This method was widely used to solve lattice-like problems, discussed in Section 5.3.1, before the rise of computerized methods.

systems and to study complex three dimensional problems¹⁰. These methods also provide generic tools to derive or integrate functions, by geometrically estimating surfaces, weighing cut out surfaces or using measuring devices as the planimeter [19, p.374].

Leaving aside drawing methods for less abstract analogous models, analogy has historically been used by Antoni Gaudí to design arches with evenly distributed load, with reverse scale model made of chain assemblies and weight sand bags [175, p.46]. A more generic example is the once widespread use of electric circuits to solve ordinary differential equation, tuning resistors, capacitors and inductors to adapt to the problem parameters [19, p.374]. A similar electric analogy was even used to model the partial differential equations of fluid dynamics in two dimensions [143]. For similar problems but with a distinct approach, Atanasoff designed a carving tool to iteratively solve the Laplace's equation with a cube of wax [39, p.876]. An example from more fundamental physics is the study of the atomic structure of liquids by scale models¹¹, first with assemblies of balls and rods [29] and then by direct measurements on heaps of ball-bearings [30] (Figure 3.3).

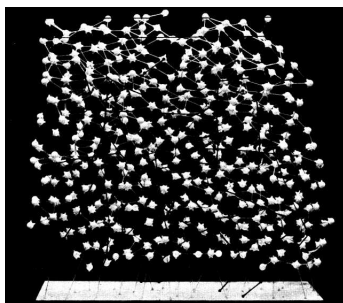


Figure 3.3: Example of analogous scale model. Scale model of a heap of 10^3 ball-bearings, used to investigate the structure of molecules in liquids. Illustration from [30, plate 15].

More focused on solid mechanics, examples include the determination of elastic stress fields by photoelasticimetry [231, p.312–320] and the simulation of metal hot forming by room temperature forming of lead or plasticine, tracking local strain field by engravings of colored layers: the visioplasticity technique [51, p.227].

In the absence of powerful numerical solving capacities, those models were based on experimental techniques on a physical setup. An analogous model can also be purely conceptual: a conceptual model designed to simulate another one, and where dominant phenomena are expected to be analogous [105, 72, 166].

3.1.3 Numerical Resolution

Numerical approximated and discretized approaches rely on the subdivision of a complex system – often in time and space – into many easily solvable elementary problems. From a conceptual point of view, key features of numerical methods appeared very early in history. However, the quantitative computing power available deeply shapes the qualitative design of the model.

¹⁰ For example the descriptive geometry [142, p.20–25] and its application to the drawing and the development of surfaces in boilermaking [125].

¹¹ Similar goals have been investigated with numerical models, including at a contemporary period [7].

3.1.3.1 Halley's Comet

The idea that complex models can be solved by numerical discretization is not recent in history. Many typical characteristics of modern numerical methods appeared long before massive computing power was available.

Limiting ourselves to models somewhat similar to the method used in this PhD, a pioneer attempt to discretize an otherwise unsolvable physical model can be traced back to 1757, when Clairaut, de Lalande and Lepaute attempted to study the dynamics of the system {Sun, Jupiter, Saturn, Halley's Comet} by numerical approximation [92, p.20].

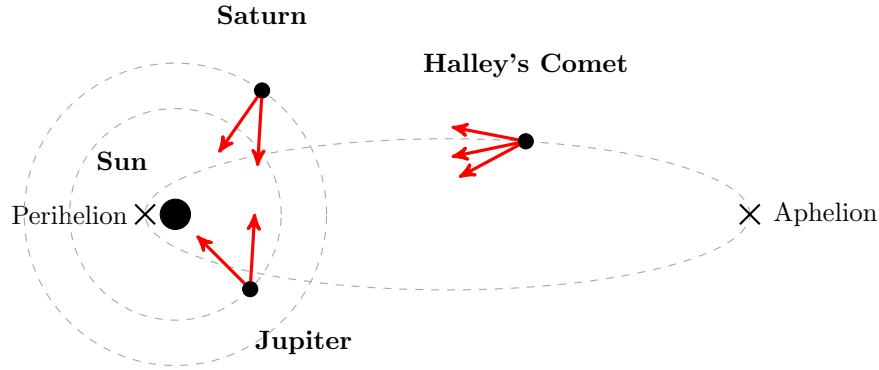


Figure 3.4: Sketch of the 1757 attempt to compute the orbit of Halley's Comet, with considered gravitational forces. The modeling objective was to predict the next perihelion of the Comet. The Sun is considered fixed, the actions of the Comet on Saturn and Jupiter are neglected. The trajectories of Jupiter and Saturn are computed in parallel with the trajectory of the Comet.

A remarkable characteristic is that key features of modern numerical methods were displayed:

- They chose a dominant physical phenomenon (gravitational attraction between massive points) and simplification assumptions (the Sun is fixed, Halley's Comet actions on other bodies are negligible).
- They discretized time, updating the state of the system every 1 or 2° on the orbits of Jupiter or Saturn.
- They designed a calibration procedure for their approximated method, based on experimental data from 1531, 1607 and 1682 previous perihelia of Halley's Comet.
- They estimated the error on the result, which proved to be of the correct order of magnitude¹².
- They parallelized the computing effort: de Lalande and Lepaute computed the state of the system {Sun, Jupiter, Saturn} and passed their results to Clairaut, at the other side of the table, for him to compute Halley's Comet position.

¹² The prediction was 1758/04/15 \pm 1 month, the actual perihelion occurred on the 1758/03/13, after 76 years of revolution. Even though, Jean le Rond d'Alembert considered that their numerical method was not of any help in understanding Halley's Comet, they respected the predicted order of magnitude of the time precision of 0.1%, while periods range from 74 to 79 years, more than 6% of variation.

Although all the calculations were done by hand – it took them five months of efforts to calibrate their method and predict the Comet’s next perihelion – their approach is in essence very close to modern numerical model resolution.

3.1.3.2 Model design and Computing Power

The computing power available deeply influences the modeling choices. Since the 40s, the computing speed of sequential machines increased by more than ten orders of magnitude. Such a quantitative leap made possible otherwise unreasonable approaches. In the last decade, the sequential speed tends to stagnate but parallel computing leads to a massive increase in computing power. To take advantage of this technological shift, specific care is required in the design of a model.

Initially purely manual, calculations were progressively machinized and standardized [70, p.6], to improve speed and accuracy, while limiting errors¹³. Among classical tools for human calculators we find tables and abacus – regrouping precomputed values of functions¹⁴ – slide rules [19, p.374], counting frames and diverse computing machines.

In parallel to technical improvements, a strong emphasis on labor division and organization allowed to distribute large computing tasks in teams of calculators [91]. Such techniques remained competitive in the infancy of numerical computers, before being rendered obsolete by this much faster challenger.

The chaotic history [101] of numerical computers developing speed, available memory, versatility, standardization and ease of programming will not be detailed here. It must however be underlined that the order of magnitude of the computing speed (Figure 3.5) possible with digital electronic computers are altogether impractical with other techniques, human computation or mechanical devices [39, p.877].

For decades up to the beginning of this millennium, around 2004 [74], the progresses of the computing hardware were mostly driven by frequency scaling: the increase of the processor frequency increased the number of operations executable by a machine in a given time (Figure 3.5). Arising issues, among which power consumption of machines, led to a shift from frequency to parallel scaling (Figure 3.6). Without a radical technological breakthrough, the number of operations per cycle and the frequency of the processors may not dramatically increase, leaving unchanged the current order of magnitude of executable operations per second on a single processing unit.

To increase the computing power, industrial companies shifted from building faster processing units to design machines with more units: a shift from frequency scaling to parallel scaling. Faster machines are built associating multiple processing units, from a pair to 10^7 for Sunway TaihuLight¹⁵, the current fastest machine of the TOP500 list [219].

In consequence, gains in computing power do not imply any gain in the running time of a sequential implementation. An additional development effort is required to parallelize the codes and take advantage of the machine resources: portions of the algorithm have to run simultaneously on various processors. Actual computing time gains can be rather limited [16] and the nature of the algorithm sets an asymptotic limit [101, p.41], to possible time gains with parallel scaling. The irreducible critical path is intrinsically

¹³The approximation of π to 707 decimals – manually computed between 1853 and 1873 – was only found in 1946 to be erroneous after the 527th decimal place [71]. The automation of the computations and the use of machines considerably reduce such errors, although flaws directly stemming from the hardware can still be found on modern computers. See for example the Pentium FDIV bug discovered in 1994, leading to incorrect results in floating-point division at the 5th decimal place [159], and the Intel Skylake processor bug, recently found, where hyper-threading activation leads to segmentation faults [127].

¹⁴Including printed tables of random numbers [205, p.625–629], strange-looking collections in a modern context.

¹⁵Sunway TaihuLight uses SW26010 chips, shown on Figure 3.5, operating at moderate frequency by modern standards.

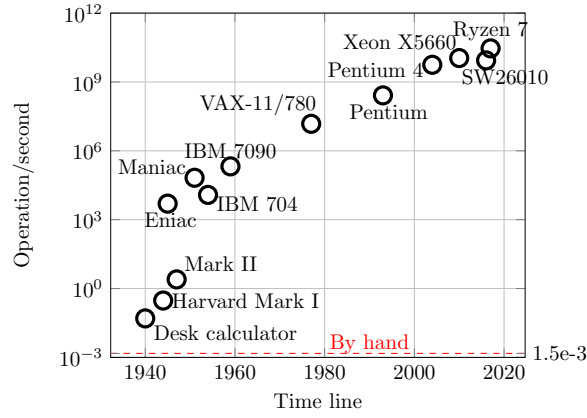


Figure 3.5: Rough orders of magnitudes of computing speed for single processing units. The performance of modern machines can no longer be measured on this scale, they are designed to use multiple units in parallel. Mixed data for multiplication and addition of 10-digit and 16-digit numbers, from [88, p.137], [39, p.887], [148, p.33], [63, p.14], [67] and [219].

sequential¹⁶. Whatever the strategy used to save *time* by massive parallelization, the required resources in terms of *energy* is rather intrinsic to a resolution tool: modern computers require roughly 10^{-9} J to execute a floating point operation [3].

From handmade calculation to massively parallel architectures of modern clusters, many key principles are shared by the numerical methods, independently of the tool used for the resolution. The main invariant issue remains the compromise¹⁷ between accuracy and computation time, which is sometimes closely related to cost. Regardless of the available machines, computing times approaching 10^4 h are impractical, computing times of 10^6 h are ridiculous [238, p.249].

Thus, conceptual numerical methods can only become of any practical interest when an adequate computing power is available. Very small test cases used during this PhD, for example running within minutes on a Xeon X5660, would take about 10^7 years to compute by hand and would have taken months to compute on Maniac, the fastest machine of the 1950s¹⁸. A quantitative order of magnitude change in the available computing power implies a qualitative change in modeling approaches that are *possible* [19, p.376] [33].

¹⁶As a quick example applied to DEM simulations: the particles can be distributed on several processing units, but the explicit time integration is sequential. See also Section 8.5 for a discussion regarding potential algorithmic work-around.

¹⁷The fact that technical progress does not change the nature of key compromises seems rather generic [43, p.95].

¹⁸Maniac was the machine used to implement and test for the first time DEM-like algorithms (refer to Section 5.3.2.2 and [166]).

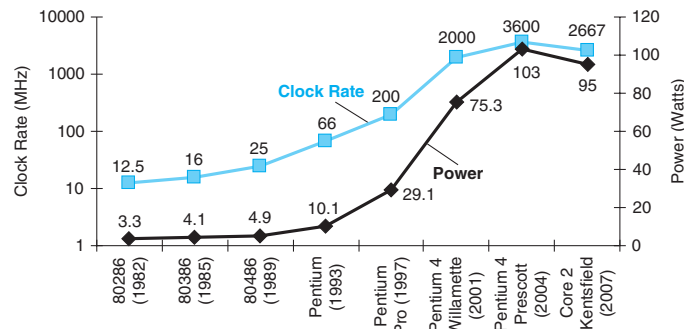


Figure 3.6: Frequency and consumption of Intel x86 processors: the Prescott consumed much more for little performance improvement. Thermal issues led to the abandonment of the Pentium 4 line, replaced by chips with lower frequency and multiple processors. Illustration from [63, p.14].

3.1.3.3 Computerized Numerical Approach

The massive computing power available is not a guarantee of the quality of a model. For complex models and modern computer architectures, a comprehensive and systematic “error-proof” model is technically unreasonable and contradictory with computing performances. Specific safeguards are required to assess the quality and the potential domain of application of a model.

Modern computing techniques open the route toward the modeling of complex systems. The model life cycle is a long route starting with a physical observation of a dominant physical phenomenon, turned into an idealized conceptual model, translated in a mathematical formalism, algorithmically discretized to be solvable by numerical means and finally implemented in a given programming language.

Although a clear distinction cannot always be respected, the issue of evaluating the credibility of a computed result can be divided into three sub-problems [203]:

- The evaluation of the accordance of the studied physical phenomena with the proposed conceptual model, the *qualification* of the model¹⁹.
- The fidelity of the final computed results to the conceptual model, the *verification* of the model.
- The *validation*, which studies the accordance of predict behavior and measurements of the physical phenomena.

Validation and verification are common practice in model design. Qualification is rarer when a strongly based literature is available for the studied phenomena. On the generic sketch for credibility assessment in Figure 3.7, an alternative route is proposed to solve a conceptual model: an analogous model can be designed, typically in the objective of an easier resolution. Such an approach introduces a supplementary “layer” between the model and the physical phenomena. In this specific case, the exact status (validation or verification?) of the benchmark of the results of the analogous model with reference results can be somewhat ambiguous. More generally, the strict respect of this terminology

¹⁹ The *qualification* terminology, introduced in [203], seems little used in the literature.

is not always practical. Its main merit lies in the attempt to systematically shed light on various potential pitfalls.

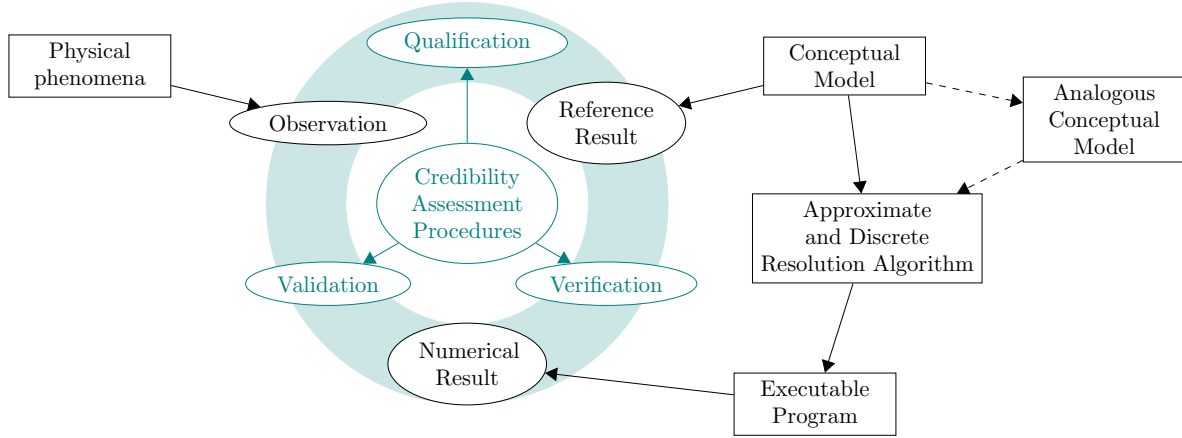


Figure 3.7: Possible credibility assessment methodology in the modeling process. An alternative route is proposed for the resolution of the conceptual model: an intermediary analogous conceptual model. All the steps to move from one block to the other are potential error sources. Inspired from [203].

The use of computerized numerical approaches introduces specific issues in the verification phase, where the accordance of the results to the conceptual model is checked. Errors can be classified as stemming from distinct sources [162, p.9]:

- Discretization (spatial and temporal);
- Iterative procedure convergence [211, Chap. 6 and 8];
- Computer round-off [87];
- Computer programming²⁰.

For many numerical model implementations, a rigorous mathematical approach – proving existence and uniqueness of the solution, computing convergence rates and accumulated round-off errors – is impractical. Similarly, robust programming methodology are time and memory consuming, both at implementation and at run time [189, p.15]. Not only the accuracy of a computerized model can be questioned, but – even with improved accuracy and stability – the reproducibility of the results is not guaranteed²¹ and is contradictory with the quest of fast computations [185].

Somewhat like in the design of models of chaotic physical behavior, a nonspecialist in computing science is bound to evaluate the reasonable metrics of interest of an executable program and delimitate ranges of sensible use, confronting it with test cases and physical measurements.

²⁰ “There are two ways to write error-free programs; only the third one works.” [168, §40]. In the context of numerical modeling of physical phenomena, the likeliness of bugs in elementary processing units (see footnote 13 on page 43) can be considered negligible with respect to programming errors: “the chips are one of the least likely sources of error; user input, application software, system software, and other system hardware are much more likely to cause errors.” [159, Q11].

²¹ A standard library as Intel’s MKL is only reproducible under drastic conditions [185, p.7].

3.2 Description of the Studied Phenomena

The observed physical phenomena can be described and idealized, which is a preliminary step in a modeling approach. Some overall effects of the motion and interactions of interfaces of solids are briefly looked into. Classifications are proposed for interface types and topological events. Such taxonomies are not always of practical interest, but can serve as a guide in the design of a model.

As a rough conceptual model, based on the observations of the physical phenomena of interest (Section 2.5), we consider a collection of finite solid continuous media, seen as distinct objects. The objects can undergo finite geometrical transformations, i.e. translation, rotation and strain [192, p.59] that are not considered infinite, as the intimate mixing of two fluids, nor infinitesimal²². In addition, the strains are considered irreversible and isochoric and the elastic effect are considered negligible.

The objects can also mechanically interact with one another – or with themselves – through contacts of their material boundaries, the interfaces. The interfaces are spatial discontinuities. A canonical dichotomy of discontinuities in solids, is to describe them as *weak* or *strong* (Figure 3.8).

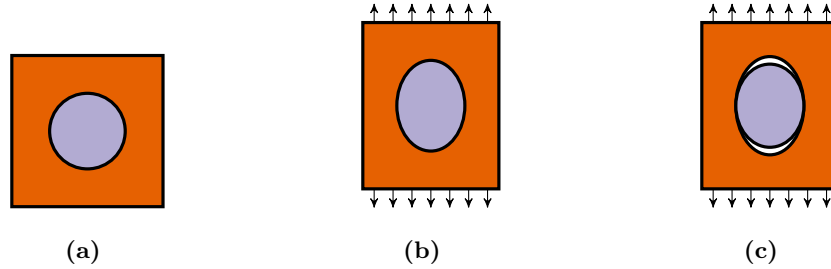


Figure 3.8: Weak/strong typology of discontinuities, example of a bi-material. (a) Reference configuration of a bi-material. (b) Weak discontinuity: the displacement field is continuous across the interface. The derivatives of the displacement field, e.g. the strain field, may be discontinuous across the interface. (c) Strong discontinuity: the displacement field may be discontinuous across the interfaces. A strong discontinuity does not necessarily imply decohesion.

A weak discontinuity is a model where no relative motion is possible across the interface. Although it is often an oversimplification, a weak discontinuity is a convenient way to idealize cohesive interfaces between objects. Material properties, strain and stress fields can be discontinuous across the interface; but the displacement field is described as continuous. Conceptually, a single object is modeled, with changing properties at the interface. A strong discontinuity model allows arbitrary relative motion between objects, conceptually modeled as distinct. The displacement field can be discontinuous across the interface, allowing interacting interfaces to model contact phenomena.

A weak discontinuity description can often be a first modeling step, allowing the use of a continuous topological framework, by changing only material properties from one side of the interface to the other. A model describing strong discontinuity must include a discrete formalism to compute the mechanical reaction to the interaction phenomena, typically to prevent inter-penetration of the objects [54]. Such algorithms must cope with

²² Infinitesimal transformation theories consider that the geometrical variations between the deformed and reference configurations are negligible with respect to the studied length scale.

displacements and changes in shapes of the objects: motion and potential interaction of the interfaces must be handled.

From the point of view of their detection in a modeling framework, two categories of interface interaction can be distinguished (Figure 3.9). A first category is the interaction between distinct objects, hereafter referred to as *contact*. Objects involved in a contact are enclosed within their own boundaries. The distinct objects can be described by non intersecting sets of material points. At a finite precision, it is theoretically possible to explicitly list all members of the objects, and to test membership to detect contact. In contrast, in the *self-contact* category, the boundaries of a unique object are interacting, the listing of material points membership is not sufficient anymore to detect contact.



Figure 3.9: Contact/self-contact typology of discontinuity interactions. (a) Contact between two distinct inclusions in a matrix. (b) Self-contact of the boundary of a hole in a matrix.

In the case of limited strain of the materials, the tracking of material points located at the interfaces could be a sufficient strategy to detect contacts and self-contacts. In finite transformation context, the potential migration of material points makes it challenging to design fully automated contact and self-contact detection algorithm. Indeed, the deformation of an object can constrain material points to migrate toward or away from a boundary (Figure 3.10), making the contact detection task non-trivial for modeling tools working at a finite precision.

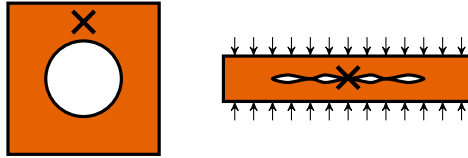


Figure 3.10: Contact detection in finite transformation context, example of a hole in a matrix. Material points initially far from interfaces might migrate toward them. Thus, using a finite precision, tracking material points initially on the interfaces is not sufficient to detect self-contacts.

The potential creation, motion and interaction of interfaces can lead to topological changes of the objects. A topological event does not necessarily stem from radical changes in the shape of an object. From a mathematical point of view and illustrated with a mechanician's words, topological events occur when successive configurations are not:

- Homotopy equivalent [235, p.163], when the number of objects or of holes changes (e.g. Figures 3.11b and 3.12).

- Homeomorphic [235, p.57], when the configurations are homotopy equivalent, but the number of endpoint changes (e.g. Figure 3.11a).



Figure 3.11: Examples of topological events with negligible strain of the solids. (a) The initiation and the branching of the crack are not homeomorphic configurations. The sole propagation of the crack is mathematically not a topological event. Regardless, the phenomenon may be delicate to model. (b) The inclusion fragmentation or healing of the inclusion are not homotopy equivalent. No topological change for the matrix.

For example, Figure 3.11b, an inclusion is split into two parts: the number of “objects” changes, the configurations are not homotopy equivalent. In Figure 3.12a, as the pore opens, the configurations are not homotopy equivalent: the object remains unique, but now presents a “hole”.

In contrast, the configurations before and after the branching of a crack (Figure 3.11) are not homeomorphic: only the number of “end points” varies. In the final configuration, the theoretical removal of the bifurcation point would separate the crack into three “parts”. Before the branching, two “parts” at most can be obtained by the removal of a single point on the path of the crack.

In the context of solid mechanics and working at finite precision, rigorous distinctions are probably too abstract to be practical, both with respect to the observation of the physical phenomena and the numerical models. Typical configurations of interest are:

- The creation of the removal of holes (Figure 3.12a) or endpoints (Figure 3.11a) in objects.
- The merging and splitting processes of objects or holes (Figure 3.12b).

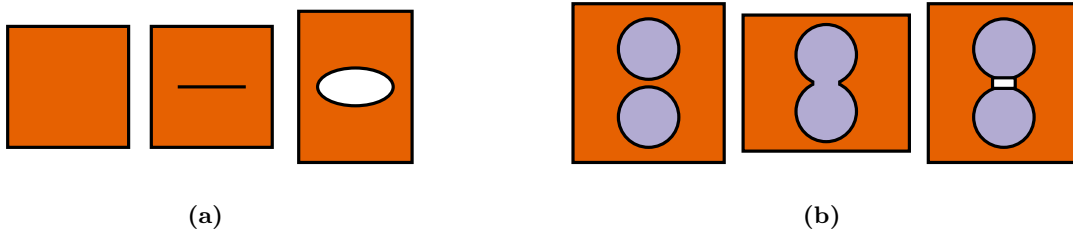


Figure 3.12: Examples of topological events. (a) Pore opening or closure. (b) Neck creation and breakage: merge and split of the inclusions, merge of the two holes in the matrix.

More than the mathematical typology of topological events, our conceptual model must handle operations on boundaries of the objects, including their creation, destruction, merge and split.

3.3 PhD Objective: Requirements for a Modeling Tool

Based on the description (Section 3.2) of the observed phenomena (Section 2.5), practical guidelines can be drawn for the modelization objective.

A model and its resolution strategy shall be designed to:

1. Describe the quasistatic deformation of cohesive and continuous media.
2. Describe irreversible finite transformations in continuous media. The possibility of tracking transformations from a reference configuration may be of interest²³. However, the assumption of small displacements will not be considered reasonable: reference and deformed geometries will be distinguished. Elastic reversible processes will be neglected.
3. Describe cohesive materials, coping with tensile and compressive loads, displaying a resistance to shape modification. Mechanical transformations shall conserve volume. The volume conservation will be a metric of quality of the model.
4. Describe inelastic strains, with typical plastic or viscoplastic behaviors. The accuracy of the representation of a targeted stress/strain behavior will be a metric of the quality of the model. Norton law and perfect plasticity will be typically represented behaviors, with little focus on more complex constitutive laws.
5. Describe multi-material flows, with an arbitrary finite number of distinct continuous phases.
6. Provide a detection method for contact events, the interaction between distinct phases, and for self-contact events, the interaction of a phase boundary with itself. The reliability of the interaction detection algorithms will be a metric of quality of the model.
7. Handle an arbitrary finite number of distinct interactions occurring simultaneously. The model shall handle simple interaction behaviors as impenetrability or cohesion. The ease of extension toward more complex interaction behavior, both for weak and strong discontinuities, will be a subjective indicator of quality of the model.
8. Handle topological modification of the continuous phases: merging and splitting. Topological events also concern holes in the phases: opening, closure, merging and splitting. The ease of handling of such events will be a subjective indicator of quality of the model.

²³For example to model history dependent phenomena, which is not a major focus of our study.

Part II

Review of Modeling Strategies

4	Key Modeling Choices	57
5	Lecture Grid: Lagrangian Methods	61
5.1	Overview	62
5.2	Continuous Constitutive Law	64
5.2.1	Variational/Global Resolution	64
5.2.1.1	Basis Function Built on a Mesh	64
5.2.1.1.1	[FEM] Conforming Mesh	65
5.2.1.1.2	[MPM] Lagrangian Markers	66
5.2.1.1.3	[XFEM] Extra Basis Function	67
5.2.1.2	[EFG] Basis Function Built on a Cloud of Nodes	67
5.2.2	[SPH] Direct/Local Resolution	68
5.3	Discrete Constitutive Law	70
5.3.1	[Lattice Model] Global Matrix Resolution	70
5.3.2	Particle-Wise Local Resolution	71
5.3.2.1	[NSCD] Backward Time Scheme	72
5.3.2.2	[DEM] Forward Time Scheme	72
6	Comparison of Selected Methods	75
6.1	FEM, EFG, SPH, DEM and lattice model	75
6.2	Ongoing Challenges	77

In Part I, the general context of the PhD has been introduced. Initiated from an experimental perspective, the study of composite forming led to specific modeling needs. Toward the understanding of the deformation mechanisms, a numerical method able to handle interface interactions and topological events is complementary to the designed experimental setup.

Part II reviews potential numerical methods in computational solid mechanics to describe finite transformation in multi-phase materials. The part is organized in three chapters:

- Chapter 4 briefly presents two key modeling choices: the kinematical standpoint and the topology of the description. Only Lagrangian methods are further considered.
- Chapter 5 is a lecture grid of diverse Lagrangian numerical methods, based as much as possible on algorithmic features. A graphical outline of the chapter is proposed on page 63.
- Chapter 6 focuses on the comparative description of selected methods and their potentiality.

Highlights - Part II

Review of Modeling Strategies

- Simulation strategies are shaped by two key modeling choices: the kinematical description of the flow and the topology of the constitutive law of the material.

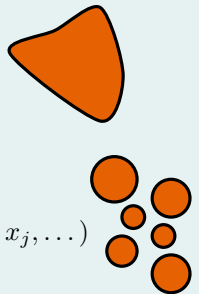
Eulerian kinematics are considered as too costly to accurately track numerous interfaces. Among Lagrangian kinematics, two topological approaches are studied: approaches based on a continuous constitutive behavior (solving a partial differential equation) and approaches based on a discrete law (mimicking continua with a set of interacting objects).

- A partial lecture grid, focusing on algorithmic features, can help to highlight distinctions and similarities for a selection of numerical methods.

Methods based on a continuous topology display a variety of strategies to include the description of discontinuities in their framework. Methods based on a discrete topology can in turn propose an analogical route to mimic the behavior of continuous media.

- Among potential modeling tools, the discrete element method (DEM) is innately suited to handle numerous contacts and topological events.

Its versatility and the ease of implementing arbitrary behaviors makes it an appealing tool. To our knowledge, no existing DEM algorithm can describe the inelastic finite strain of incompressible continuous media.

$$\mathcal{L}(x) = f$$


$$f_i = \sum_j f_{j \rightarrow i}(x_i, x_j, \dots)$$

Figure 4.1

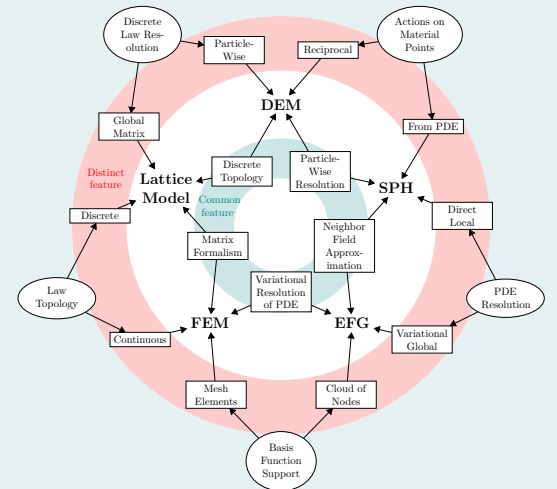


Figure 6.1

Chapter 4

Key Modeling Choices

In this short chapter, the conceptual distinction between Lagrangian/Eulerian and Discrete/Continuous modeling approaches is examined. Examples of numerical methods are given and the DEM, used in this PhD, is located.

Two key modeling choices will deeply shape the strategy to model deforming materials. We focus here on potential techniques to model the continuous ideal phases described in Section 3.3.

The first key modeling choice is the topology of the material constitutive law (see Figure 4.1): an idealized material phase can either be considered as a *set of discrete interacting objects*, or as a *continuum*¹.

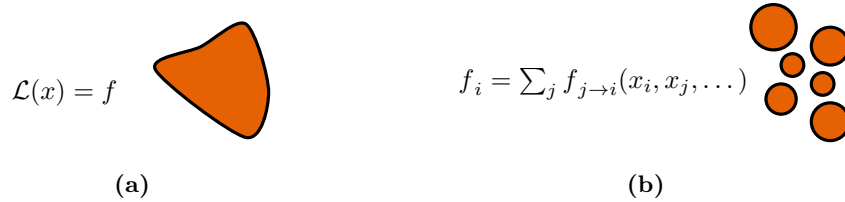


Figure 4.1: Key modeling choice: the conceptual topology of the idealized material. In both cases, the unknown field of the system is x . (a) Continuous material: within a domain a continuous constitutive law is valid, typically described by a partial differential equation (PDE), with a differential operator \mathcal{L} and a second member f . (b) Discrete material: distinct elementary entities interact with each other. The action f on a given entity i is the summed effects of all its interactions with surrounding entities j .

The second key modeling choice is the kinematical standpoint used to describe the flowing material [156, p.45] (see Figure 4.2): a *Lagrangian* (or *material*) approach, where the position of specific material points is tracked over time; or an *Eulerian* (or *spatial*) approach, where the flow, seen from fixed points, is measured.

To fix ideas giving concrete examples of numerical methods (see also Table 4.1), Lagrangian finite element method (FEM) is a dominant modeling tool in solid mechanics simulations, using the motion of a mesh to track the continuous material deformation.

¹In a numerical context, the effective resolution of a continuous description of a material necessarily relies on a discrete re-formulation. The distinction is drawn here at the level of the conceptual model. Algorithmically similar methods may stem from discrete and continuous topologies of the constitutive law (see also Section 6.1).

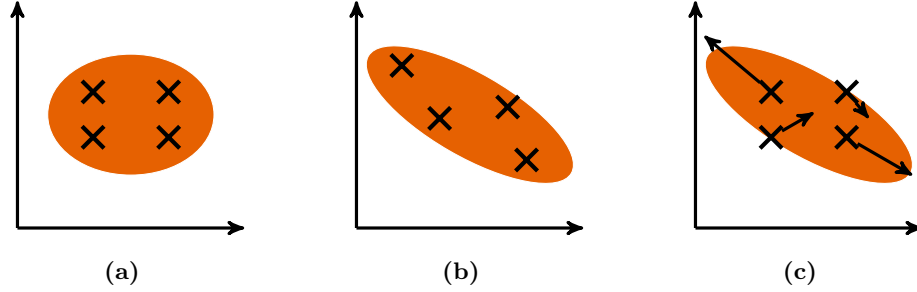


Figure 4.2: Key modeling choice: the kinematical standpoint. (a) Reference configuration, initial computation point position. (b) Lagrangian (or material) standpoint. The position of the material points is explicitly tracked in the deformed state by the motion of the computing points. (c) Eulerian (or spatial) standpoint. The flow of the material is measured at fixed computing points.

Eulerian finite volume method (FVM) is widely applied to fluid simulations, the flow of the continuous fluid is observed from a fixed mesh². The lattice Boltzmann method (LBM) [41] and cellular automata (CA) [174, Chap. 6] [2] are Eulerian descriptions of the flow of discrete objects. The discrete element method (DEM) is a Lagrangian method tracking explicitly the position of sets of discrete objects.

	Kinematical Standpoint	
	Lagrangian	Eulerian
Constitutive Topology	Continuous FEM	FVM
	Discrete DEM	LBM

Table 4.1: Key modeling choices: kinematical standpoint and topology of the constitutive law. Examples of typical (refer to footnote 2) numerical method for each strategy. Positioning of the DEM, the main numerical tool used in this work.

In the context of solid mechanics, typically to model metallic materials at the structural product scale, a framework based on a Lagrangian standpoint and continuous constitutive law – as the classical Lagrangian FEM implementations – often appears more natural [20, p.315]. *A priori*, Eulerian kinematics seem better suited to fluid dynamics, as fluids do not possess a preferential reference configuration [75, p.131]. A discrete constitutive law finds more straightforward applications in the simulation of finite sets of objects, typically granular materials or atomic interactions.

Eulerian kinematical standpoints can be successfully applied to solid mechanics problems. Their intrinsic ability to handle arbitrary strain without mesh distortion are appealing to model flowing-like phenomena, and such frameworks are also popular to model

²It must be emphasized that FVM and FEM are not intrinsically bound to Eulerian and Lagrangian kinematical standpoint, respectively. Both methods are generic variational formulations to solve PDE by weighted residual, closely related and that can be written with common mathematical formalism [108, p.3323]. They differ mostly by the basis function choice: polynomial for the FEM and constant for the FVM [69, p.443]. Both numerical methods can be formalized from both kinematical standpoints and the four possible combinations have been applied to both fluid and solid simulation. Intrinsic ease of application to partial differential equation (PDE) types [108, p.3325] and strong traditions in distinct communities lead to the predominance of Lagrangian FEM for solid simulation and Eulerian FVM for fluid simulation.

fluid-solid interactions. Specific numerical techniques have been developed to allow the tracking of materials and interface positions. Examples in finite transformation context include volume of solid method [5], pseudo-concentration technique [25], reference map technique [116]... Mixed strategies, trying to take advantage of both Eulerian and Lagrangian standpoints were developed, as the arbitrary Lagrangian Eulerian (ALE) [58] and the deforming spatial domain/stabilized space time (DSD/SST) [230] formalisms.

However, the handling of complex three-dimensional interfaces of arbitrary shape are still challenging using Eulerian or mixed Eulerian-Lagrangian standpoints. A heavier computing effort is required to match the accuracy of the interface description reachable with Lagrangian standpoints. Eulerian kinematical modeling option will not be investigated further in this work³. Our analysis will thus focus on Lagrangian strategies for solid mechanics, which natively and explicitly track the phase motion in space with respect to a reference configuration⁴. Both discrete and continuous topological descriptions of the materials will be considered. In the context of modeling continuous media, topologically discrete simulation tools necessarily rely on an analogous approach to build phenomenological models.

³However, Eulerian descriptions of interface motion, within Lagrangian frameworks, are now classical and efficient extensions of Lagrangian methods. They are briefly described in Section 5.2.1.

⁴An *updated* Lagrangian formulation takes the current – or at least a recently computed – state as reference; a *total* Lagrangian formulation is always written with respect to the initial state [20, p.335].

Chapter 5

Lecture Grid: Lagrangian Methods

In Chapter 4, two key modeling choices (topology and kinematics) were examined. To meet our modelization objective, we choose an approach associated in the literature to the prolific “meshless” and “particle” methods.

This chapter is an attempt to build a lecture grid of a subset of the literature landscape of the Lagrangian methods. The objective is to better locate the specificity of our approach. Far from being a strict classification attempt, the lecture grid aims to provide potential comparison features between methods. The chapter is split into three sections:

- Section 5.1 is a short overview of the chapter.
- Section 5.2 describes methods using a topologically continuous constitutive law.
- Section 5.3 describes discrete approaches.

A reader without specific interest in taxonomy issues arising in the vast realm of meshless and particle methods might skip to Chapter 6, where selected methods are briefly compared.

5.1 Overview

Some Lagrangian methods are generic means to solve PDE, accounting for a continuous constitutive law. Specific strategies must be designed to deal with discontinuities. Other Lagrangian methods solve the behavior of systems of discrete interacting bodies. Modeling a continuum is possible, but not straightforward. Up to some point, the two strategies can be converging, especially in the context of a numerical resolution.

As a very general description of the variety of strategies developed in the vast Lagrangian family of numerical methods for solid mechanics, two complementary, and up to some point converging, routes can be drawn. From one side, extensions within continuous constitutive law frameworks were developed to handle discrete phenomena as discontinuities and their motion and their interaction. From the other side, discrete constitutive behavior frameworks were used and modified to phenomenologically mimic continuous behaviors. The naive continuous/discrete distinction is challenged in the context of numerical resolution, by essence discrete at the lowest computing level. As it will be discussed, some borderline strategies may display ambiguous features.

Following the first route – developing discrete phenomena management within a continuous framework – extensions are built on top of a PDE solver. Prolific developments tried to stick as closely as possible to the FEM framework, adding supplementary numerical ingredients for discontinuity management. These strategies, described in Section 5.2.1, take advantage of a strong physical, mathematical and numerical background. Taking one step aside, as discussed in Section 5.2.1.2, the methodology dependency on the mesh can be alleviated building the basis function of the Galerkin method on a cloud of nodes, without prior definition of their connectivity, as in the element-free Galerkin method (EFG). More radically, the FEM variational formulation and matrix formalism can be abandoned, in methods related to the smooth particle hydrodynamics (SPH), by solving directly and locally the partial differential equation, in methods presented in Section 5.2.2.

The second route, more confidential but yet historically early, is described in Section 5.3. Interactions between large sets of independent objects are designed to mimic macroscopic continuous behaviors, thus phenomenologically reproducing continuous behavior with a discrete framework, for example the DEM. Although not as straightforward to describe the continuous phenomena, this approach is innately suited to handle an arbitrary large number of discrete interactions.

The classical sets of methods named in the literature – first of which meshless methods¹ and particle methods² – are broad and provide little help to classify resolution strategy from the algorithmic and conceptual point of view. Indeed, many historical bridges between rather distinct methodologies, and concurrent formalization of essentially similar approaches tend to create a somewhat fuzzy literature landscape. From a practical point of view, the numerical side tools and algorithms used for secondary tasks have a major contribution on the overall exploitability of a conceptual method. The denomination of the method is often heavily influenced by minor algorithmic choices, or even implementation details.

In order to better locate the methodology developed in our work, the lecture grid proposed hereafter attempts to rely as much as possible on algorithmic features rather than final applications³. The objective is not to propose a strict and comprehensive

¹ Meshless methods, or meshfree methods, regroup numerical approaches as global resolution on cloud of nodes (Sections 5.2.1.2), direct and local resolution on independent points 5.2.2, over-impression of Lagrangian markers in a continuous framework 5.2.1.1.2 and topologically discrete constitutive law 5.3.

² Particle methods can include practically any modeling approach handling data at nodes or material points.

³ In the literature, the claimed filiation of the introduced methods sometimes focus on the modeled phenomena. For example, the work on elasticity in continuous media of Hrennikoff [105] is frequently

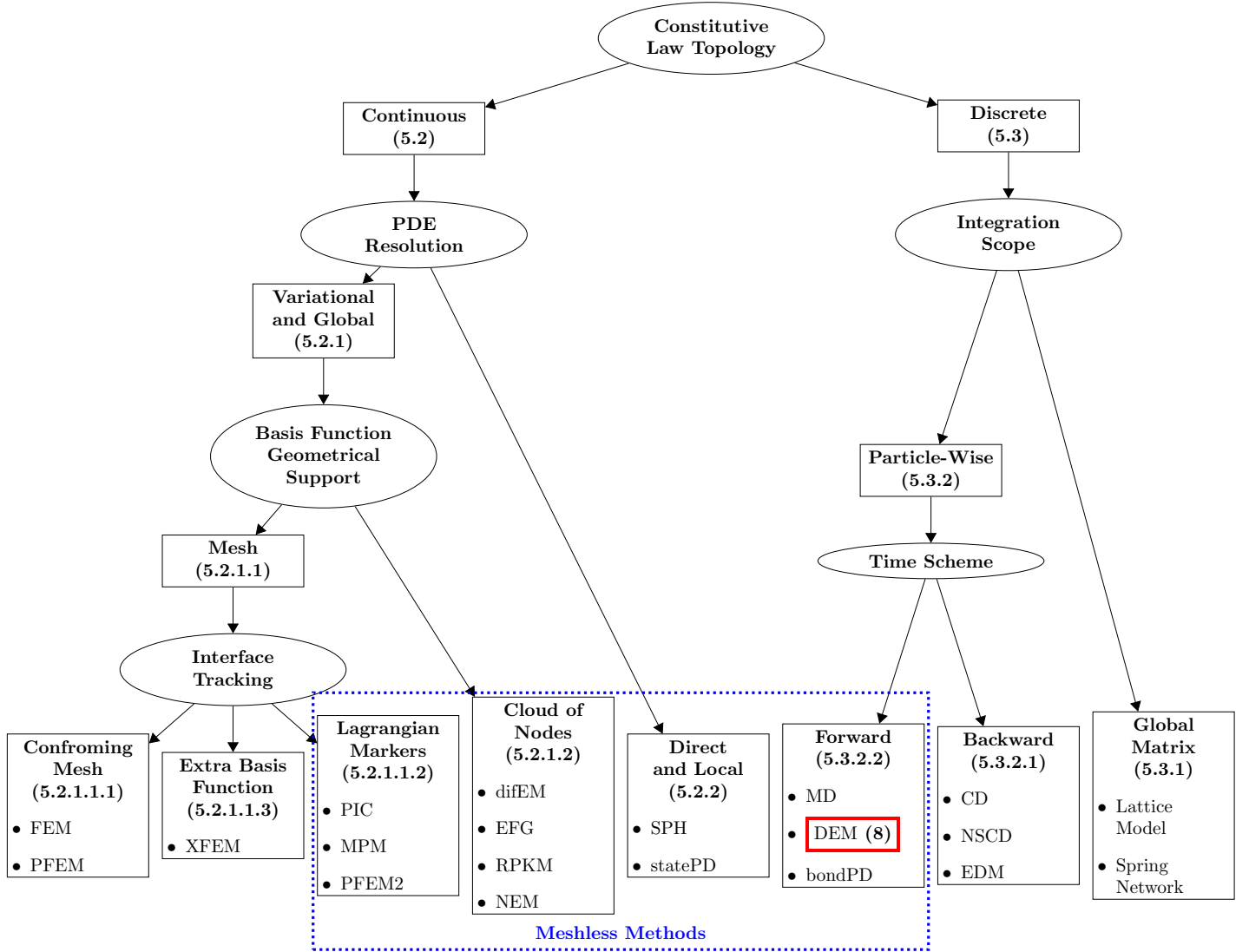


Figure 5.1: Reading grid for the comparison of some numerical strategies based on a Lagrangian kinematical standpoint. The outline of Chapter 5 is based on this graph, from which the corresponding sections to the modelization choices are cross-referenced. Positioning of the meshless methods set and the discrete element method (DEM), the main numerical tool used in this work. See also Figure 6.1 for a pairwise comparison of a selection of methods.

classification, nor a historical review⁴, but to contribute to the identification of similarities and specificities of numerical approaches.

5.2 Continuous Constitutive Law

A solidly grounded approach to model the deformation of solids is to solve the PDE describing their continuous behavior. To handle discrete and topological events, numerous extensions within these continuous constitutive law frameworks were developed. Two main resolution strategies of the PDE are considered: variational approaches (Section 5.2.1) and direct resolutions (Section 5.2.2). It must be kept in mind that in the context of a numerical resolution, such continuous models will be discretized at some stage.

5.2.1 Variational/Global Resolution

The variational resolution of a PDE is based on energy minimization. Assumptions are made on the local “shape” of the fields of interest, typically the displacement for solid mechanics. The solution is found by minimizing the energy, within the a priori chosen “shape”. Powerful mathematical frameworks are designed, based on linear algebra, to aggregate local contribution and solve the system globally. Two potential geometrical supports for the shape functions, used for spatial discretization, are dealt with: a mesh (Section 5.2.1.1) and a cloud of nodes (Section 5.2.1.2).

5.2.1.1 Basis Function Built on a Mesh

The de facto dominant tool to solve PDE in solid mechanics is the FEM. The method is a variational resolution based on a mesh as geometrical support. After a short introduction, three potential strategies to cope with discontinuities will be investigated: a conforming mesh (Section 5.2.1.1.1), the use of Lagrangian markers (Section 5.2.1.1.2) and the use of extra basis functions (Section 5.2.1.1.3).

Since decades, the FEM [55] is a strongly established tool for solid mechanics, and more generally to solve partial differential equation systems. The method is based on a finite set of degrees of freedom, used to describe the motion of a continuous medium by a judiciously guessed interpolating function. The solution is found by minimizing the energy of the system, using a matrix formalism.

In the first related attempts by Rayleigh and Ritz the deforming shapes of linear vibrations were guessed globally, for the whole system [184, Chap. 4][188]. The methodology proved to be impractical for arbitrary geometries⁵. To deal with arbitrary geometries, Courant thus chose to subdivide the system into small domains and used polynomial shape functions in each subdomain [46]. The rise of computing power triggered the development of such methodologies, and led to the success of the FEM [233].

The continuous unknown fields are discretized on a mesh, with a polynomial approximation on its elements. The resolution is based on variational formulation – energy minimization principles – integrating and summing-up local contributions from the elements into global grand matrices and vectors, solving the overall problem by linear algebra. Versatile and generic, the FEM relies on strongly grounded physical and mathematical tools, built or adapted to meet the mechanicians’ needs in a wide range of applications.

cited as an early precursor of the FEM, although it conceptually is much closer to the DEM [98, p.149].

⁴ One could refer to [244, p.3] or [70] for the FEM and to [98, p.149] for the DEM.

⁵See footnote 7 on page 40.

Historical challenges – regarding finite transformation, interface and contact handling – triggered intense research and development of work-around and extension for the method.

In the proposed reading grid, the FEM extensions are organized based on the numerical approach chosen to track mobile interfaces. This key issue can for example be addressed by building a conforming mesh (Section 5.2.1.1.1) or⁶ by adding additional data to the continuous description with:

- Lagrangian markers (Section 5.2.1.1.2);
- Extra basis functions⁷ (Section 5.2.1.1.3).

Contact detection and handling are not examined here, as they do not seem to be as closely bound to a specific method than the interface tracking.

5.2.1.1.1 [FEM] Conforming Mesh The most straightforward strategy to track interfaces is to explicitly build their geometrical description, directly using the nodes of the mesh: a conforming mesh. Given a meshed geometry, any standard FEM implementation can handle the description of limited strain. Typically, one can model composites – considering ideal weak discontinuities, perfectly cohesive – and cellular materials. In a conforming mesh context, the description of strong discontinuities can be enriched, for example by the use of a cohesive zone model (CZM) [65]⁸.

In this standard configuration, describing larger strains involves a periodical remeshing procedure of the geometry (Figure 5.2). Indeed, an excessive distortion of the elements leads to numerical errors and ultimately computation failure. The remeshing involves the transfer of the state parameters, which is not trivial and needs specific numerical care to maintain a controlled accuracy [45, 167].

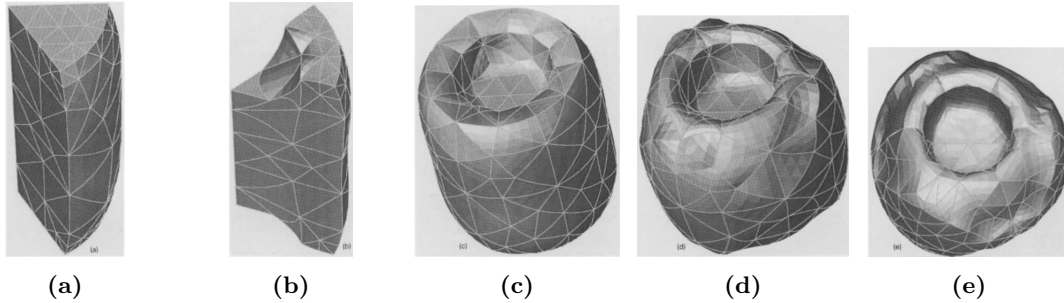


Figure 5.2: Large strain in forging process simulation, remeshing procedures. (a) Initial state and first mesh. The geometry is axisymmetric, only one sector is modeled. (b) Deformed state of the first mesh. (c) Remesh: a second mesh is built and replaces the first one. The axisymmetry will be lost, the full part has to be modeled. (d) Intermediary state with the second mesh. (e) Deformed state and a third mesh is built. Illustration from [45, p.127–128].

An alternative, used in the particle finite element method with moving mesh (PFEM)⁹ [107], is to store all the state parameters at the nodes, and to conserve all nodes at each remesh-

⁶Our lecture grid is by no means comprehensive, the immersed boundary method [170] is another example, among many, of introduction of a dual Lagrangian / Eulerian description.

⁷ Extra basis functions are often collaboratively used with conforming meshes or Lagrangian markers.

⁸This strategy was used to study the room temperature behavior of a composite [73, p.146] similar to our model material (Section 2.3).

⁹ Our reading grid distinguishes the particle finite element method with moving mesh (PFEM) and the particle finite element method with fixed mesh (PFEM2). The PFEM, described in this section, uses

ing iteration. Arbitrary numerous remeshings can be performed without transfer-related accuracy loss, but the cost of the remeshing operations is not lessened.

5.2.1.1.2 [MPM] Lagrangian Markers A somewhat exotic strategy to extend the FEM is to use a set of Lagrangian markers to represent the position and motion of the materials. This supplementary description of discontinuities is overprinted on top of the continuous mesh (Figure 5.3).

Two coordinate systems are simultaneously used:

- A mobile set of Lagrangian points accounting for the material displacements;
- A fixed Eulerian or pseudo-Eulerian grid to perform computations.

Lagrangian material points move from grid cell to grid cell over time and at each step computations are based on the material points present in the cells.

This particle in cell method (PIC) – or material point method (MPM) – strategy has been originally designed for fluid dynamics and was applied very early to two-dimensional multiphase flow, with interface tracking [66]. The method was later transposed to two-dimensional solid mechanics [224, 225]. In this solid mechanics applications, the “fixed” mesh is not always strictly Eulerian anymore. At each step, the mesh is deformed with the material in a Lagrangian way, as in the standard FEM. However, the same initial undeformed mesh is used again at the beginning of the following step, the overall material deformation being represented by the material point motions. A cheap regular mesh can thus be used, without remeshing procedure during simulation.

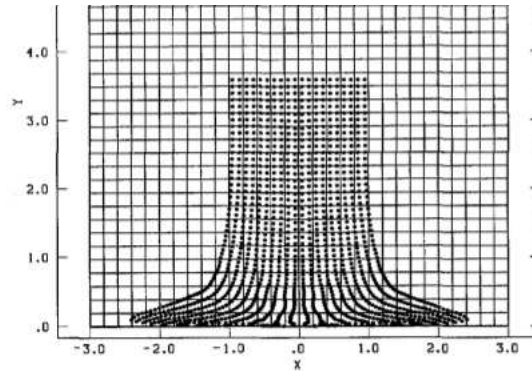


Figure 5.3: Illustration of the PIC from [225, p.247] on a 2D axisymmetric model. Impact of an elastoplastic rod, initially cylindrical, on a rigid surface. Superimposition of the Lagrangian markers and the computing grid for a deformed state.

An alternative formulation aimed at fluid-structure interaction simulations – the particle finite element method with fixed mesh (PFEM2)¹⁰ – was also initially designed for fluid [190] and extended to solid [21]. In contrast to MPM, the computing mesh is purely Eulerian and fixed.

To our knowledge, the use of Lagrangian markers to track the interfaces themselves instead of the material phases has not been transposed from fluid to solid simulations. See for example the moving Lagrangian interface remeshing technique (MLIRT) [37].

a conforming mesh strategy to track interfaces. The PFEM2, described in Section 5.2.1.1.2, uses a set of Lagrangian markers over an Eulerian computing grid. As a side note, both methods also use extra basis function to improve the accuracy of interface definition [107, p.1763].

¹⁰ Refer to footnote 9 for disambiguation between particle finite element method with moving mesh (PFEM) and particle finite element method with fixed mesh (PFEM2).

5.2.1.1.3 [XFEM] Extra Basis Function A more mainstream approach to overprint discontinuities on top of the continuous mesh is to enrich the elements by extra basis functions. Now widely used in both fluid and solid mechanics, this strategy typically uses singular functions to describe interface positions within the elements. Often based on the level set methodology [163], this approach gives an Eulerian description of interfaces within a Lagrangian mesh of the material [157]. In FEM context, a current denomination in the literature is the extended finite element method (XFEM). This strategy allows the study, without remeshing procedures, of problems where interfaces move without perturbing too much the global geometry (Figure 5.4).

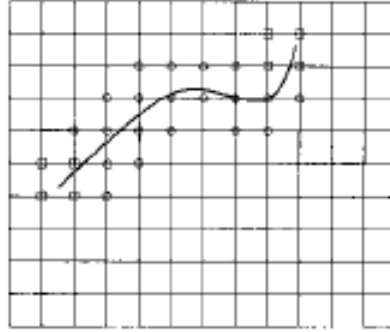


Figure 5.4: Illustration of the principle of the XFEM from [157, p.138]. Arbitrary crack over a mesh. At the circled nodes, enriched shape functions are used.

It must be emphasized that from a mathematical point of view, the level set methodology offers efficient tools to deal with topological events [163, p.26], whose handling is much easier to implement and automate than with Lagrangian descriptions, especially those involving a connectivity table. The drawback, lessened by numerous numerical techniques, is a tendency to lose track of mass or volume [163, p.82].

In the context of finite strains, frequent global remeshing cannot be avoided if a conforming mesh or an Eulerian description of interfaces are used. In addition to potentially prohibiting computing cost, the reliable automation of remeshing and the proper handling of topological events are algorithmic challenges. More generally, the overall quality and efficiency of the methods are often bottlenecked by these side tools. Developing techniques rely on parallelization paradigms, and the simultaneous use of conforming mesh and Eulerian interfaces, as an auxiliary tool for remeshing [45, 28].

5.2.1.2 [EFG] Basis Function Built on a Cloud of Nodes

To alleviate the FEM dependency on a mesh, while conserving the framework and the main methodology, the building rules of the basis function can be modified to rely on a cloud of nodes, instead of a mesh.

As in classical FEM, the partial differential equation system is solved in a weak form, using a Galerkin method, *via* a matrix formulation of the discretized problem. Local contributions are integrated and assembled to build global grand matrices, numerically executed using a background cell structure, with a matrix formalism numerical resolution. However, instead of building polynomial basis function over elements of a mesh, which requires an *a priori* specification of the connectivity of the nodes in the FEM, they rely on a neighborhood of nodes.

To minimize the importance of implementation details, these methods can all be con-

sidered as instances of the partition of unity method (PUM) [150]. The pioneer attempt, the diffuse element method (difEM)¹¹ [158], used moving least-square approximation as partition of unity and was applied to heat conduction. Very close refactoring, the EFG [23] and the reproducing particle kernel method (RPKM)¹² [136], were applied to solid mechanics.

In the natural element method (NEM), the partition of unity is based on the Sibson (or natural neighbor) interpolation [208]. Initiated for geophysical problem of large scale mass transports and geometry changes [232, 199], the NEM was then applied to solid mechanics [223].

Generalized finite difference method (GFDM) approaches [134, 135], working on unstructured grids, are closely related methods [24, p.4].

Although this cloud of nodes approach has been quite popular among FEM-background mechanicians, a major drawback is the numerical cost [24, p.45], intrinsically higher than the FEM, with which it shares a common methodology, and fortuitously higher than SPH, its main challenger. Firstly, to integrate the stiffness matrix [150, p.29]. Secondly, for the matrix system resolution. Indeed, the grand stiffness matrix built in the FEM has a banded structure, stemming from the connectivity table; leading to lower resolution cost. The grand matrices built with cloud of nodes are generally not banded, or at best with larger band width – as nodes being bound to interact with a greater number of neighbors, to ensure numerical stability — inducing higher computational cost of the numerical resolution of the system.

5.2.2 [SPH] Direct/Local Resolution

A historically early attempt to handle drastic changes in size and shape consists in solving directly the partial differential equation without rewriting it in a variational formulation. A cloud of points discretizes the material, but it is not used as a support to build basis function of a Galerkin method. The material behavior, the continuous medium constitutive law, is directly solved locally for each computing point. Although the methods developed are algorithmically discrete, this is a numerical work-around to compute the behavior of a continuum.

A now classical approach uses probabilistic approximation of the local field derivatives, based on a kernel function (Figure 5.5). From this computation, the “efforts” acting on the material points is used to integrate explicitly¹³ their positions and velocity over time. This approach is conceptually and mathematically close to the EFG – they use a similar approximation of field derivatives to solve continuous PDE – and algorithmically close to the molecular dynamics (MD), allowing implementations of the formalism in MD solvers [165, 4]. Initially developed for astrophysics and more generally for compressible fluids [141, 86], the SPH was later on transposed to solid mechanics [131, 130] and successfully applied to finite transformation in three-dimensional context [79].

As a side note, the algorithmic proximity with the MD can prove challenging to our naive distinction: discrete versus continuous constitutive law. The classical SPH requires the use of a kernel to approximate the derivatives, the “efforts” are conceptually not computed pairwise as interaction forces. Some methods are more ambiguous, as the particle and force method (PAF) [52], where pairwise interactions are directly derived from the conservation of energy in a fluid [52, p.15]. More recently, the Mka3D formalism [154]

¹¹ In this work, to avoid confusions, the acronym difEM is used for the diffuse element method. The acronym DEM is reserved to the discrete element method.

¹² It is not always clear in the literature whether RPKM is to be considered closer to SPH (a method described in Section 5.2.2) or to EFG. The former is the claimed filiation in the original paper [137, p.1082] and the latter can for example be read in Sukumar [222, p.3]. The fact that matrices are built within a Galerkin method [136, p.1667] [31, p.1262] oriented our choice.

¹³ Variants using fully [138] or partially [122] implicit integration schemes have also been proposed.

solves the elastic behavior of a continuum in a DEM framework using only geometrical assumptions. Both examples illustrate the limits of our lecture grid.

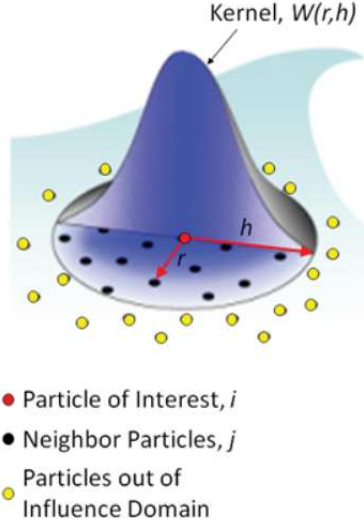


Figure 5.5: Illustration of the kernel function of the SPH [79, p.33]. Fields are locally approximated for each computing point, using the kernel.

Numerous variants of the SPH were developed, using diverse kernel function type, mathematical formalism, stabilization technique and side tool. As an example among many from the literature, the state-based peridynamics (statePD)¹⁴ [210, 77] is an instance of SPH based on a total Lagrangian formulation [83]. Recent developments of the statePD propose an updated Lagrangian formulation [26], as in the original SPH.

A typical numerical issue of the method is the so-called *tensile instability*. This potential inconvenient arises when material points move too far or too close to each other [130, p.73], potentially causing a purely numerical fragmentation [153, p.290]. For example, in solids, when an initially homogeneous spatial distribution is anisotropically deformed, the number of neighbors, within a constant smoothing distance, can become too small in the direction of the preferential strain. The integration of the constitutive behavior becomes unreliable, and the continuity of the media might be numerically interrupted.

As a side-note, this problem – well documented in SPH because of its common use in finite transformation context – can arise very similarly in EFG-like methods¹⁵ and PIC-like methods [66, p.17]. Tensile instability can also be considered as closely related to mesh distortion issues in FEM.

¹⁴In our perspective, state-based peridynamics (statePD) is distinct from bond-based peridynamics (bondPD). The statePD, described in this section, locally integrates a continuous constitutive law to solve a PDE. The bondPD, presented in Section 5.3, relies on a discrete constitutive law, computing interactions between sets of objects.

¹⁵The natural element method (NEM) was designed partially to overcome such problems: nodes are not interacting within a fixed radius as in EFG, but with neighbors defined by Voronoi tessellation. This approach could be considered as a meshing procedure.

5.3 Discrete Constitutive Law

Instead of solving a PDE, a discrete topology of the constitutive law can be chosen. The material is conceptually described as a set of interacting objects. After a general introduction, two resolution strategies are examined: global for the whole system (Section 5.3.1) or particle-wise, like in the DEM (Section 5.3.2).

Radically rooting the description of the material in a distinct formalism, numerical methods can be built on a discrete constitutive law. The common denominator of these approaches is to consider a finite set of interacting discrete elementary objects. Unlike methods relying on a continuous constitutive law (see Section 5.2), discrete approaches are not designed to solve partial differential equation systems.

The mathematical and physical consequences of such approaches must be underlined (see also Section 3.1.1). The built models are often indeterminate when applied to static problems and, in addition, become chaotic for dynamic problems. Numerical methods are thus intrinsically¹⁶ ill-conditioned, regardless¹⁷ of the chosen modeling approach: the error on the final state cannot be bounded for an arbitrary small error of the initial state.

Such limitations imply that dynamic discrete models may, at best, model statistical collective behavior of sets of objects. The exact tracking of the state of individual material points is illusory and is not within the modeling scope.

As a historical note, the numerical resolution of a discrete model may seem more straightforward than the numerical discretization of a continuous model. Early attempts thus used discrete topology to analogically model elasticity in continuous media [242, 187, 105].

Topologically discrete methodologies share three common conceptual ancestors, all of them struggling with necessary computing power necessary to solve such systems:

- Gravitational laws (Section 3.1.3.1), Clairaut modeled only 6 bodies [92, p.20];
- Molecular systems for gases [146] or solids [119], Maxwell computed only average global effects [146, p.62] and Kirsch drew qualitative conclusions, unable to deal with the too numerous unknowns [98, p.149].
- Truss-type systems, for which the first effective computing techniques were designed for discrete systems: graphical [147] (see also Section 3.1.2) or numerical [48]. These methods remained dominant until the rise of numerical computers.

From an algorithmic point of view, drawing a clear distinction between numerical methods inspired by truss or molecular phenomena proves delicate. We thus favored the resolution methodology to build our lecture grid: either a global resolution for the whole system like in the lattice models (Section 5.3.1) or a local approach, particle-wise or interaction-wise, like in the DEM (Section 5.3.2).

5.3.1 [Lattice Model] Global Matrix Resolution

Lattice models are conceptually discrete models: a set of interacting objects is considered. Their resolution is based on linear algebra tools, the local contributions are aggregated in global matrices.

¹⁶Even without taking into account numerical round-off errors, stemming from floating-point arithmetic [87].

¹⁷ As an interesting side note, preliminary hints to bound the error on the dynamics of a discrete system *representing a continuum* have been published [154, p.1533–1536], under restrictive assumptions like infinitesimal strain.

Lattice models [206, 164], or spring networks, use topologically discrete constitutive behavior of the material: linear elastic springs connecting material points for example.

Potential non-linearities (contact, material properties...) are linearized around the studied state to build a global stiffness matrix, for the whole system. The response of the system is solved using this global linearized force/displacement law between the degrees of freedom of the discrete material points.

The methods are often focused on static or quasistatic phenomena [195, 196], efficiently solved in their framework, but can be extended to dynamic phenomena somewhat like the FEM¹⁸, writing a system including mass and damping matrices. To account for geometrical evolution of the system, the stiffness matrix can be rebuilt, or the parameters can be adapted to simulate the breakage of a bond [164, p.42].

5.3.2 Particle-Wise Local Resolution

To take full advantage of the discrete methods, it may be profitable to design algorithm to locally solve such models. For each object within the considered set, the temporal evolution is computed from the surrounding environment. After a short introduction, two temporal integration schemes are dealt with: backward schemes (Section 5.3.2.1) and forward schemes (Section 5.3.2.2).

Many numerical methods use a particle-wise resolution to solve topologically discrete material behavior: The temporal evolution of the system is solved independently for each elementary object (or particle), based on their interactions with others. This methodology allows a very free handling of local non-linearities of the behavior and the effective resolution of complex interactions between numerous objects. Conceptually, this approach is closely related to the 1757 attempt described in Section 3.1.3.1.

The evolution of the discrete system can be numerically handled using two overall time integration schemes¹⁹:

- *Backward* (or *implicit*), see Section 5.3.2.1;
- *Forward* (or *explicit*), see Section 5.3.2.2.

Although the choice of scheme can be of little influence on the final modelization metrics [112, p.2], it must be underlined that the two schemes are not only algorithmic strategies to solve identical systems: backward schemes allow the formulation of interaction laws as strict inequalities, typically for the impenetrability of the particles. In contrast, in forward schemes, such inequalities are regularized, typically with a stiff penalization of interpenetration.

From the computational point of view, the choice of scheme also implies distinct numerical behavior. Backward schemes are unconditionally stable, allowing a user-chosen compromise between computation time and accuracy, while forward schemes are strictly limited (see also Section 8.3).

¹⁸Lattice models share their matrix formalism with the FEM [70]. Even after the apparition of the FEM, at more recent periods than the pioneer papers referred to in the beginning of Section 5.3, lattice models were often proposed as alternative methods, for example to reduce computing cost [118].

¹⁹Backward and forward based methods are sometimes respectively referred to as *non-smooth* and *smooth* methods [111]. However, the degree of smoothness of the interaction laws between particles is not algorithmically constrained by the choice methodology. Although backward schemes are mathematically more satisfying to integrate arbitrarily non-smooth behaviors, forward schemes are almost always used to integrate behaviors of limited smoothness, among which the contact creations and losses.

5.3.2.1 [NSCD] Backward Time Scheme

In a backward time scheme integration, the solver tries to find a solution satisfying constraints in the previously elapsed time step. This strategy allows to properly take into account arbitrarily non-smooth behaviors. In addition, the compromise between computing time and accuracy can be chosen.

Backward (or implicit) time schemes handle, by design, inequalities and arbitrarily non-smooth interaction laws between elementary objects. The instantaneous values taken by the interactions are not computed and thus do not need to be expressly defined. At each step, for each interacting pair [112, p.11], the solvers attempt to find a final solution, satisfying the system constraints and Newton's second law in the time elapsed since the last known state.

In a first variant, referred to as the event-driven method (EDM), the time stepping is defined by the physical events [174, Chap. 3]. The system state is updated only at anticipated interactions, computed from the current state. Such approaches, that can be traced back early in the study of molecular systems [7], are of interest only for systems like sparse gases, with rare, independent, non simultaneous interactions, whose time scale is considered negligible with respect to the simulation time (instantaneous interactions). They are thus little suited to dense material modelization.

A second variant of backward scheme use an arbitrary time stepping, independent from physical events [174, Chap. 5]. Specific implicit integration schemes were developed for solid mechanics and applied to granular materials, as the contact dynamics (CD) and the nonsmooth contact dynamics (NSCD) [110, 112]. For each elapsed step, potential interactions are listed and expressed as a set of equations and inequations, numerically solved to find a satisfactory state of the system.

In the spirit of the NSCD, the local configurations of each interaction are considered of secondary order and not known enough for a detailed description [111]. Only key phenomenological aspects are modeled, *via* non-smooth mappings and inequalities, typically the impenetrability of the objects and Coulomb friction. The solver only operates on integrals of the interactions, without computing instantaneous evaluations, and can handle both smooth and non-smooth behaviors.

The intrinsic numerical stability of such schemes allows an interesting latitude in the compromise between accuracy and the cost of the computation. However, numerically efficient implicit solvers can prove more delicate to implement and parallelize than explicit solvers. It must however be noted that, to our knowledge, the largest particle simulations are run using implicit integration [197] and that specific algorithms for general-purpose computing on graphics processing units (GPGPU) have been developed [132].

5.3.2.2 [DEM] Forward Time Scheme

A forward time scheme integration extrapolates future states of the system from the current configuration. In consequence, it is conditionally stable, leading to strong constraints on the chosen time steps. The overall conceptual simplicity of the discrete method using forward integration allows an easy adaptation of the method for distinct purposes.

Forward (or explicit) time integration schemes rely on the computation, at each time step, of the interactions between the elementary objects [174, Chap. 2]. The next state of the system is then computed from the current state – positions, velocities and forces – using Newton's second law. This lean approach is straightforward to implement, including to model arbitrary complex interaction behaviors, as long as an explicit – but

not necessarily smooth²⁰ – mapping defines them. The choice of the time step, and thus the computing time, is strongly constrained by numerical stability requirements, the smoothness of the interaction laws and acceptable computing cost. However, the relative simplicity to distribute the computing load allows the simulation of large systems. This methodology, used in our work, is detailed in Chapter 8.

Historically, the computerized model of elastically connected material points from 1955 [72] lack the essential ability to arbitrarily change neighbors over time. An algorithmically more comprehensive precursor can be found in 1959 [166] (Figure 5.6). Although the implementation is somewhat rough²¹, all the key algorithmic features are already present in the main²² code: interaction forces, arbitrary changes in neighbors, explicit integration. Interestingly, those attempts from the 1950s were using a discrete constitutive law as a numerical work-around to study macroscopic continuum.

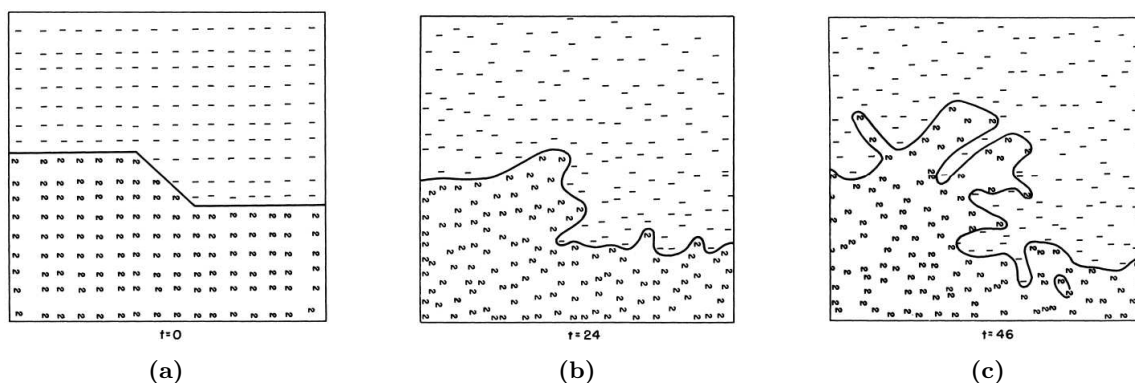


Figure 5.6: Early MD-like code illustration, with 256 particles in 2D. Discrete constitutive law analogically applied to the study of instabilities in bi-phased liquid (denser phase initially on the top) under the influence of gravity. Typical temporal evolution from initial state. Illustrations from [166, p.7–9].

Nowadays, mainstream usage of such methods focus on the simulation of phenomena that are conceptually described by a discrete topology. Slight variations of the formalism used to compute the interactions between neighboring objects stem from distinct applications. In the MD formalism [183], typically applied to study atomic interactions, interactions are defined through potentials around points, within a cutoff distance. In the DEM formalism [49, 50, 149], widely used in granular flow simulation, interaction forces are computed between interpenetrated particles.

These two classical formalisms do not differ from the algorithmic point of view, and rely on identical numerical tools. To sum up, in the MD, typically punctual bodies interact through a potential, often written as functions of the distance, within a cutoff range chosen for computational convenience. In standard DEM applications, bodies are considered to have a “physical” dimension and shape, they interact through force and torque laws, often written as functions of the indentation.

²⁰The conditional stability of forward scheme implies that the smoothness of the laws influences the overall numerical quality.

²¹All parameters were powers of two, in order to replace multiplications by binary shifts, faster operations [166, p.8]. Refer to Figure 3.5 on page 44 for orders of magnitude of the computing power of Maniac, the historical machine used for this work.

²²The authors do not describe in detail their aborted attempt to treat incompressibility more accurately than with binary repulsion [166, p.11]. Although unsuccessful, this second code was probably an ancestor of the SPH [86, p.376].

Implemented codes can practically be used for both, with minor adaptations²³. Built upon a historically distinct mathematical formalism, but essentially relying on the same principles, the bond-based peridynamics (bondPD) [209]²⁴ comes to similar results and limitations. Other variants found in the literature are the so-called movable cellular automata (MCA) [177, 176], using CA algorithms to define the interaction laws between the elementary particles in a MD framework.

In the perspective of modeling continuous phenomena, choosing a discrete constitutive law, from the start, can be seen as an alternative to numerically discretizing and solving continuous equations. In a more modern context than the pioneer attempts cited above, such methodology has been applied to elastic and brittle macroscopic behaviors [60, 59, 124, 209, 202, 109, 10, 98, 114, 217]. The general trend is to build *ad hoc* interaction laws between imaginary particles, used to discretize a continuous medium, in the objective of phenomenologically reproduce a targeted macroscopic behavior. Most examples in the literature focus on the objective of modeling dynamics brittle fracture and fragmentation of elastic solids (Figure 5.7). More unusual applications can be found, for example volumetric plastic strain [113] or quasistatic buckling [123].

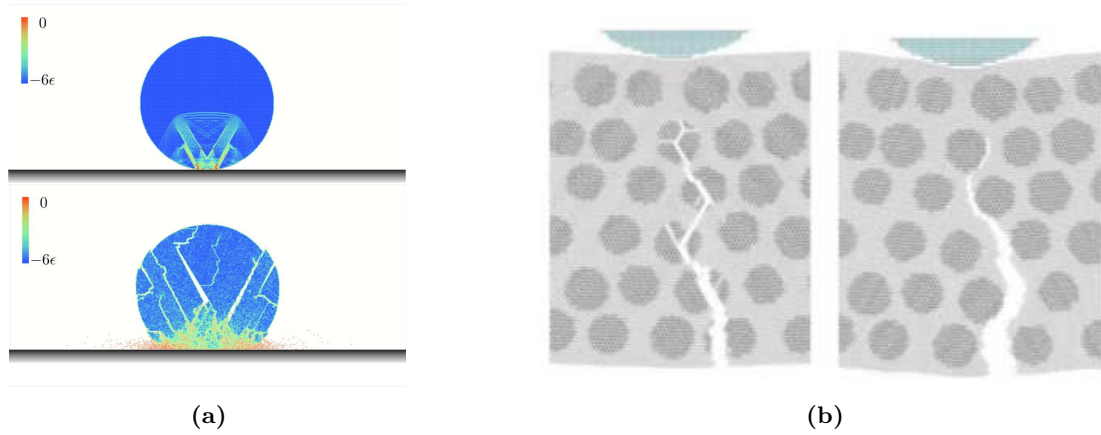


Figure 5.7: Examples of (a) Disk fragmentation under impact [202, p.5]. (b) Crack propagation in composite [217, p.9], with typical path depending on the relative properties of the phases.

²³For example, the DEM solver liggghts is a fork from the MD solver lammmps, mainly adding new features and reusing the main structure and algorithms.

²⁴Refer to footnote 14 on page 69 for disambiguation between bond-based peridynamics (bondPD) and state-based peridynamics (statePD).

Chapter 6

Comparison of Selected Methods

In Chapter 5, a partial lecture grid of Lagrangian numerical method was proposed to locate the DEM in a somewhat comprehensive context.

This chapter proposes elements to illustrate the specificity of our approach. The chapter focuses on a few selected methods to underline similarities and differences with the DEM. The chapter is split into two short sections:

- Section 6.1 proposes a pair-wise comparison of the conceptual and algorithmic features of FEM, EFG, SPH, DEM and lattice model.
- Section 6.2 reviews some potentialities and issues linked to the different methods, with specific emphasis on the DEM.

6.1 FEM, EFG, SPH, DEM and lattice model

This section is an attempt to sum up the key similarities and distinguishable features, from an algorithmic and conceptual point of view, identified in the lecture grid proposed in Chapter 5. The section is limited to widespread methodologies: FEM, EFG, SPH, DEM and lattice model. A graphical synthesis of the section is proposed in Figure 6.1.

Lattice models share their core mathematical tool with the FEM. Both methodologies build-up global grand matrices from local contributions, to globally solve the response of the system *via* linear algebra [70, p.3]. Unlike the FEM, lattice models are not designed to solve PDE representing the behavior of continuous media.

The FEM is a variational formulation¹ of partial differential equation (PDE). The domain is subdivided into elements, formed by explicitly interconnected nodes, on which fields are approximated by polynomial shape functions. The local contributions to the energy are integrated and the problem solved in a global matrix formalism by minimizing the energy.

The EFG shares a common main methodology with the FEM: the variational formulation and global matrix formalism are identical. However, a distinct strategy is used to build the shape functions. Fields are statistically approximated in the neighborhood of the nodes, without a prior table of connectivity.

The SPH shares a mathematical tool with the EFG: the statistical approximation of fields in a neighborhood. However, the SPH does not rely on a variational formulation,

¹ The system is solved by minimizing its energy rather than by direct resolution.

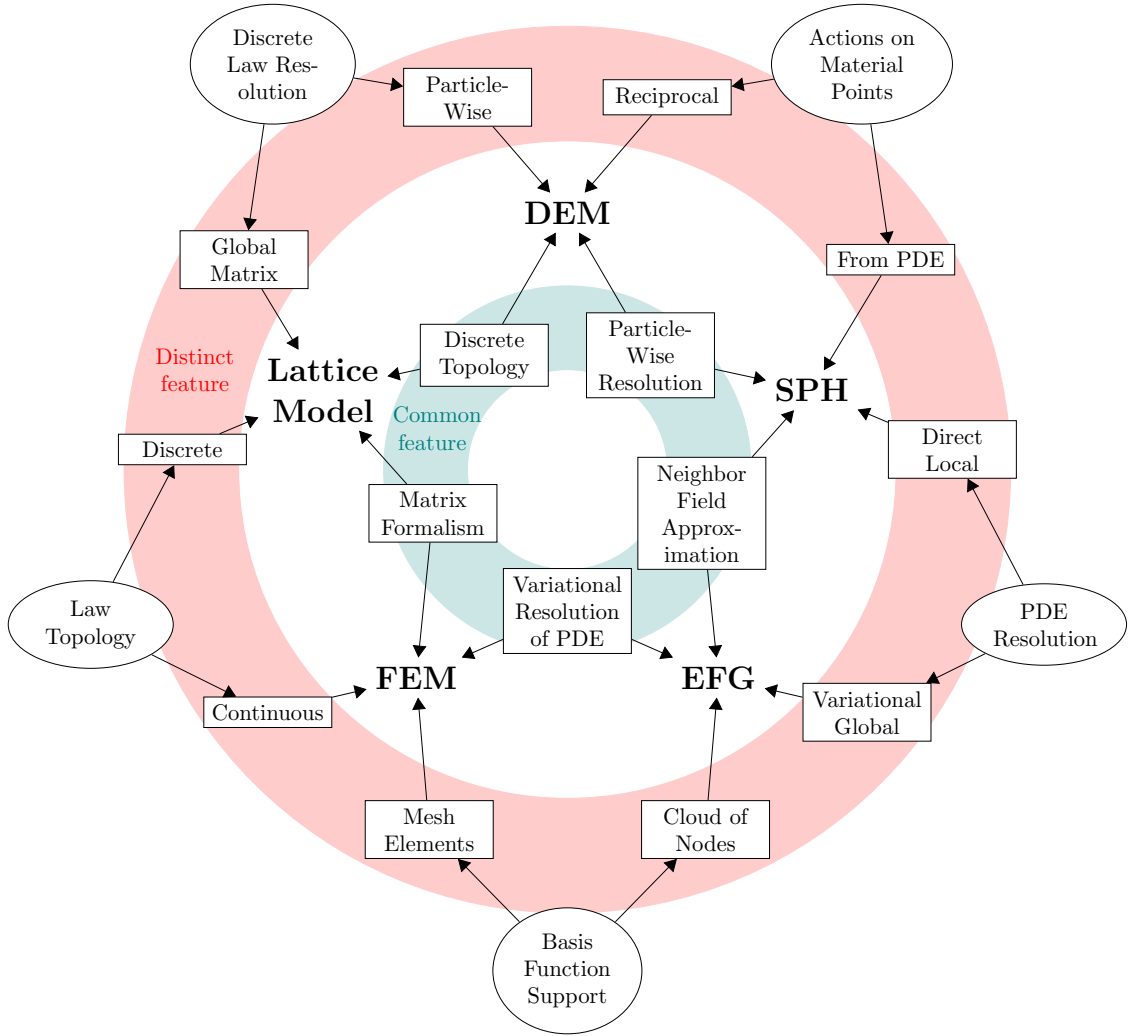


Figure 6.1: Graphical view of the comparison, proposed in Section 6.1, between discrete element method (DEM), smooth particle hydrodynamics (SPH), element-free Galerkin method (EFG), finite element method (FEM) and lattice model. Emphasize on pairwise and circular relationship. For each pair of methods, a key common feature (in the center) and a key distinct feature (in the periphery) are highlighted. Refer to Figure 5.1 for a more comprehensive and systematic lecture grid.

the resolution of the partial differential equation is direct. No global matrix is built, the resolution of the continuous behavior is executed locally and independently for each material point by computing “efforts” exerted by the medium and explicitly integrating position and velocity.

The DEM is algorithmically related to the SPH. They both use a forward (explicit) time scheme to integrate the position and velocity of the material points, based on Newton’s second law and both use neighbor detection algorithms. The salient distinction is the determination of the “efforts” exerted on material points [86, p.376]. In the DEM, the forces on material points stem from interactions with its neighbors. In the SPH, efforts on the material points are computed as the solution of a PDE, with a continuous constitutive law. Conceptually, many-body (or non local) interaction laws implemented in DEM-like frameworks to account for the local volume free for the particle [178] are steps toward the SPH methodology to solve a continuous constitutive law.

Lattice models are conceptually related to the DEM. They share a topologically discrete constitutive behavior: interactions between members of a set of objects. However, they are algorithmically distinct: while the DEM locally computes and integrates the behavior of the objects, independently from one another, lattice models rely on a matrix formalism and a global resolution.

6.2 Ongoing Challenges

As-is, numerous powerful methods could be used for our modelization objective. We remark that the strategy of using the DEM, well suited to model contacts and topological events, is hindered by the lack of models describing isochoric inelastic strains.

Within the Lagrangian family, the accumulated efforts of the past decades opened wide fields to modeling tools. While methods related to the EFG still suffer from comparatively higher numerical cost, the FEM and the SPH now allow the modeling of finite transformation of complex multiphase three dimensional geometries, including topological changes. Successful developments of the FEM are based on the massive use of automated remeshing techniques and tracking and reconstruction algorithms of the interfaces. Even though parallel implementations are developed, these approaches are limited by computing power to configurations where the discrete events do not become predominant with respect to the continuous behavior. The SPH is now a promising challenger, for example with hybrid SPH/DEM models, implemented to handle both discrete and continuous behaviors in a common framework [227, 121]. Advanced developments remain specialist tools, and are not widely available in mainstream commercial or free/*libre* codes.

Strategies using from the beginning a discrete constitutive law, like the DEM, are now widespread and well established. The design of interaction laws and the calibration procedures of the numerical parameters allow the modeling of targeted continuous elastic macroscopic behaviors. The handling of fracture and contact mechanics has very little additional computational cost, including in case of generalized fragmentation, one only needs to define *ad hoc* interaction cases.

However, to our knowledge, two closely related drawbacks hinder the modeling of finite strain in metallic alloys with the DEM. Firstly, the developed models for the continuous medium rely on initially pairwise bonded neighbors, as in a fixed grid. Most implementations can handle the breakage of such bonds, and arbitrary contacts afterwards, but do not accept neighbor changes within a continuous behavior. This is understood here as a total Lagrangian formulation of the constitutive behavior of the continuous medium, as the initial state – the initially bonded neighbors – is taken as reference. Secondly, the introduced plasticity model intrinsically induces volumetric strain. Indeed, plasticity

and viscoplasticity are computed as a pair interaction between particles, allowing them to inelastically interpenetrate [113] [210, p.153]. Summing-up, the discrete constitutive law models lack isochoric plasticity mechanisms and are limited to total Lagrangian descriptions of the continuous behavior.

Part III

Modeling Approach

7	PhD Objective: Research Question and Strategy	83
8	The Discrete Element Method	85
8.1	Neighbor Detection	86
8.2	Interaction Forces	87
8.3	Time Integration	88
8.3.1	Velocity Verlet Algorithm	88
8.3.2	Time Step Choice	89
8.3.3	Parameter Adaptation	90
8.4	Boundary Conditions	92
8.5	Parallel Computing	94
8.6	Control Metrics	95
9	Principle of the Developed Method	99
9.1	Finite Inelastic Transformation	101
9.2	Boundary Conditions	104
9.3	Discretization of Continua	105
9.4	Contact Event Detection	106
9.5	Self-Contact Event Detection	107
10	Chosen Numerical Tools	111
10.1	DEM Tools	112
10.2	FEM Tools	113

In Part I, the global objective of modeling inelastic strain in multi-material was stated. In Part II, various candidate numerical methods to describe interface interactions and topological events were reviewed and compared.

Part III describes the chosen algorithmic strategy and the effective numerical tools used or developed. It is divided into four chapters:

- Chapter 7 states the research question, closely built upon the specific numerical approach, developed in the DEM framework.
- Chapter 8 reviews the basic algorithms and principles of the classical DEM.
- Chapter 9 presents the algorithmic principles of the developed numerical methods. The chapter can be considered as a “cookbook”, presenting as briefly as possible the introduced methods. The discussion of the choices will not be found in this chapter, but in the corresponding applicative sections, in Parts IV and V.
- Chapter 10 sums-up the choices of numerical solvers for DEM and FEM simulation, as well as pre- and post-processing tools.

Highlights - Part III

Modeling Approach

- The PhD focuses on the development of a discrete element method (DEM) algorithm for finite inelastic transformation of incompressible multi-material.

The framework is trusted to natively handle discrete interaction and topological events. Efforts are concentrated on the assessment of the description of the deformation of continuous media.

- The principles of the model is a set of attractive-repulsive spherical particles, discretizing a continuum.

Under external loads, the packing of particles collectively cope with strain. The rearrangements, with arbitrary neighbor changes, account for irreversible strain. *Ad hoc* interaction law can be designed to cope with compressive and tensile load.

- Algorithms to detect physical events like contact and self-contact are proposed.

The self-contact detection, based on an approximation of the free surfaces, is to our knowledge a novel approach.

- The chosen DEM simulation engine is liggghts. Custom features are implemented within this open-source code.

The compromise between modularity and performance leads to the choice of the solver, able to massively parallelize large geometries. The studied problem can thus be scaled up to the study of full real samples.

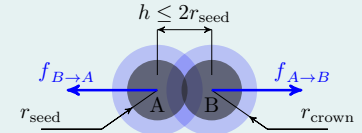


Figure 9.3a

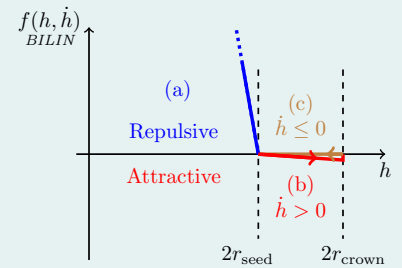


Figure 9.4a

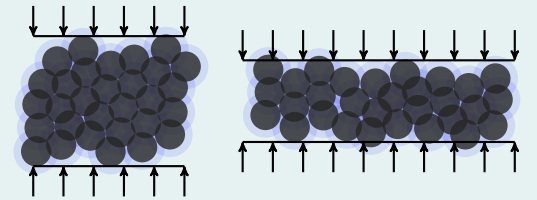


Figure 9.2

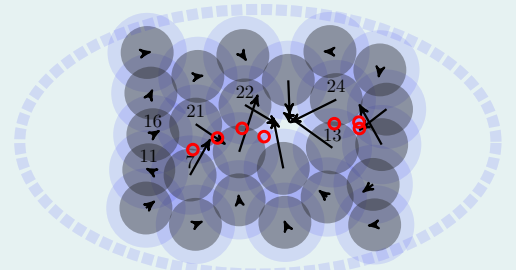


Figure 9.9b

Chapter 7

PhD Objective: Research Question and Strategy

This short chapter proposes a research question in relation with our modelization objective (Section 3.3) and the current limitations (Section 6.2) of the discrete element method (DEM).

The global purpose of this PhD is to propose a numerical tool handling the modelization objective defined in Section 3.3. In short, a model able to describe quasistatic finite inelastic and isochoric strains in solids, handling interface motion, interaction and self-interaction.

Among existing modeling strategies, a topologically discrete constitutive behavior is chosen. It will not be attempted to solve locally a continuous constitutive behavior. The formalism and numerical framework are taken from the DEM. As an innate drawback of the modeling choices, the powerful tools from the continuum mechanics formalisms cannot be exploited. The model will thus be phenomenological. The macroscopic continuous behavior of the material will be approximated *via* the collective interaction of sets of particles, without *a priori* qualitative likeness between elementary interaction laws and macroscopic behavior.

Nevertheless, foreseen applications include strongly tortuous mesostructures and generalized interactions of interfaces. In such circumstances, the actual local continuous behavior is often uncertain, a rough phenomenological approximation can capture well enough its deformation. The predominance of physical contacts also advocates for a discrete numerical method, designed to handle an arbitrary large number of complex contacts.

Similar approaches have successfully been applied to elastic and brittle behavior (refer to Section 5.3.2.2). However, to our knowledge, no inelastic finite strain model has been proposed (Section 6.2).

The research question is thus exploratory: can a phenomenological DEM model be developed for inelastic and incompressible finite strain of continuous media?

The nature of the question implies a twofold proposition:

- An operational algorithmic proof of concept.

It shall be illustrated that the chosen strategy can effectively be applied. Computational issues will not be considered as secondary, as they may drastically restrain the potential of the method.

- A critical scope of validity assessment.

The quality of the description of the captured continuous phenomena shall be clearly delimited, based on numerical or experimental references. The ill-posed nature of the model will not be eluded.

Chapter 8

The Discrete Element Method

The research question (Chapter 7) is closely built upon a specific numerical method, the DEM. This chapter describes the classical framework of the DEM. After a short introductory overview, each section is dedicated to an algorithmic feature.

The DEM is a dynamic Lagrangian method, conceptually closely related to molecular dynamics (MD)¹, aiming to model the collective motion of sets of interacting objects, interpenetrating one another, governed by the conservation of momentum [149] [174, p.13–134]. At each time step (Figure 8.1), overlaps between objects are detected and used to compute interaction forces. Forces, positions and velocities are integrated using an explicit scheme, independently for each object, to define the next state of the system.

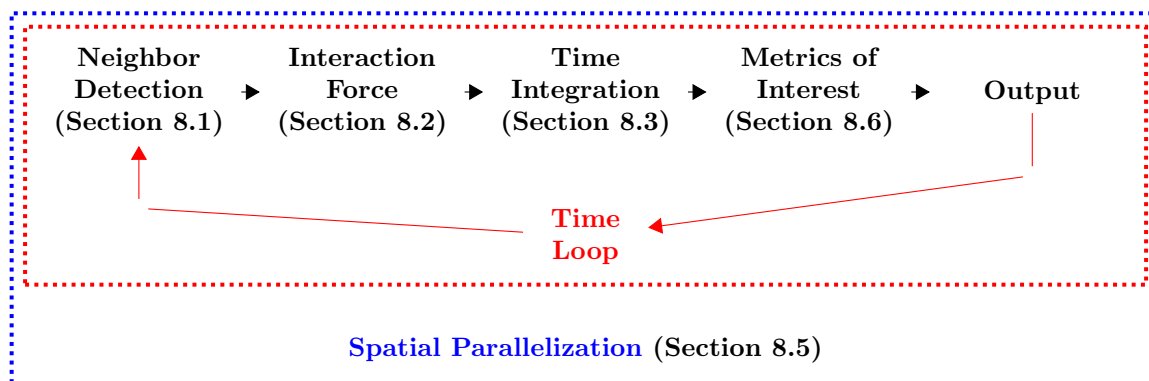


Figure 8.1: Schematic DEM time loop. Cross-references to the corresponding sections. The boundary conditions of the system are examined in Section 8.4. A more comprehensive graph can be found Figure 9.1 on page 100.

From the conceptual point of view, the elementary objects could be of arbitrary shape. This option, aiming to refine the geometrical description of the objects, has been implemented following various routes, among which² polyhedral objects [62], triangulated bodies [213], sums of primitives [64] or superquadric particles [173]. However, neighbor search and overlap computation are algorithmically much heavier than with the exclusive use of spheres as elementary objects. In each modeling context, the compromise between

¹A major distinction between MD and DEM is cultural: the DEM community cites the reference paper of Cundall [50]. See also Section 5.3.2.2.

² For more possibilities, refer for example to [64, p.143].

computing efficiency and accurate description can be debated, It must not be forgotten that the method is *a priori* ill-posed, which sets an asymptotic limit on the accessible local details. In our work, particles have no proper physical sense: they are an arbitrary discretization of a continuous medium. Our description will thus be limited to methods using spheres as elementary objects, favoring the computing efficiency and modeling complex shapes with larger collections of spheres.

The elementary objects could be designed to be able to deform or change size during the simulation. This can be used for example to account for chemical reactions [76] or to describe a finer mechanical behavior of the objects [64]. In the objective of focusing on the collective behavior of numerous elementary objects, the shape and size of the objects will be considered fixed.

Along with an elementary geometry, we will not describe nor use algorithms that explicitly track the angular position of the particles. The algorithms used here can, with little effort, be extended to implicitly take into account rotational effects, but our models will rely on non oriented and non rotating particles, and will thus not describe torques between objects.

We thus limit ourselves to methodologies where a particle is defined by its mass and diameter, constant in time, and its position and velocity at a given instant. Additional state parameters, including history variables, can be Lagrangianly defined and stored at the particle level. In the following sections, we describe the ground features and algorithms of the classical DEM framework, and associated numerical issues. The three generic steps of the DEM time loop, (neighbor detection, computation of the interaction forces and temporal integration) are described respectively in Sections 8.1, 8.2 and 8.3. Ways of constraining the studied system, through boundary conditions, is described in Section 8.4. In our perspective, the parallel implementation of a code, described in Section 8.5, is not a mere computing add-on and must be thought and designed along with the model itself.

8.1 Neighbor Detection

To define which pairs of particles are interacting, it is unreasonably costly to check every potential pair between all particles. The number of potential pairs is reduced by using geometrical criteria.

The systematic check of all possible pairs in a whole system, to detect interacting particles, is costly: $\mathcal{O}(N^2)$ for N particles. Among various algorithms, a simple and efficient method for large systems is to build cell lists [7, p.465]. The simulation domain is divided into cubic subdomains, the cells. All the particles are geometrically sorted and attributed to the cells. For each particle, a list of potential neighbors can periodically be built, using only the particles attributed to the surrounding cells (Figure 8.2). Implementations can execute the effective interaction check at this stage, using the particles sizes and positions, or leave it up to the interactions computation algorithm (see Section 8.2).

This approach can efficiently handle large systems – $\mathcal{O}(N \cdot N_n)$ for N_n particles in a neighborhood of 27 cells – as long as the dispersion of the diameters of the particles is limited. Specific variants are designed for systems with an excessive dispersion, using multiple grids.

The performance of the detection can be tuned and adapted to the context, with sensible choices of the cell sizes – typically 10 % larger than the largest particle diameter – and of the update frequency of the neighbor lists. Systems with slow motions of the particles and rare neighbor changes, quasistatic packing of dense aggregates for example, accept a lower update frequency than rapid flows.

The neighbor detection is, from the functional point of view, related to the meshing procedures used in the finite element method (FEM). It is much cheaper and easier to

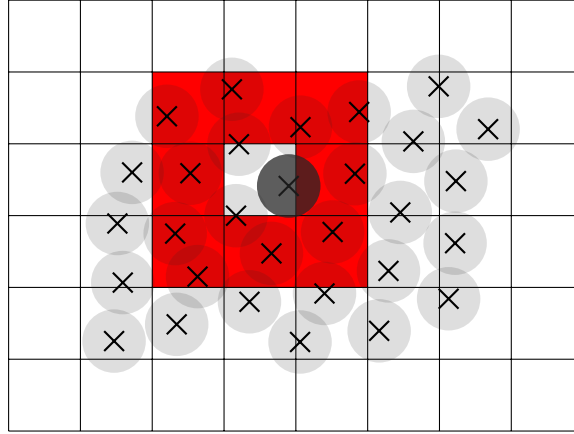


Figure 8.2: Neighbor detection algorithm. Potential neighbors for the dark particle are only searched for in its own cell and in the neighboring cells (shaded in red).

implement for arbitrary configuration, as no elements – with quality requirements – are built.

8.2 Interaction Forces

Algorithmically, a vast flexibility is possible on the design of interaction laws. The implementation of complex behavior is often straightforward. For the model to simulate the collective effect of local phenomena, the interaction laws should be computed from local metrics.

The possibility of implementing arbitrary interaction laws is a major strength of the methodology. They only have to be explicitly computable, based on the current or past state of the system. Depending on the physical principles that inspired them, they thus take a variety of shapes.

In the most classical case, the relative kinematics – relative position and velocity – of the particles are used to compute pairwise interactions, all pair interactions being considered as independent (Figure 8.3). A major strength of the DEM is that little implementation efforts are required to design arbitrary interaction laws, based on the current relative positions and velocities of two interacting particles³.

A useful algorithmic add-on is the introduction of state parameters, for example variables linked to the solicitation history, at two distinct levels:

- The particles, storing a numerically Lagrangian data;
- The pairs, limited in time to the lifespan of the interaction, for example to model a bonding behavior between particles.

More complex behaviors can involve dependencies between the interactions. Computing interactions from neighborhoods, for example for compaction of powders [99, p.48], is referred to as many-body potentials in a MD context, and can be understood as non local

³Classifications are often used in the DEM to subdivide the interaction law in independent contributions, considered to represent distinct physical phenomena: normal and tangential forces; rolling and twisting torques; repulsive and cohesive effects; elastic and damping effects.



Figure 8.3: From kinematics to forces. (a) Position and velocity of a pair of interacting particles (A, B). (b) Reciprocal forces: $\bar{f}_{B \rightarrow A} = -\bar{f}_{A \rightarrow B}$, typically computed from relative position and velocity of the particles.

behaviors in solid mechanics. To some extent, from the algorithmic point of view, such interactions [178] can be considered as a step toward continuous methods as the smooth particle hydrodynamics (SPH), presented in Section 5.2.2.

Regardless of the design of the interaction laws – instantaneous or history aware, pairwise or many-body – all external forces, applied by the neighbors, are summed for each particle. Using Newton’s second law, the acceleration \bar{a} of each particle can be computed from its mass m and the sum of external forces \bar{f} :

$$\bar{f} = m \cdot \bar{a} \quad (8.1)$$

although, in the effective implementations, the acceleration is not explicitly stored (see Section 8.3).

In the design of complex interaction laws, it must be remembered that the collective behavior may be qualitatively distinct from what is intuitively expected from the elementary behavior. In addition, for the model to effectively simulate the collective effect of local phenomena, artifacts involving metrics that are not locally computed must be considered with care.

8.3 Time Integration

The state of the system is explicitly integrated in time. The conditional stability of the schemes imposes a conceptual upper bound on the choice of the time step. Computationally, the theoretical time step is often unreasonably costly to be respected. Numerical parameters are sometimes adapted to allow a faster resolution. The change of the behavior of the system must be considered.

8.3.1 Velocity Verlet Algorithm

A classical explicit integration method is the velocity Verlet scheme. More complex algorithms can be designed to minimize the numerical errors, but will not be considered here.

At each time step, the state is updated particle-wise, from Newton’s second law, using an explicit scheme. The current position, velocity, mass and external forces acting on a particle are used to define the next position and velocity, independently from other particles. Arbitrary integration schemes can be built by discretizing Taylor expansion of Newton’s second law.

An elementary time integrator is the velocity Verlet scheme. At the beginning of a given step i , the position is updated from the previous velocity and acceleration. The

current acceleration is then computed from the sum of the external forces. The current velocity is computed from the previous position and the average of previous and current acceleration.

$$\bar{x}_i = \bar{x}_{i-1} + \bar{v}_{i-1} \cdot \Delta t + \frac{1}{2} \bar{a}_{i-1} \cdot \Delta t^2 \quad (8.2a)$$

$$\bar{a}_i = \frac{\bar{f}_i(\bar{x}_i, \bar{v}_i, \dots)}{m} \quad (8.2b)$$

$$\bar{v}_i = \bar{v}_{i-1} + \frac{1}{2} (\bar{a}_{i-1} + \bar{a}_i) \cdot \Delta t \quad (8.2c)$$

A similar scheme can be applied for the rotational behavior, if needed, using the angular positions and velocities and the moment of inertia of the particles.

In the classical form presented in Equation 8.2, the velocity Verlet algorithm would require the storage of two accelerations: current and previous. A leaner variation may be implemented, avoiding the explicit handling of the accelerations:

$$\bar{v} = \bar{v} + \frac{1}{2} \cdot \frac{\Delta t}{m} \cdot \bar{f} \quad (8.3a)$$

$$\bar{x} = \bar{x} + \Delta t \cdot \bar{v} \quad (8.3b)$$

$$\bar{f} = \text{Interaction law}(\bar{x}, \bar{v}, \dots) \quad (8.3c)$$

$$\bar{v} = \bar{v} + \frac{1}{2} \cdot \frac{\Delta t}{m} \cdot \bar{f} \quad (8.3d)$$

The velocity Verlet's local truncation error – caused at each step by the scheme only, typically excluding numerical round-off errors – is $\mathcal{O}(\Delta t^2)$ for the velocity and $\mathcal{O}(\Delta t^4)$ for the position. As the DEM relies on a large number of steps, the global truncation error – accumulated over time, ideally the effective difference between the retrieved result and the exact solution – is a more relevant metric to compare schemes. Both velocity and position have a $\mathcal{O}(\Delta t^2)$ global truncation error.

More sophisticated algorithms could be chosen to enhance the time integration accuracy, but their computational efficiency is debatable [191, p.878]. In addition, the intrinsic ill-posed and chaotic nature of the motion of an assembly of objects must be remembered. A lower truncation error of the integration scheme will not help to overcome this limitation and may not significantly affect the metrics of interest, measured at larger scale than the individual position of the particles.

8.3.2 Time Step Choice

The conditional stability of the explicit schemes imposes a conceptual upper bound on the choice of the time step. A practical estimation of this critical time step is based on the evaluation of the natural period of an imaginary spring-mass system.

A canonical numerical choice, for explicit integration schemes, is the size of the time step [47]. As an image, the data must be able to travel through the domain faster than the objects. A concrete example in the DEM context: given a sufficient velocity, a particle can jump through another one within a time step. Thus, the detection of the interactions – and the computing of its effects – was slower than the motion of the particles. This configuration – the instability of the explicit integration – must be avoided as it leads to unpredictable behavior, the numerical scheme failing to capture the key modeling features.

A standard DEM criterion, to choose an appropriate time scale, is the spring-mass analogy. This idealized case refers to a system of particles, with a minimal mass m ,

interacting elastically with a maximal stiffness k . The natural oscillation period t_0 of an imaginary spring-mass system (Figure 8.4), of stiffness k and mass m can be computed as follows:

$$t_0 = 2\pi\sqrt{\frac{m}{k}} \quad (8.4)$$

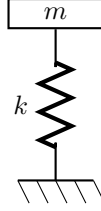


Figure 8.4: Elementary ideal spring-mass system of stiffness k and mass m (see also Equation 8.4).

Although interactions between particles can be simultaneous and unilateral, changing quite radically the vibratory behavior, setting the time step Δt to a fraction – typically a hundredth – of the natural period t_0 ⁴ is often a first reasonable choice.

As a short justification that collective effects will not dramatically increase the typical oscillation time of the particles, one can consider a periodic one-dimensional column of N masses m , connected by springs of stiffness k . The N natural periods t_0^i of such a system are given by the relation [13, p.432]:

$$t_0^i = \pi\sqrt{\frac{m}{k}} \frac{1}{|\sin \frac{\pi n}{N}|} \quad ; \text{ with } n \in \mathbb{Z}^* \quad (8.5)$$

The case $n = 0$ is the rigid body motion of the whole system, allowed by the periodic boundary conditions. The system is not oscillating, the natural period is infinite. The shortest natural period⁵ of the system is thus half of natural period of the one degree of freedom system:

$$t_0^{\min} = \pi\sqrt{\frac{m}{k}} = \frac{t_0}{2} \quad (8.6)$$

Not only the natural periods do not dramatically drop for large systems, but the propagation of external solicitations is limited by high-frequency cutoff. Indeed, waves with lower period than t_0^{\min} are evanescent [155, p.5], no power will be transmitted to the system and no significant motion will be observed away from the application point. This can be seen as a numerical advantage for the stability of the method, but is also a limitation on the observable time scales.

The final choice of the time step needs to be further tuned to match specific requirements, empirical procedures often being necessary for complex interactions.

8.3.3 Parameter Adaptation

Computationally, the critical time step is often unreasonably costly to simulate the state of the system on large time scales. Numerical parameters can be adapted to allow a faster resolution, but this necessarily modifies the behavior of the system.

⁴ Sometimes called the critical time step.

⁵ In the corresponding normal mode, adjacent masses move in opposite directions.

The order of magnitudes of the stiffnesses and masses of physical objects sets another strict limitation for discrete numerical methods as the DEM. To illustrate this, we will consider spherical elastic homogeneous physical objects, interacting with Hertzian forces [102]:

$$f_{\text{Hertz}} = \frac{4}{3} i^{3/2} r^{1/2} E^* \quad (8.7)$$

With i the indentation between two objects, r the radius and $E^* = \frac{E}{1-\nu^2}$, computed from the Young's E and the Poisson's ratio ν of the material.

An average stiffness k for a given indentation can be computed as:

$$k = \frac{f_{\text{Hertz}}}{i} = \frac{\frac{4}{3} i^{3/2} r^{1/2} E^*}{i} = \frac{4}{3} i^{1/2} r^{1/2} E^* \quad (8.8)$$

Introducing the relative indentation $i_r = i/r$, the stiffness k becomes:

$$k = \frac{4}{3} i_r^{1/2} r E^* \quad (8.9)$$

The mass m of an spherical object of density ρ can be computed:

$$m = \frac{4}{3} \pi r^3 \cdot \rho \quad (8.10)$$

The natural period spring-mass system can be estimated using Equation 8.4:

$$t_0 = 2\pi \left(\frac{m}{k} \right)^{1/2} = \frac{2\pi^{3/2}}{i_r^{1/4}} \left(\frac{\rho}{E^*} \right)^{1/2} \cdot r \quad (8.11)$$

For a given relative indentation i_r , the natural period is thus proportional to the radius r of the object and to the square root of the ρ/E^* ratio of the material. Starting from a map of the existing materials in the space (ρ, E) (see Figure 2.1 on page 20), orders of magnitudes of typical natural period can be computed (Figure 8.5).

The time step required to model realistic masses and stiffnesses are often unreasonably small⁶ to compute phenomena on the relevant time scales, as the system is governed by stiff differential equations. From molecular dynamics to granular flows, physical parameters are sometimes modified numerically to maintain the computing time within realistic bounds⁷.

In the objective of a faster execution of each time step, the spatial characteristics of the system can be altered, typically the number of objects or the distance of interaction. To reduce the number of time steps, two main options can be considered, with similar macroscopic results:

- Acting on the solicitations of the system:
 - Decrease the total simulation time, for example by increasing the velocity at which the domain is deformed. This is a common strategy in MD.
- Acting on the response of the system, allow the use of larger time steps by:

⁶ The computing effort required to simulate a time step depends on the number of interacting neighbors. As very rough order of magnitude, with a processor frequency of 2.3 GHz on an Intel Xeon E5520, the DEM code liggghts very roughly requires 10^{-6} cpu second per particle and per time step, for less than a dozen of effective neighbors per particle.

⁷ Such tuning of physical parameters are used in other numerical contexts, as FEM dynamic simulations of quasistatic processes. The increase of the prescribed velocity and the density of the material are common in deep drawing models [43, p.469].

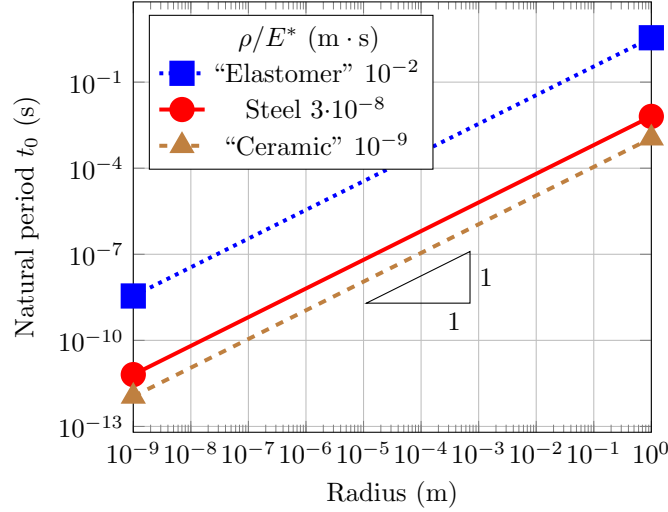


Figure 8.5: Rough orders of magnitude of oscillation time scale of hypothetical spherical objects with Hertzian contacts, for a relative indentation $i_r = 1\%$ (Equation 8.11). The values for “elastomer” and “ceramic” are not representative of real materials. They are made-up using extreme ρ and E values for the class of material [12, p.5], to illustrate theoretical critical cases reachable with dense materials. Materials as aluminum and silica display very similar tendency and they cannot be distinguished from steel at the chosen scale of the graph.

- Decreasing the forces, typically *via* artificially low stiffnesses. Commonly used in granular flows [95, p.535], where the mass flow is a metric of interest.
- Increasing the masses. This approach is specifically relevant to study quasi-static phenomena, as the forces tend to be realistic when the accelerations become reasonably small.

The various strategies are often combined. It must be emphasized that these numerical work-around can influence, not only quantitatively but also qualitatively, the response of the system. When the forces or masses are modified, the similarity between the considered system and the one that is effectively solved is *a priori* lost. In addition, the modification of the length scales (natural period or time step) can altogether prohibit bifurcation phenomena occurring at smaller time scales than the chosen time step [90, p.7]. In this sense, it is misleading to name “normalization” the parameter modification procedures often used in DEM: they modify the nature of the solved problem.

8.4 Boundary Conditions

As for the interaction laws, a lot of flexibility is granted to design of boundary conditions. New objects, in addition to the elementary particles, can for example be introduced.

The most trivial constraint on the system is to let particles fly through the domain boundaries. Particles can either be altogether lost, and be removed from the system, or the boundary positions can be adapted to follow the motion of all particles.

Coming from the MD, where systems are often considered infinite, periodic boundary conditions are a common feature in DEM codes. Particles are made to interact through

the opposed boundaries of the domain. This implies simple modifications of the neighbor detection and the interaction computation. Distances have to be taken into account *modulo* the dimension of the domain. The state of the system can be controlled by modifying the boundary positions, leading the particles to interact and rearrange.

A classical work-around to impose more diverse constraints is to describe boundaries with particles. The "boundary" particles are typically excluded from the standard integration scheme, to control their position. Requiring little developing efforts, the description of even trivial geometries requires a large number of particles, which is potentially costly. The intrinsic "roughness" of the modeled boundary can be a desired feature but is often a handicap.

Arbitrary geometries can also be modeled by the introduction of undeformable objects, either geometrical primitives – as infinite planes – or meshed surfaces. An additional implementation effort is required to include these objects in the neighbor detection and interaction computation algorithms. In addition to static interacting objects, various levels of complexity can be implemented:

- Geometries can follow prescribed motion;
- They can be used as "sensors", to measure the forces acting on them;
- Their motion can be controlled *via* the particle actions.

Going further, though this strategy was not used in our work, coupling strategies of the DEM with solvers designed for other physical models can be used. In some cases and up to some extent, such couplings can be seen as complex boundary conditions. Classical strategies rely on the communication of data between distinct codes, with a distinct time stepping, adapted to the physical models. The most widespread and fast developing area is the coupling with fluid dynamic solver based on the finite volume method (FVM) [89]. The coupling with solid dynamic models was also successfully implemented, whether for undeformable bodies with solid dynamic solver [103] or deformable bodies with FEM solvers [151, 61]. Endless combinations can be imagined, for example solid/fluid/particle models [34].

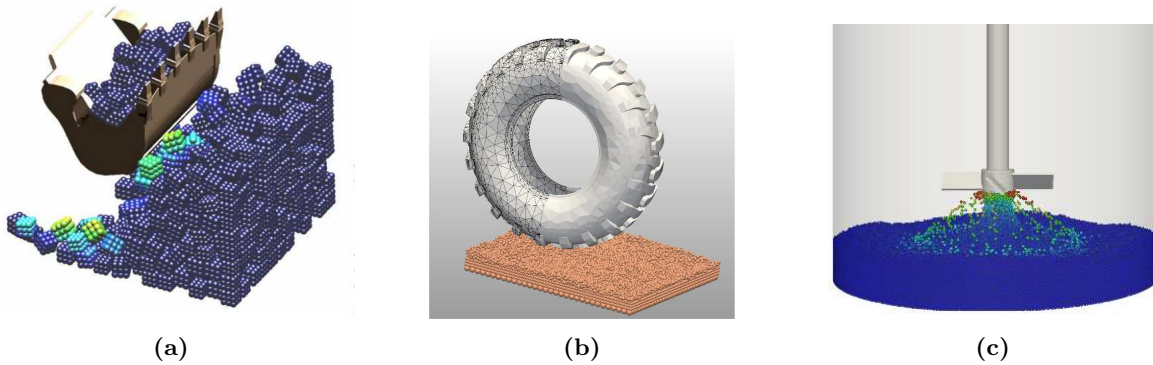


Figure 8.6: Examples of DEM coupling. (a) Solid dynamics: excavation of rocks [94, p.3]. (b) FEM: interaction between a tire and a snow layer [151, p.167]. (c) FVM: effect of the motion of an impeller in a fluid on a bed of particles [34, p.35].

8.5 Parallel Computing

Parallel computing is only a technical work-around to distribute the computing load on several processing units, operating simultaneously, in the objective of minimizing the total computation time. Although the modeled phenomena should theoretically remain unchanged, the technical limitations are decisive in the algorithmic and modeling strategies.

The most straightforward way to implement a DEM code – and the most understandable for both developers and end users – is to follow as closely as possible the governing physical model, sequentially programming all operations, looping over time step, particles and neighbors. Algorithmic optimization and performance issues are thus considered as secondary. As mentioned in Section 3.1.3.2, progresses of computing hardware shifted from frequency scaling – over time a code would tend to run faster without modifications – to parallel scaling. To take advantage of improvements, portions of the algorithms have to be modified to run simultaneously on various processors.

The nature of the algorithm sets an asymptotic limit, Amdahl’s Law [101, p.41], to potential gains using parallel programming. The limiting, purely sequential part of the algorithm is known as the critical path and cannot be parallelized. In the DEM, the explicit time integration of the state of a particle, is such a critical path: the knowledge of the current state at each step is necessary to compute the following.

Counter intuitively, somewhat exotic parallel-in-time algorithms can be designed for such initial value problems [133] and have been applied to particle simulations [218]. These methods are iterative and based on the resolution on various temporal grids. Although all steps need to be computed, and are in fact all computed various times, such algorithms provide a work-around to distribute the computing load. This strategy is still confidential and will not be further considered here, we will only look into the spatial division of a system into mostly independent subdomains. If the boundaries between these subdomains are appropriately dealt with, the simultaneous computation of the state of the subdomains allows to deal with a larger number of particles within a given time.

These purely technical limitations have a major impact on the modeling choices. From one side, the algorithm critical path – strictly constrained by the time integration – cannot be made to run notably faster. From the other side, the size of the system can theoretically be arbitrarily large, only being limited by machine capacity. In the limits of what the physical model accepts, efficient codes can be written by designing rough interaction law and time integration, and expanding the system number of objects. Anyhow, the mechanicians must adapt their modeling strategy, only being able to choose at which level they accept to deal with computing issues, software and hardware.

A first approach to parallel computing is open multi-processing (OpenMP). This interface allows to distribute computing load between various elementary processors, using a shared memory for all of them. Starting from a purely sequential implementation, OpenMP can allow a progressive parallelization of the code, as an add-on, without requiring in depth rewriting work. The main sequential algorithm is run as a unique master process. For critically time consuming operations, for example the loops on the particles in a DEM code, the master process calls subprocesses to split the computing load. Once all subtasks are over, the master process gathers all data and carries on. The scalability of OpenMP codes is strictly limited by the hardware memory architecture capacities, thus making impossible massive parallelization on classical machines. However, more sophisticated hybrid parallelization techniques [27] and hardware can take profit of the OpenMP interface.

To take full advantage of parallel and massively parallel computing, implementations must be designed from the root to fit the chosen parallel paradigm requirements⁸. The

task used to be quite delicate when parallel programming started to spread, implementations being hardware dependent, and hardware changes being frequent, diverse and often experimental. Heavy programming effort used to be required, with little guarantee on the lifespan of the code. With time, *de facto* standards progressively emerged. For applications similar to the DEM, potentially massively parallel computing is mostly shared between the message passing interface (MPI) and general-purpose computing on graphics processing units (GPGPU).

The MPI relies on a distributed main memory and runs on multiple central processing units (CPUs), the most common generic elementary computing devices. Each CPU works on and accesses to its own storage space, and exchanges data with others when needed. The same code is executed on all CPUs, in the case of the DEM each CPU affecting itself to a specific geometrical subdomain. The CPUs exchange data during the execution of the code, for example to properly handle particles lying close to the subdomain boundaries. It is often necessary to entirely rewrite a specific MPI implementation of an algorithm. A key advantage of this parallelization strategy is that distinct operations can be performed simultaneously on distinct sets of data. As a very practical example applied to the DEM, complex interaction laws with numerous conditional statements. The widespread MPI standard is adaptable to diverse hardware architectures, and is used on classical machines and clusters.

The GPGPU attempts to take advantage of elementary graphics processing unit (GPU), originally designed for image processing purpose. As a rough description, with respect to CPUs, the GPUs can only handle less complex operations, work at lower frequency, but can be massively assembled. The memory is shared between the GPU, allowing a fast access to data. The GPGPU programming consists in allowing bidirectional communication between CPU and GPU, in contrast with the classical unidirectional flow, from the CPU, to the GPU and finally to the display. This parallel approach is somewhat akin to the OpenMP interface, with a master process running on CPU calling subroutines on GPUs, the kernels. Being distinct physical devices managing their own memory, the CPUs must send the data along with the instructions to the GPUs. In comparison to a pure CPU parallelization strategy, GPGPU cannot handle distinct tasks simultaneously: the benefits are restricted to identical operations on large sets of data. For scientific computation as the DEM, specific hardware is required and devices are far less common than classical machines.

The domain of parallel computing still undergoes heavy developments, both from hardware and software standpoints, and may well take new radical shifts. Cutting edge techniques require heavy development effort and the resulting code may only have a short lifespan.

8.6 Control Metrics

The choice of relevant metrics is often a sensitive issue in modeling. Spatial and temporal averages are almost always necessary in the cases of the DEM. Adimensional metrics can be defined and may help to identify the system key characteristics.

The ill-posed nature of the DEM (Section 3.1.1) implies that specific care must be taken to define relevant metrics. In general, all local metrics defined at the level of the particles can only be interpreted after suitable time and space averaging.

For computational mechanics, fundamental metrics of interest are the stress field $\bar{\sigma}$ and the strain field $\bar{\epsilon}$. Macroscopic uniaxial stresses can be evaluated by the measure of

⁸“Adapting old programs to fit new machines usually means adapting new machines to behave like old ones.” [168, §120]

the total force acting on a boundary (Section 8.4) and the cross-section of the packing. Likewise, macroscopic strain can be estimated by comparing, in a given direction a , the evolution of the typical length A : $\varepsilon_{aa} = \log \left(\frac{A}{A(\text{ref})} \right)$.

The local fields, at the level of the elementary particles can be approximated and reconstructed in local neighborhoods. The components of the local stress tensor are classically computed as follows [44, p.162] [140, Chap. 2]:

$$\sigma_{ab} = \frac{1}{V} \sum_{i=0 \dots n} f_i^a \cdot l_i^b \quad (8.12)$$

With V the typical volume occupied by the particle, n the number of contacts, f_i^a the components of the force vector due to the interaction with particle i , l_i^a the components of the branch vector⁹ and a and b the coordinate system indices.

The algorithm choice for the strain field estimation heavily depends on the expected strain type, many metrics being targeted to describe the perturbation of a regular lattice. For inelastic strain, with respect to a reference configuration, the components ε_{ab} of the local strain field $\bar{\varepsilon}$ can be approximated in a neighborhood as follow [68, p.7197]:

$$\begin{aligned} \varepsilon_{ab} &= \sum_k X_{ak} Y_{bk}^{-1} - \delta_{ab} \\ \text{with } X_{ab} &= \sum_{i=0 \dots n} l_i^a \times l_i^b(\text{ref}) \\ \text{and } Y_{ab} &= \sum_{i=0 \dots n} l_i^a(\text{ref}) \times l_i^b(\text{ref}) \end{aligned} \quad (8.13)$$

More generally, to try to characterize the overall behavior of a discrete system, numerous adimensional metrics have been defined and used. Among them:

- A relative density of particles \mathcal{D} :

$$\mathcal{D} = \frac{N \cdot \pi d^3}{6 \cdot V_{\text{tot}}} \quad (8.14)$$

With V_{tot} the total volume of the system and N the number of particles. This metric is typical of the DEM, where the dimension of the numerical particle is supposed to be linked to a physical dimension of the elementary objects.

- A stiffness level \mathcal{K} [194, p.215]:

$$\mathcal{K} = \frac{k}{P \cdot d} \quad (8.15)$$

With k the elastic stiffness of the contacts, d the diameter of the particles and P the confinement stress.

The confinement stress can be approximated with a typical force acting on a particle and its diameter. Considering linear elastic contacts we can write:

$$P \propto \frac{d^2}{k \cdot i} \quad (8.16)$$

With i the indentation between two particles. The stiffness level is thus interpreted as a simple geometrical parameter, accounting for the relative penetration of the particles:

$$\mathcal{K} \propto \frac{d}{i} \quad (8.17)$$

This dimensionless quantity is thus closely related to the density \mathcal{D} .

⁹The branch vector \bar{l}_i is the geometrical vector from the center of the particle to the center of its neighbor i .

- An inertial number \mathcal{I} [194, p.207]:

$$\mathcal{I} = \dot{\epsilon} \sqrt{\frac{m}{Pd}} \quad (8.18)$$

With $\dot{\epsilon}$ the prescribed strain rate and m the mass of the particles.

Using the same approximation as previously for the confinement stress P , and remembering the definition of the natural period (Equation 8.4), the order of magnitude of the inertial number can be written:

$$\mathcal{I} \propto \dot{\epsilon} \cdot t_0 \sqrt{\frac{d}{i}} = \dot{\epsilon} \cdot t_0 \cdot \sqrt{\mathcal{K}} \quad (8.19)$$

The salient criterion is the comparison of the characteristic times of the external load and the packing, respectively the inverse of the strain rate and the natural period.

- A global equilibrium criteria \mathcal{Q} [44, p.155], intended for quasistatic configurations:

$$\mathcal{Q} = \frac{E_k}{P \cdot d^3} \quad (8.20)$$

With $E_k = \frac{1}{2}m \cdot v^2$ the average kinetic energy. In this adimensional parameter, the $P \cdot d^3$ factor represents the product of a typical force and a typical length, respectively $P \cdot d^2$ and d . To allow a more direct mechanical interpretation, the criteria can be re-written as the ratio of the kinetic and elastic potential energies:

$$\mathcal{Q} \propto \frac{E_k}{E_p} \quad (8.21)$$

With $E_p = \frac{1}{2}k \cdot i^2$. The criteria could be re-written to highlight the role of the natural period t_0 (Equation 8.4):

$$\mathcal{Q} = \frac{\frac{1}{2}m \cdot v^2}{\frac{1}{2}k \cdot i^2} = \left(\frac{t_0}{2\pi} \cdot \frac{v}{i} \right)^2 \quad (8.22)$$

This layout has less direct physical sense but gives a hint about the role of t_0^2 in the comparison of distinct configurations.

More generally, the use of such a criterion implies that the absolute value of the velocity can be interpreted, which may be doubtful in a system where only the relative velocities between interacting particles are of interest. Correction procedures to take into account the average velocity have been designed.

To sum up, numerous generic or specialized metrics have been designed to study granular flows, at a macroscopic or local scale, this short proposed list being by no means exhaustive. The characterization of granular flow is still to some extent an open question [152, p.15]. More than a specific metric, the general methodology of identifying driving time and length scales proves efficient, including for systems alien from traditional models.

Chapter 9

Principle of the Developed Method

Chapter 7 introduced the grounding choice of using the DEM framework to meet our modelization objective. Chapter 8 introduced the classical algorithms of the method. The present chapter is dedicated to the synthetic description of the principle of the developed method. The introduction of the chapter sums up the effective algorithmic “ingredients” used for the different phenomena, with an indication of the specific contribution of this PhD. Each section is then dedicated to a specific physical phenomenon. The discussion, justification and test of the algorithmic choices are to be found in Parts IV and V.

Although it is somewhat artificial to strictly distinguish between conceptual model, algorithm and implementation, this section focuses on the principles of the method. Implementation issues with the chosen tools are briefly dealt with in Appendix B.1.

A brief overview of the algorithmic “ingredients” used in the developed models can be found in Table 9.1. These introduced features are contextualized in a generic DEM framework in Figure 9.1. Our method relies on custom interaction laws, particle and pair state variables and non-local behaviors.

Physical model	Algorithmic feature and choice	Principle	Application
Inelastic strain	Purely geometric and instantaneous interaction law	9.1	IV, V
Prescribed strain	Mobile rigid mesh/particle interaction law	9.2	IV, V
Material discretization	Set particle state variable from 3D image	9.3	IV, V
Contact (distinct objects)	Particle state variable	9.4	/
Self-contact	Non-local interaction law, pair and particle state variable	9.5	V

Table 9.1: Algorithmic ingredients used in the developed models. Cross-reference to the sections describing the choices (Principle) and to the applicative parts (Application).

Algorithmically, the designed interaction laws (particle/particle in Section 9.1 and particle/mesh in Section 9.2) introduce little novel features. Only the conditional attractive force may be an introduced feature. However, the adaptation and tuning of our laws to phenomenologically model large inelastic strains is to our knowledge not found in the literature.

The detection of contact between distinct objects (Section 9.4) is taking advantage of the intrinsic properties of the DEM and has already been investigated. This aspect has

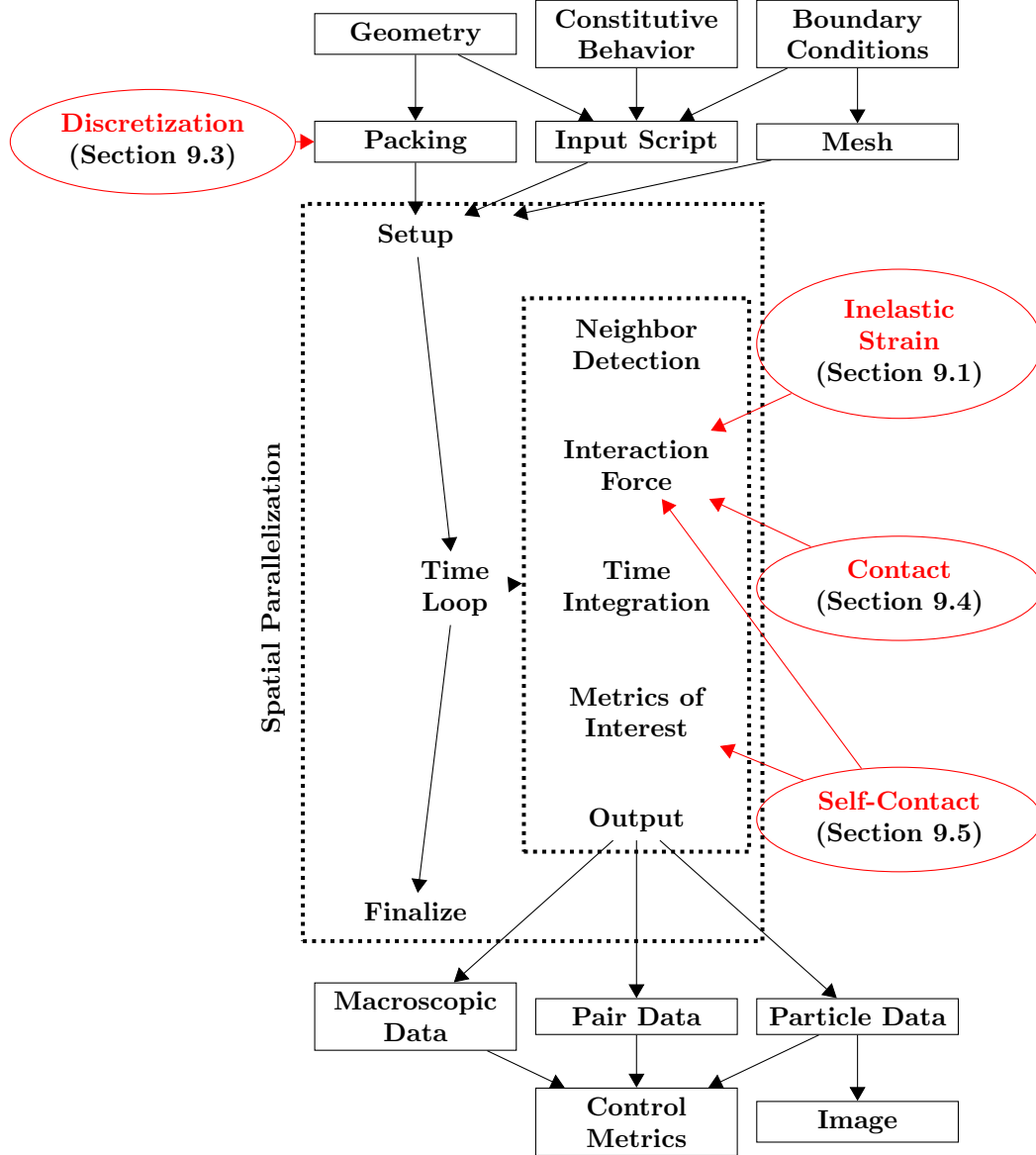


Figure 9.1: General DEM framework of the introduced features (circled), with cross-reference to the corresponding sections. The chosen numerical tools are presented in Section 10.1, see also Table 10.1.

not been a major focus of the PhD and was added for the sake of completeness.

The material discretization (Section 9.3) is now rather standard in the context of our laboratory [193]. Personal contributions concern the handling of periodic packings and the definition of arbitrary state variables.

To our knowledge, the self-contact detection algorithm (Section 9.5) is a novel contribution to the field.

9.1 Finite Inelastic Transformation

The basic assumption of our modeling approach is the analogy between the motion of a collection of spheres, with attractive/repulsive behavior, and incompressible inelastic strain in a continuum. The ad hoc interaction laws are chosen and tuned for the collective rearrangements of the particles to mimic key features of inelastic incompressible transformation of a continuous medium. The choice and behavior of the two interaction laws introduced here are examined respectively in Chapters 11 and 14. A more in depth investigation of the calibration of the numerical parameters can be found in Section 11.2.

Conceptually, a continuous object is discretized using a dense collection of particles, interacting with pairwise attractive/repulsive reciprocal forces. Repulsive forces prevent excessive indentation and attractive forces provide some cohesiveness. This behavior aims at controlling the overall volume variations.

When an external load is applied to this collection, the particles move with respect to one another, the packing rearranges. The particles are freely allowed to change neighbors, thus modeling arbitrary deformation of the object.

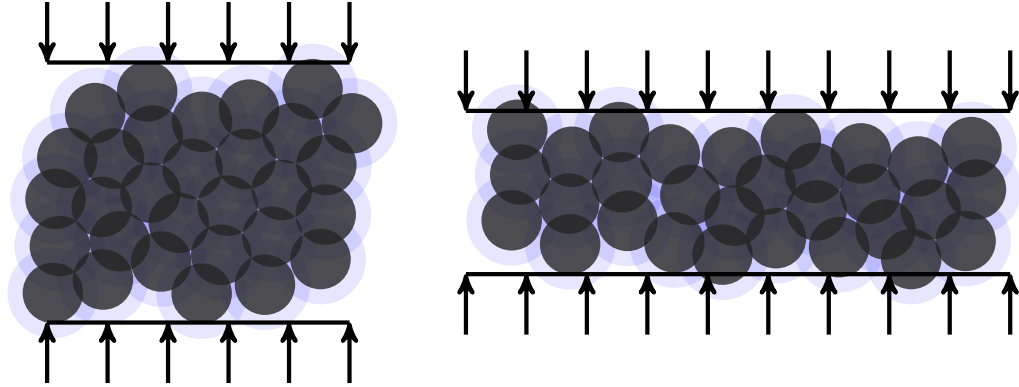


Figure 9.2: Principle of finite inelastic transformation modeling. The particles collectively rearrange to cope with the strain. Particles may arbitrarily change neighbors and the overall volume is meant to be conserved.

An early modeling choice was to consider particles subdivided into two concentric and spherical interaction zones with distinct behaviors (Figure 9.3):

- A repulsive seed, mimicking incompressibility, of radius r_{seed} ;
- An attractive crown, adding cohesiveness, of radius $r_{\text{crown}} > r_{\text{seed}}$.

Reciprocal interaction forces $\bar{f}_{A \rightarrow B} = -\bar{f}_{B \rightarrow A}$ are computed for each pair of overlapping particles (A, B), based on the distance between their center h and relative normal velocity \dot{h} .

This geometric strategy¹ is closely akin to MD methodology, where punctual bodies interact through potentials, with an equilibrium state. The algorithm allows an interaction management without the use of history parameters and lists of neighbors, that would need to be stored between time steps.

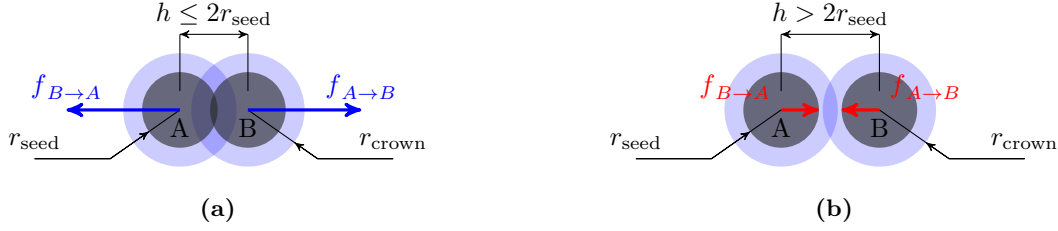


Figure 9.3: Geometry of the interaction of a pair of particles (A,B) and normal forces \bar{f} . (a) Seed interaction, repulsive normal forces. (b) Crown interaction, attractive normal forces.

In this PhD, interaction laws are governed by normal forces \bar{f} , piecewise linear with the pair distance h and linear with the introduced parameters (Figure 9.4). The signed norm f of the force will be used, following a classical DEM convention: repulsive forces are positive and attractive forces are negative. Two distinct laws, but similar in many aspects, are used:

- “*BILIN*”
This interaction law is suitable only for compressive loads (Figure 9.4a, Equation 9.1, Part IV). The repulsive forces are strongly dominant and a calibration procedure of the strain rate sensitivity is proposed.
- “*TRILIN*”
This interaction law can mimic tension and compression (Figure 9.4b, Equation 9.2, Part V). Repulsive and attractive forces are more balanced, no control over the strain rate sensitivity if proposed.

The interaction law *BILIN* relies on two stiffness parameters k_{rep} and k_{att} , accounting respectively for repulsion and attraction:

$$f_{\text{BILIN}}(h, \dot{h}) = \begin{cases} k_{\text{rep}}(2r_{\text{seed}} - h) & \text{if } h \leq 2r_{\text{seed}} \\ k_{\text{att}}(2r_{\text{seed}} - h) & \text{if } 2r_{\text{seed}} < h \leq 2r_{\text{crown}} \text{ and } \dot{h} > 0 \\ 0 & \text{and } \dot{h} \leq 0 \end{cases} \quad (9.1a)$$

$$(9.1b)$$

$$(9.1c)$$

The attractive force is dependent to the relative normal velocity² \dot{h} : the attractive force is only activated if a pair has a tensile motion, and is canceled in case of compressive motion. This behavior helps to smooth the creation of new contacts between particles and introduces a dissipative effect of the total energy, numerically sufficient within the

¹In DEM, the algorithms for cohesive behavior are often based on explicit lists of pairs that have cohesive interactions even if they are not overlapping. Although identical numerical results could be obtained using such algorithms, they conceptually define preferential neighbors, which will be avoided in our work.

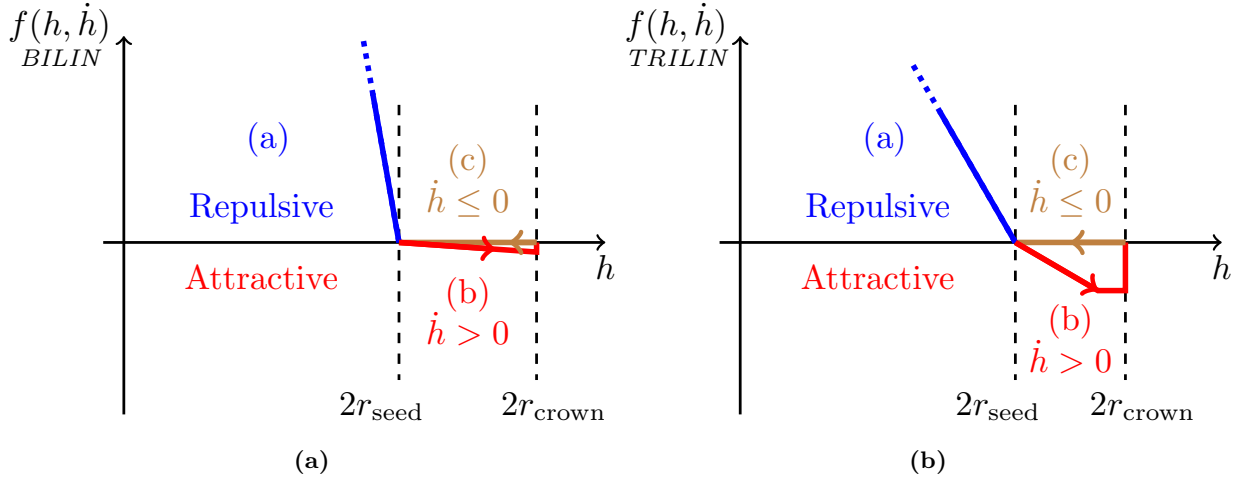


Figure 9.4: Interaction laws: signed force f versus distance h . Classical DEM conventions are applied: repulsive forces are positive. For each graph, attractive to repulsive force ratio and radii to scale. Refer to Table 9.2 for the constant used. (a) *BILIN*: compression dominated loads, tunable strain rate sensitivity (Part IV, Equation 9.1). (b) *TRILIN*: tension-compression loads, fixed strain rate sensitivity (Part V, Equation 9.2).

strain rate validity range of the model, linked to the frequency of oscillation of the pairs.

At the pair level, no damping, shear or torque interaction laws are implemented. In the tested configurations, such interactions only introduce second-order effects on the macroscopic behavior of packings.

The behavior of the interaction law *TRILIN* is very similar to *BILIN*, with the introduction of an additional parameter f_{att} , a bound on the magnitude of the attractive force:

$$f(h, \dot{h}) = \begin{cases} k_{\text{rep}}(2r_{\text{seed}} - h) & \text{if } h \leq 2r_{\text{seed}} & (9.2a) \\ \max(k_{\text{att}}(2r_{\text{seed}} - h), f_{\text{att}}) & \text{if } 2r_{\text{seed}} < h \leq 2r_{\text{crown}} \text{ and } \dot{h} > 0 & (9.2b) \\ 0 & \text{and } \dot{h} \leq 0 & (9.2c) \end{cases}$$

A second additional parameter X_{wall} , will be described in Section 8.4 as it concerns only particle/mesh interactions.

Most numerical parameters are set to a fixed value (Table 9.2) for all simulation and considered as constants of the models *BILIN* and *TRILIN*:

- All simulations are run using identical radii, for all particles. The dimensions are totally arbitrary and do not represent a physical metric, they are chosen for numerical convenience.
- The relative magnitude between attractive and repulsive forces is fixed.

It will be discussed (Section 11.2) that the linearity of the interaction laws with the introduced parameter allows an independent tuning of the kinematics of the packing and

²Computed for a pair $\{i, j\}$ as the projection of the difference of the velocities $\bar{v}_i - \bar{v}_j$ on the unit vector \bar{e}_n pointing from a particle to the other.

Model	Driving variable	Numerical constant							
		r_{seed} mm	r_{crown} mm	t_0 s	Δt s	m g	k_{att} $\mu\text{N} \cdot \text{mm}^{-1}$	f_{att} μN	X_{wall} /
<i>BILIN</i>	t_0, k_{rep}	0.5	$1.5 \cdot r_{\text{seed}}$		$5 \cdot 10^{-4}$	$k_{\text{rep}} \left(\frac{t_0}{2\pi} \right)^2$	$\frac{k_{\text{rep}}}{10}$	/	/
<i>TRILIN</i>	k_{rep}	0.5	$1.4 \cdot r_{\text{seed}}$	1	10^{-1}	$k_{\text{rep}} \left(\frac{t_0}{2\pi} \right)^2$	$\frac{k_{\text{rep}}}{1.8}$	$\frac{3}{4} k_{\text{att}} (r_{\text{crown}} - r_{\text{seed}})$	3

Table 9.2: Constants defining the models. The force levels are driven by k_{rep} for both models. For *BILIN*, the kinematics are driven by $t_0 \in [10^{-2}, 1]$ s. For *TRILIN*, two additional constants are introduced: f_{att} and X_{wall} . Arbitrary unit system: (mm, g, s), forces thus expressed in μN and stresses in Pa.

the forces acting upon it. The effective driving parameter of the kinematics is the natural period $t_0 = 2\pi\sqrt{k_{\text{rep}}/m}$. The value of t_0 must be chosen with respect to the targeted arbitrary windows, of limited sizes, of macroscopic strain rate. This behavior was only investigated for *BILIN*, t_0 is considered as a constant in *TRILIN*. At a given natural period, the time step Δt is fixed (Section 11.3.2).

The driving parameter of the forces level is k_{rep} , allowing arbitrary stress levels to be modeled. The mass m of the particles is thus computed from k_{rep} and t_0 .

9.2 Boundary Conditions

The boundary conditions are rather classical in DEM simulations: free surfaces and mesh-constrained surfaces driven by prescribed displacement. These boundary conditions are used throughout Parts IV and V.

Two types of boundary conditions are applied to the packings:

- Free boundary, where particles are not constrained by any means;
- Kinematically constrained boundary, using rigid meshes interacting with the particles.

Interaction forces between the mesh elements and the particles are computed with a very similar contact law as particle/particle interactions. In the interaction laws (Section 9.1) h is defined as the shortest distance between an element and a particle (Figure 9.5a), and r are to be used instead of $2r$.

For uniaxial loads, the following boundary conditions are applied (Figure 9.5b): top and bottom planar meshes and free lateral sides. Meshes are used to apply prescribed macroscopic true strain rate. The total forces acting on the meshes are measured to evaluate the macroscopic flow stress. To smooth the force signal, the total force is averaged over a temporal sliding window. The width of the sliding window is defined in strain, in the range $5 \cdot 10^{-3} - 10^{-2}$ for all applications.

Only planar meshes were used, although any arbitrary surfaces meshed with triangular elements are accepted.

A potential conceptual limitation is that each particle has only one interaction with a mesh element³ and multiple interactions with other particles. Under compressive loads, the model behaves correctly without further modifications: the particle/mesh interaction tends to indent more to balance with the numerous particle/particle interactions. Under tensile loads, two modifications are applied to the particle/mesh law:

³Each particle interacts only once with a planar mesh.

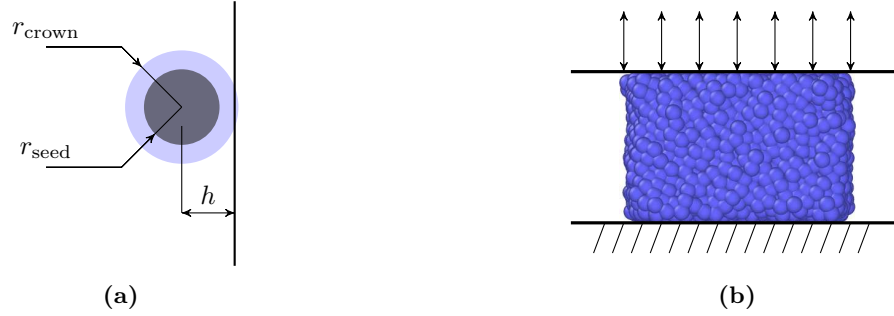


Figure 9.5: Boundary conditions. (a) Geometry of the mesh/particle interaction. Crown interaction example. (b) Typical boundary conditions for uniaxial loads. Top and bottom planar meshes with prescribed motion, free lateral sides.

- The normal relative velocity dependency is removed: crown interactions are always attractive.
- A multiplicative factor X_{wall} is applied to the attractive forces.

These numerical recipes are *ad hoc* means to apply a macroscopic load. Their influence is limited “far”, i.e. a few diameters away, from the application points of the load⁴.

9.3 Discretization of Continua

The generic spatial discretization of continua is based on large random packings of particles. The procedure is used throughout Parts IV and V. The modeling of complex mesostructures, starting from 3D images from X-ray tomography, is more specifically applied in Chapters 13 and 16.

The basic components used to discretize continuous media are cuboids of randomly packed particles. The packings are built using classical interaction laws, with elastic repulsion only, in periodic domains. The procedure is classical in the DEM [236]:

1. Random granular gas generation;
2. Triaxial⁵ compaction;
3. Relaxation.

The compaction is performed at prescribed strain rate, up to a fixed density \mathcal{D} (Equation 8.14), approximately corresponding to the equilibrium density of the foreseen interaction law⁶. The initial state of the random packing and the elaboration route, in the context of the large strains studied here, seemed to have little influence on the compression results, and are not detailed here.

Distinct geometric configurations are built starting from random packings:

1. The desired configurations are built using geometrical criteria (setting the material type of the particles, removing particles to create voids...);

⁴Although it would be bold to pretend to apply Saint-Venant’s principle [237].

⁵Isotropic in the case of cubic domains.

⁶The effective density for a given interaction law varies by a few percent with the strain rate, see for example Figure 11.8 on page 128.

2. The attractive-repulsive model and the boundary conditions are applied;
3. A short relaxation is run (typically 500 steps).

The actual tests are run after this last relaxation procedure.

Complex material mesostructures are modeled using 3D segmented images (Figure 9.6). The image is reconstructed using a simple box filter [216, p.5] which is used as a mask on the random packing of particles. For each particle, the color at its center is used to set its type or remove it. This has a low algorithmic cost and for very large sets of data, a smaller periodic packing can be replicated in all direction, minimizing the cost of generation of this initial packing.

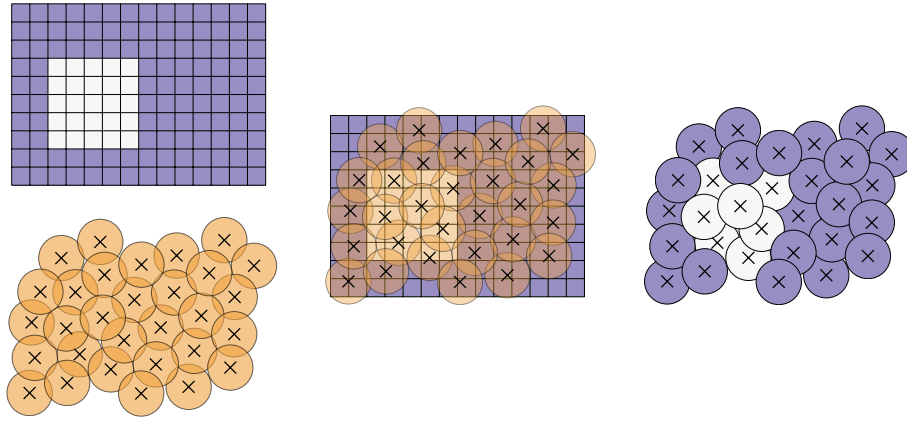


Figure 9.6: Principle of the discretization from a segmented image. A segmented image is used as a mask on a dense random packing to individually set the properties of the particles.

Arbitrary particle state variables (in addition to the particle type) can be set using this procedure, typically to define distinct objects made of the same material [36].

9.4 Contact Event Detection

The detection of the contact of distinct objects is implemented by testing the membership of the numerical particles to predefined clusters. The membership can be defined from a 3D tomography image. This algorithm is presented for the sake of completeness but does not introduce key novelties and thus will not be further examined.

The detection of contact events between distinct physical objects (Figure 3.9a) is straightforward: a particle state variable represents the membership to a specific physical object (Figure 9.7).

This state variable is initialized at the beginning of the simulation, typically using the procedure described in Section 9.3. When two particles interact, their respective memberships are compared, the attractive forces are canceled for particles from distinct objects. Arbitrary behavior can be readily implemented.

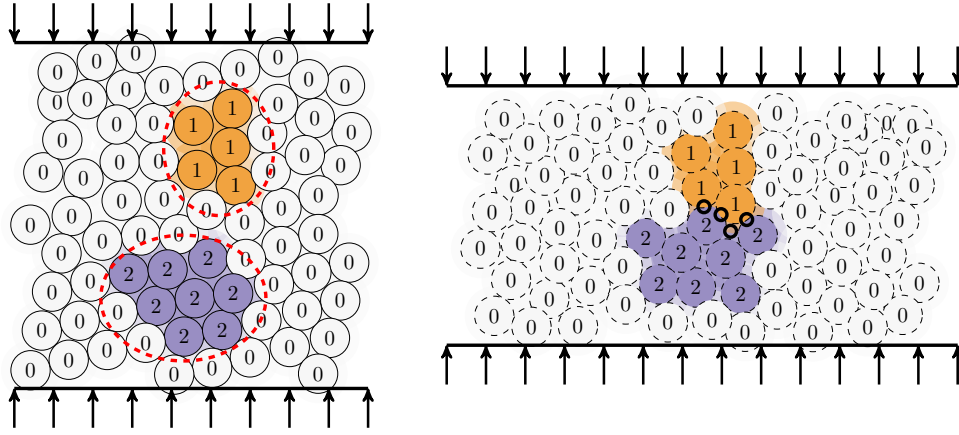


Figure 9.7: Principle of the detection of contact between distinct objects. Each particle is assigned to an object at the initial state object, here 0, 1 or 2. In the deformed state, four pair interactions (circled) are considered as events between objects 1 and 2. The neighbor changes in object 2 are simply accounting for inelastic strain in this solid, they are not considered as physical contacts.

9.5 Self-Contact Event Detection

The detection of the self-contact of the interface of a single object is based on a local approximation of the free surfaces. The metrics are computed for each particle from the position of its neighbors. This algorithm is only used in Part V. More specifically, its tuning, discussion and extension are to be found in Chapter 15. For applications to complex mesostructures, refer to Chapter 16.

In the context of our method, neighbor changes between elementary particles are accounted on to describe inelastic strain in continuous media. A dedicated discrimination algorithm must detect the new pairs that must mimic the physical interaction of interfaces.

The detection of self-contact events (Figure 3.9b on page 48) cannot rely on the initial state of the system (Section 3.2):

- By definition, physical self-contacts involve particles that are members of the same object;
- In a finite transformation context, particles may migrate away or toward the free boundaries: the self-contact detection cannot rely on the particle initial position.

The conceptual work-around is to compute a local metric accounting for the existence, and the orientation, of a free boundary in a neighborhood: a metric somewhat analogous to the classical outward pointing normal.

The outward vector \bar{n} is computed for each particle i , and is the opposite of the sum of the branch vectors⁷ \bar{l} of its neighbors j (Figure 9.8):

$$\bar{n}_i = - \sum_j \bar{l}_j \quad (9.3)$$

The outward vector is not normalized to unity, as its magnitude roughly quantifies the existence of a free boundary.

⁷The branch vector \bar{l} is the geometrical vector from the center of a particle to the center of a neighbor.



Figure 9.8: Computation of the outward vector \bar{n} of a particle, from the positions of its neighbors. (a) A particle surrounded by neighbors. The centroid of the neighbors is very close to the center of particle. The magnitude n is small and its orientation is meaningless. (b) A particle at a free surface. The magnitude n is large and \bar{n} points “outwards”.

In our method, the interaction force in a pair depends not only on the state of the pairs, but also on the state of their neighbors. This methodology is referred to as a many-body law in a MD context.

A pair state variable is initialized at the beginning of the simulation: all initially interacting pairs are considered to represent an “internal” interaction (Figure 9.9a). When the packing deforms, new pairs are created: the outward vectors (magnitude and orientation) of the two particles are compared and the pair state variable is set to:

- “Internal” if the neighbor change is considered to be a normal effect of inelastic strain (Section 9.1), “internal” pairs follow the standard interaction law;
- “Interface” if the new pair interaction is considered to be the result of a self-contact event. The attractive forces are canceled for “interface” pairs.

Qualitatively a self-contact event is detected if the outward vectors have a large magnitude and point toward the new neighbors. In subsequent time steps, if the pair state variable shared with a neighbor is set as “interface”, this neighbor is excluded from the centroid computation for the evaluation of the outward vector.

Algorithmically, for a new pair of particles $\{i, j\}$, the following variables are considered (Figure 9.10): the respective outward vectors $\{\bar{n}_i, \bar{n}_j\}$ and a unit vector \bar{e}_n pointing from the center of i to the center of j .

Three parameters are introduced: two angle threshold α_{ij} and α_{e_n} and a magnitude threshold N_{mag} . The numerical values used can be found in Table 9.3. A new pair is classified as “interface” only if:

$$\left(\cos(\bar{n}_i, \bar{n}_j) \leq \cos \alpha_{ij} \right) \quad (9.4a)$$

and

$$\left(\left(\cos(\bar{n}_i, \bar{e}_n) \geq \cos \alpha_{e_n} \text{ and } n_i \geq N_{\text{mag}} \right) \text{ or } \left(\cos(\bar{n}_j, -\bar{e}_n) \geq \cos \alpha_{e_n} \text{ and } n_j \geq N_{\text{mag}} \right) \right) \quad (9.4b)$$

In short, an interface pair is detected if the outward vectors are not too parallel and at least one outward vector is large and points toward the other particle.

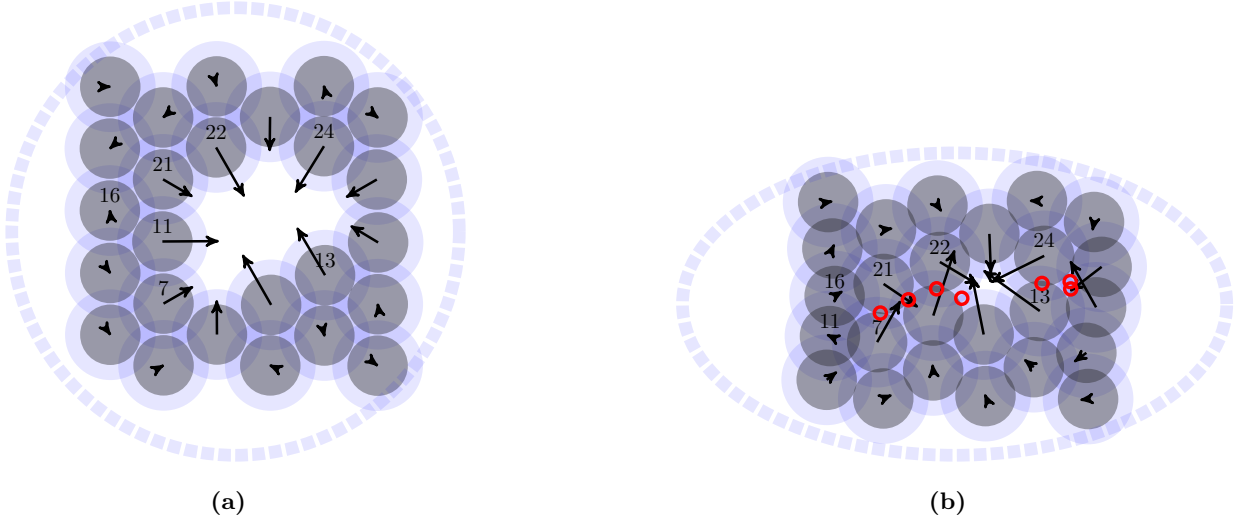


Figure 9.9: Principle of the detection of self-contact events. Example of a hole in an infinite packing. (a) Initial state. All pairs are considered as “internal” interactions. The outward vectors \bar{n} of each particle are computed. (b) Deformed state. The newly created pairs $\{i, j\}$ are classified as “interface” or “internal”, based on the respective orientation and magnitude of \bar{n}_i and \bar{n}_j . The circled pairs, e.g. $\{7, 21\}$ and $\{13, 24\}$, are classified as “interfaces”. In contrast, the new pair $\{7, 16\}$ is detected as “internal”. Particle 11 moved away from the free surface, the magnitude n_{11} has reduced.

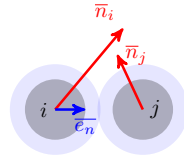


Figure 9.10: A newly created pair of particles $\{i, j\}$, with the respective outward vectors $\{\bar{n}_i, \bar{n}_j\}$.

N_{mag}	α_{e_n}	α_{ij}
mm	°	°
2.3	80	65

Table 9.3: Self-contact detection parameters used with *TRILIN*.

Chapter 10

Chosen Numerical Tools

Chapter 9 exposed the conceptual principle of the developed method to meet our modelization objectives. This chapter presents the choice of effective numerical tools to implement it. The introduction gives a few guidelines followed in the choices of the tools and two short sections focus respectively on DEM and FEM tools.

The main objective in the choice of the numerical tools was to limit as much as possible the programming effort to the implementation of new features. It was also hoped to reach reasonable computing efficiency of the codes without requiring too specialized skills in computing science and too much programming effort. Both objectives imply to rely as much as possible on preexisting tools – software solutions and libraries – adaptable to the need with at most minor modifications. In a more global perspective, the limitation of the number of manipulated programming languages was also considered. In order to favor reproducibility¹, distributed and collaborative development and task automation, command-line interfaces were often preferred.

Tools were thus chosen on the criteria of being scriptable, expendable, reproducible, tested, documented and supported by a reactive community of developers and users. Key time consuming operations, first of which DEM resolution, needed to be scalable, both from the computational and the license point of view. A technically efficient choice² can be the orientation toward open-source and free/*libre* tools. The basic choice of GNU/Linux operating systems was also favored by platform coherence considerations, to smoothly move from different types of machines, including computing clusters.

The choice of the software, language, version control system and parallelization paradigm was iterative, based on trial and error and progressive coherence construction with the adopted simulation software solutions. A specific effort was also devoted to pool tools and portions of code, although it proved delicate. Octave, scilab and cuda were abandoned in favor of python, git, C++ and MPI. The python language not only imposed itself as the tool for the development of numerical methods and pre/post processing procedures – both for DEM and FEM simulations – but it was also the scripting language of a vast majority of the used software solutions.

The main software solutions used are summed-up in Table 10.1, a short description and argumentation of their choice is given in Section 10.1 for DEM related tools and in Section 10.2 for FEM related tools.

¹Reproducibility in time and between machines and users. Although, in a strict sense, numerical repeatability seems illusory [185].

²May Richard Stallman have mercy for such a blasphemy.

Software	Assigned Task	Implementation language	License	Personal Work
liggghts	DEM solver	C,C++ (MPI)	GPL	develop, document, use, bug report
ovito	DEM postprocess	C++, python	GPL	use, bug report
code_aster	FEM solver	fortran, python	GPL	use, bug report
salome	FEM preprocess	C++, python	LGPL	use
paraview	FEM postprocess	C++, python	BSD-3	use
python	Swiss army knife		PSFL	use

Table 10.1: Main software solutions.

10.1 DEM Tools

The chosen DEM solver is liggghts, selected for its interesting compromise between performance and ease of development. The specialized postprocessing and visualization tool ovito proved well suited to our needs.

Although the conceptual model used should theoretically not depend on the chosen implementation, the performance and the internal structure of a code largely drive its possible uses and developments.

Existing DEM and MD code include: dp3d [145], edem³, esys-particle [240], granoo [9], gromacs [1], lammmps [172], liggghts [120], pfc⁴, rocky DEM⁵, woo [214], yade [215].

In this work, it was from the start assumed that massively parallel paradigms had to be used, in order to scale-up to larger problems without further development effort. The choice of liggghts [120], parallelized in MPI, as the DEM solver led to leave aside an attempt to develop and adapt an embryonary in-house GPGPU parallelized code. Although no in-depth systematic comparison of the existing codes was carried out, the liggghts code was chosen for its interesting compromise between performance, ease of development and modularity.

The code is a fork from lammmps, a popular MD code, from which it inherits its modularity, its sequential and parallel performance [200], and an active community of users and developers. Developed by a private company stemming from – and still closely related to – the academic world, the public version of the code is open-source and freely available.

Liggghts is currently a reference in terms of available features for the DEM⁶. A vast user community, both industrial and academic, can be relied on to test and share issues. From the user point of view, the documentation is overall comprehensive and sufficient for autonomous use.

Documentation regarding the development of new features is somewhat scarce outside the most standard procedures. By design, the code is in large parts modular, allowing the development of features as simple add-ons. For example, a new interaction law can simply be introduced by adding an autonomous file in the sources. Some features, as the management of pair history, happen to be written in old-fashioned C language⁷, with syntaxes that can be hard to grasp for the neophyte and restrain the flexibility of their implementation. Regular improvements, extensions and bug fixes are published and the

³Proprietary code commercialized by DEM Solutions Ltd.

⁴Proprietary code commercialized by Itasca.

⁵Proprietary code commercialized by ESSS (engineering simulation and scientific software).

⁶ Multi-physics coupling, first of which with fluid dynamics, is also a strong emphasis, but is outside the scope of our work.

⁷

developers are accessible and reactive regarding requests.

The use of liggghts imposed the implementation of the developed features in C++ in a MPI framework. It also promoted the use of git as version control system, in first place to easily take profit of the periodical evolution of the code. The software was extensively used on numerous machines, using from 1 to 24 CPUs. In addition to this use and developments of interaction laws, metric evaluation routines and generic procedures⁸, our work also involved bug detection in both sequential and parallel issues – and their reporting with eventual resolution – and documentation [96], more specifically regarding development procedures. Liggghts was the most deeply used tool within our work.

Ovito [220] rapidly imposed itself as the visualization tool for particle data. More tailored to discrete simulation needs than a generic visualization software, as paraview for example, it is developing at a rapid pace, regularly introducing useful features, is fully scriptable in python and is thoroughly documented, both for scripted and graphical use. The developer is extremely reactive⁹ and has a clear view of the needs of its community. Our work mainly focused on both graphical and command-line use of the software, with periodic bug reports and feature requests.

10.2 FEM Tools

The code_aster FEM solver is chosen for its comprehensive documentation and its “out-of-the-box” handling of viscoplasticity and finite strain. Contact, and remeshing are not considered. Side tools salome and paraview are designed and adapted to the needs of code_aster.

The FEM solver was chosen with less emphasis on performance and modularity. The code needed to readily handle finite transformations and viscoplastic behavior in quasi-incompressible cases¹⁰. The objective was to provide a reliable reference for the designed DEM models more than to easily implement new concepts.

The choice of code_aster [245] as the FEM solver was strongly triggered by the outstanding quality of its documentation. Not only it comprehensively covers and illustrates the use of the code itself, but it can also be considered as a standalone and didactic review over many mechanical and numerical issues arising in the FEM, including for advanced features¹¹. Similarly, code_aster provides a huge collection of test cases, illustrating numerous configurations and potential syntaxes and uses.

In addition, code_aster can handle a large variety of mechanical behaviors¹², is fully scriptable in python and almost systematically provides explicit and insightful error messages, directly pointing to recommended readings in the documentation. The vast and active community is helpful and the developers can provide fast add-ons or patches¹³.

Our work in code_aster focused on its use for finite transformation simulations and contributions to the community *via* bug reports.

⁷This was a choice of the developers: “the C++ routines that do the serious computations in LAMMPS are written in a simple C-like style, using data structures that are nearly equivalent to Fortran arrays. This was done to try and avoid any performance hits.” [201, FAQ 2.6].

⁸In files respectively named `normal_model_*.h`, `compute_*.cpp` and `fix_*.cpp` in the code conventions.

⁹Bug fixes have been provided within hours.

¹⁰In comparison, appealing libraries like for example deal.II [17] require much more development effort to set-up a specific case.

¹¹The quality of the documentation being only moderated in case of the use of the automatic translation, from French to English, often altering the readability of the document. The same remark applies to error messages. Along the same line, command and variable names are only meaningful in French and the release notes don’t seem to be translated.

¹²The mechanical behavior described by the MFront tool can be used.

¹³A feature request was treated within days.

FEM pre-processing was based on salome¹⁴ which readily produces formats compatible with code_aster, is well documented and is used by an active community. The visualization of the FEM results were based on a fork¹⁵ of paraview, provided by salome. Although documentation of paraview is chaotic, it is somewhat compensated by its huge user community. Work with salome and paraview was limited to mere usage, taking advantage of both tools being fully scriptable in python.

¹⁴ The salome-meca platform includes code_aster as the solving engine and salome for numerous pre- and post-processing tool.

¹⁵ Implementing the MED format used in code_aster.

Part IV

Compression of Dense Bi-Material

11 Single Material	119
11.1 Interaction Law Choice	119
11.2 Calibration Procedure	120
11.2.1 Strain Rate Sensitivity Calibration	121
11.2.2 Stress Level Calibration	123
11.2.3 Scope of Validity	123
11.3 Verification	124
11.3.1 Equilibrium	124
11.3.2 Temporal Convergence	125
11.3.3 Spatial Convergence	126
11.4 Macroscopic Behavior	127
12 Bi-Material Test Cases	131
12.1 FEM Reference	131
12.2 Parallel Configuration	132
12.3 Series Configuration	133
12.4 Spherical Inclusion Configuration	134
12.4.1 Spatial Convergence	136
13 Complex Multi-Material Mesostructure	139
13.1 Computation on a Full Sample	139
13.2 <i>In Situ</i> Configurations	141

In Part III, the conceptual and algorithmic principles of the developed method were described. To apply the DEM to the simulation on inelastic incompressible strain, a phenomenological model is designed. Numerous spherical particles discretize continuous media and their collective re-arrangements mimic inelastic strains. Innate and powerful handling of discontinuities and topological events are expected.

Part IV effectively illustrates the methodology on uniaxial compression of single and multi-materials. It is structured in three chapters:

- Chapter 11 presents the simulation of single phase materials, including the calibration procedure of the numerical parameters and the quantification of the expectable accuracy.
- Chapter 12 is dedicated to test cases on simple geometries for bi-materials. DEM results are compared to reference data from FEM simulations.
- Chapter 13 applies the methodology to a real composite mesostructure. The morphology of a 3D full sample, starting from X-ray tomography image, is discretized and compressed. *In situ* data and simulations are compared for local configurations.

The main results of this part were submitted as an article to the International Journal of Mechanical Sciences. Minor revisions were requested for publication and the amended version is proposed in Appendix D.

Highlights - Part IV

Compression of Dense Bi-Material

- A calibration procedure allows the tuning of the numerical parameters to mimic targeted macroscopic behaviors: plasticity and viscoplasticity.

The strain rate sensitivities are limited to a maximal value and are valid only for a given strain rate range. Plastic behaviors are valid for various decades of strain rate.

- Single material can be compressed up to large strain ($\varepsilon = 1$) with limited relative error on the volume and the flow stress (typically around 10%).
- Simple bi-material configurations are compared to results obtained by the finite element method (FEM).

On macroscopic metrics (flow stress and morphology), the error of the developed model is of the same order of magnitude as the error on a single material. No dramatic errors seem to be introduced when simulating multi-material configurations.

- The model is applied to a 3D mesostructure of a full sample, obtained by X-ray tomography.

The discretization procedure from a binarized 3D image is algorithmically cheap. The proposed methods can be scaled up to large and complex geometries.

- The comparison procedure between the numerical results and *in situ* measurements is possible, but the amorphous phase crystallization in the experiments hinders further analysis.

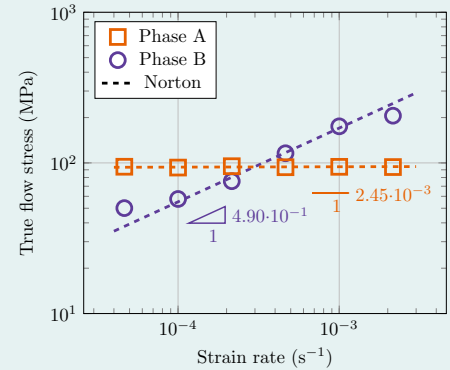


Figure 11.10b

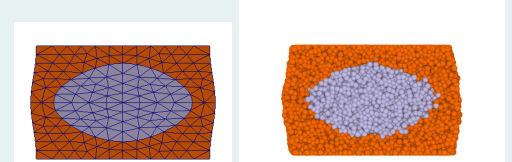


Figure 12.5

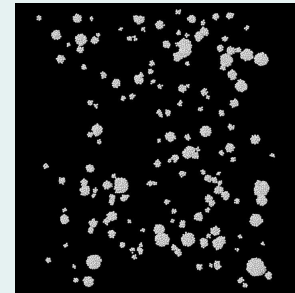


Figure 13.1

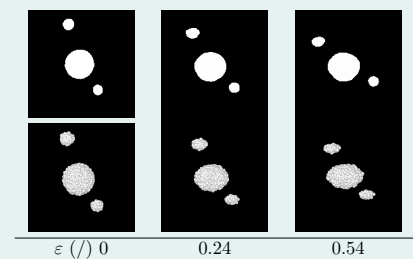


Figure 13.8a

Chapter 11

Single Material

In this chapter, our developed method is applied to the simulation of single materials under compressive loads. It is organized in four sections:

- Section 11.1 gives some elements of discussion regarding the interaction law *BILIN*, defined in Section 9.1.
- Section 11.2 describes the calibration procedure, to choose the correct numerical parameters to mimic a target viscoplastic behavior.
- Section 11.3 provides some guidelines for the verification of the model.
- Section 11.4 concerns the application of the model to two qualitatively distinct behaviors, with high and low strain rate sensitivities.

A first approach to the study of the local stress fields is proposed in Appendix A.

11.1 Interaction Law Choice

The model BILIN is attractive-repulsive. The forces are elastic linear and the attractive force is only activated for tensile motions of the pair.

In Part IV, limited to compressive loads, the *BILIN* interaction law (Figure 11.1) is applied and tested.

The choice of the elementary interaction law is somewhat arbitrary. Indeed, the local interactions are not individually meant to have physical sense. In counterpart, their collective effects must fulfill the modelization objective. *A priori*, no direct qualitative link can be established between the individual or pairwise behavior and the collective result. Very practically, the tested configurations often proved to display counter intuitive trends and the macroscopic behavior seems to be dominated by steric effect, with secondary regard for the details of its prescription¹. Interaction laws are thus chosen conceptually as simple as possible and of reasonable cost from a computational point of view.

A useful restriction to creativity in the design of interaction laws is to respect the linearity of the forces with the introduced parameters. This linearity allows a straightforward calibration of the modeled stress level. In the discrete element method (DEM),

¹To some extent, although they are poorly integrated by explicit schemes, constant attractive and repulsive forces behave well enough to be effective.

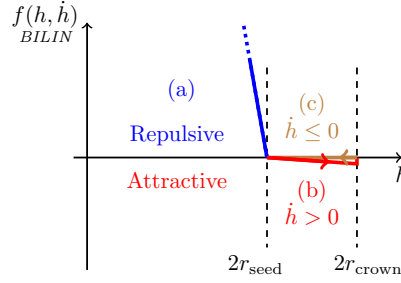


Figure 11.1: Pairwise interaction law *BILIN* used in this part. Force f versus distance h . Attractive to repulsive force ratio and radii to scale. Classical DEM conventions are applied: repulsive forces are positive. Refer to Section 9.1 for the full description of the law.

for each particle, the time integration is based on the ratio of force and mass f/m (Section 8.3). The overall kinematic behavior will thus be kept statistically² unchanged if both masses and forces are scaled by a common factor. (Section 11.2).

The relative size of the crown $r_{\text{crown}}/r_{\text{seed}} = 1.5$ is chosen to avoid the interaction of two particles “across” another one. The choice of using identical radii for all particles is not an algorithmic limitation, the proposed implementation can readily accept a limited dispersion. The effect of such a dispersion has not been investigated, but may be a good strategy to limit excessive numerical crystallization. It was not necessary in this model, as attractive forces are much weaker than the repulsive forces, the distance between two particles is sufficiently free.

11.2 Calibration Procedure

This section describes the calibration procedure of the numerical parameters of the interaction law to mimic perfect viscoplastic behavior. It is divided into two steps, tuning respectively the strain rate sensitivity and the stress level. Two sections each examines one step and a third short section draws some limitations.

The objective of our calibration procedure is to model a perfect viscoplastic behavior, described as a relation between the scalar macroscopical strain rate $\dot{\epsilon}$ and the flow stress σ , by the unidimensional³ Norton law [126, p.106]:

$$\sigma = K|\dot{\epsilon}|^M \cdot \text{sign}(\dot{\epsilon}) \quad (11.1)$$

. Where M is the strain rate sensitivity and K is the stress level. All cases presented in this chapter being in compressive state, strain, strain rate and stress are given in absolute value.

As the DEM does not rely on a continuous framework, the numerical parameters cannot be derived *a priori* from the targeted macroscopic behavior. We work here at fixed ratio $r_{\text{crown}}/r_{\text{seed}} = 1.5$, to allow a large overlap zone without catching second neighbors. The seed radius is arbitrarily set to a size of $r_{\text{seed}} = 1$ mm. The ratio between attractive and repulsive stiffnesses is set to $k_{\text{rep}}/k_{\text{att}} = 10$, to guarantee a numerically predominant repulsion.

²The numerical errors are not negligible: the position of individual particles does notably change with the f/m ratio.

³Refer to Equation 2.1 on page 24 for the full tensorial form.

During this PhD, the study of the behavior of our system promoted the idea of an intimate relationship between numerical time step, natural period of the packing and prescribed strain rate. The overall behavior of the system seems to be driven by these three parameters. However, no clear trend could be established regarding the role of the time step. We thus work at fixed time step, set for the model *BILIN* to $\Delta t = 5 \cdot 10^{-4}$ s (see also Section 11.3.2). Our calibration procedure thus takes advantage of the relationship between strain rate and natural period, it probably could be improved by taking into account the role of the time step.

The remaining parameters to be chosen are the repulsive stiffness of the interactions k_{rep} and the mass of the particles m . We propose a two-step calibration procedure, based on uniaxial compression test simulations, on cubes of single materials:

1. Calibrate the strain rate sensitivity M , tuning the ratio between mass and repulsive stiffness m/k_{rep} .
2. Calibrate the stress level K , applying a common multiplicative factor to both mass m and repulsive stiffness k_{rep} .

The numerical parameters, obtained independently for each phase, are used in multi-material simulations without further fitting procedure.

11.2.1 Strain Rate Sensitivity Calibration

The strain rate sensitivity is calibrated by adjusting the natural period with respect to the prescribed strain rate. The ratio mass/stiffness of the model is thus chosen.

The strain rate sensitivity M of a packing depends on its ability to quickly rearrange itself, with regards to the prescribed strain rate. To quantify an image of the reaction time, we use the natural period t_0 of an ideal spring-mass system of stiffness k_{rep} and mass m :

$$t_0 = 2\pi \sqrt{\frac{m}{k_{\text{rep}}}} \quad (11.2)$$

This value is not meant to match the actual oscillation period of particles, but to quantitatively compare sets of parameters.

Packings of $5 \cdot 10^3$ particles with natural periods ranging from $1 \cdot 10^{-2}$ to 1 s are compressed at strain rates from $3 \cdot 10^{-6}$ to 1 s^{-1} . For each natural period, the flow stress σ is normalized by the flow stress at the lowest strain rate σ_{low} . The results (Figure 11.2) exhibit a clear influence of the natural period. All packings follow a similar trend: the influence is first limited, then the flow stress increases with the strain rate up to a limit value. Above the limit, the strain rate is too high for the packing to collectively cope, the deformation is localized to the particles near the moving mesh. The overall behavior is shifted to various ranges of strain rate by the value of the natural period.

To sum up, the strain rate sensitivity M , i.e. the slope in the space $(\dot{\epsilon}, \sigma/\sigma_{\text{low}})$, is driven by the relation between the natural period and the strain. The common trend for all configurations (Figure 11.3a) is clearly exhibited in the space $(t_0 \sqrt{\dot{\epsilon}}, \sigma/\sigma_{\text{low}})$.

To quantify the observed trend, the data are approximated by least-square fitting, using a sigmoid of generic expression:

$$\sigma/\sigma_{\text{low}} = a + b/(1 + \exp(c - d \cdot t_0 \sqrt{\dot{\epsilon}})) \quad (11.3)$$

The fitting parameters used here are $(a, b, c, d) \approx (0.9048, 4.116, 3.651, 210.0)$. Using this fitted common trend, a master curve is built in the space $(\dot{\epsilon} \cdot t_0^2, M)$, thus describing an

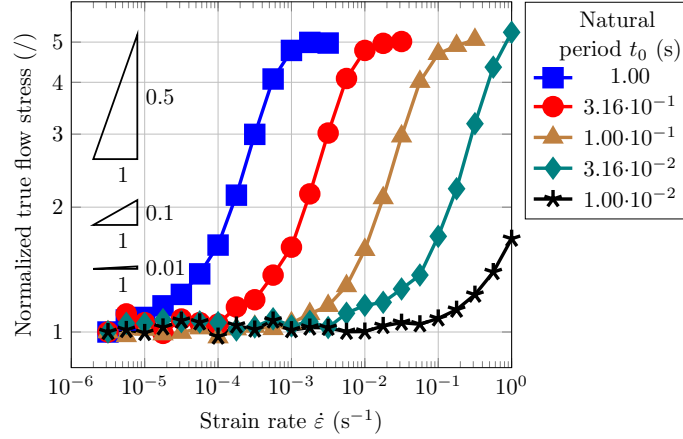


Figure 11.2: Normalized flow stress at a strain of 0.3 for $5 \cdot 10^3$ particles versus prescribed strain rate. Influence of the natural period on strain rate sensitivity.

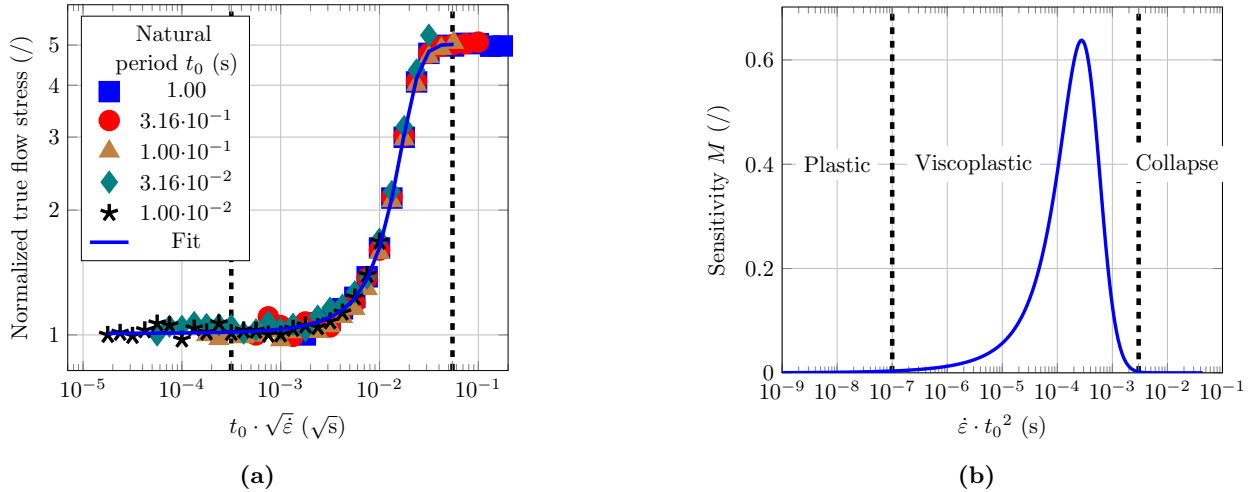


Figure 11.3: Calibration of the strain rate sensitivity. Successive steps toward the master curve. (a) Normalization in the $(t_0\sqrt{\dot{\epsilon}}, \sigma/\sigma_{\text{low}})$ space. Common trend for all natural periods. Sigmoidal fit, see Equation 11.3. (b) Master curve of strain rate sensitivity. Three flow regimes.

overall behavior of the model *BILIN* for all tested configurations⁴ (Figure 11.3b).

Three flow regimes, in terms of strain rate sensitivity, can be identified in Figures 11.3b and 11.3a:

- Plastic: for $\dot{\epsilon} \cdot t_0^2 < 1 \cdot 10^{-7}$ s the strain rate sensitivity is negligible ($M < 4 \cdot 10^{-3}$). A plastic behavior can thus be represented, with stress variation of the order of magnitude of the expected precision of the model, valid over various orders of magnitude of strain rates. The packing rearranges quickly enough when deformed, so

⁴ As a side note, although the analysis was not pushed further, our proposal exhibits the role of t_0^2 to compare the behavior of distinct sets of parameters. This parameter seems to have a driving role in the global equilibrium criteria \mathcal{Q} described in Section 8.6.

that variations of strain rate does not affect the flow structure.

- Collapse: at higher values than $\dot{\epsilon} \cdot t_0^2 > 3 \cdot 10^{-3} \text{ s}$, the packing is not reactive enough for the particles to collectively cope with the strain. The strain localizes next to the moving planes, the flow stress drops and the macroscopic equilibrium is lost. Such configurations are not suitable for our purpose.
- Viscoplastic: in the intermediate window, the $\dot{\epsilon} \cdot t_0^2$ value governs the sensitivity of the packing, up to a maximum of 0.6. In this configuration, when the strain rate increases, the particles are forced to indent more to rearrange, leading to higher flow stress. However, the sensitivity is strongly strain rate dependent, An actual viscoplastic behavior can only be modeled *via* an averaged strain rate sensitivity, with a scope of validity limited to a narrow range of strain rates.

The master curve (Figure 11.3b) allows to directly choose the natural period approximating the desired sensitivity at the targeted strain rate. The m/k_{rep} ratio is thus fixed. If the strain rate range is known *a priori*, the master curve also gives an approximation of the variation of the strain rate sensitivity within the strain rate range. For example, in order to model a high strain rate sensitivity $M \approx 0.5$, the value $\dot{\epsilon} \cdot \Delta t^2$ must be chosen close to $2 \cdot 10^{-4} \text{ s}^{-1}$. If the targeted strain rate is $2 \cdot 10^{-4} \text{ s}^{-1}$, the chosen natural period would be chosen as $t_0 \approx \sqrt{2 \cdot 10^{-4} / 2 \cdot 10^{-4}} = 1 \text{ s}$.

11.2.2 Stress Level Calibration

The mass and stiffness are adjusted to meet the required stress level. Their ratio remains unchanged to respect the expected strain rate sensitivity.

For a given kinematical behavior of a packing, the stress level can arbitrarily be set. The integration of motion, for each particle, relies on the acceleration computed from Newton's second law. Hence, a multiplicative factor applied to both forces and masses leaves the kinematics of a packing, and its strain rate sensitivity, unchanged. Since our interaction laws are linear with the introduced parameters, we can use a common multiplicative factor on stiffnesses and masses.

The stiffnesses k_{rep} and k_{att} are scaled up to match the desired flow stress at the targeted strain rate. The mass m is proportionally adjusted, in order to maintain the correct strain rate sensitivity.

11.2.3 Scope of Validity

Arbitrarily high strain rate sensitivities cannot be modeled. Low strain rate sensitivities are valid over several decades of strain rate. High strain rate sensitivities are only valid on narrow ranges.

This two-step calibration allows us to reach arbitrary stress level, but displays limitations regarding the reachable strain rate sensitivity and strain rate.

We cannot model arbitrary strain rates with a given set of parameters. The numerical strain rate sensitivity depends on the strain rate. This effect can be controlled for very low sensitivities: a negligible sensitivity can be respected over various orders of magnitude of strain rate. However, a large tolerance must be accepted on higher sensitivities, which can only be reasonably approximated on narrow ranges of strain rate. The model also has intrinsic limits regarding the reachable strain rate sensitivities. Reaching higher sensitivity would require lower natural periods, for which the packings collapse and are unable to cope with the strain.

As a general conclusion for this section, our calibration procedure allows to choose independently the stress level and the strain rate sensitivity, tuning the mass of the particles and the stiffness of the interactions. The scope of validity, for controlled sensitivity, is limited to narrow strain rate ranges.

11.3 Verification

Three effects are investigated in dedicated sections: the mechanical equilibrium of the packings, the dependency to the time step of the model and the effects of the size of the packings.

11.3.1 Equilibrium

The DEM is a dynamic method. To model quasistatic phenomena, the mechanical equilibrium must be verified within a sufficient accuracy. Excessive dynamic effects are not suitable for our purpose.

In order to simulate quasistatic phenomena, the behavior of the packing must be independent from the way the strain is applied. The total forces⁵ acting on the boundary conditions, the top and bottom meshes, respectively mobile and fixed, must balance. If a mesh moves too fast, the macroscopic equilibrium is lost and the strain localizes next to the moving plane.

At a given strain rate, the equilibrium relative error depends on the natural period, but is of the same order of magnitude for all strains. The equilibrium errors (Figure 11.4) are always inferior to 0.1 % for both phases in the studied strain rate range. The macroscopic behavior is indeed kept unchanged by inverting the mesh motions.

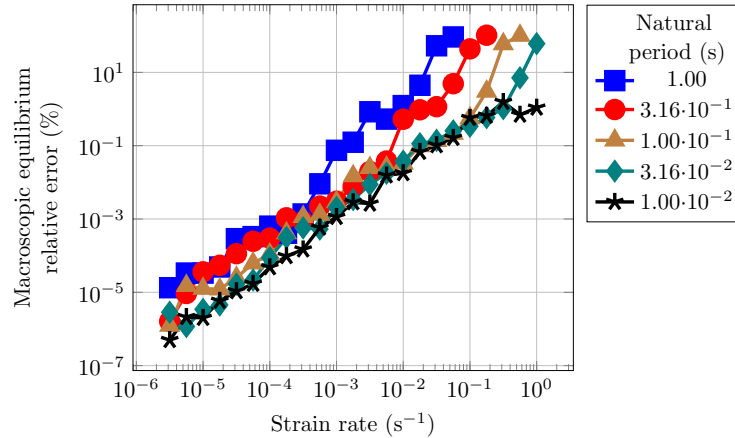


Figure 11.4: Macroscopic equilibrium for single material. Relative error versus strain rate for $5 \cdot 10^3$ particles, strain 0.3. Effect of the natural period.

⁵The torque balance is of little interest in this test case, as resulting torques on the meshes are only numerical noise.

11.3.2 Temporal Convergence

The choice of the time step can induce quantitative and qualitative changes in the response of the system. A proper temporal convergence seemed excessively costly and is not strictly necessary for our purpose. The time step is chosen to provide a correct integration of the motion, but does not meet convergence requirements.

Although our modeling objective is to capture quasistatic phenomena, the used numerical relies on an explicit dynamic framework. A fundamental algorithmic issue is thus the choice of the time step (refer to Section 8.3.2). The chosen metric to study the effect of the time step on a packing behavior is the time step normalized by the natural period $\Delta t/t_0$.

An upper bound to the value of the time step is set by the proper integration of the motion of the particles. Using the *BILIN* interaction law, if $\Delta t/t_0 > 2 \cdot 10^{-1}$, the coarse time discretization leads to totally unpredictable behavior, particles get massively lost during the simulation. This configuration must thus be avoided.

Below this threshold, the influence of the time step on the stress/strain response of packings of 500 particles⁶ was tested (Figure 11.5). The time step being a purely numerical parameter, it should not influence the metrics of interest of the studied physical phenomena. From a traditional perspective, a model would be expected to meet time convergence requirements, understood as independence of the macroscopic flow stress with respect to the time step.

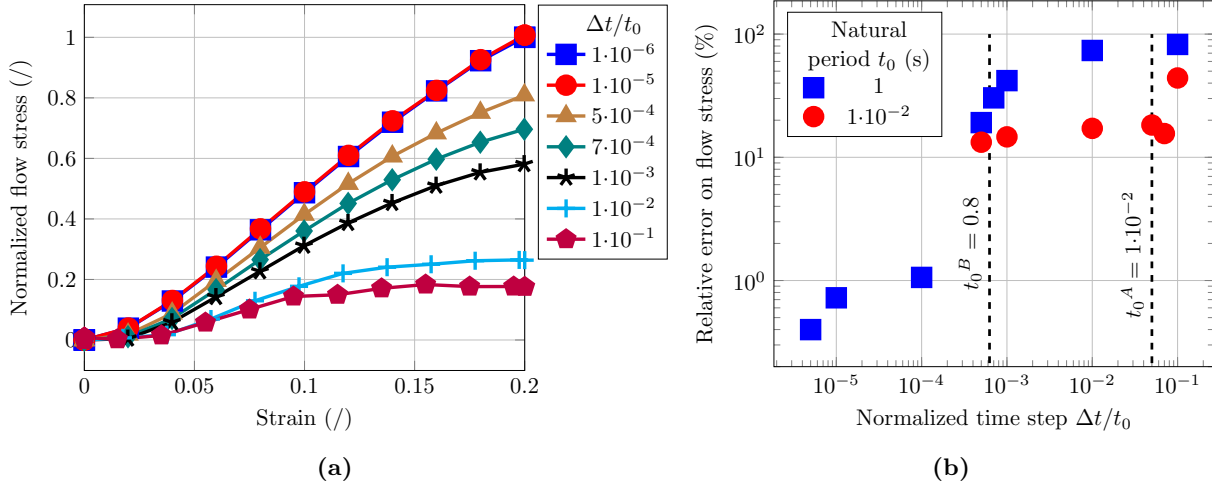


Figure 11.5: Temporal convergence for $5 \cdot 10^2$ particles. (a) Normalized flow stress versus strain for a natural period $t_0 = 1$ s. (b) Relative error on the flow stress at a strain of 0.2 versus the ratio of the time step and the natural period $\Delta t/t_0$. The reference for $t_0 = 1$ s is $\Delta t/t_0 = 10^{-6}$. The reference for $t_0 = 10^{-2}$ s is $\Delta t/t_0 = 10^{-4}$. The $\Delta t/t_0$ values are highlighted for the phases A and B, used later on.

It must be emphasized that the time step influences the stress/strain behavior both quantitatively and qualitatively (Figure 11.5a). For the studied systems, the macroscopic strain/strain time convergence is only reached for time steps smaller than $\Delta t/t_0 < 10^{-5}$,

⁶Although the behavior is not considered geometrically converged for packings of 500 particles (refer to Section 11.3.3), the relative error due to the rough discretization is limited. A partial repetition of the test with bigger packings led to similar results. The use of little packings allows a wider range of time steps to be studied.

with a relative error below 1 % (see $t_0 = 1$ in Figure 11.5b). This constraint would be unreasonably time consuming for our purpose, especially for packing with low natural periods: simulation times would be longer than the PhD duration.

It must be here remembered that we use an analogous model. Requiring the time convergence would be legitimate if a *physical* system of attractive/repulsive spheres were to mimic the inelastic deformation of continuous media. In such a context, the DEM would be a numerical means to simulate the behavior of the physical analogous model. It would thus be necessary for the numerical model to effectively represent the behavior of the physical model: the time step would be required to meet convergence requirements.

In our case, nothing prohibits to check whether the temporally non-converged numerical model itself⁷ displays sufficient analogy to the physical phenomena of interest, the inelastic strain of a continuum. The time step is merely a numerical parameter, whose choice is driven by the respect of a reasonable numerical behavior (typically in our case $\Delta t/t_0 < 2 \cdot 10^{-1}$) and the macroscopic overall behavior of the system. For computational convenience, some freedom can be allowed on the time step choice. In counterpart, the time step must be set to a fixed value from calibration to simulation.

Summing-up, the time step is considered as a numerical parameter, fixed at an arbitrary value, compatible with a reasonable numerical behavior but free from physical constraints at the scale of the elementary particles.

11.3.3 Spatial Convergence

The behavior of our model must be independent enough from the chosen spatial discretization. Packings of a few hundred of particles can roughly provide the correct order of magnitude of flow stress. The error on the flow stress is quantified with respect to a spatially converged state to provide guidelines on the choice of the number of particles.

As our model relies on a collective motion of particles, too small a packing will not display the expected behavior. The kinematical behavior of a single material cube in uniaxial compression is roughly observed with a few dozen of particles (Figure 11.6). With a few hundred of particles, the stress fails to represent the expected plastic trend, but already exhibits a correct order of magnitude (Figure 11.7a). A few thousand of particles allow a controlled relative error, around 10 %. Single material configurations are typically run with $5 \cdot 10^4$ particles. (Figure 11.7b). The relative error is computed with respect to a packing of $1 \cdot 10^6$ particles, for which spatial convergence is considered to be reached.

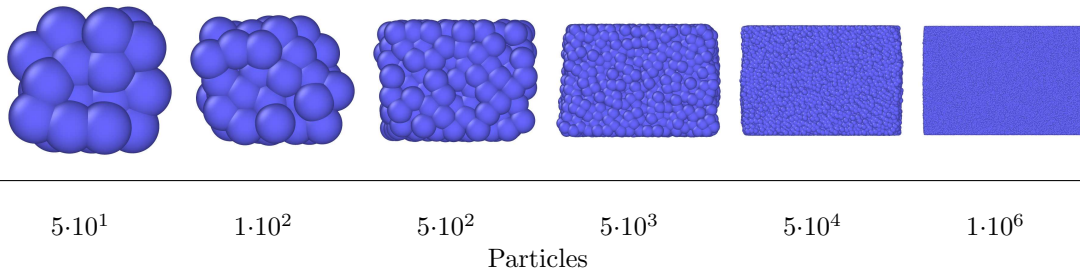


Figure 11.6: Spatial convergence. Cross-section. Strain rate 10^{-3} s^{-1} , true strain of 0.3.

⁷ “It must be considered that a numerical method is in itself a model.” [111, p.65]

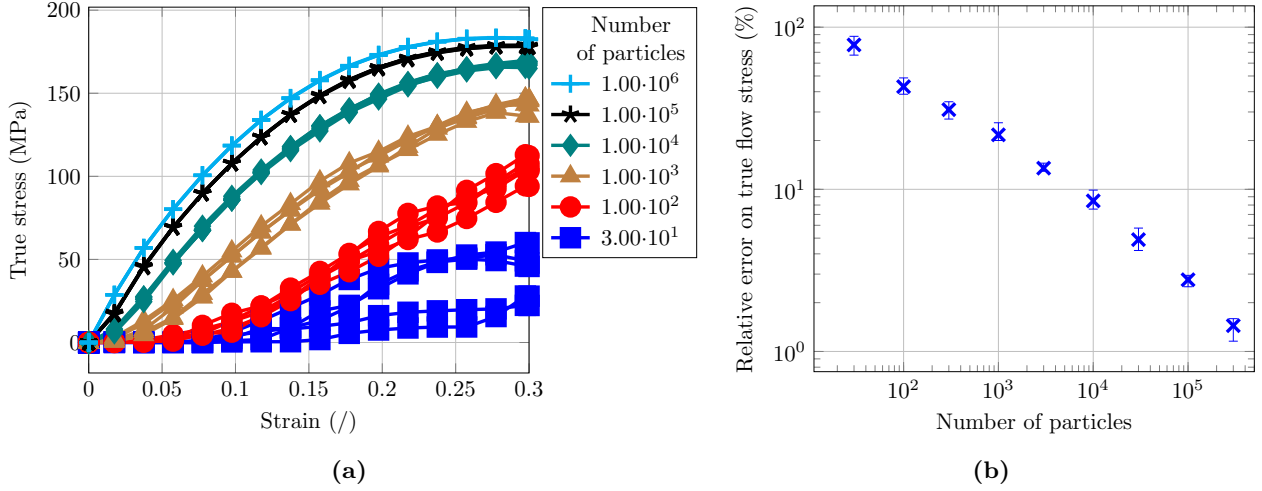


Figure 11.7: Spatial convergence for single material, from 30 to $1 \cdot 10^6$ particles. (a) Flow stress versus strain, using five distinct initial random packings for each packing size. (b) Relative error versus packing size, in regard to the converged simulation ($1 \cdot 10^6$ particles packing).

11.4 Macroscopic Behavior

Under compressive strain the model BILIN allows the packing to rearrange collectively. Large strain can be modeled with controlled volume variation. After a transient regime, the flow strain-stress behavior is precise enough for our purpose. Very distinct behavior can be modeled depending of the strain rate sensitivity, from a plastic-like behavior to a highly strain rate sensitive viscoplasticity.

The two phases behavior are inspired from the experimental model material (Section 2.3). In the identified forming window, around 400°C , The phases both have a flow stress close to 100 MPa in the strain rate window $1 \cdot 10^{-4} - 1 \cdot 10^{-3} \text{ s}^{-1}$, but with drastically distinct strain rate sensitivities. The negligible strain rate sensitivity phase is referred to as A, with a low natural period, the high sensitivity phase is referred to as B, with a high natural period. The corresponding numerical parameters are given in Table 11.1.

A key feature expected for a set of parameters is the conservation of the packing volume. The volume of the packings is estimated reconstructing a polyhedral mesh, using an algorithm implemented by Stukowski [221], based on the alpha-shape method. As a side note, the definition of the boundaries of the modeled objects are somewhat blurry. They are here defined as the envelope of the centers of the particles, for practical

Phase	Discrete parameters			Continuous parameters	
	Scale of the pairs			Macroscopic behavior	
	k_{rep} $\mu\text{N} \cdot \text{mm}^{-1}$	m g	t_0 s	M /	K $\text{MPa} \cdot \text{s}^M$
A	$6.23 \cdot 10^9$	$1.58 \cdot 10^4$	$1 \cdot 10^{-2}$	$2.45 \cdot 10^{-3}$	$9.59 \cdot 10^1$
B	$2.65 \cdot 10^9$	$4.29 \cdot 10^7$	$8 \cdot 10^{-1}$	$4.90 \cdot 10^{-1}$	$5.02 \cdot 10^3$

Table 11.1: Numerical parameters for the two phases. Time step $\Delta t = 5 \cdot 10^{-4} \text{ s}^{-1}$. Radii $r_{\text{crown}} = 0.75 \text{ mm}$, $r_{\text{seed}} = 0.5 \text{ mm}$. Stiffness ratio $k_{\text{rep}}/k_{\text{att}} = 10$.

convenience⁸.

For both phases, the volume variation depends little on the strain rate. The prescribed compression decreases the volume, typically about 5 % for A and 10 % for B (Figure 11.8). Before reaching a somewhat stable flow regime, the packing volume decreases in first 0.2 of strain. Most of the volume variation occurs within this initial stage, the volume then stabilizes on a plateau before a final increase of the error at larger strains, above 0.6. This trend, and its initial transient regime, will also be observed for the flow stress (Figure 11.10a).

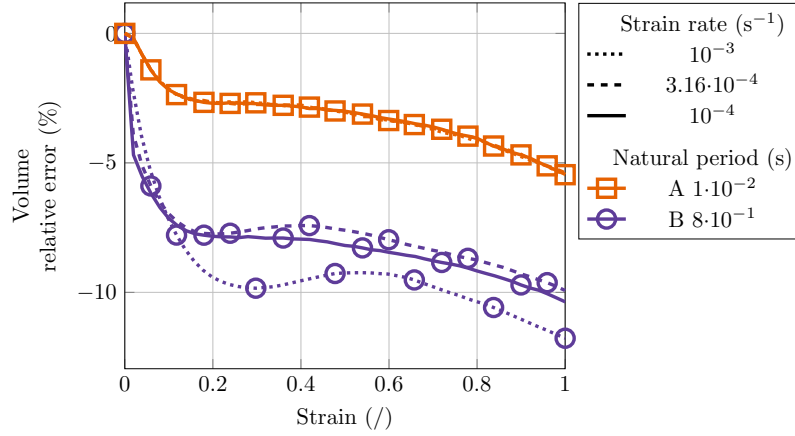


Figure 11.8: Volume conservation for single material. Relative error on volume versus strain for $5 \cdot 10^4$ particles.

Regarding the kinematical behavior of a packing (Figure 11.9), the overall cuboidal shape is conserved, but the sharp edges tend to be blurred along with the strain. This is understood as an effect of the surface tension induced by the attractive component of the interaction law. As the discretization by particles creates local defects in the geometry, the initially flat faces becomes slightly wavy.

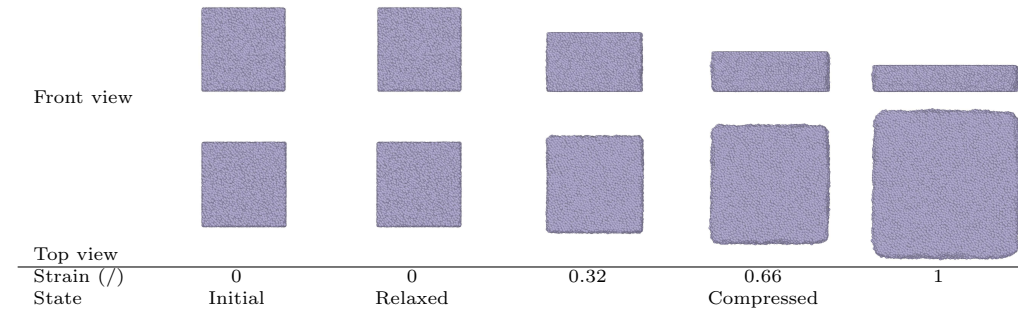


Figure 11.9: Single material packing. Natural period 0.8 s, strain rate $3.16 \cdot 10^{-4} \text{ s}^{-1}$, $5 \cdot 10^4$ particles.

Typical profiles of stress-strain curves are presented Figure 11.10a. In this section, the true stress is computed using an estimation of the cross-section, based on the current

⁸The envelope of the radius of the particle implies an explicit choice of this radius. Two are defined in our model, none of which has intrinsic physical sense.

macroscopic strain and the initial volume, assuming its variations (Figure 11.8) are acceptable. As for the volume evolution, a transitory stage can be observed at the beginning of the deformation, where the stress rises to reach the plastic plateau. The flow stress then oscillates around a fairly constant value. An overshoot effect of the stress can be observed at higher strain rates for the B phase.

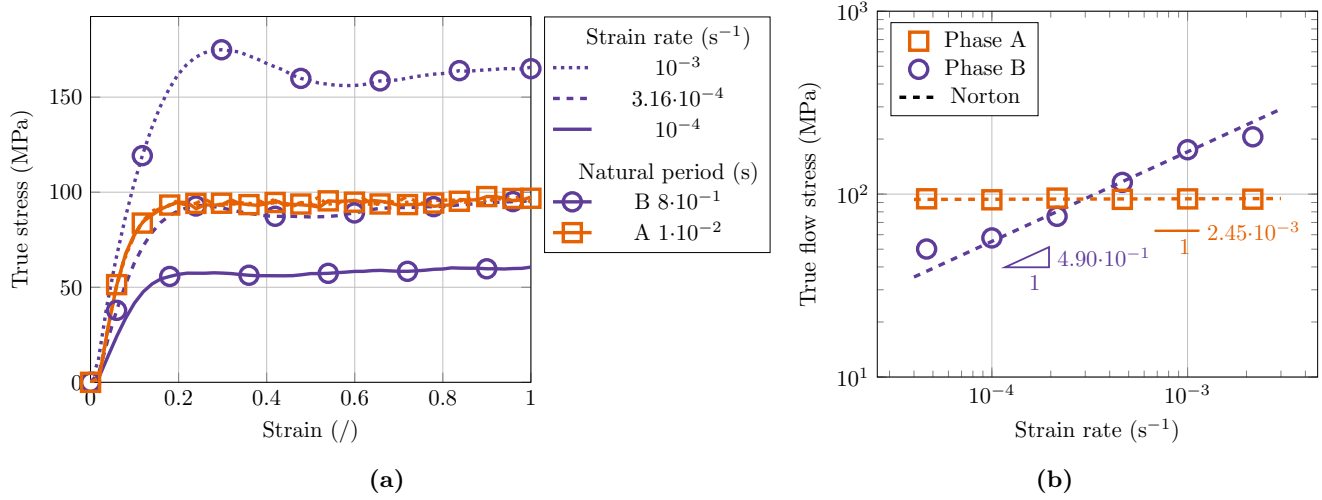


Figure 11.10: Single materials. Phases A and B. Flow stress and strain rate sensitivity. (a) Flow stress versus strain. Effect of the strain rate. (b) Norton law approximation, based on flow stress values at a strain of 0.3. Strain rate sensitivity of both phases in the range $1 \cdot 10^{-4} - 1 \cdot 10^{-3}$ s $^{-1}$.

For each phase, the value at a strain of 0.3 is used to compute the Norton approximation, by least-square fitting (Figure 11.10b and Table 11.1). As discussed in Section 11.2.3, the high sensitivity phase, B, is only valid within one order of magnitude of strain rate, the approximation is not reasonable when the strain rate is out of the studied range.

Chapter 12

Bi-Material Test Cases

In Chapter 11, the numerical parameters of the DEM model have been calibrated, independently, to mimic two qualitatively distinct behaviors.

Keeping in mind the limitations of the single material model, we evaluate in this chapter the reliability of the model for bi-material configurations. The shape of the phases and the engineering macroscopic stress are metrics compared with results from analytical and finite element method (FEM) references, briefly presented in Section 12.1. Three simple geometrical bi-material configurations are studied, each examined in a specific section:

- Parallel (Section 12.2);
- Series (Section 12.3);
- Unique spherical inclusion (Section 12.4). This test case has been further investigated, in terms of spatial convergence and of local fields.

The three geometries are discretized with $5 \cdot 10^4$ particles and uniaxially compressed up to a strain of 0.3, at prescribed strain rates. In the studied configurations, interaction parameters at the interfaces had little influence on the macroscopic results, they have been set to the average of the phase parameters.

12.1 FEM Reference

FEM simulations are used as numerical reference of our bi-material tests. Some key discrepancy sources are looked into: the handling of interfaces, the use of symmetry and the constitutive behavior parameters.

Total Lagrangian FEM simulations, well suited for our elementary geometrical configurations and limited strains, are run using Code_Aster [245]. The visualization of the FEM results are rendered using paraview [100]. The elastic strain is numerically negligible with respect to the inelastic strain. This incompressibility is handled with a mixed formulation displacement-pressure-swelling. A finite transformation formulation is used, taking into account potentially large strains, rotations and displacements. A logarithmic metric of the strain is chosen and the geometries are meshed using quadratic tetrahedral elements. Refer to Appendix C for the exact syntax details and choices.

Top and bottom nodes follow prescribed vertical motion, lateral sides deform freely. The geometrical models are reduced using the symmetries of the problems, while the DEM simulates the full geometries. In FEM, at the interface between two phases, the

nodes are shared, prohibiting any relative motion, which is the most severe difference with our DEM simulations. In the experimental background of this study, the phases have very little adhesion at the interface. Both materials follow a Norton law (refer to Equation 2.1 on page 24), semi-implicitly integrated using a theta-method. The B phase uses the continuous parameters identified in Section 11.4 (Table 11.1). To allow an easier numerical convergence of the model, the numerical strain rate sensibility of the A phase is slightly increased for the FEM simulations ($M = 3.05 \cdot 10^{-2}$ and $K = 120 \text{ MPa} \cdot \text{s}^M$). In the range $1 \cdot 10^{-4} - 1 \cdot 10^{-3} \text{ s}^{-1}$, the induced relative error on the flow stress is $\pm 3\%$.

12.2 Parallel Configuration

In parallel configuration, the phases are mostly independent. The DEM captures the linear mixture law accounting for the macroscopic flow stress.

A cube is vertically divided into two cuboidal phases, for various volume fractions, and vertically compressed at constant strain rates. The engineering stress is compared to a mixture law, linear with the volume fraction.

In this simple configuration, little interaction should take place between the phases, and in ideal conditions, an homogeneous strain for both phases is expected. In the DEM simulations, the global geometry of each phase remains close to a cuboid along the deformation (Figure 12.1).

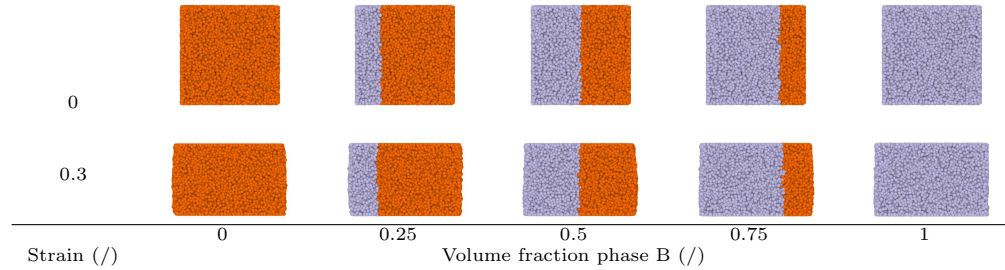


Figure 12.1: Bi-material parallel configuration. Front view, transverse cross-section. Strain rate $4.64 \cdot 10^{-4} \text{ s}^{-1}$.

At given strain rate $\dot{\epsilon}$, the true stress in the phases being independently defined by the Norton law, the global true stress σ_{true} can be computed with an elementary mixture law [78, p.99], linear with the volume fraction f of the phase B:

$$\sigma_{\text{true}}(f, \dot{\epsilon}) = f \cdot K_B \dot{\epsilon}^{M_B} + (1 - f) \cdot K_A \dot{\epsilon}^{M_A} \quad (12.1)$$

To provide a consistent metric for all configurations, the engineering stress σ_{engineer} is used as reference. It is computed at a given strain ϵ (Equation 12.2), based on the true stress and the volume conservation:

$$\sigma_{\text{engineer}}(f, \dot{\epsilon}, \epsilon) = \exp(-\epsilon) \cdot \sigma_{\text{true}}(f, \dot{\epsilon}) \quad (12.2)$$

The engineering stress-true strain profile, as in single material configuration, displays a transitory regime, typically in the first 0.15 of strain, with a progressive rise towards the flow stress (Figure 12.2a).

The DEM model is able to capture, after a transient regime and at the precision of the single phases, the linear pattern of flow stress with the volume fraction (Figure 12.2b). With a rougher discretization, for example only a thousand particles per phase, the result remains qualitatively close, degrading the accuracy by a few percent.

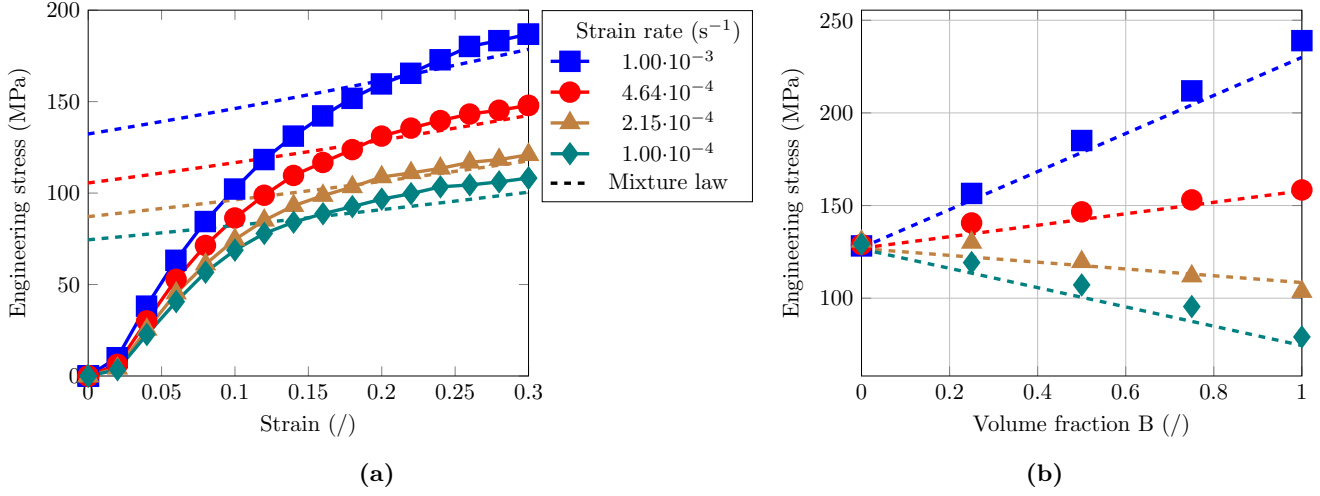


Figure 12.2: Bi-material parallel configuration. Effect of the strain rate on the flow stress. Theoretical reference: mixture law from Equation 12.2. (a) Engineering flow stress versus strain. Fixed volume fraction: 0.5. (b) Linear trend of the engineering flow stress, at a strain of 0.3, with the volume fraction.

12.3 Series Configuration

In series configurations, the deformation is not homogeneous. Depending on the strain rate, a phase will preferentially deform. The macroscopic flow stress modeled with the DEM model comes in good agreement with the FEM reference.

A cube is horizontally divided into two cuboidal phases, for various volume fractions, and vertically compressed at constant strain rate. Using the symmetries, one fourth of the geometry is modeled with the FEM, using approximately $1.3 \cdot 10^3$ nodes. For the full geometry, the ratio DEM particles to FEM nodes would be a little under 10.

In this geometrical configuration, the strain is not *a priori* homogeneous anymore. Due to distinct strain rate sensitivities, one phase preferentially deforms depending on the strain rate, which is qualitatively observed both in FEM and DEM simulations. Qualitatively (Figure 12.3), the B phase (bottom phase) deforms more at lower strain rate. The A phase, at high strain rates, deforms more homogeneously in DEM than in FEM. The “mushroom” shape is slightly blurred in this strain rate range.

FEM and DEM are in good agreement, after the transient regime observed in DEM, within a few percent of relative error (Figure 12.4a). In the strain rate validity range, the DEM model is thus able to capture the final flow stress evolution with respect to the volume fraction (Figure 12.4b). As a side note, the heterogeneity of the strain in the series configuration is responsible for a nonlinear variation of the flow stress with respect to the volume fraction. This effect of the geometry of the bi-material, clearly displayed at $1 \cdot 10^{-3}$ s $^{-1}$ (Figure 12.4b), is correctly reproduced.

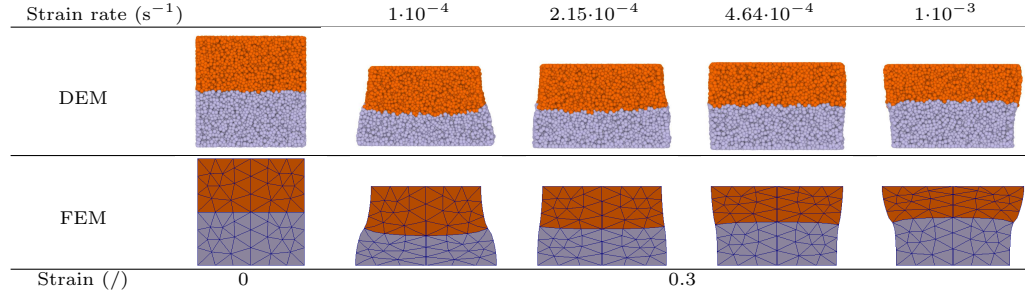


Figure 12.3: Bi-material series configuration. Volume fraction 0.5. Bottom phase: B (high sensitivity). Front view, transverse cross-section.

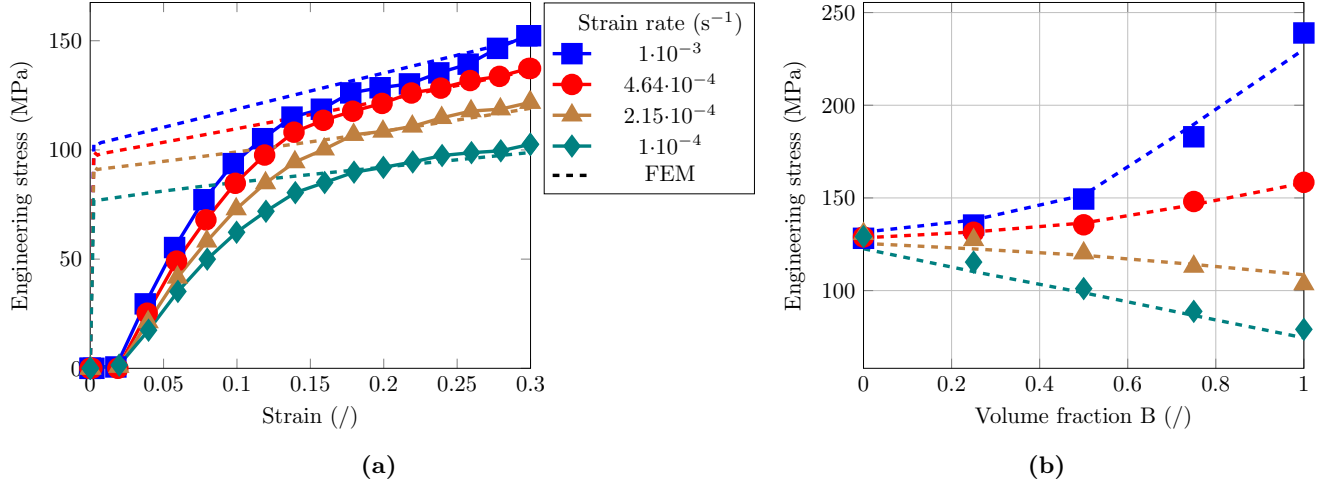


Figure 12.4: Bi-material series configuration. Effect of the strain rate on the flow stress. Numerical reference: FEM simulations. (a) Engineering flow stress versus strain. Fixed volume fraction: 0.5. (b) Non-linear trend of the engineering flow stress, at a strain of 0.3, with the volume fraction.

12.4 Spherical Inclusion Configuration

The deformation of a spherical inclusion in a matrix is qualitatively close to the experimental configuration of interest. Overall, the evolution of the flow stress and the shape factor are correctly captured. A specific section is dedicated to the study of the influence of the discretization on the results. A first approach to the study of the local stress fields is proposed in Appendix A.

A single spherical inclusion of phase B is placed in the center of a phase A cube, with a fixed volume fraction of 20 % of phase B inclusion. Using symmetries, one eighth of the geometry is modeled with the FEM, using $2.1 \cdot 10^3$ nodes. For the full geometry, the ratio DEM particles to FEM nodes would be a little under 3.

Qualitatively, two typical kinematical tendencies of the matrix are displayed in FEM (Figure 12.5), with an intermediary state of homogeneous co-deformation:

- A barrel shape of the sample, when the flow stress of the inclusion is low, at lower strain rates;

- An hourglass shape, when the flow stress of the inclusion is high, at higher strain rates.

In the FEM simulations, the hourglass shape of the matrix is strongly emphasized by the non-sliding interface between phases. While the barrel shape is easily displayed at low strain rates in DEM, the hourglass shape is only clear at higher strain rates, outside of the validity domain studied range. Potential sources are the more permissive contact conditions between phases, the rough discretization (see for example the finer discretization on Figure 12.8) and the lower local stress field seen in the inclusion (refer to Appendix A).

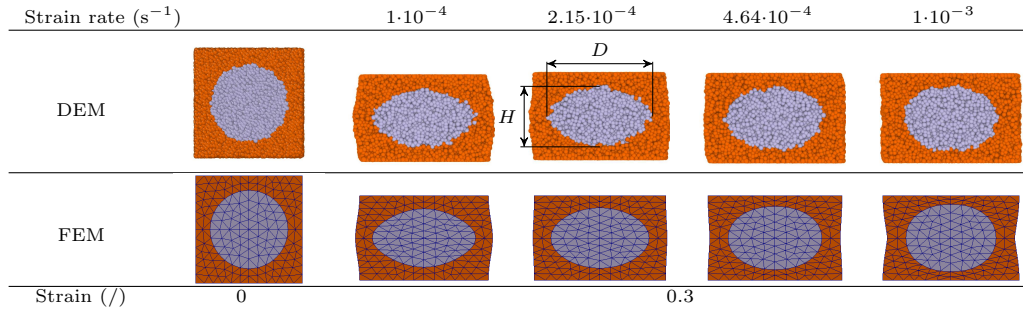


Figure 12.5: Bi-material spherical inclusion (phase B) configuration. Front view, transverse cross-section.

Although we would expect lower stresses with a less constrained system, the flow stress is overestimated (Figure 12.6b), by about 10% on the studied strain rate range, even if the tendency is acceptable after the rise strain (Figure 12.6a).

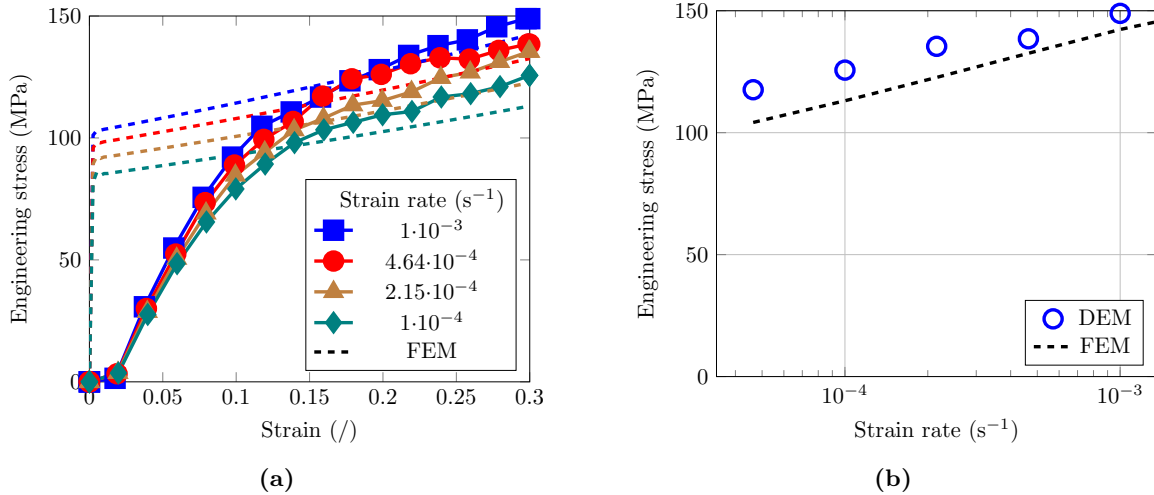


Figure 12.6: Bi-material spherical inclusion configuration. Effect of the strain rate on the flow stress. Numerical reference: FEM simulations. Unique volume fraction of inclusion (phase B): 0.2. (a) Engineering flow stress versus strain. (b) Engineering flow stress versus strain rate, at a strain of 0.3.

To quantitatively compare the models from a kinematical perspective, we study the

macroscopical shape factor S_f of the B inclusion, which is less sensitive than the matrix shape to the interface definition. This factor (Equation 12.3) is the ratio of the inclusion height H , in the compression direction, and diameter D , averaged in all perpendicular directions:

$$S_f = H/D \quad (12.3)$$

For the DEM simulations, this value is approximated computing the shape factor of an equivalent ellipsoid, having the same inertia matrix as the cloud of particles modeling the inclusion. At all strain rates, at the very beginning of the applied strain (Figure 12.7a), the inclusion remains roughly spherical for a few percent of strain, and follow a similar trend as in FEM after a rise strain. In the validity range of the B phase, the final shape factor (Figure 12.7) is underestimated with a relative error of about 5 %, the inclusion deforms more in DEM than in FEM.

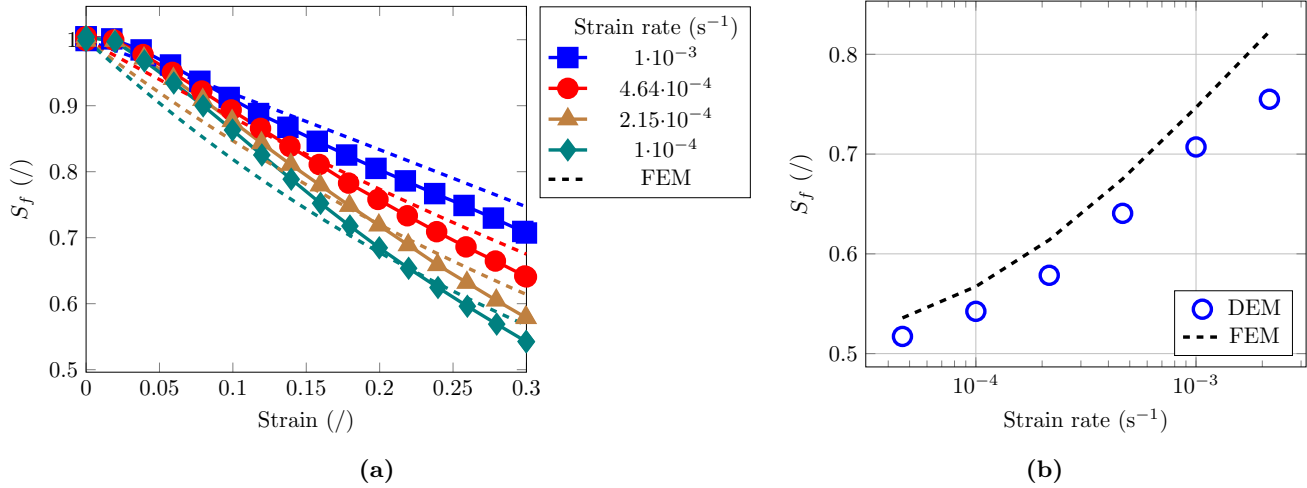


Figure 12.7: Bi-material spherical inclusion configuration. Effect of the strain rate on the shape factor of the inclusion. Numerical reference: FEM simulations. Unique volume fraction of inclusion (phase B): 0.2. (a) Shape factor versus strain. (b) Shape factor versus strain strain rate, at a strain of 0.3.

12.4.1 Spatial Convergence

The effect of the discretization is studied for the case of the unique spherical inclusion. The quantification of the error on the modeled shape factor gives direct guidelines for the discretization of geometries from experimental data.

To evaluate the error on the shape factor depending on the roughness of the discretization, an identical geometry is modeled with packings of various sizes (Figure 12.8). The relative error for the final shape factor is computed using the $1 \cdot 10^6$ particles configuration as reference, where $1 \cdot 10^5$ particles discretize the inclusion. For each size, five random initial packings are tested. The chosen test case is harsh for our model: for the smaller packings, the meshes may interact directly with the particles of the inclusion at the end of the deformation.

A very rough description of the inclusion, with 20 particles for example, remains too inaccurate to catch more than an order of magnitude of the deforming trend (Figure 12.9a), and the initial shape factor is already far from a perfect sphere, with little

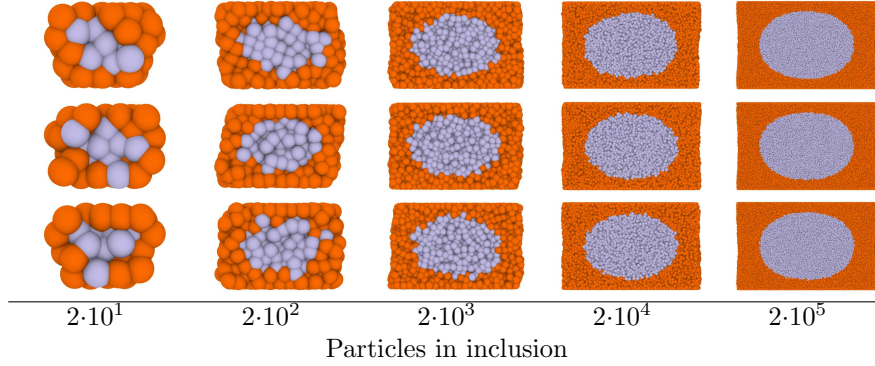


Figure 12.8: Bi-material spherical inclusion configuration. Spatial convergence. Cross-section of three out of the five random configurations tested.

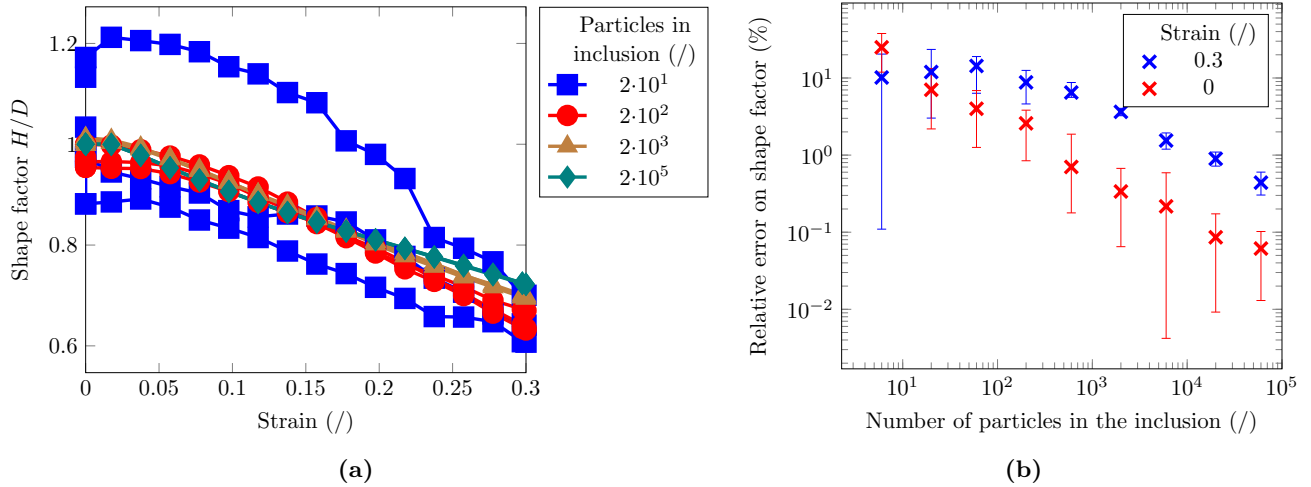


Figure 12.9: Bi-material spherical inclusion configuration. Spatial convergence: effect of the number of particles discretizing the inclusion on the shape factor. (a) Shape factor versus strain. Typical results for three (out of a total five computed) random packings. (b) Error on the shape factor versus the number of particles used to discretize the inclusion. Reference for relative error: $1 \cdot 10^5$ particles used to discretize the inclusion. Minimum, maximum and average error for five random packings.

repeatability. In a realistic context, such a rough discretization can only reasonably be used to capture the position of an inclusion in a composite. With a finer discretization, starting with a few hundred particles, the qualitative trend can be captured and the repeatability improves: it becomes possible to estimate the necessary discretization for an arbitrary precision (Figure 12.9b). The purely geometrical error, on the initial state, is about an order of magnitude smaller than the final error, after compression. For a final error under 10%, more than 200 particles must discretize the inclusion.

Chapter 13

Complex Multi-Material Mesostructure

In Chapter 12, the behavior of our DEM model is compared to numerical reference on very simple geometries. However, our objective is to model complex morphologies of metallic composites.

This chapter applies the model to experimental setups, starting from the discretization of a tomography 3D images (Section 9.3). The chapter is split into two sections:

- Section 13.1 is an illustration of the potentiality of the method on a large data set. The agreement between modeled and observed behaviors is not considered. An initial state is discretized and arbitrarily compressed.
- Section 13.2 applies the methodology to compare the numerical results to *in situ* observations, globally and for local configurations of interest. Numerical and experimental issues are briefly examined.

13.1 Computation on a Full Sample

To test the method on large geometry, a full sample of our composite model material is discretized and compressed. The objective is to assess the numerical scaling to larger models than the previous test cases.

As an illustrative example, the methodology is applied to the 3D mesostructure of a full sample, obtained by X-ray microtomography at the European synchrotron radiation facility (ESRF) (beamline ID19). The studied material is a metallic composite, with a crystalline copper matrix and spheroidal inclusions of amorphous zirconium alloy. The total volume of the sample is approximately 0.5 mm^3 , containing a volume fraction of inclusion of 15 %, with diameters up to a few dozens of micrometers. The voxelized image has a size of $594 \times 591 \times 669$ voxels, with a voxel size of $1.3\text{ }\mu\text{m}$. The purpose of this section is not to compare quantitatively numerical and experimental results, but to underline the potential of the method for large arbitrary data sets.

Starting from a three dimensional voxelized image, the discretization of the geometry has a low algorithmic cost (Section 9.3). The segmented image is used as a mask on a random packing of particles. For each particle, the color of the voxel geometrically corresponding to the center defines the material type: matrix A or inclusion B.

The image is here binarized in two phases and used as a mask on a packing of about $3.36 \cdot 10^6$ particles, with a number of voxels to number of particles ratio of 70. As shown in

Figure 13.1, about 170 physical inclusions are discretized, using between 500 and 5 discrete element particles each. The rough discretization of the smallest inclusions, for example the further left inclusion in Figure 13.2, is not precise enough to allow a strain evaluation, only the inclusion position can be tracked. For the biggest inclusions, estimated numerical error on the shape factor after 0.3 of strain is around 10 % (Figure 12.9b).

The sample is uniaxially compressed up to 0.3 at $1 \cdot 10^{-3} \text{ s}^{-1}$. Local and global illustrations are given in Figures 13.1 and 13.2, with relative displacement and deformation of the inclusions. The computation was run on an Intel Xeon E5520, using 8 processors. The $6 \cdot 10^5$ steps for $3.36 \cdot 10^6$ particles were executed in $8 \cdot 10^5 \text{ s}$, less than 10 days. The computation time is linear with the number of steps and of particles. As long as the load is properly balanced between processors and provided that the geometry of the sub-domains keeps the volume of the communications between processors reasonable, the DEM solver scales properly with the number of processors. On the studied geometries, the roughly cuboidal overall shape of the samples allows simple dynamic balance of the load between processors. The computing time can thus be reliably estimated on a given machine, roughly $3 \cdot 10^{-6}$ cpu second per particle and per time step, for a single processor in the given example.

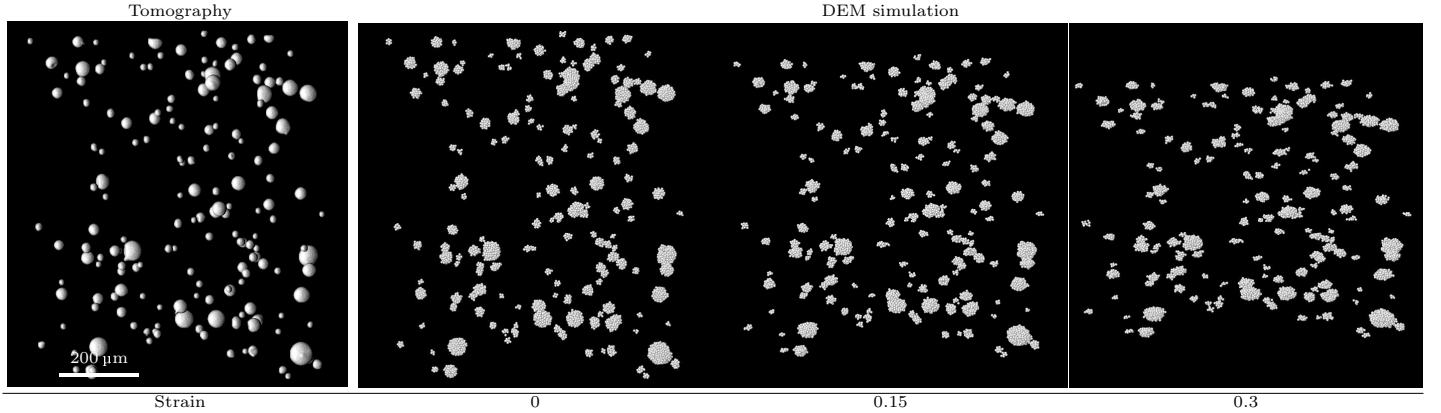


Figure 13.1: Discretization and compression of the full sample. 3D view of the inclusions only, the matrix is hidden. Vertical compression axis.

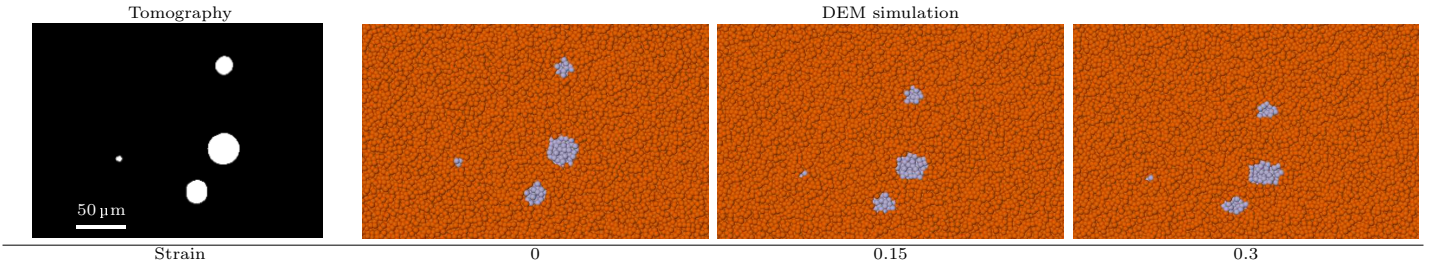


Figure 13.2: Zoom on a local configuration. Discretization and compression. Cross-section of the matrix and the inclusions. Vertical compression axis.

This section illustrated that the proposed methodology can be applied to large arbitrary realistic mesostructure data. The discretization has low algorithmic cost, but the model does not allow yet a simple way to locally adapt the discretization roughness.

The cost of the computation can be reliably estimated as the model does not depend on non-linear resolutions.

13.2 *In Situ* Configurations

The results from the DEM simulations are compared to the temporal evolution of the morphology obtained by in situ X-ray tomography. The discrepancy is stemming from limitations of the model, first of which in terms of strain rate sensitivity, and from the crystallization of the amorphous alloy, delicate to control in this experimental environment.

The maximal strain rate sensitivity that can be modeled with *BILIN* is limited to $M \approx 0.5$, on a limited strain rate range. Classical sensitivities for metal creep, typically 0.2 [117], can thus readily be modeled. However, the amorphous phase of our model material (Section 2.3) exhibits unusually high strain rate sensitivities, up to Newtonian behavior $M = 1$ (Figure 2.7b on page 26).

Experimentally, in the temperature and strain rate range of interest $M_{\text{exp}} \approx 0.73$ (Figure 13.3a). Numerically, the amorphous phase is modeled in DEM with an underestimated $M_{\text{num}} \approx 0.49$. FEM simulations were run on the single inclusion test case, to quantify the effect of this underestimation (Figure 13.3b). The variation of the shape factor S_f of the inclusion with the strain rate is indeed influenced by M . By design, at the center of the range, the difference is very limited. A typical relative difference of a few percent can be observed at the extrema of the considered strain rate range. However, the order of magnitude of this introduced error is reasonable: from a strictly numerical point of view, the typical FEM to DEM error is similar or higher; experimentally, it is acceptable with respect to the numerous uncertainties. Qualitatively, the behavior of the model is governed by the association of two phases with respectively high and low strain rate sensitivities. Within a restricted strain rate range, the exact values of strain rate sensitivities are of secondary order¹.

To limit the uncertainty on the prescribed strain rate, the nominal strain rate experimentally applied to the samples is not directly used. Several pairs of particles, above and below the region of interest, are tracked (Figure 13.4). The overall estimated strain rate is polynomially fitted (second order typically suited well this series of data) and used as input in the DEM simulations. The error introduced by using rigid planar meshes, over-constraining the system, proved to be of secondary order.

A preliminary series of simulation is run applying the overall macroscopic strain rate, polynomially fitted for full experimental samples, to a single numerical inclusion. The configuration is very similar to the test case studied in Section 12.4, using $1 \cdot 10^4$ particles and a volume fraction for the inclusion of 0.2. The expected relative error of the shape factor with respect to a spatially converged packing is around 1 % (Figure 12.9b). The numerical shape factor was compared to the average measured shape factors for all inclusions (Figure 13.5).

It must be emphasized that the modeled boundary condition may be quite alien to the studied system. Our test case leaves free lateral boundaries. In the sample, in the neighborhood of a physical inclusion, the lateral flow of the matrix is more constrained. As it will be discussed, the introduced discrepancy is of secondary order with respect to the encountered issues.

Two nominal strain rates, $5 \cdot 10^{-4}$ and $2.5 \cdot 10^{-4} \text{ s}^{-1}$, are tested and experimentally have very similar overall behavior. A large discrepancy can be observed between experiments and simulations after a strain of 0.2 – 0.3. At both nominal strain rates, the deformation

¹ See also Section 12.1 for a discussion regarding the low strain rate sensitivity phase.

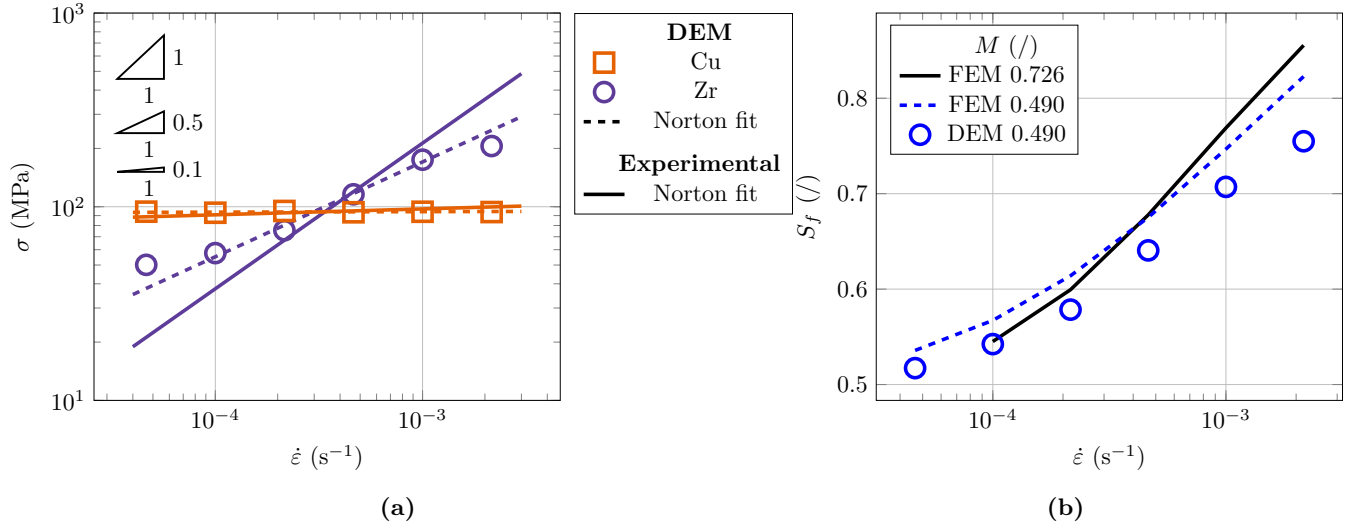


Figure 13.3: Quantification of the error introduced by the underestimation of the strain rate sensitivity. (a) Norton law fit for numerical and experimental data. (b) Shape factor of an initially spherical inclusion after a strain of 0.3 at various strain rates. Effect of the strain rate sensitivity for the FEM simulation.

of the inclusion is largely overestimated by the simulations. The discrepancy potentially introduced by the free lateral boundary conditions would on the contrary let the matrix flow freely around the inclusion. We should overestimate the shape factor. In addition, the experimental deformation seems to slow down after a strain of 0.2 – 0.3.

A probably dominant effect in this discrepancy stems from the crystallization of the amorphous alloys of the inclusion. Uniaxial compression test, carried-out on a sample of the amorphous alloy up to the beginning of the crystallization (Figure 13.6), can provide a first set of indications. On Figure 13.6, the macroscopic mechanical effect of the crystallization (see also Figure 2.4a on page 23) can be detected 1·10³ s after the introduction of the sample in the furnace, with the increase of the flow stress.

Despite these preliminary data, the effective kinetics of the crystallization is not well understood² and the effects of the thermomechanical elaboration process (Section 2.3) are

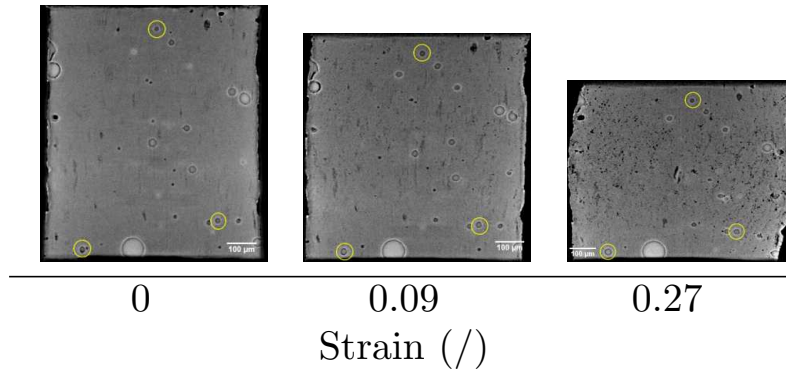


Figure 13.4: Estimation of the strain rate from the distance between tracked inclusions.

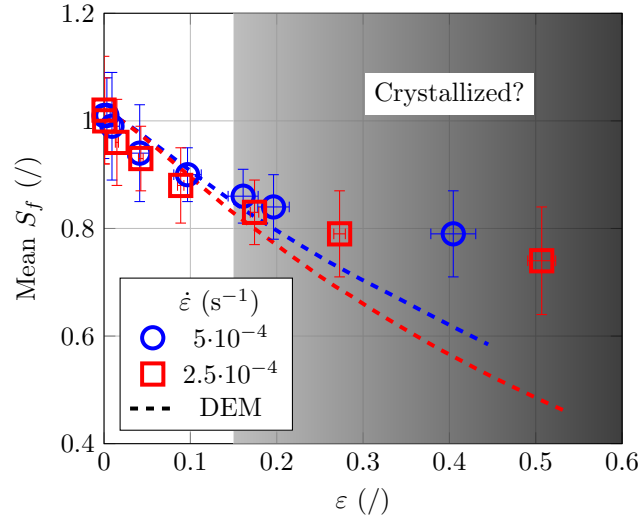


Figure 13.5: Comparison of the experimental average shape factor for all inclusions in a full sample to DEM simulation of a single inclusion. The crystallization of the amorphous phase increases the flow stress of the inclusion during the experiments.

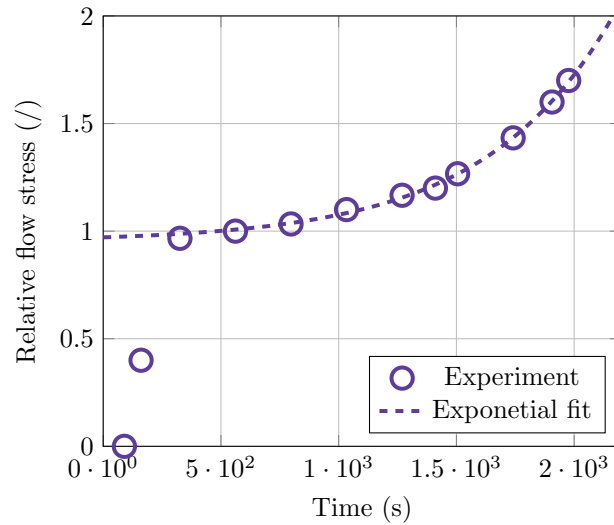


Figure 13.6: Mechanical effect of the crystallization of the amorphous phase, at 400 °C and $4.24 \cdot 10^{-4} \text{ s}^{-1}$. The effective compression starts at 90 s after the sample is introduced in the furnace. Exponential fit of the temporal evolution of the stress, normalized by the reference flow stress (Equation 13.1).

not known. The data from Figure 13.6 are obtained on a millimetric sample, obtained by casting. The alloy in our composite is obtained by atomization and is then hot co-extruded with the copper powder. In addition, the set-up for *in situ* measurements at the ESRF imposes at least several minutes between the introduction of the sample in the furnace and the beginning of the test, typically from 5 to 10 min (Section 2.4).

A naive attempt to account for these numerous unknowns is implemented by exponentially fitting the experimental crystallization (Figure 13.6) and using the function (Equation 13.1) to implement an explicit time dependence of the numerical parameters of the amorphous phase.

$$\sigma/\sigma_{\text{ref}} = a \cdot \exp(b \cdot t + c) + d \quad (13.1)$$

The fitting parameters used here are $(a, b, c, d) \approx (2.91 \cdot 10^{-4}, 1.82 \cdot 10^{-3}, 4.24, 9.51 \cdot 10^{-1})$. A somewhat arbitrary initial offset of 600s is used in the simulation, to account for the effects of the set-up time and the elaboration process. The flexibility of the DEM frameworks allows to implement such models with relative ease.

As-is, the effect of this correction attempt is not sufficient to capture the tendency observed on the experimental results (Figure 13.7). The temporal evolution of the numerical parameters limits the modeled deformation with respect to the “plain” simulation. However the deformation is still too high for a satisfactory description of the experimental observation. At this stage, too little is known to propose a more detailed approximation. More in-depth comparison between the *in situ* data and the simulations are thus hindered. The end of this section will thus be limited to a qualitative test of the methodology on typical geometrical configurations of interest, without using the temporal correction. For three chosen configurations, a locally estimated polynomial evolution of the strain rate is applied to the discretized geometry.

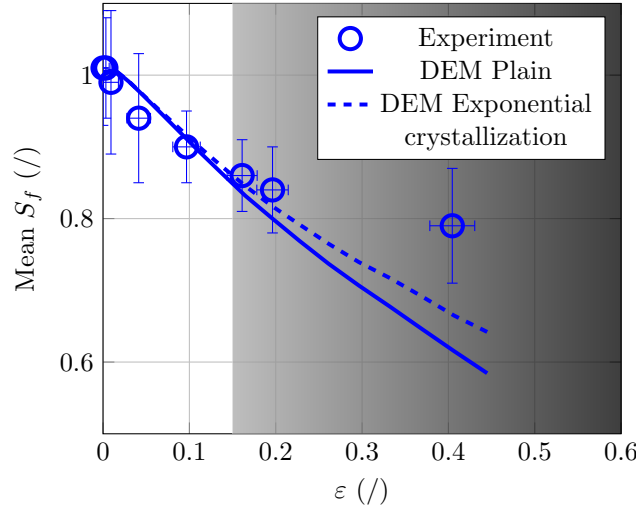


Figure 13.7: Attempt to apply the temporal evolution measured on Figure 13.6 to the evolution of the shape factor of a single inclusion. In the simulation taking into account a crystallization effect, the initial time offset of 10 min is considered. Nominal strain rate $5 \cdot 10^{-4} \text{ s}^{-1}$.

A first example (Figure 13.8a) of local configuration is the deformation of a large inclusion surrounded by smaller inclusions, to allow an estimation of a local flow of the

² The reproducibility of the test can be questioned, and the fully crystallized mechanical behavior has not been tested. For some samples, the increase of the flow stress is more progressive and starts earlier.

matrix. Mainly due to the crystallization, the evolution of the shape factor of the main inclusion is largely overestimated in the simulation with respect to the experimental measurements (Figure 13.8b), the quantitative analysis is thus not of interest. Quantitatively, it can be observed that the deformation trend is also quite distinct. In the simulation, the roughly homogeneous deformation leads to a limited relative motion of the particles. Experimentally, the small inclusions tend to flow around the central inclusion, with a large displacement perpendicularly to the compression axis.

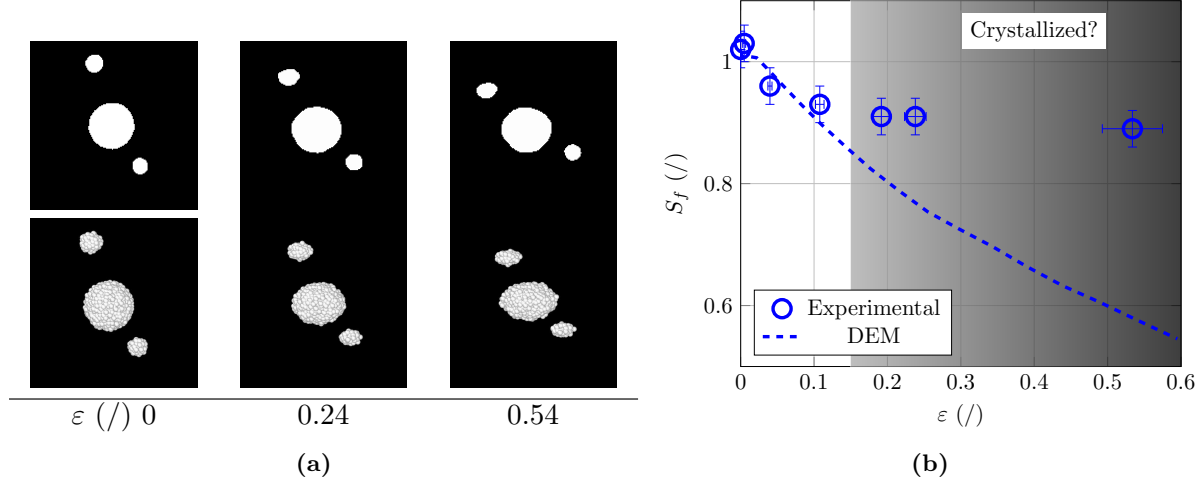


Figure 13.8: Simulation of a local configuration at a nominal strain rate of $5 \cdot 10^{-4} \text{ s}^{-1}$. Initial diameter of the central inclusion $55 \mu\text{m}$. (a) Cross-section of the segmented tomography volume. Hidden matrix for the DEM simulation. (b) Evolution of the shape factor S_f of the central inclusion with the strain.

A second configuration is the behavior of two inclusions of similar size, initially close to one another (Figure 13.9). An interesting feature of this test case is the numerical behavior at this rough discretization. This initial matrix between the inclusion is only modeled by a pair of particles. As isolated particles cannot display the expected, the effect of the thinnest layer of matrix numerically vanishes. Depending on the dominant physical phenomenon driving the experiment, for example if thin film with high mechanical properties is formed, such a numerical behavior can be quite misleading. In addition, as contact phenomena are not included, this example is a limit case of the model.

A last example, typically observed in experimental context, is the presence of defects in the structure of the inclusion. The presence of a large hole in the inclusion (Figure 13.10) is a consequence of the atomization process and influences their overall behavior. Although the behavior seems qualitatively satisfactory, the model reaches two major limitations. Firstly, a further deformation would lead to the self-contact of the interface of the hole, not used in this simulation. Secondly, the model *BILIN* is only suitable for compressive loads and the stress state is probably more complex on a “porous” mesostructure, potentially with local tensile loads.

The handling of contact events of tensile loads will both be examined in the next part.

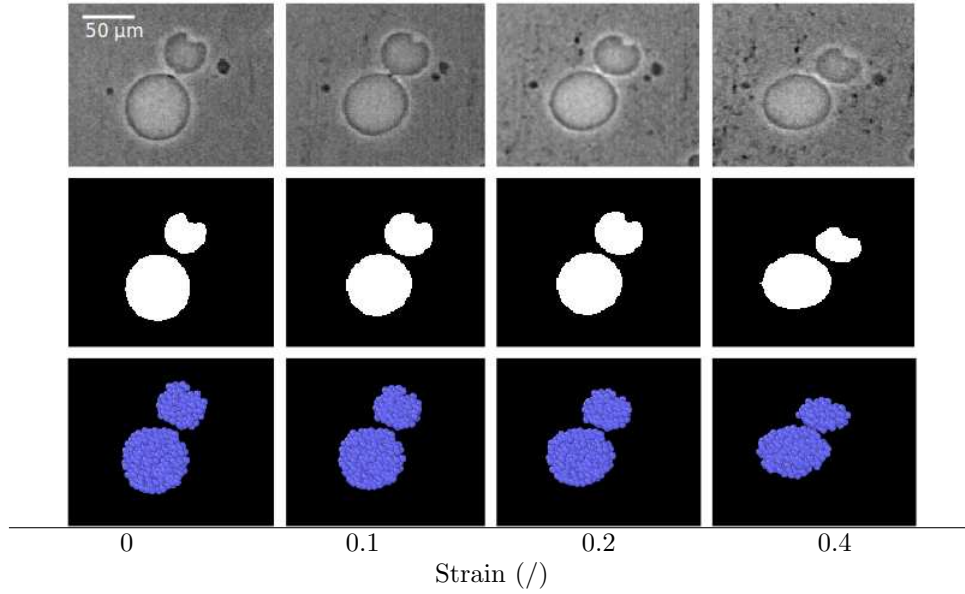


Figure 13.9: Example of a pair of close inclusions [35, p.19]. Cross-sections of the reconstructed and segmented tomography volume. Hidden matrix for the DEM simulation.

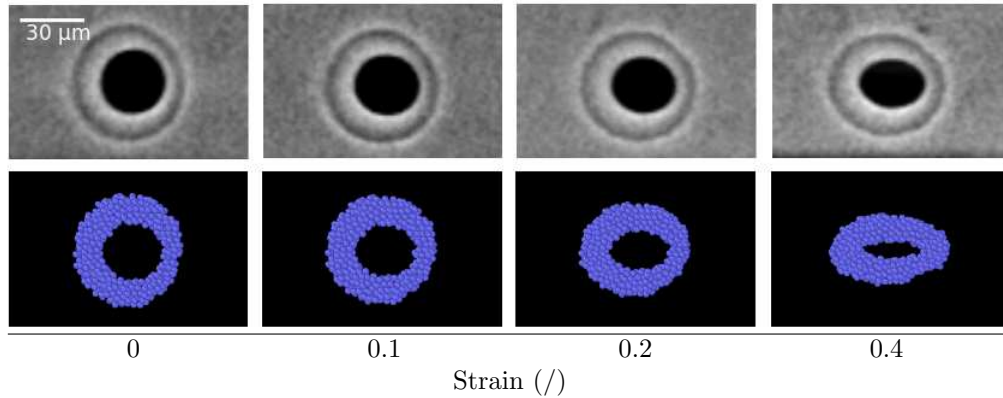


Figure 13.10: Example of a hollow inclusion [35, p.19]. Cross-sections of the reconstructed tomography volume. Slice of the inclusion, with hidden matrix, for the DEM simulation.

Part V

Tension-Compression of “Porous” Material

14 Dense Material	151
14.1 Interaction Law Choice	151
14.2 Macroscopic Behavior	152
14.3 Influence of the Number of Particles	158
15 Self-Contact Detection	161
15.1 Threshold Choice	161
15.2 Compression-Tension of a Spherical Pore	166
16 Complex “Porous” Mesostructures	169
16.1 Casting Pores in Aluminum Alloy	169
16.2 Aluminum Open Cell Foam	173
16.2.1 Qualitative Behavior	174
16.2.2 Macroscopic Stress	178
16.2.2.1 Effect of the Relative Density	180
16.2.2.2 Effect of the Strain Rate	182
16.2.3 Local Field	185

In Part III, the conceptual and algorithmic principles of the developed method were described. In Part IV, focused on loads dominated by compression, the methods were applied and tested on dense bi-materials. The interaction law used was the model *BILIN*, whose application is limited to compression only.

Part V extends the model to less restrictive loads, using the model *TRILIN*. Potential uses for the detection of self-contact events are illustrated on “porous” geometries, as opposed to the dense configurations studied previously. The part is split into three chapters:

- Chapter 14 is about the behavior of the interaction law *TRILIN* under tensile and compressive loads.
- Chapter 15 describes the self-contact detection procedure and its effect on simple geometries.
- Chapter 16 illustrates the methodology with “porous” mesostructures, obtained by X-ray tomography.

Highlights - Part V

Tension-Compression of “Porous” Material

- The *TRILIN* interaction law can cope with compressive and tensile loads.

The overall behavior of the packing is satisfactory regarding the flow stress and the volume conservation. The symmetry error between compressive and tensile behaviors is around 20 %. The necking and the rupture under tensile load are displayed but not controlled.

- The self-contact detection algorithm allows the tracking of interface interactions.

The closure and re-opening of pore is natively displayed in uniaxial compression-tension tests. The parameters of the self-contact detection are tuned to choose an acceptable compromise between contradictory objectives.

- The proposed framework allows a flexible and controlled handling of topological events.

The implementation of a healing time of the interfaces illustrates the potential of the method to control the evolution of the topology of the sample. Given a locally computable metric, arbitrary behavior can be developed with ease.

- The behavior of the model is illustrated using complex mesostructures, obtained by X-ray tomography: casting pores and a low relative density foam.

Qualitatively, the self-contact algorithm allows the tracking and the re-opening of numerous interface interactions. Some macroscopic metrics are derived for the tested geometries. Good agreement is obtained for the flow stress of the foam. Local deformation mechanisms can be observed and studied.

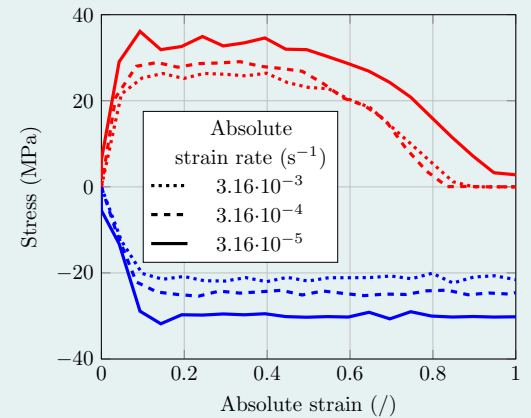


Figure 14.4

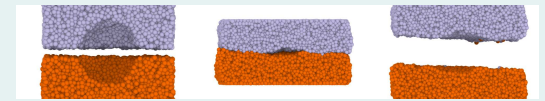


Figure 15.5

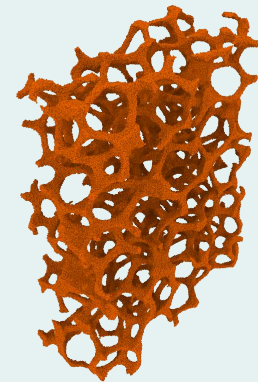


Figure 16.5



Figure 16.10

Chapter 14

Dense Material

In Part IV, the interaction law *BILIN* was only suitable for compressive loads. This model is thus not suitable for porous mesostructures.

In this chapter the behavior of the interaction law *TRILIN* (introduced in Section 9.1) is described. This interaction law is able to cope with compressive and tensile loads and is used throughout Part V. The chapter is divided into three sections:

- Section 14.1 briefly presents the modeling choices for the interaction law *TRILIN*.
- Section 14.2 describes the behavior of a dense packing, without any self-contact detection algorithm.
- Section 14.3 studies the numerical effect of the number of particles on the behavior of a packing.

14.1 Interaction Law Choice

The model TRILIN is attractive-repulsive. The repulsive forces are elastic linear. The attractive forces are linear up to a threshold and are only activated for tensile motions of the pair. In the objective of modeling a constant strain rate sensitivity over large ranges of strain rate, a limited strain rate sensitivity is chosen.

The model *BILIN* was chosen in Part IV for its simplicity. The fixed ratio of stiffnesses $k_{\text{rep}}/k_{\text{att}} = 10$ could only reasonably cope with compressive loads. The model *TRILIN* is designed to be more generic, coping with tension and compression. For the two configurations, a similar absolute macroscopic flow stress is sought for. Keeping an elementary bi-linear interaction law, as the *BILIN* model, no satisfactory modification of the radii or the stiffness ratio was found. Indeed, the attractive stiffness k_{att} must increase to provide a more cohesive behavior and the crown radius r_{crown} must provide a wide enough geometrical range of interaction: the attractive forces become excessive.

Among potential force profile, a threshold on the attractive force was chosen for the model *TRILIN* (Figure 14.1). We have no claim whatsoever that this choice is optimal in any respect, this configuration was only the first to provide a sufficiently satisfactory behavior to be used as proof of concept.

In Part IV, it proved impractical to use time-converged model. The required simulation time being excessive, the time step was treated as a mere numerical parameter (Section 8.3.2). Accepting fully this assumption, the time step for *TRILIN* was chosen quite large with respect to the natural period: $\Delta t/t_0 = 10^{-1}$.

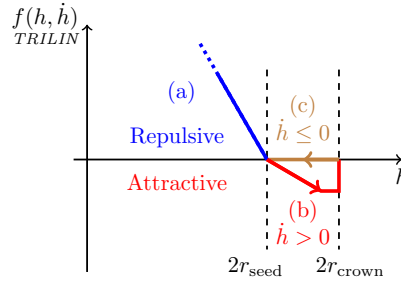


Figure 14.1: Pairwise interaction law *TRILIN* used in this part. Force f versus distance h . Attractive to repulsive force ratio and radii to scale. Classical DEM conventions are applied: repulsive forces are positive. Classical mechanical convention will be applied in the discussion: tensile stress will be positive. Refer to Section 9.1 for the full description of the law.

The linearity of the interaction force with respect to the introduced parameters is respected. The flow stress, for a given kinematic behavior can thus be arbitrarily chosen at fixed k_{rep}/m ratio. Although it has not been investigated, the increase of the strain rate sensitivity at higher strain rate seems to be displayed¹. It is thus probably possible to apply the calibration procedure applied in Part IV to tune the strain rate sensitivity. A new calibration chart (Figure 11.3 on page 122) would have to be computed, the shapes of the interaction laws and the ratio $t_0/\Delta t$ being distinct. A similar behavior is to be expected: a large plastic-like domain and a narrow tunable viscoplastic domain.

However, the co-deformation configurations studied in Part IV had, by design, a limited strain rate dispersion in a given sample. Although the limited validity range of strain rate was a handicap, the model could be used to study configurations of interest, where both phases deform notably. By contrast, in the objective of modeling porous material, the strain rate field is necessarily very heterogeneous. In the cases of a truss-like system for example, strains may be concentrated at the joints.

The natural period of the model is thus fixed ($t_0 = 1$ s) in the objective of mimicking a low strain rate sensitivity, valid over a wide range of strain rates. With the chosen time step ($\Delta t = 10^{-1}$ s), the average strain rate sensitivity over the strain rate range $10^{-5} - 10^{-2} \text{ s}^{-1}$ is $M = 7.14 \cdot 10^{-2}$. The validity range of the domain in strain rate can be arbitrarily shifted by modifying the natural period and the time step at fixed $t_0/\Delta t$ ratio. The mimicked stress level used is loosely inspired from typical flow stresses of Al-7075 at 400 °C, typically 30 MPa at 10^{-3} s^{-1} (Figure 2.14 on page 31). The chosen repulsive stiffness in the pair interaction of the particles $k_{\text{rep}} = 1.42 \cdot 10^9 \text{ } \mu\text{N} \cdot \text{mm}^{-1}$ leads to a macroscopic stress level $K = 49.1 \text{ MPa} \cdot \text{s}^M$ in the Norton approximation.

14.2 Macroscopic Behavior

With the interaction law TRILIN, the particles of the packing can collectively cope with tensile and compressive load. Overall, the stress-strain behavior and volume conservation is satisfactory. At large strain, tensile loads induce the necking and the rupture of the sample. Some parasitic numerical crystallization is observed and is promoted by the planar meshes.

Under compressive load, pushed by the moving meshes, the particles using the model *TRILIN*

¹This point will be discussed in Section 14.2. Refer also to Figure 14.5b.

rearrange collectively to cope with strain (Figure 14.2). Compared to the behavior of the model *BILIN* (Figure 11.9 on page 128), the deformation is less homogeneous: the lateral free surfaces are not as regular and smooth². The overall reorganization is however effective.

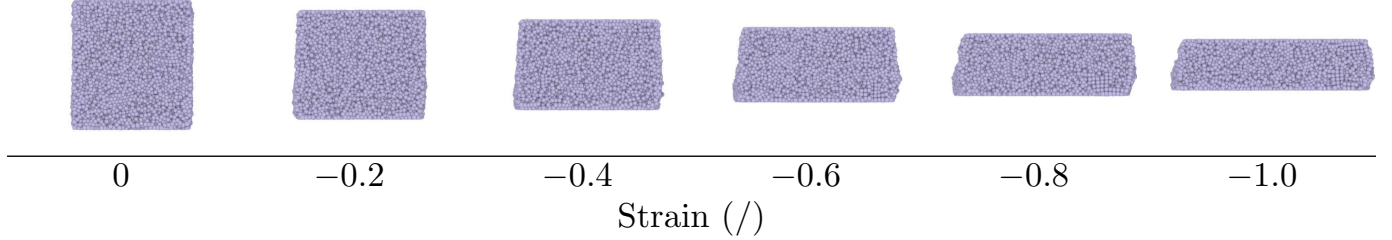


Figure 14.2: Cross-section for uniaxial compression. Packing of $5 \cdot 10^4$ particles at $-3.16 \cdot 10^{-4} \text{ s}^{-1}$. See also Figure 14.6.

Under tensile load (Figure 14.3), the deformation of the sample is at first roughly homogeneous. On the presented configuration, qualitatively, a necking behavior then appears ($\varepsilon \approx 0.4$) and develops until the final rupture ($\varepsilon \approx 0.9$). Incidentally, this behavior somewhat mimics the ductile necking of metallic polycrystals [78, p.121].

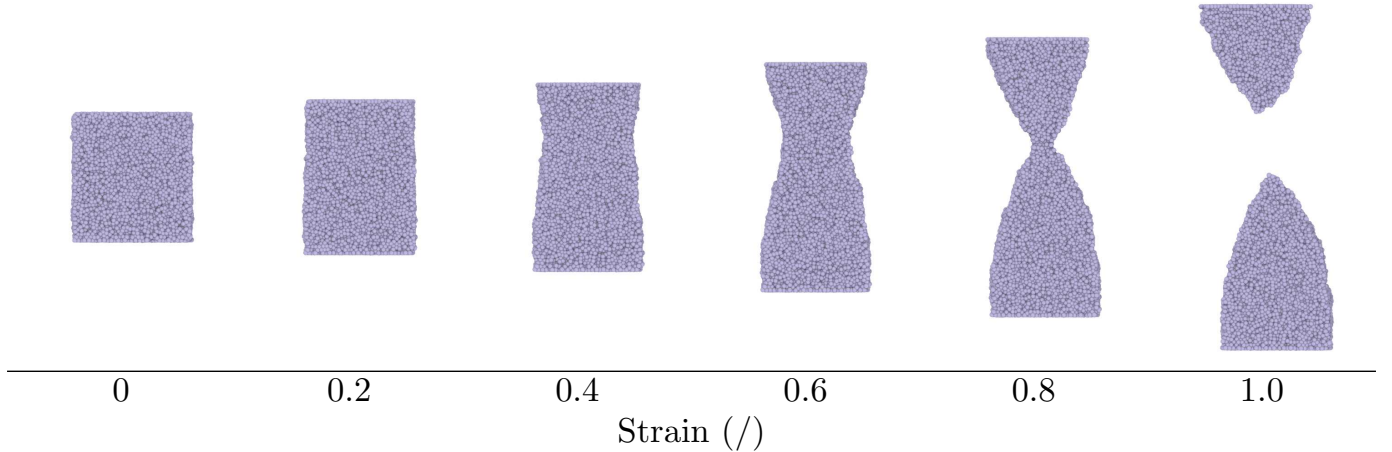


Figure 14.3: Cross-section for uniaxial tension. Packing of $5 \cdot 10^4$ particles at $3.16 \cdot 10^{-4} \text{ s}^{-1}$. See also Figure 14.6.

The strain-stress behavior (Figure 14.4) is coherent with the observed tendencies:

- Under compressive load (negative stress), the stress is stable up to large strain.
- Under tensile load (positive stress), the stress³ is stable up to strains of $0.4 - 0.5$, where it drops under the influence of the necking. Once the sample is broken in two parts, the stress is zero.

For both compression and tension, a transient regime is observed at the beginning of the test. The observed width in strain is roughly $0.1 - 0.15$, which is about twice shorter than

²This effect will be examined in the end of the section.

for the model *BILIN* (Figure 11.10a on page 129). At larger strain rate, the oscillations of the stress are of greater magnitude. As expected, the absolute flow stress increases with the absolute strain rate.

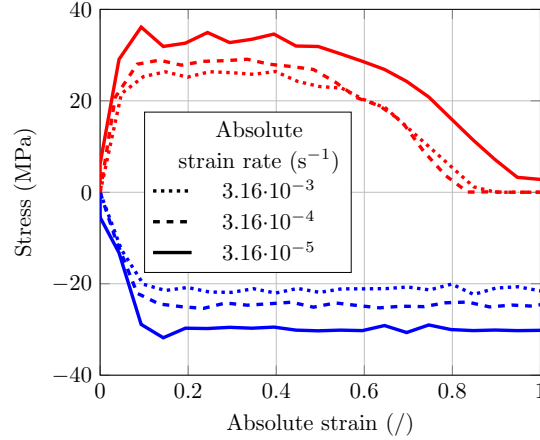


Figure 14.4: Stress-strain behavior for *TRILIN*, using $5 \cdot 10^4$ particles. Under tensile load: stress drop at necking and null stress after rupture.

To compute the approximation by a Norton law of the macroscopic behavior (Figure 14.5a), the flow stress at a given strain rate is computed as the average stress between strains of 0.21 and 0.35. Both for tension and compression, the variation of the flow stress is limited over a wide range of strain rate. The flow stress varies approximately by a factor 1.7 for three decades of strain rate. The compressive behavior is smoother and the Norton approximation is rougher for the tensile loads. The order of magnitude of the flow stress is similar, the tension stress being at most 20 % larger than compression stress.

Qualitatively, the sensitivity-strain rate behavior of *TRILIN* seems comparable to *BILIN* (Figure 14.5b). The strain rate sensitivity is stable at lower strain rates and increases rapidly at higher strain rates. A calibration procedure, finding a link between strain rate, natural period and strain rate sensitivity, seems possible. The behavior was however not quantitatively investigated. Firstly because the foreseen application – porous mesostructures – implies large heterogeneities of the strain rate field. Therefore, a viscoplastic behavior valid only on a limited strain rate range is not appropriate. Secondly, the choice of a large time step would induce an excessive motion of the meshes at each step for higher strain rates, where the model should exhibit higher strain rate sensitivity. The model *TRILIN* is thus focused at a limited strain rate sensitivity over a wide strain rate range.

The minor variation of the strain rate sensitivity is sufficient to influence the necking behavior⁴. At higher strain rates, the increase in the strain rate sensitivity tends to stabilize the necking. Qualitatively comparing the sample deformed at distinct strain rates (Figure 14.6), this effect is seen at $3.16 \cdot 10^{-3} \text{ s}^{-1}$. At a strain of 0.8, the necking is less pronounced and at 1.0, the sample is not separated into two parts yet⁵. On the stress-strain curves (Figure 14.4) the total rupture ($\sigma = 0$) of the sample occurs before a strain of 0.9 for $\dot{\epsilon} \leq 3.16 \cdot 10^{-4} \text{ s}^{-1}$.

³As in Part IV, the macroscopic stress is computed using an estimation of the cross-section based on the current height of the sample and its initial volume.

⁴Refer to Section 14.3 regarding the size effects.

⁵At a strain rate of 10^{-2} s^{-1} , the stabilizing effect is extremely efficient and the necking is very limited

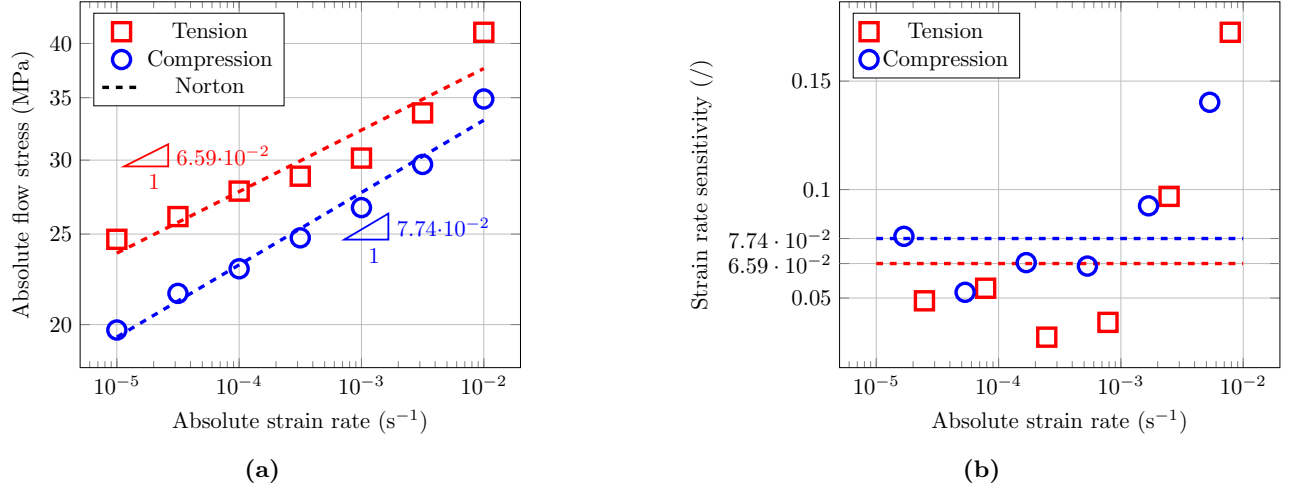


Figure 14.5: Norton law approximation for *TRILIN*. (a) Stress flow versus strain rate. Norton law parameters for the average behavior: $M = 7.14 \cdot 10^{-2}$, $K = 49.1 \text{ MPa} \cdot s^M$. (b) Sensitivity versus strain rate.

As a side note, the influence of the parameter X_{wall} on the necking behavior is not negligible. This parameter (refer to Section 9.2) is a multiplicative factor used to increase the mesh/particles interaction forces. It is an arbitrary work-around to effectively apply tensile loads. Practically, it drives the ease with which a particle interacting with the mesh will be able to leave it. The chosen value ($X_{\text{wall}} = 3$) tends to promote a rupture in the middle of the sample: the forces necessary to develop a necking are lower than the force to pull particles apart from the meshes.

The overall macroscopic equilibrium of the packings (Figure 14.7) is computed from the difference of the total forces acting on a fixed and on a mobile mesh. Under compressive load, at comparable natural period and strain rate, the relative error is a factor three larger than for the model *BILIN* (Figure 11.4 on page 124). This seems to be a direct effect of the larger time step chosen. To respect an equilibrium error under $10^{-1} \%$, as in Part IV, the absolute strain rate would need to be limited to roughly $2 \cdot 10^{-4} s^{-1}$.

Qualitatively, on dense samples, well balanced collective rearrangements seem correct up to $10^{-2} s^{-1}$. However, parasitic dynamic effects can be observed on very porous geometries at $3.16 \cdot 10^{-3} s^{-1}$. In practice, the simulations were run up to $10^{-3} s^{-1}$, corresponding roughly to an error of 1%. This choice, slightly less conservative than in Part IV, is merely a computational convenience.

A potential issue arising with the model *TRILIN* is a tendency to numerical crystallization. Indeed, as the repulsive and attractive stiffnesses are more balanced than in *BILIN*, the distance between the particles is more strictly constrained. Although the overall behavior is satisfactory, the deformation of the lateral free surfaces are not as regular and smooth as for *BILIN* (Figure 11.9 on page 128). This overall behavior mainly stems from the numerical partial and local crystallization. Some groups of particles (typically 6^3) arrange locally on a lattice and tend to follow block-wise motion with the strain. On Figure 14.8, this effect can clearly be seen on the right side of the sample.

This phenomenon is stronger at lower strain rates, as particles have more time to organize in an energy minimizing configuration. On Figure 14.6, no crystallization is

at a strain of 1.0. Such a configuration was not actually used in the simulations due to a poor equilibrium respect.

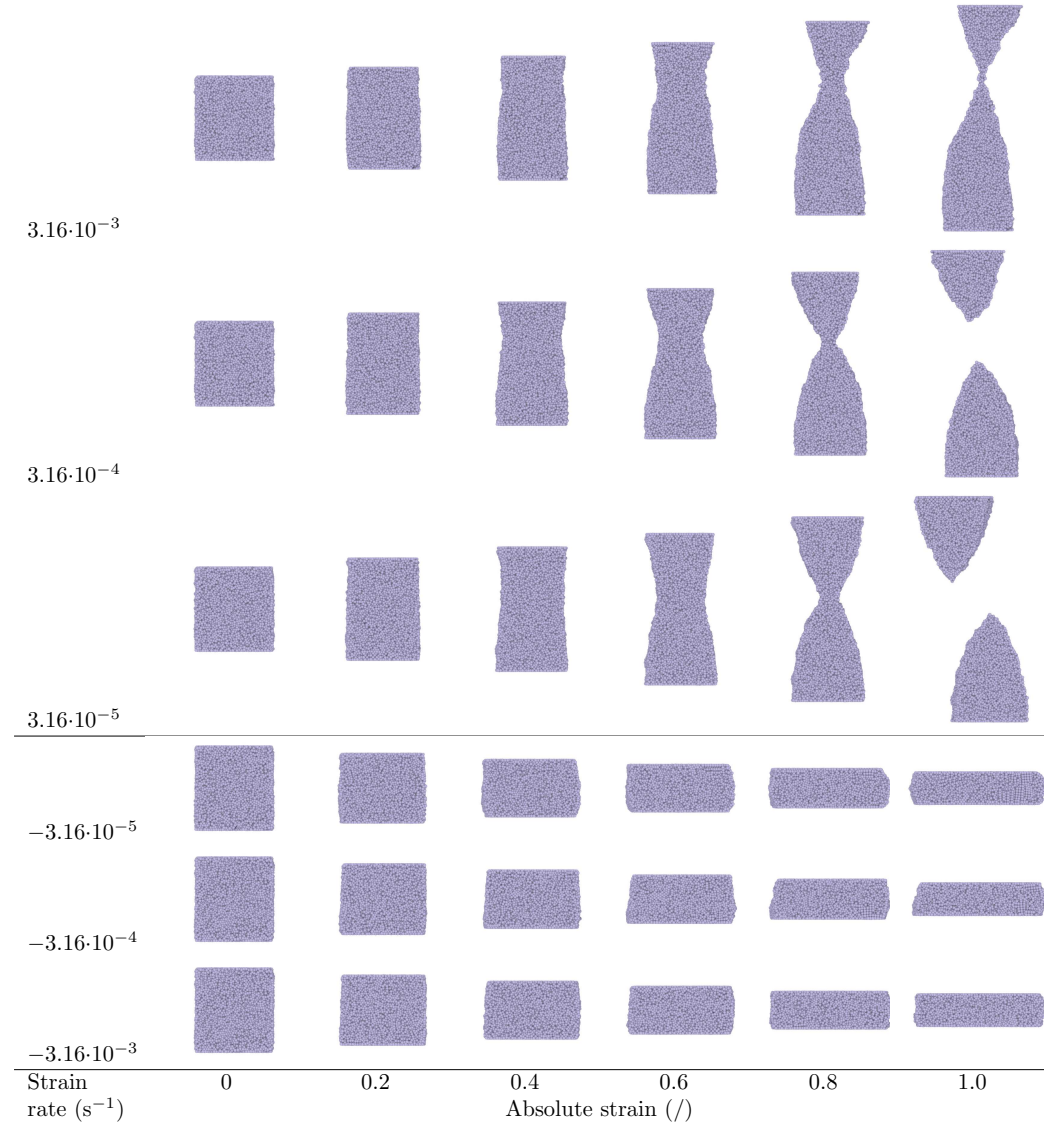


Figure 14.6: Cross-section for uniaxial test, tensile and compressive load. Packing of $5 \cdot 10^4$ particles.

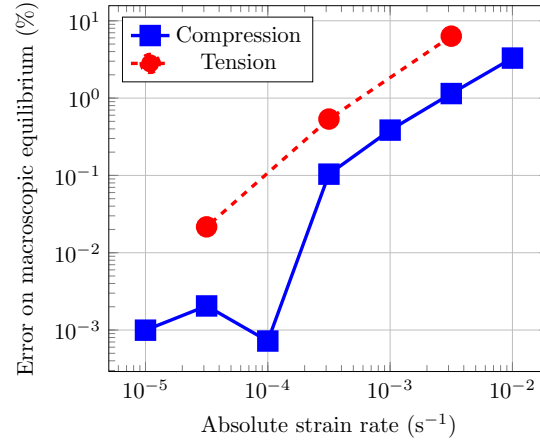


Figure 14.7: Relative error of the macroscopic equilibrium versus strain rate.

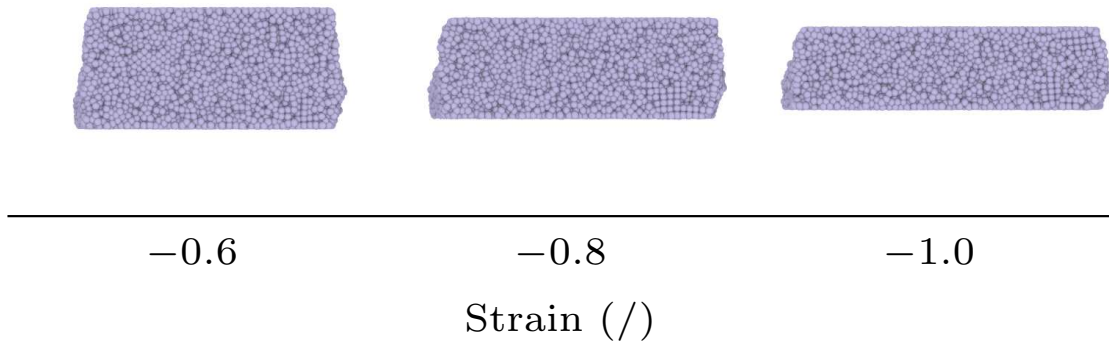


Figure 14.8: Cross-section for uniaxial compression. Packing of $5 \cdot 10^4$ particles at $-3.16 \cdot 10^{-4} s^{-1}$. See also Figures 14.2 and 14.6.

observed at $-3.16 \cdot 10^{-3} \text{ s}^{-1}$. The phenomenon is also stronger for compressive loads, where particles are closer to the meshes, as the planar meshes strongly favor this behavior. On Figure 14.6, crystallization effects under tensile load are mostly limited to the close neighborhood of the meshes.

Although the numerical crystallization seems sufficiently limited to mimic the overall rearrangement, it is a limitation for the quality of the model. Preferential strain zone form between the crystallized block and the study of local fields is thus severely limited. Numerous work-around can be imagined to improve the model, first of which the introduction of a little dispersion in the radii of the particles.

14.3 Influence of the Number of Particles

The flow stress can be captured with a limited number of particles in the packing. The error with respect to a spatially converged state is evaluated. The number of particles also influences the rupture qualitatively. This effect has not been investigated.

The choice of the spatial discretization introduces a purely numerical length scale. The number of particles in the packing thus influences the strain at which the necking is initiated and the final rupture profile (Figure 14.9). This can be understood as the influence of the ratio between the size of the sample and the size of the typical defect, corresponding roughly to the dimension of the particles.

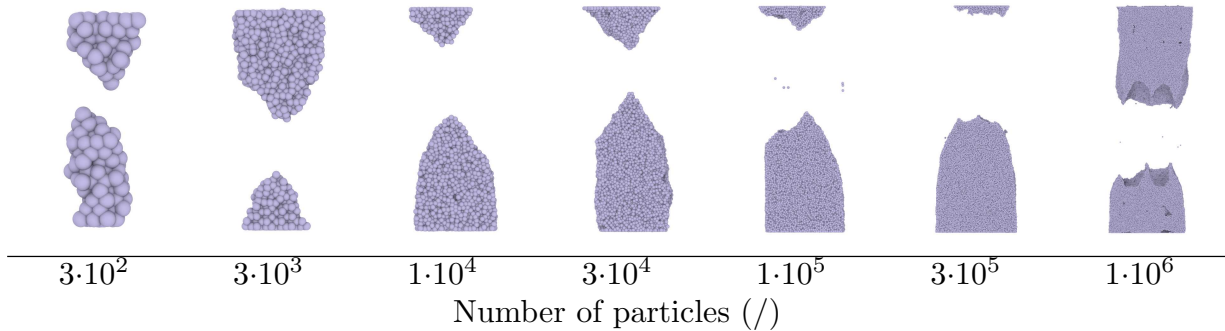


Figure 14.9: Cross-section for uniaxial tension test. Influence of number of particles on the rupture profile under tensile load at 10^{-3} s^{-1} . A size effect can be observed, with an approximate qualitative threshold between $3 \cdot 10^4$ and $1 \cdot 10^5$ particles.

For packings of limited size, typically under $5 \cdot 10^4$ particles (Figure 14.9), the necking is more or less axisymmetric and the effective resisting cross-section progressively loses its thickness and vanishes. For larger packing, multiple defects seem to nucleate in the necking zone and coalescing until the final rupture, somewhat like in the canonical dimple rupture of some ductile alloys.

This dependency of the rupture behavior on the size of the packing⁶ is not designed nor controlled to have physical sense. Although the proposed modeling approach may well be of interest to model ductile rupture, we focus here on the homogeneous response of the material. Further developments would be necessary to specifically study rupture or necking.

⁶Refer also to the effect of the parameter X_{wall} used with the planar meshes (Section 14.2).

A common stress-strain tendency is displayed for packings larger than a few thousand particles (Figure 14.10a): the transient regime is similar and the flow stress value is roughly captured even with a limited number of particles (e.g. $5 \cdot 10^2$).

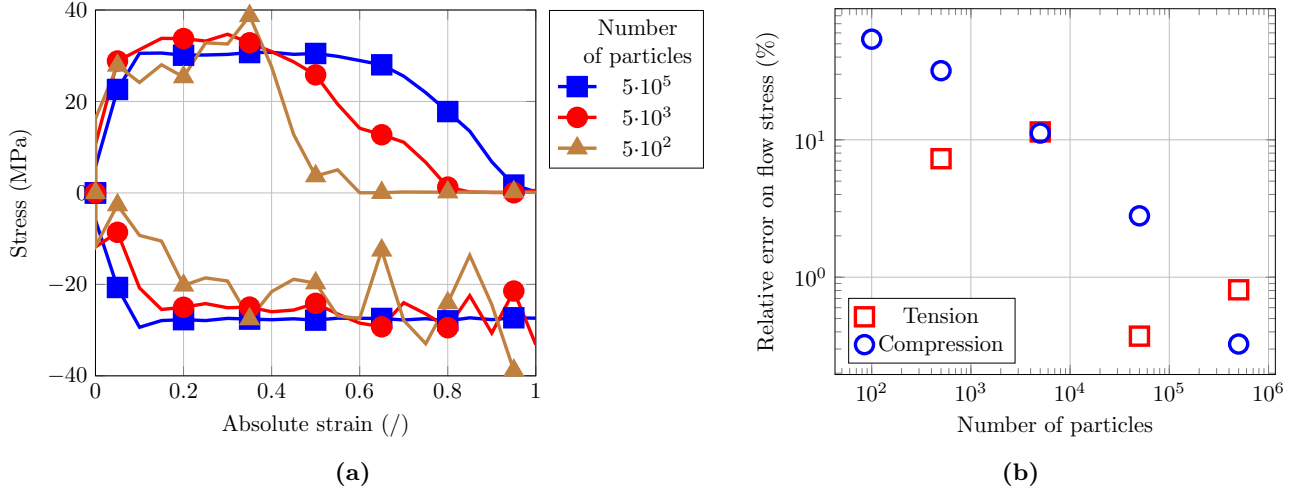


Figure 14.10: Effect of the geometrical discretization on the flow stress of the packing at a strain rate of $3.16 \cdot 10^{-4} \text{ s}^{-1}$. (a) Comparative tendencies of the stress-strain curves. (b) Relative error on the flow stress with respect to a packing of $3 \cdot 10^6$ particles.

However, with a very small packing of $5 \cdot 10^2$ particles, the crystallization under compressive load becomes excessive. This effect is caused by the harsh boundary condition imposed by the two close meshes. A small sample thus fully crystallizes and jumps from a stable crystallized configuration to another. This parasitic numerical crystallization of the packing is clearly displayed under compressive load by the flow stress oscillation at larger strain. Under tensile load, the effect is not seen on the strain-stress curve or on a cross-section of the sample (see for example the even smaller packing of $3 \cdot 10^2$ particles on Figure 14.9). With $5 \cdot 10^3$ particles, the oscillation of the flow stress due to crystallization starts at larger strain (≈ 0.7).

The size dependency of the necking and the rupture influences the strain-stress behavior under tensile load. On Figure 14.10a the strain at which the necking starts (i.e. the drop of the flow stress) increases with the size of the packing. The final rupture of the sample respectively occurs at a strain of 0.6, 0.8 and 1.0 for packings of $5 \cdot 10^2$, $5 \cdot 10^3$ and $5 \cdot 10^5$ particles.

In spite of the numerical artifacts of crystallization (mostly under compression) and size effect (mostly under tension), the spatial convergence of the flow stress (Figure 14.10b) is similar to the model *BILIN* (Figure 11.7b on page 127). The error on the flow stress relative to a considered converged packing of $3 \cdot 10^6$ particles is around 10 and 3 % respectively for packings of $5 \cdot 10^3$ and $5 \cdot 10^4$ particles.

Along with the strain-stress behavior, the volume conservation is a key objective for our model. The volume variation after relaxation (Figure 14.11) is typically around a few percent, which is comparable to the low strain rate sensitivity phase A in the model *BILIN* (Figure 11.8 on page 128). From the relaxed initial state, after the typical transient regime, the volume decreases by a few percent with compression. The larger discrepancy for compression of the samples of $5 \cdot 10^2$ and $5 \cdot 10^3$ particles stems from the crystallization, notably reducing the volume.

The tension increases the volume only by $\approx 1\%$. At the initiation of the necking,

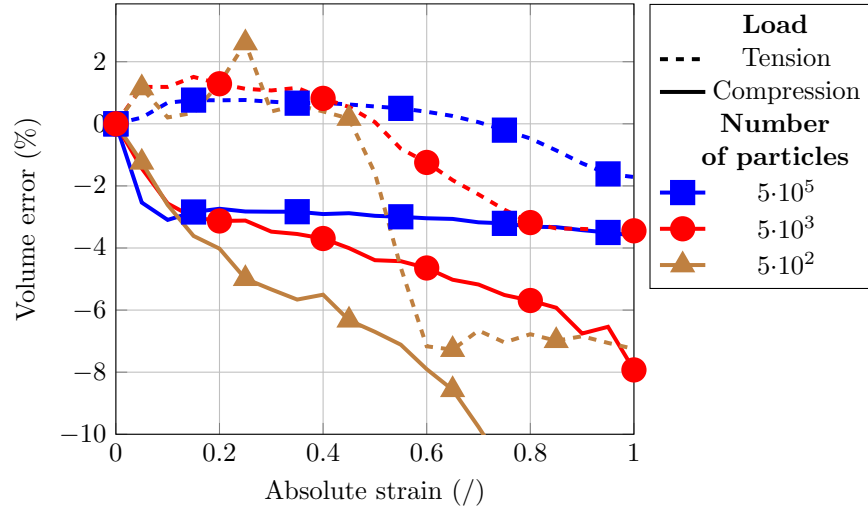


Figure 14.11: Evolution of the relative error on the volume versus strain. Influence of the packing size at $3.16 \cdot 10^{-4} \text{ s}^{-1}$.

the volume starts to drop toward lower value than the initial state at rupture. After the rupture, the volume varies insignificantly as the sample is separated into two parts.

Chapter 15

Self-Contact Detection

In the proposed model, inelastic strains are mimicked by neighbor changes in a packing. A dedicated algorithm (introduced in Section 9.5) must identify the interaction of the modeled interfaces and more specifically the self-contact events.

This chapter investigates the behavior of this self-contact detection algorithm for simple geometries. The chapter is split into two sections:

- Section 15.1 concerns the methodology to choose the threshold parameters of the algorithm.
- Section 15.2 applies the algorithm to a single spherical pore in a cubic domain. An example of controlled topological change is illustrated.

15.1 Threshold Choice

The detection of self-contact events is based on a local metric. A test case, using two blocks of material, is designed to tune the algorithm parameters. A compromise has to be chosen between contradictory types of errors. The chosen set of parameters only slightly modifies the behavior of a dense sample and misses few self-contacts.

To detect self-contact events (Figure 15.1), an “outward vector” \bar{n} is computed for each particle from the position of its neighbors. When two particles meet for the first time, the relative orientations and the magnitudes of the vectors are compared to classify the pair as “internal” or “interface”. Three thresholds are used: two on the angles (α_{ij} , α_{en}) and one on the magnitudes (N_{mag}).

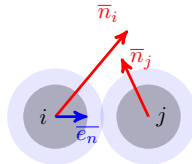


Figure 15.1: Self-contact detection algorithm. Refer to Section 9.5 for the full description of the algorithm.

Based on the chosen self-contact detection algorithm, a test case is designed to choose the threshold parameters. The principle of the test case is to perform a compression-

tension test starting with two distinct aggregates of particles (Figure 15.2). Each aggregate, chosen initially cuboidal, is explicitly labeled and tracked. Thus, when a new pair is created, the “internal” or “interface” status can be verified checking if the particles come from identical or distinct initial aggregates. The two aggregates are successively crushed onto one another and pulled apart, while all pair creations are checked.

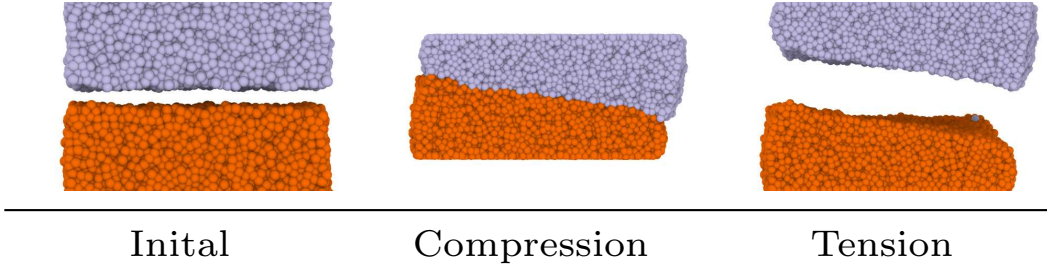


Figure 15.2: Test case to quantify the errors of the self-contact detection algorithm, used to choose the self-contact threshold parameters. Two cuboids are crushed and pulled apart. Cross-section for an arbitrary set of parameters.

Two metrics are chosen to quantify the quality of a set of parameters:

- The rate of error at the interface between the two blocks. The “interface error” is the ratio between pairs with particles from distinct aggregates mistakenly considered as “internal” and the total number of new pairs considered as “interface”.
- The rate of error inside each block. The “internal error” is the ratio between pairs with particles from identical aggregates mistakenly considered as “interface” and the total number of new pairs considered as “internal”.

It must also be considered that the denomination of these errors may be somewhat misleading and only refers to events at the particle level. For example, an “internal” error can occur at the interface and numerous “internal” errors can lead to lose the modeled interface by opening numerical ones.

The two error rates are not of equal weight. The creation of new “internal” pairs are much more frequent, thus causing a high number of errors if the two rates are balanced. However, the detection of “interface” pairs is more critical: a few percent of “interface” error is already too poor a description for our purpose, while 10 % of “internal” error can be acceptable. Indeed, particles typically have numerous “internal” neighbors and few “interface” neighbors, an error on an “internal” neighbor is more easily counterbalanced by the others: the general cohesion is sufficient.

As a guide to find a sensible threshold parameter N_{mag} , the distribution of the magnitudes n of the outward vectors is bimodal (Figure 15.3), with a transition between 2 and 3 mm. The first guess on the angle thresholds α_{e_n} and α_{ij} was found by simple trial-and-error.

A full factorial design of experiment is then executed, sweeping the domain $(N_{\text{mag}}, \alpha_{e_n}, \alpha_{ij}) = ([1.5, 2.6], [50, 105], [35, 85])$. One compression-tension test is run for each set of parameters. The explored domain favors configurations with low interface error, more critical for our purpose. As the two chosen error metrics are contradictory, the results sketch a Pareto front (Figure 15.4).

The contradictory behavior of the two error types leads to a necessary compromise in the choice of the interaction parameters. As a rough guide for the choice of a specific

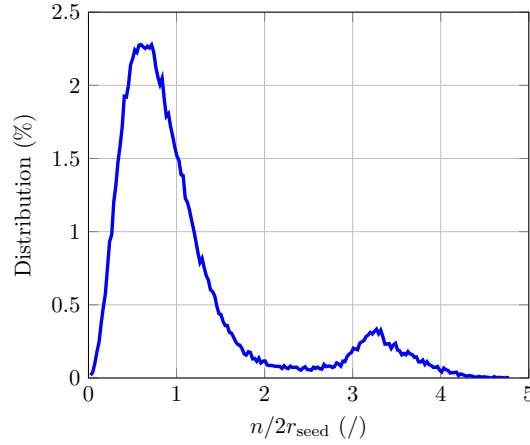


Figure 15.3: Typical distribution of the magnitude n of the outward vector. Initial state of a cuboid with a spherical hole, corresponding to Figure 15.8.

configuration on the front, the following maximal errors provide a “visually” acceptable overall behavior:

- “Interface” error $< 1\%$.
- “Internal” error $< 10\%$.

Within the range, decreasing the “interface” error to $\approx 10^{-1}\%$ notably improves the tracking of the interface. This implies an increase of the “internal” error from 2 to 3%, only very marginally modifying the overall dense behavior. The used set of parameters is thus chosen on the identified Pareto front at $(N_{\text{mag}}, \alpha_{e_n}, \alpha_{ij}) = (2.3 \text{ mm}, 80^\circ, 65^\circ)$ (crossed-out on Figure 15.4).

The chosen set of parameters leads to an “interface” error of $\approx 10^{-1}\%$. To illustrate the effects of this metric, the test case is repeated on various initial geometries (Figure 15.5): cuboids, portions of spheres, cones, cylinders with perpendicular axis and cuboids with a portion of spherical hole. Typically, a few particles are carried by the wrong aggregate after the compression-tension test. An attentive reader may spot some on the cuboid, the cylinder or the spherical hole configurations. Note that this naive metric was not used to choose the parameters and is dependent, among other factors, on the number of particles modeling the interfaces.

The chosen set of parameters leads to an “internal” error of $\approx 3\%$. The stress-strain profile for a dense sample, with and without using the self-contact detection, is shown on Figure 15.6. On this Figure, the introduced error is represented by the discrepancy between the lines (without self-contact) and the round markers (with self-contact). The stress is little affected by the chosen self-contact thresholds, including the necking behavior under tensile motion. The average flow stress computed between strains of 0.21 and 0.35 varies by less than 1%.

As a side note regarding the Pareto front, the dominant parameters to quantitatively tune the metrics of interest seem to be N_{mag} and α_{e_n} . This purely quantitative approach hides many qualitative tendencies that were not studied in depth. To improve the behavior of the algorithm, it seems equivocated to attempt a heavy quantitative optimization procedure, based on the two chosen quantitative metrics. The understanding of distinct typical configurations may be of interest to design a more comprehensive detection algorithm.

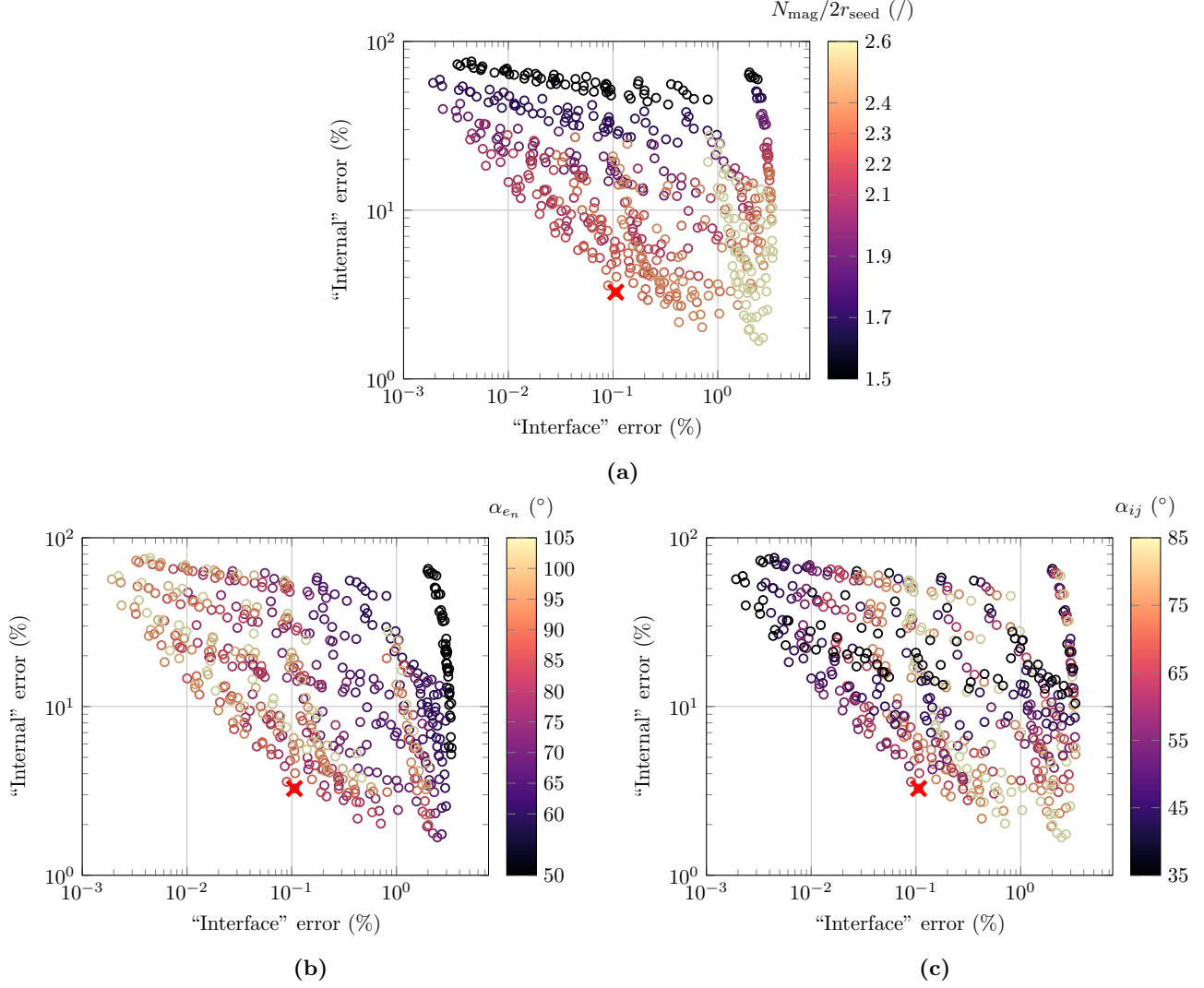


Figure 15.4: Pareto front of the two error types in the classification of new interactions between particles. The chosen configuration is crossed-out. (a) Magnitude threshold N_{mag} . (b) Angle threshold $\cos \alpha_{e_n}$. (c) Angle threshold $\cos \alpha_{ij}$.

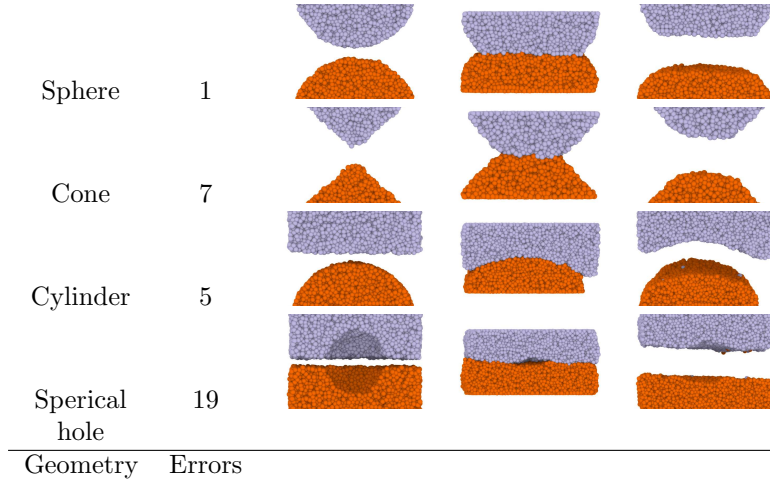


Figure 15.5: Test on various geometries of the chosen thresholds for self-contact detection. Indication of the final total number of particles carried by the wrong aggregate. Note that this metric was not used to choose the parameters. Cross-section.

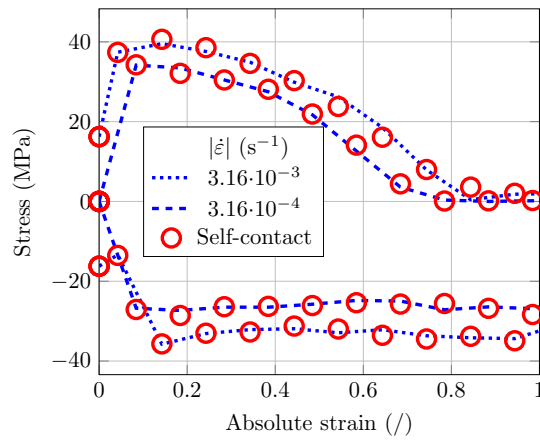


Figure 15.6: Effect of the self-contact detection on the strain-stress behavior of a dense packing.

15.2 Compression-Tension of a Spherical Pore

A cube with a single spherical pore is compressed up to the mechanical closure of the void. The interface is properly tracked. A tensile load is then applied and the pore re-opens on the cohesionless interface. A “healing” behavior is added, the interface thus becomes cohesive after a given time of contact. The pore can be partially or fully closed depending on the chosen healing time.

A simple test case for the self-contact detection algorithm is based on a cube of side $2a$ with a centered spherical hole of radius r . On a uniaxial compression-tension test, the pore is expected to be closed, while keeping track of the interface and to re-open under the tensile load.

To visualize this behavior, a handy representation is a slice of width $2r_{\text{crown}}$ across the pore displaying the outward vectors only (Figure 15.7). On the chosen example, a relative pore radius $r/a = 0.15$.

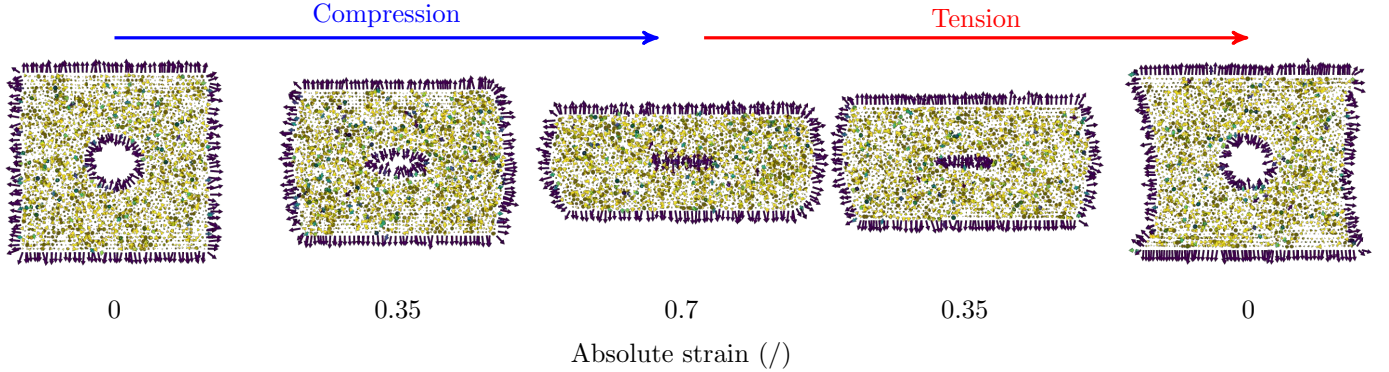


Figure 15.7: Compression-tension of a spherical pore $r/a = 0.15$. Slice of the sample and visualization of the outward vectors.

The behavior under tensile load and more specifically the strain localization around the pore is influenced by the numerical parameter X_{wall} . This arbitrary parameter, used to apply tensile loads with meshes, can emphasize or minimize the phenomenon by allowing the particles to leave the mesh more or less easily. This behavior is not intended to represent a physical phenomenon.

It must be remembered at this stage that as we work in the discrete element method (DEM) framework, arbitrary topological events can be implemented with ease. To illustrate such behaviors, a healing time is implemented in the interaction law (Figure 15.8). For each “interface” pair, a counter sums the time elapsed from the creation of the pair. After a threshold, the healing time, the status of the pair is switched back to a normal “internal” interaction. The pore is fully closed if the healing time is negligible. For intermediary healing time (like 100s Figure 15.8), the pore re-opens but is significantly smaller.

The engineering stress for a compression-tension test is compared with and without instantaneous healing for two sizes of pore (Figure 15.9). The expected tendency of a larger peak tension stress with instantaneous healing is observed. For relatively small pores, the difference is not large as the effective variation of resisting tensile section due to the pore is limited after a large compressive strain.

The ability to track or “weld” interfaces is a strength of the method. Our model used with a simple healing time model is a first approach to mimic the pore mechanical

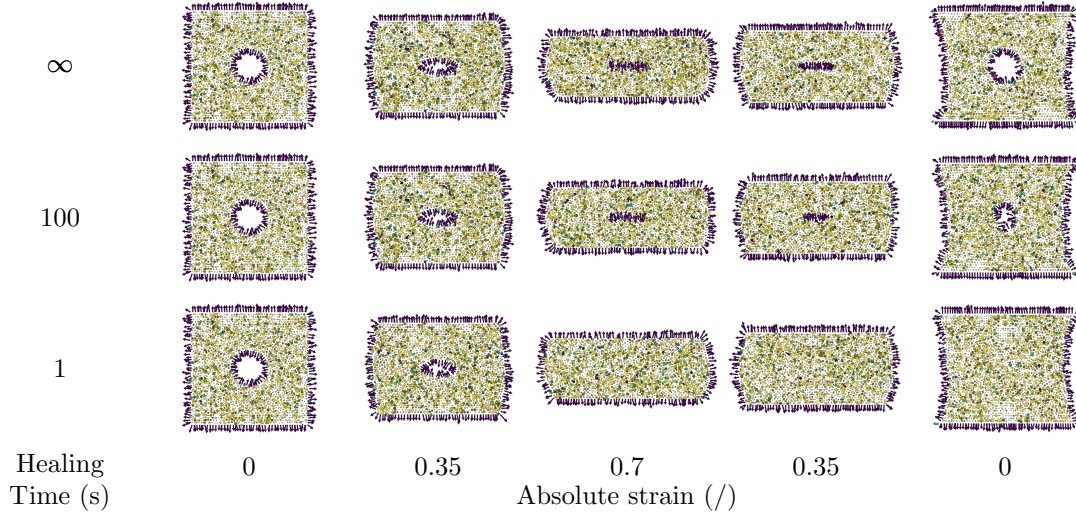


Figure 15.8: Compression-tension of a spherical pore $r/a = 0.15$. Effect of a healing time of the “interface” pairs. Slice and visualization of the outward vectors.

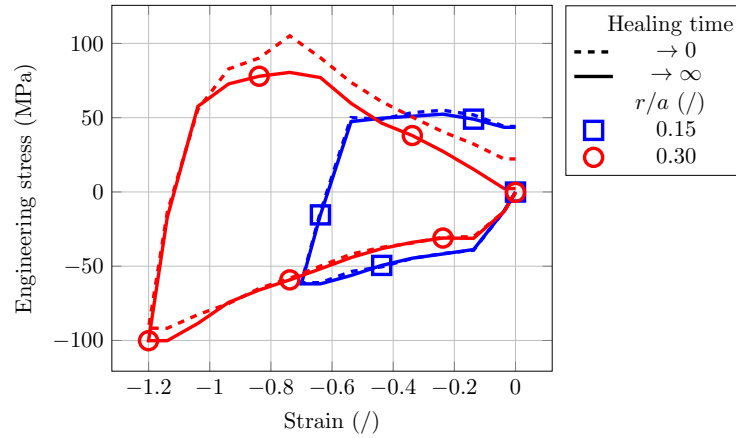


Figure 15.9

closure and healing. Qualitatively, the closure and the welding starting at the edges and progressing toward the center of the inclusion [239, p.101] is observed. Arbitrary behavior can be implemented as long as a driving metric can be locally computed during the simulation run.

Chapter 16

Complex “Porous” Mesostructures

Reminding the overall objective of this PhD, we are attempting to design a method to describe finite inelastic strain in continuous media. The method must also handle the detection, the displacement and the interaction of numerous self-contacts.

This last chapter illustrates the potential of the method on complex geometries obtained by X-ray tomography (refer to Section 9.3). “Porous” is understood here in a loose sense, merely as materials presenting holes, in opposition to dense materials. The chapter is divided into two distinct applications:

- Section 16.1 briefly deals with the mechanical closure and re-opening of casting pores, under compressive and tensile loads.
- Section 16.2 investigates the compression of low relative density foam. The compression is performed up to large strain and the local deformation mechanisms are looked into. Quantitative macroscopic metrics are computed.

16.1 Casting Pores in Aluminum Alloy

The morphology of casting pores in aluminum, obtained by X-ray tomography, is discretized. A compression-tension test is applied: the pores are mechanically closed and re-opened. In spite of the large applied strain and the tortuous geometry, the interfaces are tracked. At large tensile strain, a coalescence phenomenon is observed.

The pores shown on Figure 16.1a are formed during the casting of an aluminum alloy. These pores are considered as defects and limit the mechanical properties of the material [239, p.95], for example leading to fatigue rupture under cyclic load. In subsequent steps of elaboration, the pores can be closed and “welded” by thermomechanical processes as hot rolling, improving the mechanical properties. The phenomenon is classically divided into two steps: the purely mechanical closure of the pores [198] and the growth of a cohesive interface. We will in this section focus on mechanical effects.

The initial image has a resolution of $2.4\mu\text{m}/\text{pixel}$ and a random packing of roughly $5 \cdot 10^5$ particles is used. The chosen ratio $\text{pixel}/r_{\text{seed}} = 2$ is coherent with the discretization algorithm, For a finer discretization, a less crude reconstruction than an elementary box filter (Section 9.3) could be used. The pertinence of using the center of the particles can also be discussed.



Figure 16.1: Discretization of a set of porosities. (a) 3D reconstruction of the tomography. The largest pore is roughly $100\text{ }\mu\text{m}$ wide. (b) Mesh reconstruction on the particles with large outward vector magnitude n inside the sample.

The reconstructed meshed pores on the DEM model are slightly smaller than the pores on the 3D image. During the relaxation procedure, after particles are removed using the image as a mask, the tensile forces in the packing retrieve the global equilibrium resulting in a contraction of the pore. This test case is specifically sensitive to that effect as a few particles are removed in a large packing. To improve the geometrical description, an iterative procedure may be of interest, repeatedly removing particles after a relaxation procedure. During the iterations, it may be useful to start with smaller pores and progressively increase the size toward the final targeted shape.

The objective of the simulation is to compress the sample containing pores up to their mechanical closure. The model must not lose track of the interface. In a second stage, a tensile load is applied on the compressed sample to pull it back to its initial height. The pores must re-open on the tracked interfaces.

A visualization method of the pores must be proposed as our method is fully implicit: no explicit “interface” object that could be shown is defined. The visualization proposed here relies on a mesh reconstruction on the particles with an outward vector magnitude $n > 2.3\text{ mm}$ (Figure 16.2). The particles from the exterior of the domain are manually removed. All particles are overprinted with a strong transparency to represent the packing. Some noise resulting from this visualization procedure can be seen on Figure 16.2, for example the small and sharp mesh in the lower part of the initial state.

Little holes are appearing on the reconstructed mesh under tensile load. This “sieve” aspect is a reconstruction artifact, stemming from the increase of the total surface of the interfaces: particles are migrating toward the free surface. As an arbitrary threshold on n selects the particles for the mesh reconstruction, some migrating particles are missed.

Under compressive load (Figure 16.2), the volume of the porosity effectively reduces, until all interfaces are in contact. The interface and their interaction are effectively tracked, in spite of the large prescribed compressive strain. The interfaces that come in contact early actually undergo a large strain while being closed.

The pores re-open during the tensile phase and are not lost during the whole compression-tension process. The pores also coalesce, from a strain of 0.3 to finally form a single big pore in the final state. This complex topological change seems to be described with ease by the method. It must however be remembered that the necking and rupture phenomena are influenced by size effects. At this stage, the behavior is thus illustrative of the potential of the model but not necessarily representative of a physical event.

On the shown example, the discretization is too rough to describe properly the deformation of the smallest pore (nicely rounded and near the center on the initial state). The

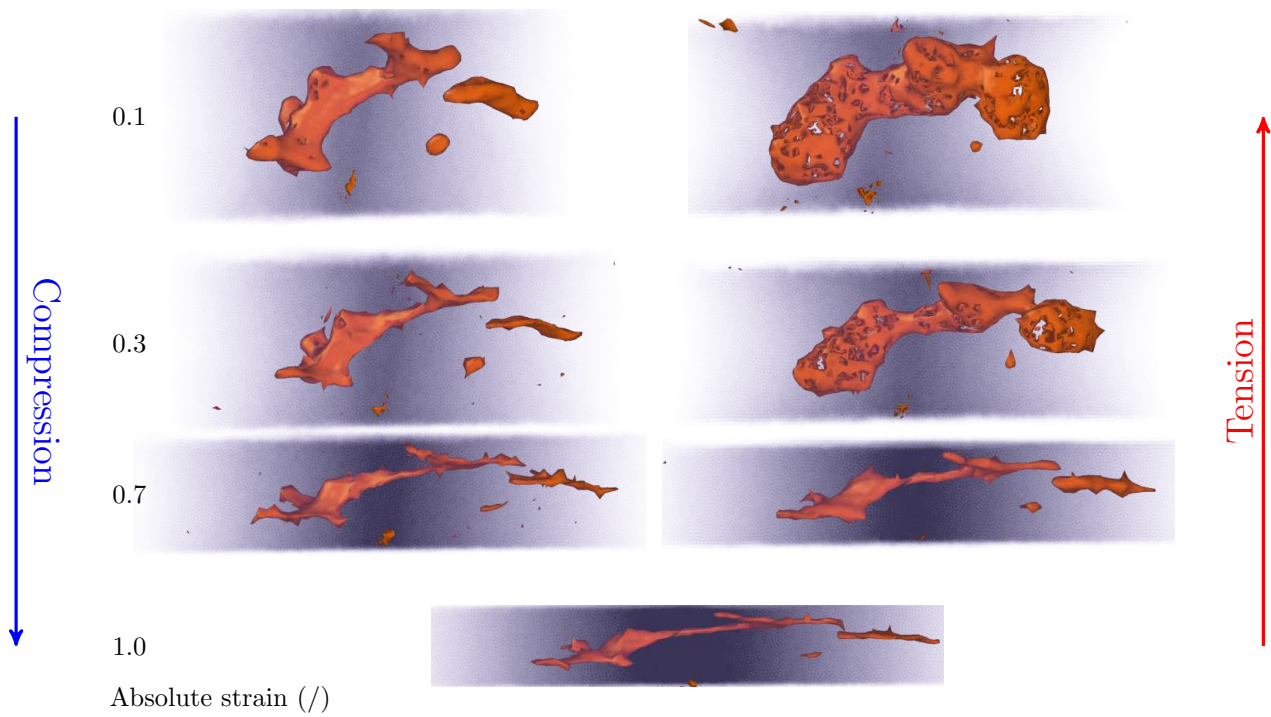


Figure 16.2: Mesh reconstruction of the pores during a uniaxial compression-tension test.

thickness of the mesh, when the pores are totally closed (for example at a strain of 1.0) stems from the mesh reconstruction: the mesh goes through the center of the particles, located at one side or the other of the interface.

To verify that the pores do indeed re-open on the tracked interfaces, it is possible to visualize thin slices of the sample (Figure 16.3). As the sample deforms, the interfaces move through the fixed slice position. It is however possible to see the re-opening of the pores, knowing from the 3D views that the interfaces are not altogether lost.

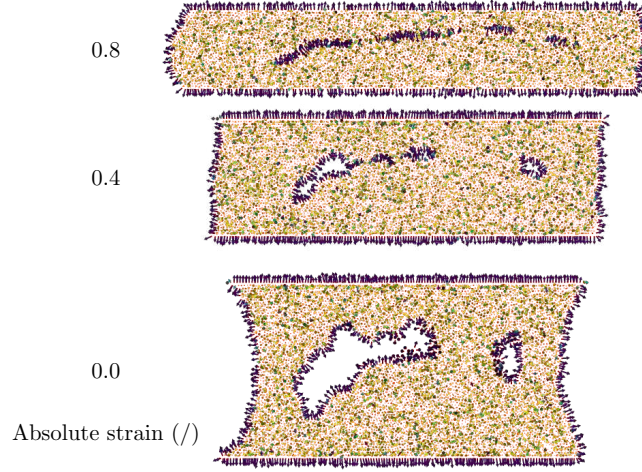


Figure 16.3: Re-opening of the pores during the tension phase of a uniaxial compression-tension test.

A typical experimental metric of interest is the evolution of the pore volume. An estimation of this metric (Figure 16.4) can be computed using mesh reconstructions. The numerical parameter driving the mesh reconstructing in ovito is a probing radius [221].

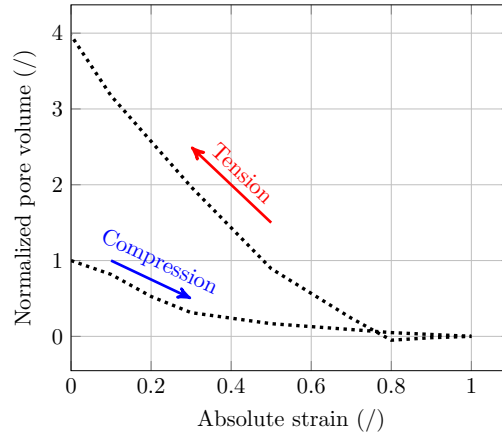


Figure 16.4: Evolution of the relative volume of the pores from the initial state.

Two reconstructions, using all particles, are executed: one with a very large probing

radius (5 mm) and a second one with a small probing radius (0.9 mm). The difference between the volumes of the meshes is influenced by the greater roughness of the surface with low probing radius. The evolution of the volume of the pores is thus corrected using a reference configuration, where the pores are considered closed (here a strain of 1.0). This first approximation is rough: at the beginning of the tensile load, the estimated volume is slightly under zero at a strain of 0.8, which is within the measurement noise.

Regarding the general trend, the pore does not seem to instantly re-open. This may not be unrealistic but may also stem from the harshness of the test case for the model. Indeed, the change of velocity of the plane is instantaneous, from a time step to the other. As shown for example on Figure 14.4 on page 154, the initiation of the flow from an equilibrium state starts with a transient regime, roughly for the first 0.1 of strain. A similar effect, but of greater magnitude is to be expected here. The potential effect of the “interface” errors (Section 15.1), i.e. pairs erroneously considered as cohesive, has not been investigated.

16.2 Aluminum Open Cell Foam

The geometry of a metallic foam with low relative density is discretized. The foam is compressed up to large strain, for various relative densities and strain rates. After a short description of the discretization procedure, three sections respectively deal with the qualitative deformation mechanisms, the quantification of the macroscopic flow stress and a first approach to the estimation of the local stress field.

To some extent, an open cell foam could be considered as a material where pores are fully percolated. From the point of view of our model, it makes no conceptual difference to consider an arbitrary low density. The limit is the choice of the discretization, only packings display the expected behavior, not isolated particles. A slender beam or a thin shell must be discretized with at least a few particles in their smallest thickness.

We study here the case of a low relative density structure. Figure 16.5a shows a tomography reconstruction [243] of an ERG aluminum foam¹ of relative density 6.6%. The dimension of the sample is $4 \times 10 \times 15 \text{ mm}^3$, and the original image resolution is approximately $13 \mu\text{m}/\text{pixel}$.

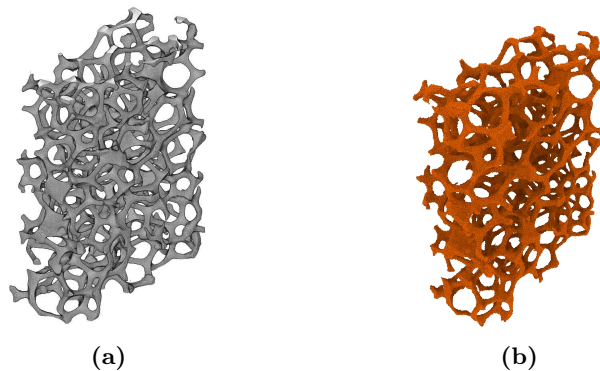


Figure 16.5: Discretization of an open cell foam. Perspective view. (a) 3D reconstruction of the tomography. (b) Packing after removal of particles and relaxation.

¹ERG materials & aerospace is a manufacturer of open cell metallic foam: www.ergaerospace.com.

At the chosen discretization (Figure 16.5b), $\text{pixel}/r_{\text{seed}} = 1.5$, the typical width of an arm of the foam is discretized with less than a dozen of particles. The total number of particles in the simulation is slightly under $5 \cdot 10^5$.

16.2.1 Qualitative Behavior

Bending mechanisms dominate the deformation of the foam. Weak zones are preferentially deformed until a self-contact event hinders their deformation. Large strain, displacement and rotation are modeled.

When such a geometry is crushed, the deformation ultimately leads to interactions of the arms on the foam: self-contacts. Our method readily allows the deformation up to large strains (Figure 16.14a).

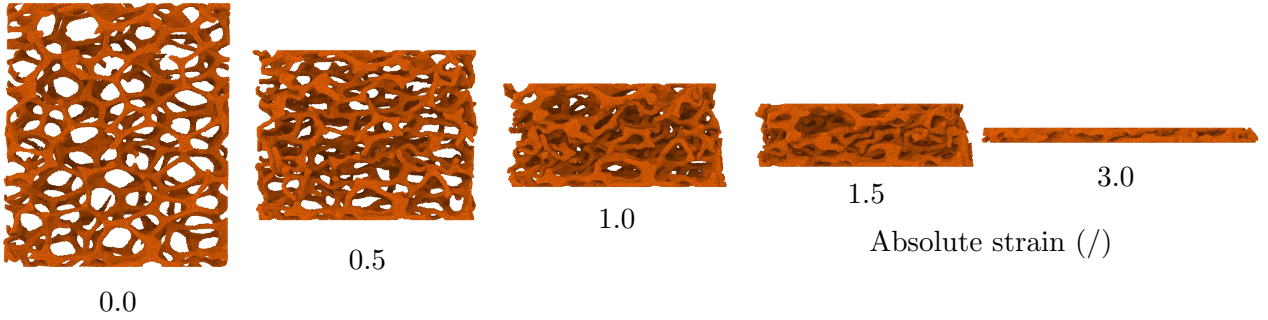


Figure 16.6: Compression up to a strain of 3.0 of the foam at $1 \cdot 10^{-3} \text{ s}^{-1}$. Left view, relative density 6.6 %.

The chosen geometry is very thin and delicate on the boundary of the domain: isolated arms must distribute the load to all the sample. Practically, during the first percent of strain (Figure 16.16), the arms directly in contact with the meshes locally deform and cope with most of the strain. Local details are thus crushed until a sufficient surface of interaction to distribute the load is created. This is not a numerical artifact, but rather the consequence of the geometry choice.

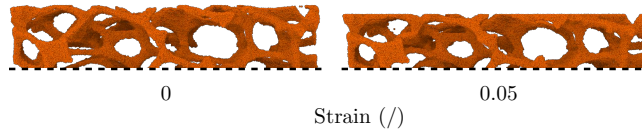


Figure 16.7: First percent of strain. Crushing of isolated arms by the plane until the effort is distributed enough to be transmitted to the structure.

Within the accuracy of the self-contact detection algorithm, the numerous interactions of the arms are captured and tracked as interfaces (Figure 16.8). The arms are well segmented in the model and their interactions are effectively non-cohesive.

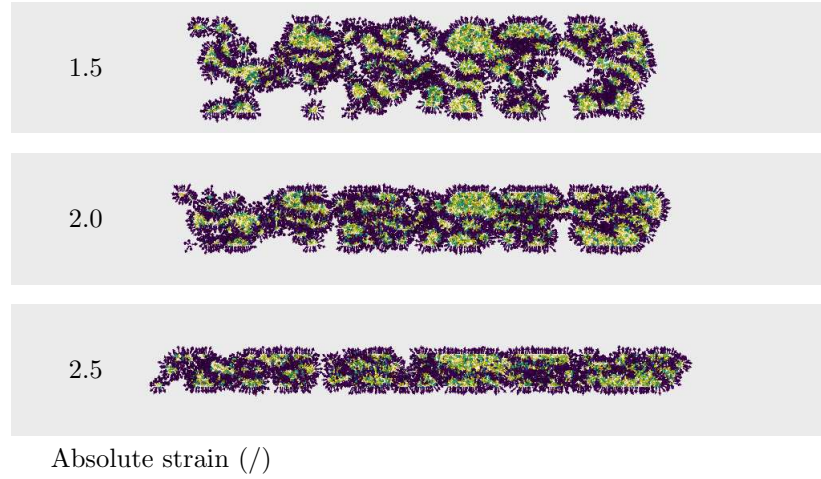


Figure 16.8: Slice through the middle of the sample (width= $2r_{\text{seed}}$), with representation of the outward vectors.

Regarding deformation mechanisms, the crushing of the foam is dominated by bending and instabilities. Illustrative examples are taken from the beginning of the deformation (strain range 0 – 1) at $-1 \cdot 10^{-3} \text{ s}^{-1}$.

A first example is an isolated arm (dashed on Figure 16.9). Very early in the deformation (before a strain of 0.2), the off axis compressive load (arrows at a strain of 0.05), imposed by the surrounding, induces a severe bending. Being of small section and isolated, the arm cannot withstand the load. At a strain of 0.3, two arms above and below come into contact (circled at 0.30). The deformation of the arm is thus temporarily hindered. This flexion mechanism of local weakness, until the deformation is halted by a contact event, is widespread in the structure.

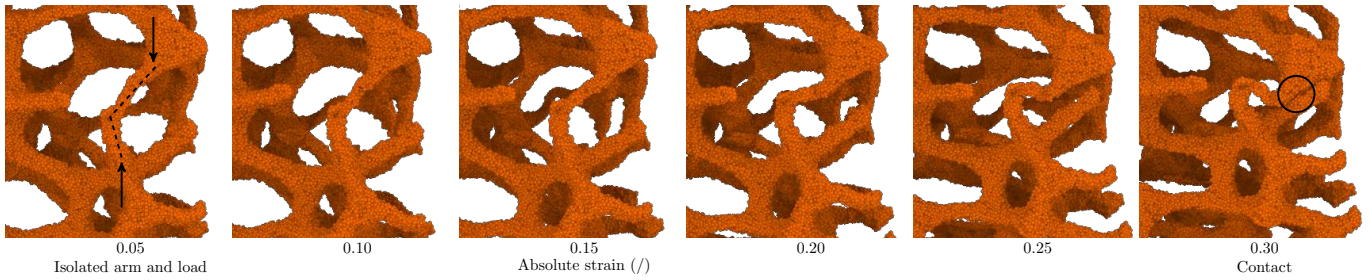


Figure 16.9: Bending of an isolated arm. Detail from Figure 16.12, on the right side of the sample.

The second example is the collapse of a cell (Figure 16.10). Initially roughly hexagonal² (sketched at a strain of 0.05), the cell at first partially withstands the load imposed by the lower and upper vertical pillars (arrows at 0.05). The superior part of the hexagon copes with most of the deformation and progressively deforms, up to a strain of 0.2. The summits of the hexagon behave like hinges and the arms bend to comply with the varying

²Qualitatively, the configuration is similar to the honeycomb loading shown Figure 1c in [8, p.2854].

angles. The deformation then accelerates, the shape degenerates to a rectangle (sketched at 0.35), with the pillar pushing in the middle of the horizontal sides. The cell collapses rapidly, while the vertical pillars coped with very little deformation from the beginning. Several typical traits are displayed: the rotation of the arms around hinge-like joints of the foam, the preferential deformation of horizontal arms by “three point” bending.

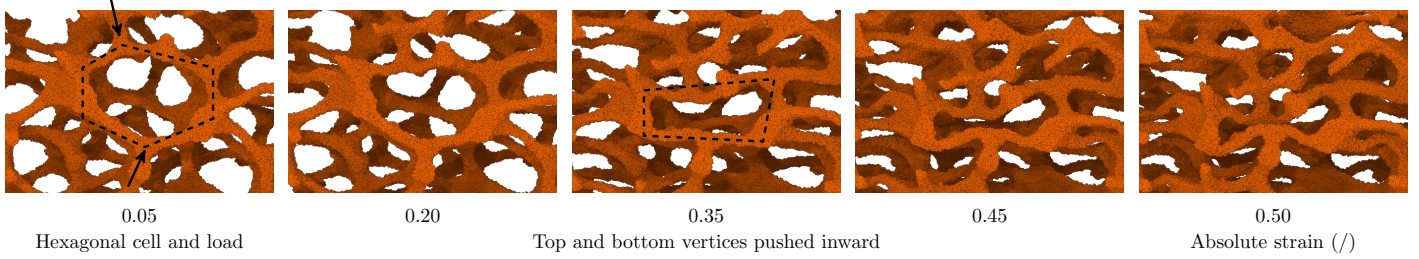


Figure 16.10: Collapse of a cell, by bending of the horizontal arms. Detail from Figure 16.12, a little above the center of the sample.

A third example is the illustration of the large rotation occurring in the sample (Figure 16.11). The lower dashed circle (at a strain of 0.5) represents a “hole” in the foam, which is not propped by arms of material. During the compression, this volume is thus drastically reduced. In contrast, the upper dashed circle is a strongly sustained cell. It is slightly deformed during the compression, but it keeps its original shape in this first stage of compression. The overall effect is the relative rotation of blocks of material (dashed lines). Successive events of this type occur simultaneously and sequentially, until the motion of weaknesses of this type are all blocked by contacts, and the “strong” cells have to deform.

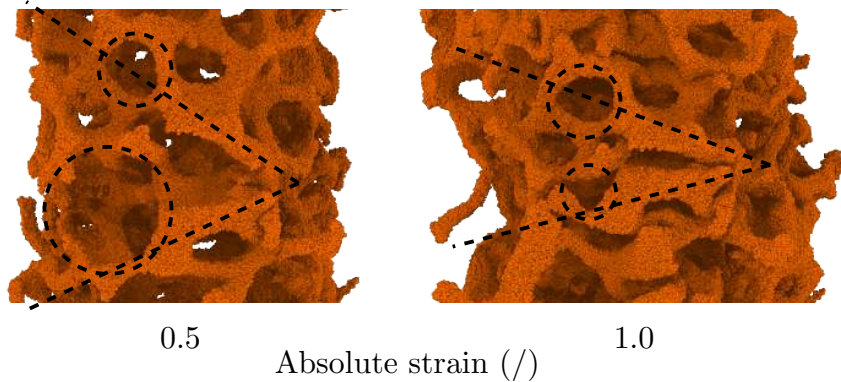


Figure 16.11: Large rotation, by deformation up to self contact of an “empty” zone. Front view.

Overall (Figure 16.12), the deformation is dominated by the bending and buckling of the arms, successively deformed in the weaker zones of the foam. These zones are defined by the geometry and orientation of the arms and the cells, as well as the geometrical surrounding configuration, which will transmit or sustain the load. The deformation involves localized large strains and large rotations. The self-contact events ultimately hinder the deformation of the weaker zones, transmitting the load elsewhere.

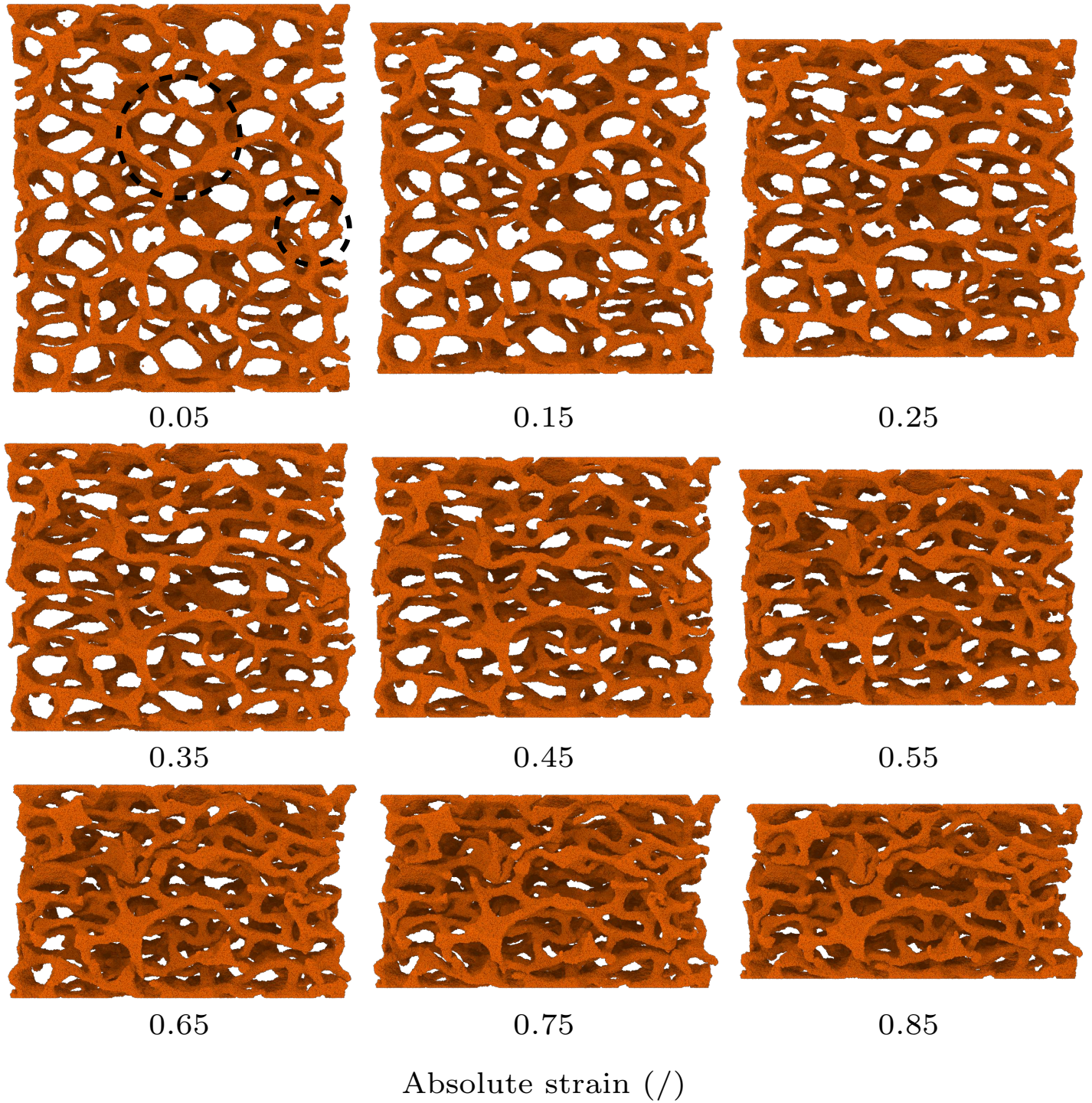


Figure 16.12: Right view, relative density 6.6 %, absolute strain rate 10^{-3} s^{-1} . Compression from a strain of 0.05 to 0.85. At a strain of 0.05, the two circled details are zoomed on Figures 16.9 and 16.10.

16.2.2 Macroscopic Stress

The macroscopic flow stress to deform the foam compares well with the literature. Initially very limited compared to the flow stress of the dense material, the stress increases as the structure collapses and the mutual support of the arms generalizes. The effect of the relative density and the strain rate are investigated in two dedicated sections.

To quantify the efforts that the foam can support, the apparent engineering flow stress can be computed, using the initial cross-section of the sample and the sum of the forces acting on the meshes.

The typical stress-strain profile of a compression up to large strain is shown on Figure 16.13. Very qualitatively, on a linear scale, the deformation of the foam requires a limited stress with respect to the flow stress of the dense material (30 MPa at 10^{-3} s^{-1}). Starting at a strain of 1, the flow stress progressively increases, rapidly increasing in the final range of strain 2 – 3.

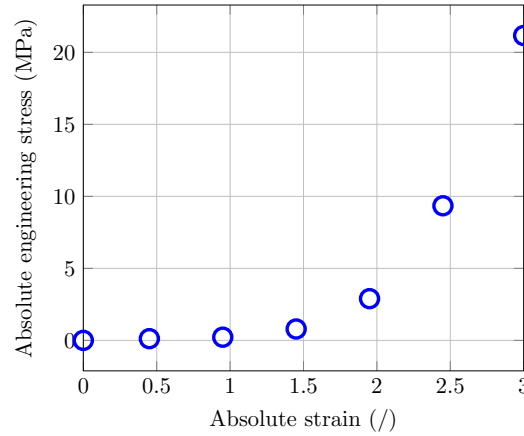


Figure 16.13: Apparent engineering stress versus strain. Linear scale. Refer to Figure 16.14a for a logarithmic scale.

Using a semi logarithmic scale (Figure 16.14a), several traits can be identified. The sharp overshoot observed in the first percent of strain will be examined in Section 16.2.2.2. Up to a strain of 0.5, the created contacts seem isolated enough for other zones to cope with the deformation without notable increase of the macroscopic flow stress. In the strain range 0.1 – 1, the flow stress is within the range 0.1 – 0.2 MPa.

To further quantify the credibility of the modeled stress, the apparent flow stress of this open cell foam is evaluated as follows [8, Eq. 9a]:

$$\sigma_{\text{foam}} = \sigma_{\text{dense}} \cdot \rho_{\text{rel}} \left(\frac{3N_n + 1}{2N_n} \right) \cdot \left[\left(\frac{N_n + 2}{0.6} \right) \frac{1}{N_n} \cdot \frac{N_n}{1.7(2N_n + 1)} \right] \quad (16.1)$$

With $\rho_{\text{rel}} \approx 0.066$, $N_n = 1/M \approx 14$ and $\sigma_{\text{dense}} \approx 30 \text{ MPa}$, the estimated flow stress for the foam is thus $\sigma_{\text{foam}} \approx 0.17 \text{ MPa}$. The order of magnitude of this apparent flow stress is thus correctly captured.

From a strain of 0.5 to 2.5, the flow stress increases with a somewhat similar tendency to the strain-density behavior (Figure 16.14b). The flow stress reaches 1 % of the dense

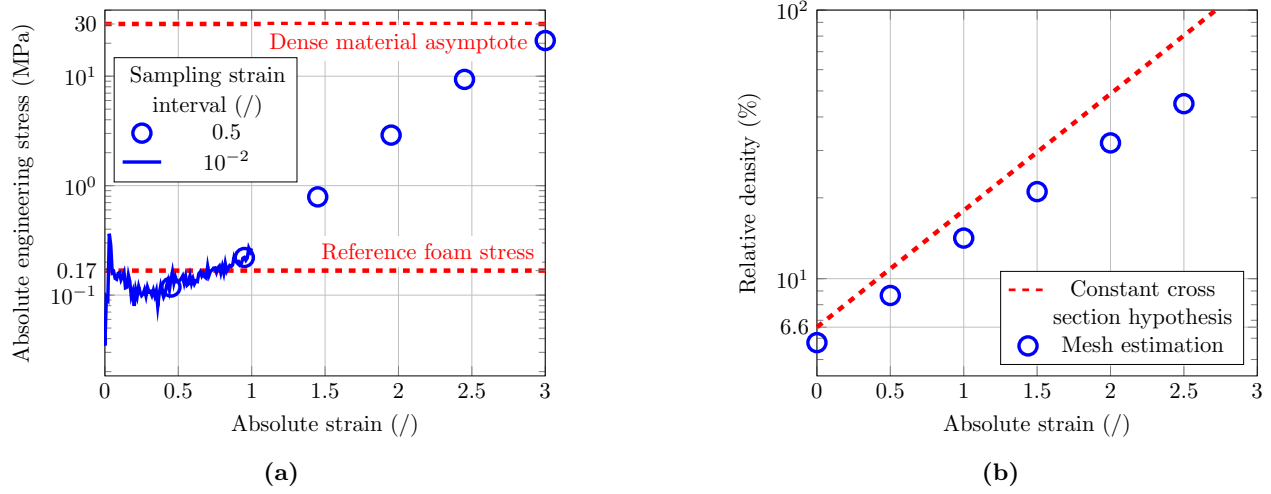


Figure 16.14: Macroscopic behavior of the foam under large compressive strain. Relative density 6.6 %, strain rate 10^{-3} s^{-1} . (a) Apparent engineering stress versus strain. In dashed lines: asymptotic true flow stress and estimated flow stress for the foam (Equation 16.1). The averaging window is identical for the two proposed sampling intervals. (b) Rough approximations of the relative density versus strain.

flow stress at a strain of 2. Self-contacts progressively generalize to the whole sample, following a somewhat stable trend in the strain range 1 – 2.5.

Note that at hypothetical full geometrical densification, the expected behavior is not one of the dense samples: the numerous discontinuities are still tracked and are not cohesive. Although our discretization with spheres introduces a surface roughness, leading to a numerical friction, the asymptotic flow stress of the “dense” foam is expected to be lower than the flow stress of a genuinely dense sample.

The instantaneous cross-section and volume of the sample are delicate to estimate, more specifically at large strains. To estimate the relative density, i.e. the compaction of the foam, mesh reconstructions are used on the particles. The overall volume occupied by the foam and its solid volume are estimated using respectively a large probing sphere and a small probing sphere. As the algorithm of mesh reconstruction of ovito has not been modified to take our interfaces into account, the results are somewhat rough and arbitrary³, hence subtle variations at large strain cannot be measured. The general compaction tendency can be captured (Figure 16.14b) and is compared to a hypothesis of constant cross-section. In the strain range 0 – 1, the slope is similar: the change of the cross-section indeed seems negligible. If some discrepancy can be found at higher strain, and is coherent with qualitative observation (Figure 16.6), the metric is a little rough for further analysis.

At large strain, close to the full compaction, using metrics based on the local fields may be more appropriate. For example the density could be estimated based on the number of neighbors of each particle. To compute the true stress close to the compaction, an averaging of the local stress field may be possible.

³ Chosen small probing radius: 0.9 mm. Chosen large probing radius: 50 mm for the initial state and 30 mm in the strain range 0.5 – 2.5. At a strain of 3, this metric measures a density of 100 %.

16.2.2.1 Effect of the Relative Density

The flow stress variation induced by the relative density is in good agreement with our reference. In the studied range of relative densities, no major change of the deformation mechanisms is observed.

Starting from the initial 3D images, with a relative density of 6.6 %, the foam is dilated to densities of 12 and 18 %. The two new images are then discretized using an identical procedure and resolution (Figure 16.15). The number of particles thus increases with the density. The behavior of the three geometries is compared in the strain range 0 – 1. With the exception of a few details, some thin links are removed at lower density due to the numerical discretization, the connectivity and overall geometry of the three structures are similar.

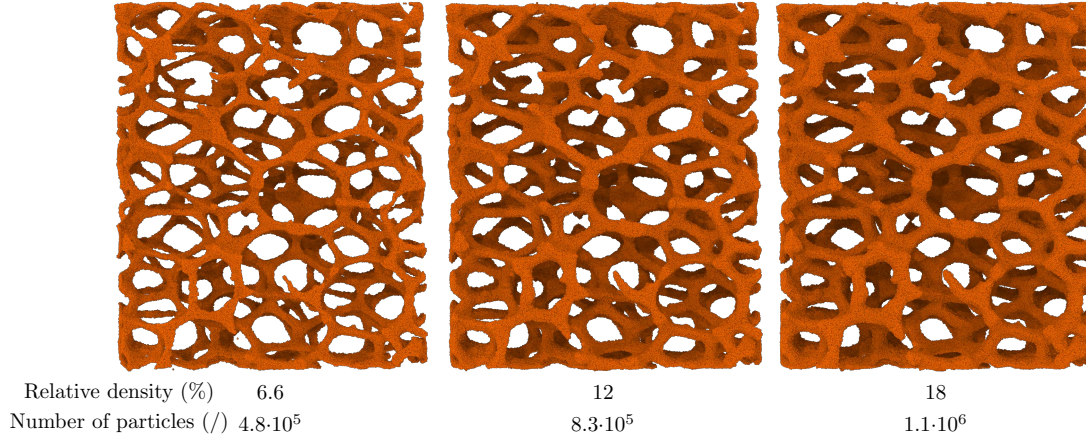


Figure 16.15: Discretization of three relative densities. Right view.

This similarity of the geometries is illustrated by the qualitative mechanical behavior (Figure 16.16). Although the bending resistance increases with the section of the arms, similar deformation is observed. On local configurations, the changes of relative resistance of the geometrical feature does influence the response. Overall, no major modification is observed within this relative density range.

The effect of the density clearly has an impact on the strain-stress behavior (Figure 16.17a). The overshoot effect, in the first percent of strain, will be examined in Section 16.2.2.2. Before numerous self-contacts occur, for example around a strain of 0.2 at the beginning of the compression, the flow stress increases with the density. Indeed, the thicker arms require more efforts to be deformed. For all three configurations (Figure 16.17a), the order of magnitude of the flow stress (computed using Equation 16.1) is correctly captured.

As the arms of the foam get thicker, they may also start to meet each other earlier in the deformation. Comparing the stresses at strains of 0.2 and 1, the overall mechanical effect seems to be secondary for the studied density and strain ranges. For all three densities, the general trend of evolution of the stress with the strain also seems similar and multiplicative factor of the stress between strains of 0.2 and 1 is comparable.

The Equation 16.1 also exhibits a power law of the flow stress with respect to the density, with an exponent depending solely on the strain rate sensitivity $N_n = 1/M \approx$

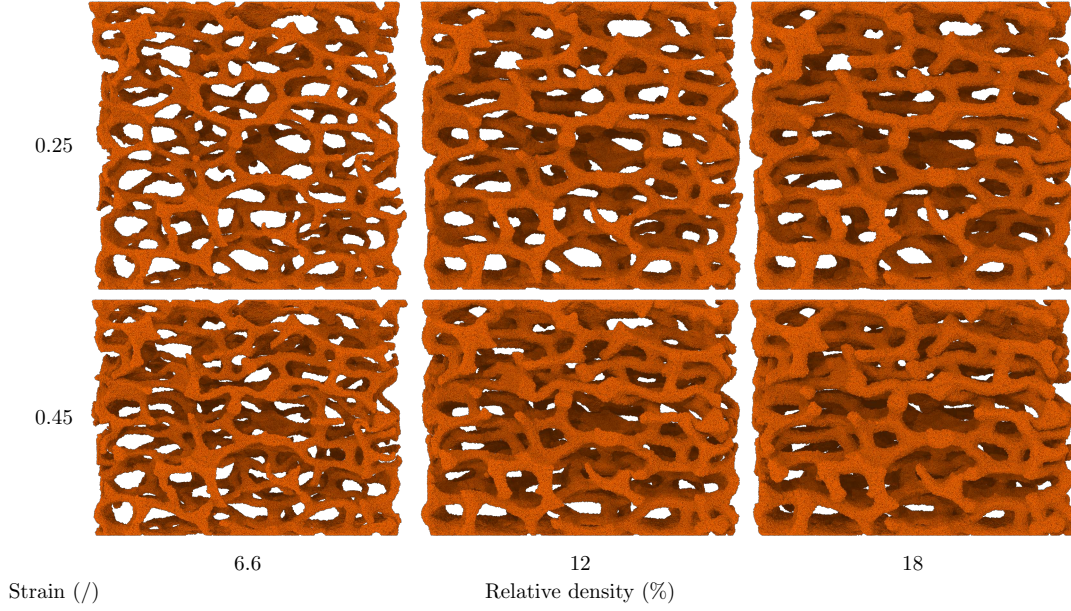


Figure 16.16: Limited effect of the relative density on the qualitative deformation. Right view.

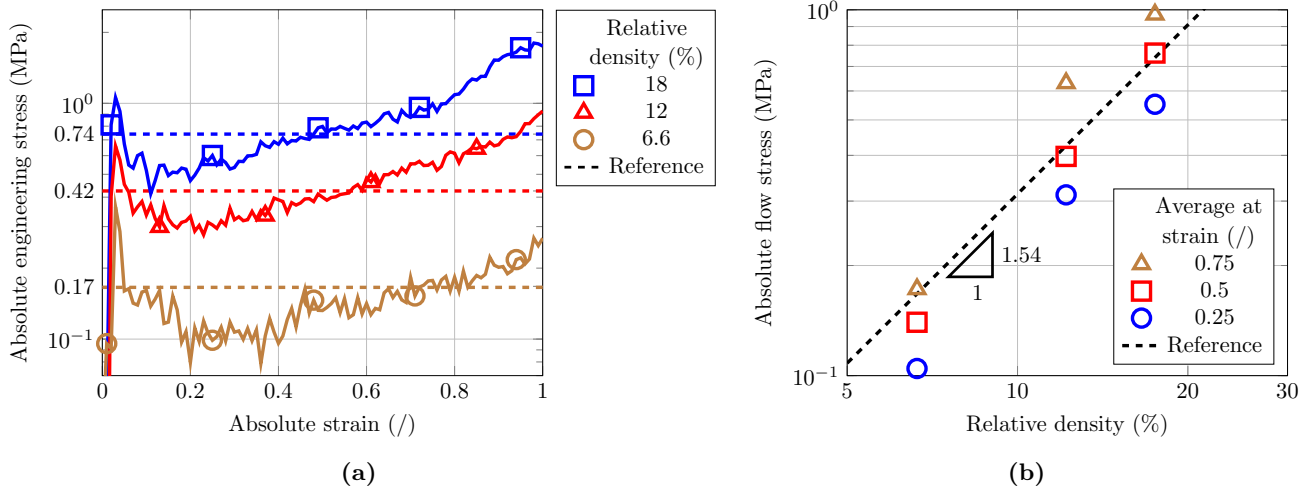


Figure 16.17: Influence of the density on the stress. Reference from Equation 16.1. (a) Apparent engineering stress versus strain. (b) Estimated flow stress versus density. Stress averaged over a window of strain of 0.1.

14.0:

$$\frac{3N_n + 1}{2N_n} \approx 1.54 \quad (16.2)$$

The order of magnitude of this trend is also correctly captured by the model (Fig-

ure 16.17b). The choice of a strain and of an averaging window to compute the flow stress is somewhat arbitrary. An exact match of the stress level of the law would thus be fortuitous. However, the measured exponent of the power law seems to be little influenced by these choices: slopes obtained with an average over the strain range $0.2 - 0.3$, $0.45 - 0.55$ or $0.7 - 0.8$ give similar results. This illustrates the similarity of the strain-stress trends for the distinct relative densities studied. The slope tends to be overestimated, numerically slightly between relative densities of 12 and 18 %, more frankly between 6.6 and 12 %.

The general study of the effect of the discretization on the flow stress was only carried out on unidirectional tension and compression tests (Section 14.3). In contrast, the deformation of the structure is dominated by bending and instabilities. To check that we indeed observe an effect of the relative density of the foam (Figure 16.17a), two distinct discretization sizes are compared. The geometry at a relative density 6.6 %, initially discretized with $\text{pixel}/r_{\text{seed}} = 1.5$, is discretized again with $\text{pixel}/r_{\text{seed}} = 1.2$. The number of particles to model the geometry thus rises from $4.8 \cdot 10^5$ to $9.2 \cdot 10^5$. Qualitatively, the rough discretization misses some fine geometrical details that are correctly captured with more particles. It is thus possible to find local configurations with some discrepancy of the exact deformation mode. Overall, the mechanisms are not affected and the final states are comparable. The order of magnitude of the flow stress at a strain rate of 10^{-3} s^{-1} (Figure 16.18) shows good agreement.

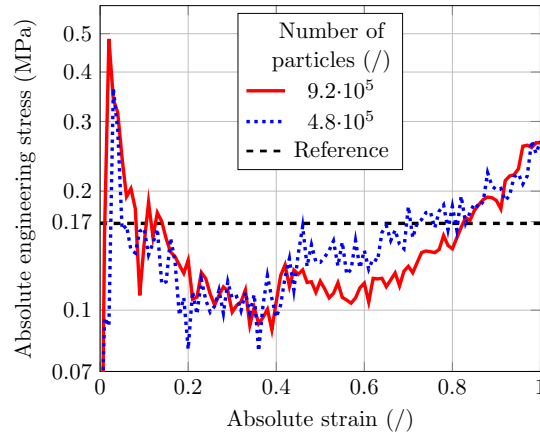


Figure 16.18: Minor effect of the chosen discretization. Strain-stress profile for a relative density of 6.6 % at a strain rate of 10^{-3} s^{-1} .

The overshoot effect, in the strain range $0 - 0.1$, is slightly more pronounced for the finer discretization, for reasons that will be examined in Section 16.2.2.2. The larger number of particles tends to smooth the oscillations of the stress. The general tendency and the stress level are similar: the observed effect of the relative density is thus not a numerical artifact stemming from the change of the number of particles.

16.2.2.2 Effect of the Strain Rate

At low strain rate, an overall buckling of the full sample is observed. Although the time dependence of creep buckling is a physical phenomenon, it will not be investigated here. The limitations of the current model in terms of strain rate are investigated.

The effect of strain rate is investigated, comparing the results at 10^{-3} s^{-1} to a lower

strain rate of $3.16 \cdot 10^{-4} \text{ s}^{-1}$, for the three relative densities tested previously (Figure 16.19). In the first few percent of strain, no overshoot effect is observed at $3.16 \cdot 10^{-4} \text{ s}^{-1}$. This effect will be examined in the end of this section.

In the range 0.2 – 0.3, the macroscopic flow stress is very similar at both strain rates. This is coherent with the fact that the foam strain rate sensitivity should be equal to the dense material strain rate sensitivity [8, Eq.9a]. With a strain rate sensitivity $M = 7.14 \cdot 10^{-2}$, the relative variation of flow stress for a factor 3 on the strain rate should be around 8%. Our model is not able to capture this subtle variation on the chosen geometry. A slight reduction of the stress at a strain of 0.2 can be observed on a supplementary strain rate of 10^{-4} s^{-1} , run for the density 6.6%. However, this direct effect of the strain rate is masked by another phenomenon and cannot be more precisely investigated.

Starting at strains between 0.3 and 0.4, the stress at $3.16 \cdot 10^{-4} \text{ s}^{-1}$ drops by a factor 1.5 (Figure 16.19). This drop is far too large to be the result of the strain rate, it is induced by a mechanical instability (Figure 16.20).

The chosen geometry⁴ of the foam is rather slender in the loading direction: the height/thickness ratio is 3.75. The chosen boundary conditions also promote such a phenomenon: the particles are free to translate on the meshes. This freedom allows an easy propagation of the instability. It is thus not physically incoherent to observe buckling behavior⁵ under compressive load. In addition, the creep buckling behavior of the materials proves to be time dependent [40]. It is thus plausible that a critical time threshold is reached at lower strain rate.

At a density of 6.6% (Figure 16.19), this hypothesis can be tested by comparing the stress profile at $3.16 \cdot 10^{-4}$ and at 10^{-4} . The stress drop characterizing the instability respectively seems to occur in the strain range 0.4 – 0.6 and near a strain of 0.2. Converted in time, this would lead to respective buckling times of $1.3 \cdot 10^3 - 1.9 \cdot 10^3 \text{ s}$ and $2 \cdot 10^3 \text{ s}$, at the considered stress.

The quantitative analysis was not pushed further and a throughout verification that we do not merely observe an unwanted numerical artifact was not carried out. For example, although we did not spot such an event, the numerical crystallization depends on the strain rate and could favor preferential modes of deformation. A general conclusion regarding creep buckling cannot be drawn from these very preliminary observations, based on a single geometry, where all buckling events occur on the same defect of the geometry. However, although we do not purposely model buckling, and we do not control it here, the method may be promising to study instability phenomena occurring in foams at large inelastic strain.

We will now examine the sharp overshoot effect, observed in the first few percent of strain at a strain rate of 10^{-3} s^{-1} . This overshoot is a numerical artifact stemming from the numerical method of prescription of the strain rate. Indeed, an interesting feature of this test case for the robustness of the model is the large dimension of the sample in the loading direction. As we work at prescribed strain rate, the absolute velocities of the meshes thus increase with the size of the sample. As we work with a fixed time step, a critical regime is met when the mesh can escape from, or go across, the geometrical zone of interaction of a particle in few time steps. The configuration shown at 10^{-3} s^{-1} is actually close to this critical numerical regime.

Overall, the global rearrangement behavior is still observed at a strain rate of $3.16 \cdot 10^{-3} \text{ s}^{-1}$. Local qualitative deformation configurations are correct. However, the packing does not manage to fully collectively cope with the deformation: the cells of

⁴The experimental sample from which the geometry is taken was indeed designed for tensile loads.

⁵We investigate here the overall buckling of the whole structure, not the local buckling of the arms of the foam [42, p.3399].

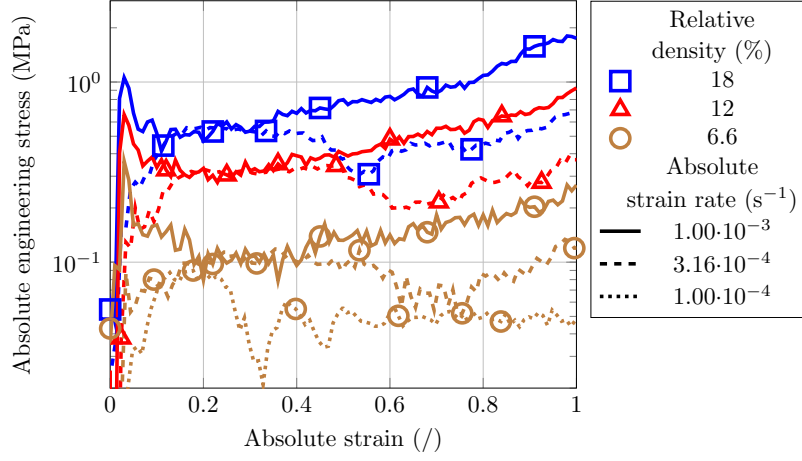


Figure 16.19: Effect of the strain rate on the mechanical behavior. Mechanical instability at low strain rate. Stress versus strain at various relative densities.

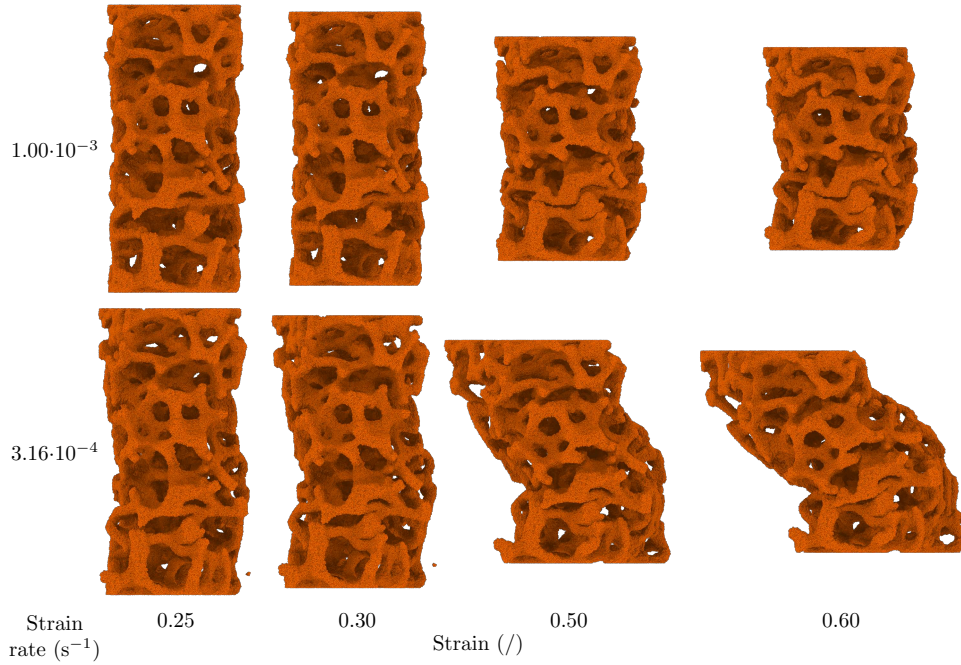


Figure 16.20: Buckling of the sample at low strain rate. Front view, relative density 12%.

the foam deform more near the meshes. This “dynamic-like” effect is not suitable for our purpose.

The initial numerical “height” H of the sample is 264 mm. At $-3.16 \cdot 10^{-3} \text{ s}^{-1}$, the initial absolute velocity of variation of height \dot{H} can be computed as:

$$\dot{H} = \dot{\epsilon} \cdot H = 3.16 \cdot 10^{-3} \cdot 264 = 8.34 \cdot 10^{-1} \text{ mm} \cdot \text{s}^{-1} \quad (16.3)$$

The variation of height ΔH at each time step Δt is thus:

$$\Delta H = \dot{H} \cdot \Delta t = 8.34 \cdot 10^{-1} \cdot 10^{-1} = 8.34 \cdot 10^{-2} \text{ mm} \quad (16.4)$$

This is numerically large compared to the dimensions of the particles ($r_{\text{seed}} = 0.5 \text{ mm}$ and $r_{\text{crown}} = 0.7 \text{ mm}$), more than one tenth of the crown radius. If the packing is initially at rest, the meshes⁶ are able to cross or leave the particles in very few time steps. Our model relies altogether on a collective motion: reasonable local relative velocities are reached after the transient regime where the flow is initiated. For large packings, a numerical work-around must be provided to initiate the flow. In the spirit of our method, it would be desirable to limit such procedures to the transient regimes⁷, at each non-smooth change in the loading of the samples.

A tested procedure was the prescription of the initial velocity (in three directions, based on the expected overall flow) of the particles, depending on their position in the sample. This naive attempt did not significantly improve the behavior, it seems necessary to at least introduce a random component to obtain suitable relative velocities between the objects.

16.2.3 Local Field

A rough estimation of the local stress field is proposed. On the studied configurations, the observed tendencies are in agreement with the qualitative behavior of the deformation. Further statistical analyses are required to provide quantitative data.

The estimation of local field is not trivial from a statistical point of view. Temporal and spatial averages are necessary, and their effect must be understood (refer to Appendix A). In addition, the presence of interfaces may lead to cumbersome procedures, although this is not a conceptual limitation.

However, the macroscopic behavior of the model is solely driven by the local phenomena. This behavior seems correct with regard to some tested metrics. It should thus be possible to define sensible intermediary scale metrics, between macroscopic and particle-wise scales. On dense samples, the behavior of packings of a few hundreds of particles could already be considered as a good approximation of the model (Section 14.3), even with the strong boundary effects of the meshes.

As a mere illustration of the interest of a local field approximation, some examples are given regarding an approximation of the stress field. For each particle, the stress is approximated with Equation 8.12 on page 96. The components of the stress tensor are average particle-wise, over a sliding window of strain of 10^{-2} . The signed equivalent Mises stress⁸ is then computed. Finally, a spatial average within a radius of 2.5 mm is performed at each particle.

⁶To balance the computing load evenly between the processors, both meshes move. Their absolute velocity is thus half \dot{H} . The result of the computation must be independent of such numerical tricks.

⁷A classical DEM procedure is to linearly remap the position of the particles when the domain is deformed. This procedure has been avoided in our work to let the system freely rearrange, to capture localization stemming from the local behavior of the material.

⁸The sign of the trace of the stress tensor is applied to the equivalent Mises stress, previously defined in Equation 2.2 on page 24.

Going back to the example of the collapsing cell (Figure 16.10 on page 176), an estimation of the stress field is given on Figure 16.21.

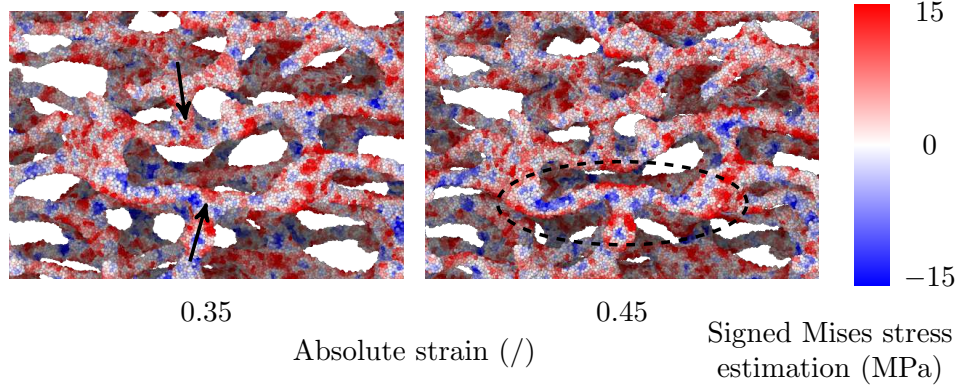


Figure 16.21: Collapse of a cell, by bending of the horizontal arms. The arrows mark the two pillars crushing the cell. The dashed ellipse highlights a bending zone of interest. Estimation of the local stress field. Same geometrical zone as Figure 16.10 on page 176. General view on Figure 16.22.

Looking at the arm bent by the bottom pillar (circled on Figure 16.21), the zones under tensile and compressive loads are qualitatively coherent with the observed bending of the structure. Opposite to the crushing pillar for example, the arm is under tensile load.

Looking at a broader view of the sample (Figure 16.22, the zones currently undergoing deformation could be identified using the map of the estimation of the stress field. At this stage, it is delicate to draw conclusions as most of the structure is bearing very limited local stress. The numerical noise is thus large compared to the observed signal.

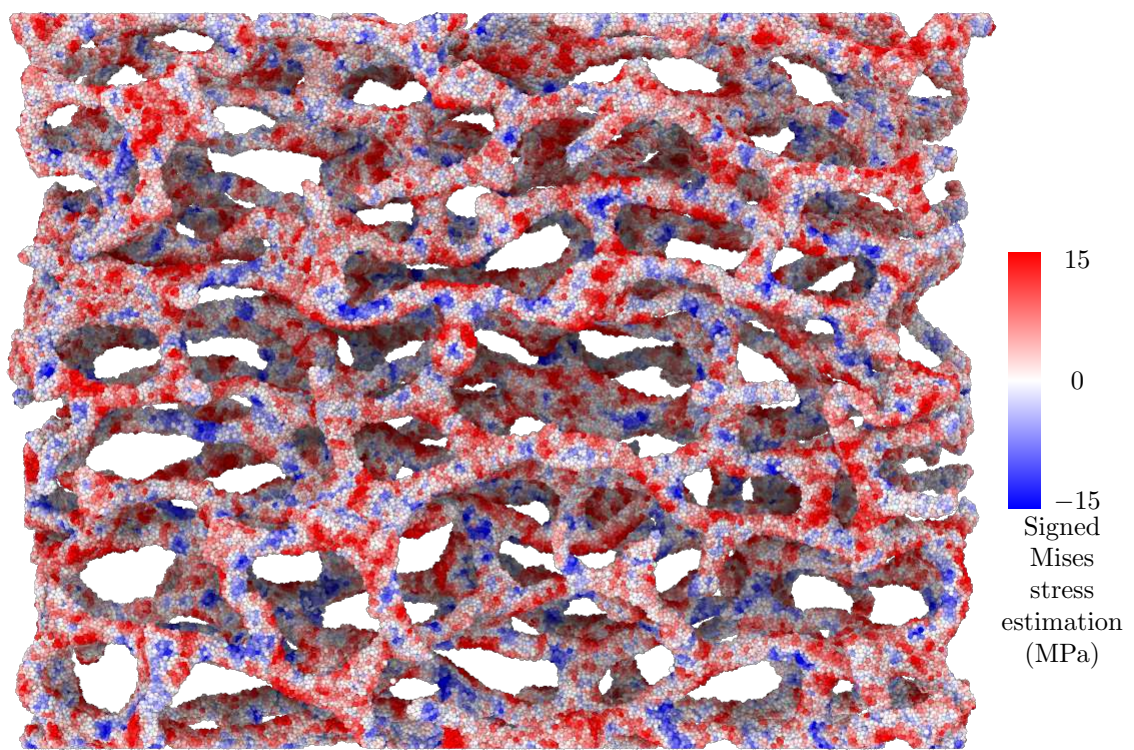


Figure 16.22: Overall view of the local stress field estimation. Strain 0.45, strain rate 10^{-3} s^{-1} and relative density 6.6 %.

General Conclusion

Chapter 17

General Conclusion

In a nutshell, this PhD questioned the potential of the DEM framework for the description of finite, inelastic and incompressible transformations of continuous media. This exploratory attempt was triggered by the intrinsic numerical properties of the method, as the flexible handling of contact events and a straightforward massive parallelization. This concluding chapter is divided into five short sections respectively dealing with the objectives of the PhD, the adopted strategy, the main results, some limits and finally potential developments.

17.1 General Objective

The general starting point of the work is to study the forming of multiphase materials. The physical phenomena of interest are the mechanisms driving finite inelastic strain in architected metallic materials, at the scale of the constitutive phases.

In order to partially decorrelate the numerous effects (e.g. morphological and rheological) a metallic composite is designed as a model material to focus on the influence of the rheology. The behavior of the composite (spheroidal amorphous $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ inclusions in a crystalline copper matrix) is experimentally studied with *in situ* X-ray tomography hot compression tests (Figure 17.1). The co-deformation of the phases is observed, with an interesting dependency of the rheological contrast on the temperature and the strain rate (Section 2.5).

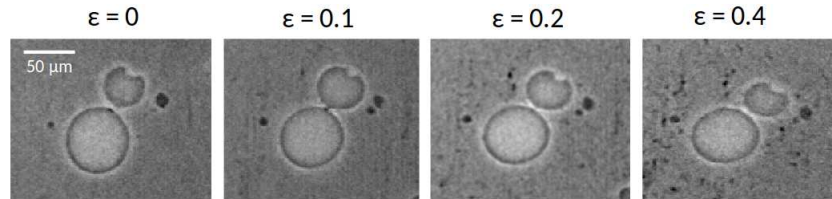


Figure 17.1: Co-deformation of the model composite material, observed by *in situ* X-ray tomography. From Figure 2.19 on page 35.

Modeling tools are seen as complementary to the experimental analysis. The modeling objective is thus to describe inelastic finite strains and interaction of interfaces, observed experimentally. The model must handle numerous interface interactions and topological events like for example pore closure and phase decohesion or fragmentation (Section 3.3).

A lecture grid, focusing on algorithmic features, can help to highlight distinctions and similarities for the selection of numerical methods. In our objective, methods designed to solve partial differential equation (PDE) display a variety of strategies to include the description of discontinuities in their framework. Methods fundamentally based on a discrete topology can in turn propose phenomenological routes to mimic the behavior of continuous media.

Among potential modeling tools, the DEM is innately suited to handle numerous contacts and topological events. The chosen modeling strategy is thus phenomenological. The research question focuses on the assessment of the pertinence of this choice to meet our modeling objectives (Chapter 7). The objective of the PhD is to develop a DEM algorithm for finite inelastic transformation of incompressible multi-material.

17.2 Modeling Strategy

More than a precise road map, it must be emphasized that the experience of this PhD advocates for a specific development *strategy*. In short, a too strict attempt to stick to expected elementary mechanisms or physical parameters can be misleading in the design of models. In the context of the DEM, two design strategies were considered:

- Adding-up elementary physical behaviors.

A tempting route in the design strategy is to build a numerical model as closely as possible to the physical model of the elementary observed phenomena: a bottom-up approach. The underlying hope is that intrinsic physical parameters may be directly fed into the numerical model. Computational and physical issues are considered separately, with thus no guarantee of their respective requirements to match. In many configurations¹, such a literal transcription may result in impractical numerical models. In such cases, the models are then marginally modified toward a more computationally suitable state. To remain coherent with its grounding assumptions, this approach can only be reasonably applied to a restricted range of phenomena. In many cases, the marginal modifications are barely sufficient for a reasonable numerical behavior, while denaturing the sought-for physics of the elementary mechanisms. This strategy was not followed in this PhD.

- Tuning an overall collective behavior.

The design process can start in a radically opposed direction, explored in this work: an *ad hoc* computationally reasonable model is built, without requirements on similarity with the physical elementary mechanisms. The design is altogether focused on the modelization objective, i.e. mimicking a physical phenomenon with a collective behavior. The constructed model, potentially highly counter-intuitive, is meant to be as simple as possible at the elementary level and to have an acceptable numerical behavior.

The two strategies may be benchmarked to compare benefits and drawbacks of the approaches for the understanding of a given phenomenon: complementary data may be accessible. The intrinsic properties and limits of the numerical tools may also promote a strategy more than the other. A common configuration in the DEM is that little faith of a “realistic” description of the elementary interaction is to be expected (Section 3.1.1).

An approach focused on collective behavior is thus a practical work-around to mathematical (does a unique solution exist?) and numerical or computational (how do errors accumulate?) unknowns. On a very practical note, the proposed model is thus an attempt

¹Refer for example to the discussion of the choice of the time step in the DEM in Section 8.3.3.

to tune the behavior of numerous and simple interacting objects to meet our modelization objectives.

17.3 Results and Applications

To our knowledge, the existing “meshless” approaches to model inelastic phenomena implement them at the level of the computing points. Our proposal relies on the neighbor changes of “undeformable” fictitious particles. *Ad hoc* interaction laws are implemented and the collective behavior of large packings are meant to mimic key features of macroscopic metallic viscoplasticity: macroscopic overall shape, volume conservation, stress-strain behavior and strain rate sensitivity.

The model is solely based on local and relative metrics, at the level of the elementary particles, with no macroscopic artifact promoting an expected solution. Oftentimes, discrete methods use global numerical artifacts to converge faster toward a presumptive solution: global viscous damping, affine transformation... In contrast, all forces acting on our particles are deduced from the relative kinematics of their neighbors of the boundaries. The potential of the model is closely linked to this grounding principle.

A somewhat contentious achievement of this PhD is the conceptual simplicity of the proposed model. Although numerous and complex configurations were investigated during these three years, the final proposal and its implementation are lean. This points out:

- The potentially complex and varied applications of simple modeling principles. If the behavior of the model had constrained us to treat each test case separately with adequate tuning and artifacts, little faith on the predictive ability of the model would be granted.
- The respect of the attempt to limit the implementation to new features, reusing existing and efficient libraries and software solutions.

Three applications of interest are summed-up in the following sections: finite inelastic strain, self-contact and complex mesostructures.

17.3.1 Inelastic Strain

The principles of the model is a set of attractive-repulsive spherical particles, discretizing a continuum. Under external loads, the packing of particles collectively cope with strain. The rearrangements (Figure 17.2), with arbitrary neighbor changes, account for irreversible strain.

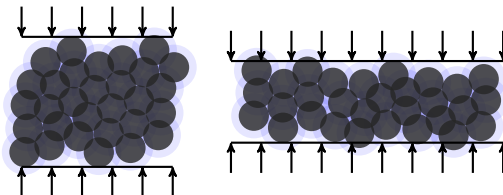


Figure 17.2: Principle of the finite strain modeling. The packing of particles rearranges to mimic inelastic transformation. From Figure 9.2 on page 101.

Two *ad hoc* interaction laws are designed and implemented in the open-source DEM solver liggghts:

- The model *BILIN*, dealing with compressive load only.
- The model *TRILIN*, able to cope with both compressive and tensile loads.

A calibration procedure allows the tuning of the stress level and the strain rate sensitivity to mimic a perfect viscoplastic Norton law. Strain rate sensitivities up to $M \approx 0.5$ can be modeled on one decade of strain rate. Small strain rate sensitivities can be correctly approximated over various decades of strain rate.

Single materials can be compressed up to large strains ($\varepsilon = 1$) with controlled relative error on the volume and the flow stress. After a numerical transient regime, the typical precision for both metrics is around 5 or 10%. In the case of tensile load, the necking and the rupture are displayed but not controlled.

Simple bi-material configurations are compared to results obtained by the finite element method (FEM) (Figure 17.3). On macroscopic metrics (flow stress and morphology), the error of the developed model is of the same order of magnitude as the error on a single material. No excessive errors seem to be introduced when simulating multi-material configurations.

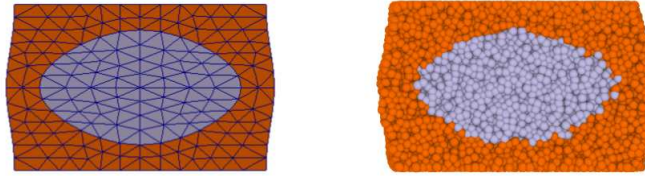


Figure 17.3: Viscoplastic bi-material. Comparison of the deformed morphology obtained by FEM and DEM. From Figure 12.5 on page 135.

17.3.2 Self-Contact

An algorithm to detect physical self-contact events, i.e. the interaction of an interface with itself, is proposed. The self-contact detection is based on an approximation of the free surfaces, for each particle, from the position of its neighbors (Figure 17.4). This is to our knowledge a novel approach.

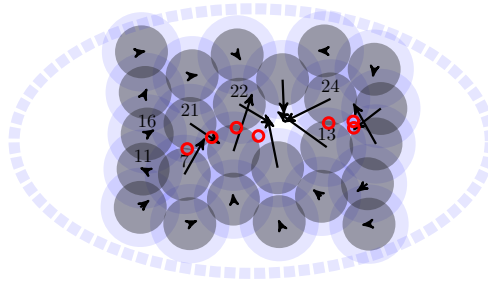


Figure 17.4: Principle of the self-contact detection algorithm. A local “outward vector” is built for each particle. New pairs are classified as “interface” or “internal”. From Figure 9.9b on page 109.

Pairs of particles modeling opposite sides of the interface are only repulsive (Figure 17.5). The mechanical closure and re-opening of holes is natively displayed in uniaxial

compression-tension tests. The tracked interfaces can be further deformed after closure without being lost.

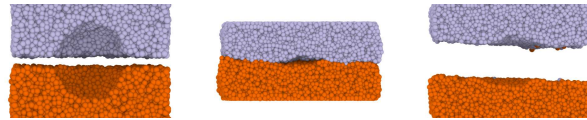


Figure 17.5: Test case for the self-contact detection algorithm. The interface is tracked under compressive load and separates under tensile load. From Figure 15.5 on page 165.

The proposed framework allows a flexible and controlled handling of topological events. As an example, a healing time of the interfaces is implemented: two particles modeling opposite sides of a mechanically closed interface will display attractive behavior after a threshold time. Given a locally computable metric, arbitrary behavior can be developed with ease.

17.3.3 Complex Mesostructure

Complex mesostructures are directly discretized for 3D images obtained by X-ray tomography. The procedure is algorithmically cheap: a segmented 3D image is used as a mask on a random packing of particles to set their properties or remove them.

A first application was the forming of the amorphous/crystalline metallic composite, that originally triggered the study. The 170 physical inclusions of a full tomography sample are discretized and the compression is simulated (Figure 17.6). The model describes the co-deformation of the phases in the material. Specific configurations of interest are compared to *in situ* measurements, applying local measured strain rates. However, the experimental crystallization of the amorphous phase hinders more in-depth analysis.

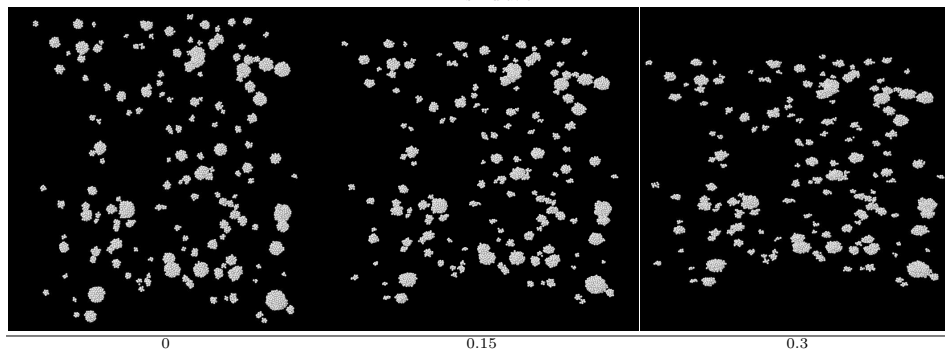


Figure 17.6: Discretization and compression of the full sample. 3D view of the inclusions only, the matrix is hidden. Vertical compression axis. From Figure 13.1 on page 140.

A second application is the simulation of casting pores (Figure 17.7). The compression-tension test performed shows the mechanical closure and re-opening of the pores. The model is able to track the self-contact of the interfaces of the pores at large compressive strain. Under tensile load, the pores re-open and eventually coalesce.

A third application is the simulation of the crushing of a foam with low relative density (Figure 17.8). The local bending and buckling deformation mechanisms of the

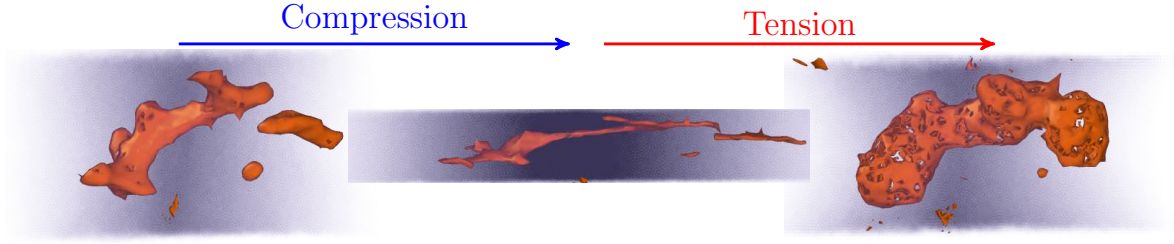


Figure 17.7: Compression-tension of a sample with casting pores. Visualization of the mesh reconstruction of the pores. Mechanical closure, re-opening and coalescence of the pores. From Figure 16.2 on page 171

arms can be observed. At large strain, numerous self-contacts hinder the deformation. The macroscopic flow stress displays a correct behavior with respect to references from the literature.

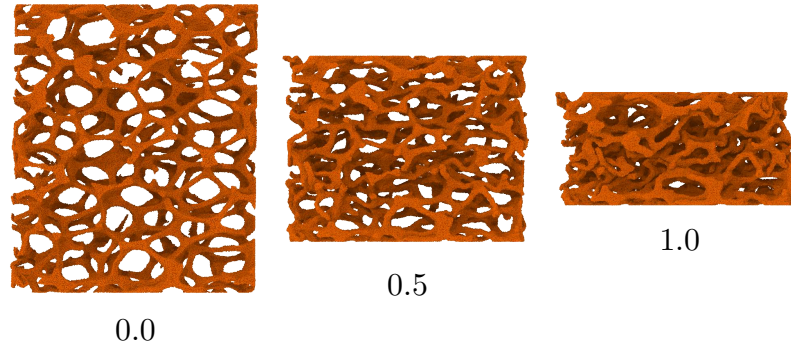


Figure 17.8: Compression of a low relative density foam. Local bending mechanisms of the arms and generalized self-contact at large strain. From Figure 16.6 on page 174.

The choice of an efficient solver leads to a low random-access memory load and relatively cheap computation. As rough orders of magnitude: $1 - 2$ KiB per particle and $2 \cdot 10^{-6} - 6 \cdot 10^{-6}$ cpu second per particle and per time step on an Intel Xeon E5520 (2.3 GHz). As an example, the compression of the ERG foam at $1 \cdot 10^{-3} \text{ s}^{-1}$ up to a strain of 1 (Part V) last 5 h on a single processing unit and use 0.8 GiB of random-access memory.

17.4 Limits and Specificities

During this PhD, the efforts were focused on producing a “proof of concept” and most of the work was exploratory. This emphasis implies that the verification, the validation and the understanding of the model is still scarce. As a very practical consequence, the proposed models probably have little robustness, and the proposed algorithms are not optimal solutions. They must be cross-checked and debugged in a necessary trial phase of development. In depth statistical analysis of the results of the model would be necessary. As-is, the estimation of the metrics and their associated errors are only orders

of magnitude. The thankless task of assessing and improving the robustness is necessary before the precision and the performance of the model can be further investigated. In this perspective, a specific care was taken to exhibit numerical artifact and work-around.

A major strength of the model is the purely local contribution of all particles to compute the material overall behavior. A shortcoming is that boundary condition cannot be enforced this way on arbitrarily large domains. For example, when enforcing a prescribed strain rate with a mesh starting from settled state, the *absolute* velocity of the mesh may become sufficient to go through or leave a particle within a single time step. The permanent flow must be established for the *relative* velocities to become acceptable. In the spirit of the method, developed numerical work-around should be limited to a transient regime, keeping the effective computation purely local.

A clear practical limit concerns the control of the strain rate sensitivity. So far, the reachable strain rate sensitivities are limited and are only valid on narrow ranges of strain rate. Although the attempt of modeling complex continuous constitutive laws is illusory, this limit is a hindrance to model generic viscoplastic configurations. The phenomenon of the numerical flow of attractive-repulsive particles, including features as the initial transient regime and the strain rate sensitivities, might be rather fundamental issues. The improvement of the model requires a more in depth understanding of these issues.

A major drawback of the model is the absence of size adaptivity. All zones of a geometry must be discretized with identical particles. The computing cost is thus set by the smallest required details to be captured. The extension of the model to handle distinct sizes of particles may be possible but is not a trivial task. As-is, the model is not suited to describe phenomena driven by the mechanics of thin films for example.

From the peculiar grounding principle of describing inelastic strain by constrained rearrangement of particles, stems a key property: the computing points are always evenly spread in the material, by design and without regard for the applied transformations (Figure 17.9b). By contrast, all Lagrangian methods based on continuous constitutive laws track the position of effective material points. The distribution of initially evenly spread computing points will thus be geometrically imbalanced after arbitrary finite strain (Figure 17.9a).

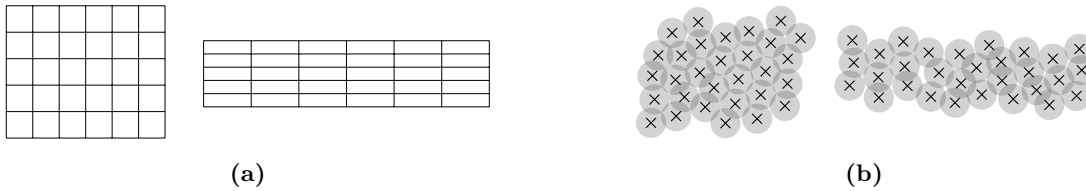


Figure 17.9: Conceptual distinction between a genuine Lagrangian tracking of material points in continuous media and the proposed DEM method. (a) Motion of material points in a continuous medium. (b) Rearrangement of particles in the proposed model.

This potential distortion of the computing “grid” applies to all methods implementing somehow a continuous inelastic behavior at the scale of the computing points, including “meshless” and “particle” methods. To control the distribution of the computing point, remeshing-like techniques or partially Eulerian models can be used.

Our fictitious particles track the materials in a looser fashion. Although our method is algorithmically Lagrangian, as the particles are tracked explicitly with time, the material points are not conceptually tracked by the particles. At a given instant, a particle does mimic some effects of an elementary portion of the material, but the material “flows”

through the particles with time. Inelasticity is implemented at a collective level and the location of a given material point may only be tracked by a group of particles, consistently with the chaotic nature of the model.

The implicit computing point rearrangement does not imply that arbitrary large strain can be modeled: if the strain is excessive the number of particles in the “thickness” of the material may become too small. From an experimental point of view, this behavior can be valuable to model processes where the mechanical influence of a phase becomes negligible under a threshold size. The initial discretization must be chosen in accordance with such an objective.

17.5 Potential of the Model

17.5.1 Direct Applications

Starting from a clear modelization objective, it is possible to choose a modeling tool displaying adequate numerical behavior with respect to the dominant physical phenomena of interest. Methods based on distinct principles are thus complementary to study a variety of configurations.

As a rough sketch of potential methods (Figure 17.10), the FEM remains a reference tool for solid mechanics, with a potentially fine description of the material constitutive behavior. It should be the favored method if the continuous behavior is the dominant physical phenomenon of interest. In pathological configurations, e.g. perfect plasticity and large strains, the convergence of the resolution may be long and unsure. It is sometimes necessary to use numerical work-around to actually find *a* solution, for example using a dynamic formalism for quasistatic problems.

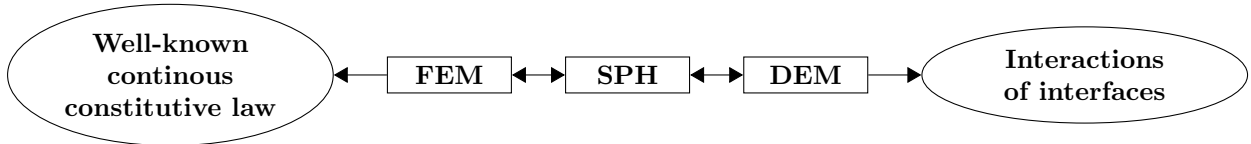


Figure 17.10: Dominant physical phenomenon and potential numerical tools.

In such cases, methods like the smooth particle hydrodynamics (SPH), may be a valuable alternative. In SPH-like methods – conceptually dynamic approaches – the continuous constitutive law is modeled at the computing points, but without relying on a connectivity table. The periodic timely and heavy remeshing procedures are thus avoided. In addition, the method allows efficient and arbitrary handling of discrete events: the behavior of the particles can be mixed between SPH-like and DEM-like interactions.

The predominance of physical interface interactions may advocate for a rougher description of the continuous material. Our DEM model conserves by design an even distribution of the computing points. This property is helpful for the detection of physical contacts. In return, less flexibility and control are possible on the continuous behavior.

In short, the proposed DEM model seems suitable for the description of numerous and simultaneous contacts and topological events in solids. However, the adaptation to fine and precise continuous constitutive behaviors would probably be delicate, or progressively evolve toward an SPH-like method.

Very practically, the proposed model seems appropriate to push further the study of

the finite strain of architected, porous or composite materials. As the focus was set in this PhD on the proposal of a functional tool, the time dedicated to more in-depth analysis of the applications was limited. The results were mostly qualitative and macroscopic, with little direct comparison to experimental results.

A systematic comparison of our model with the available *in situ* X-ray tomography measurements can readily be carried out. As an example in the laboratory context, the mechanical closure and re-opening of casting pores is an experimental study from the ongoing PhD of Pauline Gravier. The compression-tension high temperature tests have been performed *in situ* at the European synchrotron radiation facility (ESRF). In addition to the purely mechanical closure, the effective welding of the interfaces can be modeled with ease for complex geometries within our model. Based on local metrics, arbitrary healing behavior may be implemented in a straightforward way. This ability to model topological events is an innate and powerful feature of the DEM framework.

In the context of architected materials and metallic foam, the study of the deformation mechanisms and instabilities seems promising. The potential shift of mechanisms from the initial phase of the deformation to the full compaction of the sample can be studied qualitatively and quantitatively. The bending, buckling and consolidation by self-contacts can be investigated for complex mesostructures, with both local and macroscopic effects. As a more specific focus of interest, the model may also be used to study the effect of structural defects of the constitutive material of a foam. The link between failure mode, local structural defects and loading conditions is particularly appealing.

Going back to the composite that originally triggered the study, some numerical and experimental limitations were drawn. However, this does not invalidate the potential of the method for this type of study. The effect of the morphology and rheology of the phase can be modeled for the forming of metallic composites. Tortuous mesostructures can be handled, along with the potential contact or self-contact of the phases at large strain. The study of the forming of more classical multiphase material, with available *in situ* data, could help to further validate the model, in parallel to the resolution of the raised issues for our amorphous/crystalline composite.

17.5.2 Algorithmic Development

The implementation of arbitrary healing behavior is already a first hint for further algorithmic development of the model. In the short term, various extensions could help to improve the model or extend its scope.

The design of appropriate statistical procedures to study the local fields, could open interesting application. As the global behavior of the model is not macroscopically imposed, but driven by local events, collecting and interpreting data at a more local scale should thus be possible. First approaches for both strain and stress fields have been attempted, but time lacked to propose well grounded metrics.

The discretization algorithm could be improved to allow a finer description of the geometries. Currently, an image is used as a mask on a random packing to set the properties of the particles or to modify them. The relaxation that follows induces a variation of shape. An iterative procedure, with several mask/relaxation procedures, would be helpful and a limited number of iterations is probably sufficient.

The behavior of the model *TRILIN*, able to cope with tensile and compressive load, may also be studied for arbitrary load and geometries, benchmarking the results with FEM simulations. The load applied in the simulation of the foam crushing are for example far from being uniaxial. More specifically, the analysis of the effect of the discretization would allow a quantification of the local geometrical errors.

The marginal modification of the model *TRILIN* to avoid numerical crystallization should not be too cumbersome and could greatly improve its behavior. A straightforward

approach to limit the crystallization is the use of a slight dispersion on the dimension of the particles. The impact on the macroscopic and local behavior must be investigated, to check whether the interaction law has to be adapted or can be used as-is. More generally, the extension of the model to handle elementary particles of distinct size could allow an adaptation of the discretization to local geometrical details. It is algorithmically possible, but the tuning of the modification of the model may not be a trivial task. Moreover, in a finite strain context, a dynamic size adaptivity would be required, where particles split or merge. This challenge sounds unreasonably complex at this preliminary stage.

Without entering into the details of the understanding of the transient regime, numerical work-around to initiate the flow would be useful. Two practical tools could be used: a judicious initial velocity of the particles (probably including some randomness) and the progressive rise of the mesh velocities. It seems important to stick to transient procedure, to avoid supplementary numerical artifact on the solution.

Along with these short-term and practical developments, more in-depth extension and studies can be imagined.

As a first example, a tempting extension of our phenomenological description of inelastic flow is the association with a model describing elastic behaviors. Indeed, numerous functional DEM models can describe elasticity in continuous media. Such elastic-plastic models would be of interest for example to study spring-back effects in cold forming, or foam crushing at room temperature. In both examples, contacts and large strains are involved, but the elastic contribution cannot be neglected. Conceptually and algorithmically, nothing prohibits the hybridization of such elastic models and our inelastic model. However, the hard point is the practical design of a local threshold between the two behaviors. The perturbations introduced by the inelastic model are probably too large with respect to the limited magnitude of elastic phenomena. As the displacement length scales are very distinct, a hybrid model might be illusory. Such a fine description of the continuous behavior may be better handled with other numerical tools as the FEM or the SPH. It may be possible to tackle this difficulty within our DEM by investigating the stress threshold for our numerical packings to flow. It has not been quantified in our work, but tuning this purely numerical artifact may allow to mimic partially the sought for behavior.

An interesting issue from the conceptual point of view is the modeling of the contacting surfaces. Our model allows the detection and the tracking of contact and self-contact in complex configurations. However, our discretization with spheres introduces a surface roughness, leading to a potential numerical friction, which is not controlled yet. Increasing the surface friction seems possible but implementing non frictional interfaces would be a challenge.

More fundamentally, investigating toward the understanding of the flow may help in three directions to improve the models:

- Better control the strain rate sensitivity.
- Design time converged models.

The time step has been considered as a numerical parameter, which is fundamentally not physical. Although this is not necessarily a conceptual obstacle, it limits the applicability of a given set of numerical parameters: too low strain rates are excessively time consuming. Limited hope must be granted to the possibility of reducing the global computing time, as the convergence in time seems to require much smaller steps.

- Design simpler models.

The proposed interaction law may be further simplified, once the key properties are clearly identified.

The last point is a prerequisite before algorithmic or computational optimizations of the code are attempted. Although a specific care to computational issue was maintained throughout this work, the “draft ” nature of the implementation also implies limited computing efficiency. Implementation choices favored the conceptual simplicity and the ease of test of various behaviors: numerous conditional statements may be simplified, although the potential speedup should be limited. To model larger geometries, the framework of the DEM solver allows a potentially massive parallelization. For effective speedups, the computing load must be correctly balanced between the processors and further developments would be necessary. In this PhD, the needs in computing power and the architectures of the available machines did not advocate for this development.

A conceptual drawback of the followed strategy is the delicate and time-consuming design of interaction laws. The design of an improved version of the model *TRILIN* could limit the numerical crystallization and limit the force jump when a pair of particles is separated, ideally with a simpler interaction law. However, the automation of the task is not trivial: the use of optimization tools is straightforward to quantitatively tune parameters in a well defined domain. However, how to define the metrics of interest, the objective function and the parameters when the general objective is qualitative?

The rough attempts during this PhD to use optimization tools often did not lead to satisfactory results. Automated screenings of parameters were impractical, as the domains to be explored are vast and the correct configurations narrow. Promising configurations tended to be missed and numerous useless computations were run. A very manual trial-and-error approach remained the most effective strategy, specifically in the objective of finding *one* solution and not necessarily an optimum. The underlying assumption was an intrinsic doubt regarding the very definition of potential optima.

The choice of the numerical parameter is *de facto* an ongoing challenge for the DEM community, including for conventional applications of this numerical method. The proposition of tuning the interaction law itself instead of the parameters may be of interest. The key issue of choosing appropriate objective functions for optimization procedures remains open.

Overall, it seems that a shift from parametric optimization based on an *a priori* chosen algorithm is only efficient for a well understood model: a model where the link between the macroscopic flow and the elementary interactions can be described. This deeper understanding of the flow process allows to effectively restrict and define the to-be-explored domain of parameters and limits the algorithmic sensitivity.

In a situation with little understanding of the link between the scales, an *a priori* algorithmic choice may be misleading. Useful tools might be found in computing science or statistics to design *ad hoc* algorithms instead of tuning parameters. In the design of interaction laws to mimic a continuous behavior, the principal component analysis may help to discriminate the most sensitive and decisive elementary features. For classifications of pairs as in our self-contact detection algorithm, tools from data mining techniques, as supervised learning, may provide automated and reliable approaches.

Appendices

A	Local Field	205
B	DEM Implementation	209
B.1	Implementation Choices and Issues	209
B.2	Sources: Interaction Law <i>BILIN</i>	210
B.3	Sources: Interaction Law <i>TRILIN</i> , Contact and Self-Contact	212
B.3.1	normal_model_zcherry_outward.h	212
B.3.2	fix_outward.h	217
B.3.3	fix_outward.cpp	218
B.4	Typical Input Script	222
B.4.1	Input Script Template	223
B.4.2	Test Packing	225
B.4.3	Typical Planar Mesh	225
B.4.4	Packing Script Template	225
C	FEM Script	227
D	IJMS Article	231
	Notation	259
	Glossary	261
	Bibliography	265

Appendix A

Local Field

The first qualitative comparison of the discrete element method (DEM) and the finite element method (FEM) local stress fields are promising. Further statistical analysis is required to draw quantitative conclusions.

The local stress field, reconstructed at the level of the particles from Equation 8.12 on page 96, is not meaningful if taken instantaneous and particle-wise. As for the macroscopic stress, the stress is temporally averaged over a sliding window, here using a width of $5 \cdot 10^{-3}$ in strain. This procedure has to be executed component- and particle-wise, at run time. The storage of the particle-wise data at all time steps is too impractical for this procedure to be post-treated.

Even temporally averaged, the particle-wise stress is still too noisy and rough to be interpreted as representing a stress field in a continuous medium. It is then spatially averaged in post-treatment, by two means:

Implicitly: The estimation of the volume occupied by a particle (necessary in Equation 8.12) is here executed in a global fashion¹. The global relaxed volume of the packing is divided by the number of particles.

Explicitly: The stress at a particle is component-wise averaged with the values at the neighboring particles within a cutoff radius, using ovito.

The local stress field component σ_{zz} is illustrated independently for both phases: B (Figure A.1) and A (Figure A.2).

Without applying any neighbor averaging, a common displayed trend, is the presence of a thin tension zone in the periphery of the sample. The presence of the mesh also is a source of perturbation of the field. As expected in a DEM approach, the forces are carried by preferential chains and the stress is not evenly distributed. While the packing deforms, the force chains change and reorganize.

The neighbor average operates with shorter cutoff for the phase with slowest reactivity, B. For both phases, the averaged local stresses tend to be overestimated compared to the macroscopic stress. Depending on the required precision, a averaging length can be chosen. A length of $1.5r_{\text{seed}}$, corresponding to the average over the neighbor detected in the computation, is still very rough. A length between 2 and $3r_{\text{seed}}$, thus allowing the use of second neighbors, is still numerically cheap, while smoothing most of the roughest gradients. As an order of magnitude, using an averaging radius of $3r_{\text{seed}}$ and exception

¹This method is much cheaper than estimating a local volume around the particle from the position of its neighbor. In addition, the actual instantaneous distance between two particles is meaningless in our objective to model continuous media: we are only interested in the forces carried by a particle.

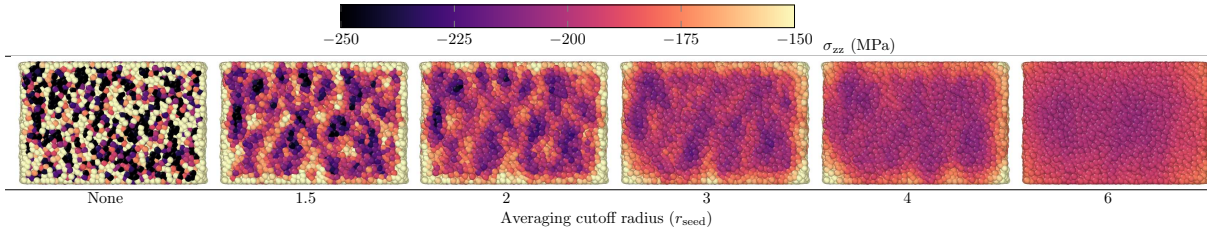


Figure A.1: Cross-section of a single material for a strain of 0.3, phase B. Map of the local stress component σ_{zz} , the measured macroscopic stress at 10^{-3} s^{-1} is 175 MPa.

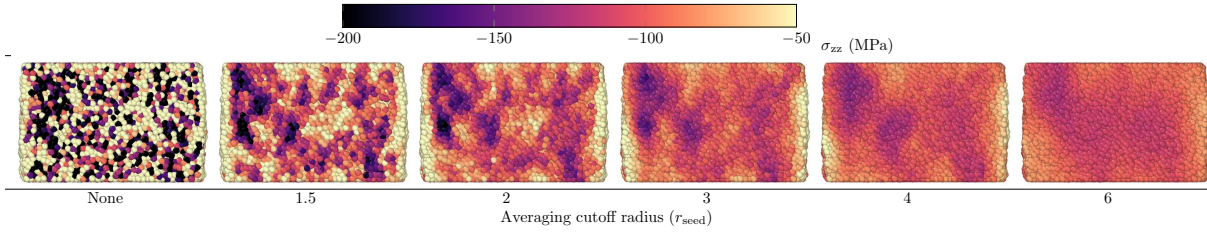


Figure A.2: Cross-section of a single material for a strain of 0.3, phase A. Map of the local stress component σ_{zz} , the measured macroscopic stress at 10^{-3} s^{-1} is 94 MPa.

made of the boundaries of the domain, the typical variation of local stress is $\pm 25 \text{ MPa}$. The relative error is thus higher for the phase A. Guidelines can thus be deduced regarding the necessary discretization of morphological features in multi-materials.

The local stress estimation is applied to the unique inclusion test case (Section 12.4). At the chosen discretization of $5 \cdot 10^4$ particles, chosen to capture the macroscopic shape of the inclusion, the smaller thickness of the matrix is only discretized by a few particle. The averaging length thus corresponds to the typical observed length. Moreover, the proximity of the planar mesh is a source of perturbation for the local fields. The stress field presented here are thus bound to be rough.

Quantitatively (Figure A.3), at 10^{-3} s^{-1} a high stress zone is captured at the north and south poles of the inclusion. At this rough discretization, a “radial” averaging procedure would greatly help to reproduce better the axisymmetric pattern, but this would mean favoring an *a priori* known solution. At 10^{-4} s^{-1} , the stress is more evenly spread, with slightly higher values in the periphery of the matrix. Although the distribution is as expected rough, the quantitative trends can be correctly modeled.

Using the interaction *BILIN*, dominated by a repulsive behavior with $k_{\text{att}} = k_{\text{rep}}/10$ (Table 11.1), the other components of the stress tensor $\bar{\sigma}$ are in principle not properly captured. However, the qualitative distribution of the von Mises equivalent stress σ_{eq} is correct (Figure A.4), as the deformation is dominated by compression. As a side note, the equivalent stress can only be captured using the spatially averaged components of the stress tensor. Computing the equivalent stress particle-wise and then spatially averaging it does not lead to any sensible result.

The distinct response of the two phases can clearly be identified (Figure A.4), even though the equivalent stress is underestimated by a factor three. An interesting critic of the model can be drawn from the comparison of the relative stresses between matrix and inclusion. In the FEM, the flow stress is higher in the matrix at 10^{-4} s^{-1} and higher in the inclusion at 10^{-3} s^{-1} . In the *BILIN* model, the stress in the inclusion does increase with

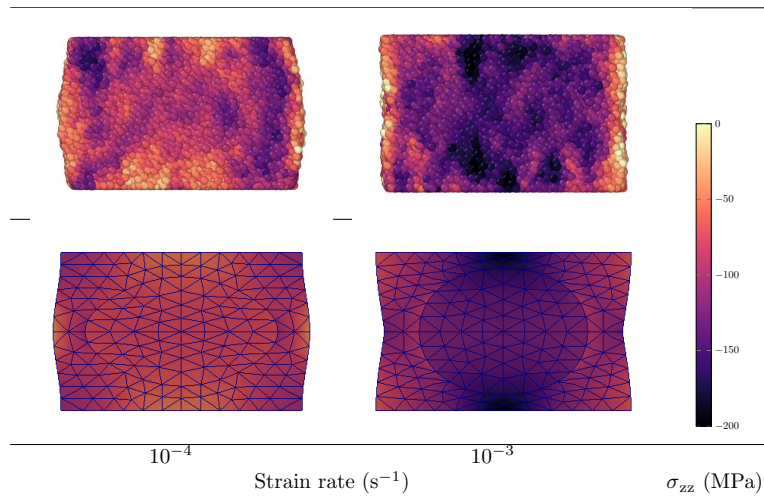


Figure A.3: Spherical inclusion test case. Local stress component σ_{zz} (in the compression direction).

the strain rate, but at 10^{-3} s^{-1} the stress in the inclusion is still lower than in the matrix. The macroscopic “hourglass” shape, clearly displayed in FEM, will only be captured at higher strain rate.

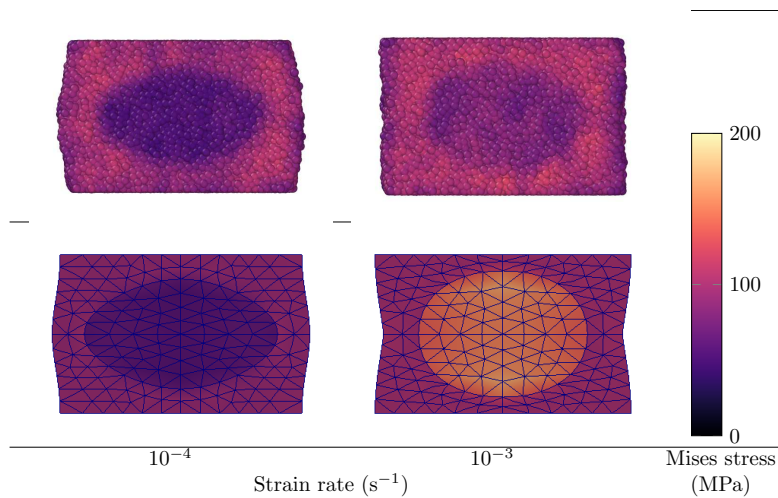


Figure A.4: Spherical inclusion test case. Local Mises equivalent stress σ_{eq} .

Appendix B

DEM Implementation

A very brief discussion of some implementation issues is proposed in Section B.1. The key source codes corresponding to Parts IV and V are given respectively in Sections B.2 and B.3. A typical template of input script, along with all the numerical parameters introduced, is proposed in Section B.4. The files were written for liggghts 3.5. No changes should be necessary for compiling and running the examples with liggghts 3.7, although it was not tested.

B.1 Implementation Choices and Issues

Along with a short discussion of some implementation issues, a brief overview of the implemented features is proposed in Table B.1.

The implementation of custom interaction laws (*BILIN* and *TRILIN*), with arbitrary parameters, is rather straightforward in liggghts: an independent file (conventionally named *normal_*.h*) is simply added to the sources.

From the implementation point of view, the mesh/particle interaction are out-of-the-box proposed in liggghts. The efforts should have been limited to the definition of the appropriate interaction behavior in the input scripts. Additional work was necessary to by-pass various errors, including segmentation faults, arising in parallel computation of geometrically large packings. The two operational work-around were:

- The definition of a domain substantially larger than the packing itself.
- The use of fixed boundaries (with the command *boundary f*) and not adaptive boundaries (*boundary s* or *m*).

The handling of pair-wise state variables (for example the time since the interaction started for the healing behavior in self-contact) is handled directly within the interaction laws. Particle-wise state variables (for example r_{seed} or \bar{n}) are introduced in interaction laws using *fix_* routines, defined in separate *fix_* files. This is necessary for the proper communication of the variables between processors when the message passing interface (MPI) parallelization is used.

Even though this procedure seemed correct for most cases, a MPI bug was spotted for the self-contact detection in the interaction law *TRILIN* and has not been corrected yet. All goes as if the \bar{n} vectors are not properly exchanged between processors (Figure B.1). The serial implementation works as expected, without detected bugs. The self-contact simulations shown in Part V are thus run on a single processor.



Figure B.1: Illustration of a MPI bug. Compression of a dense sample, using the self-contact detection. Top view, no transparency for the particles with a least one “interface” detected. (a) MPI grid $1 \times 1 \times 1$. Correct behavior: no interface detected. (b) MPI grid $2 \times 2 \times 1$. Erroneous behavior: interfaces detected at the boundaries of the processors. The self-contact variables are not correctly passed between processors.

Conceptually, the proposed self-contact modeling implies that the force in a pair of particle depends on the state of the system in a neighborhood: a non-local behavior. The outward vector \bar{n} is computed for each particle from the position of its neighbors, only taking into account the “internal” neighbors. At each step, \bar{n} must be computed, taking into account the “internal” or “interface” status of the pairs. This coupling between a particle-wise variable (\bar{n}) and a pair-wise variable (the status of the pair) is not trivial. In the proposed implementation, most of the computation is executed in the interaction law.

Physical model	Algorithmic feature	Tool	Source code in section	Implementation effort
		liggghts		
Inelastic strain	Interaction law <i>BILIN</i>		B.2	1
Inelastic strain	Interaction law <i>TRILIN</i>		B.3	1
Prescribed strain	Mesh/particle interaction		B.2, B.3	0.1
Contact (distinct objects)	Particle state variable, interaction law		B.3	1
Self-contact	Interaction law, non-local behavior, coupling between pair and particle state variable		B.3	10
		python,numpy		
Material discretization	Image/packing mask		Not provided	1

Table B.1: Overview of the implemented features. Cross-reference to the source codes, when provided. Indication of the relative implementation effort.

B.2 Sources: Interaction Law *BILIN*

The simulations of Part IV are run using the interaction law *BILIN*, defined by the file *normal_model.zcherry_faistenau.h*.

```

1  /* -----
2  file : normal_model_zcherry_faistenau.h
3  type : liggghts 3.5.0 contact model
4  ----- */
5
6  #ifndef NORMAL_MODEL
7  NORMAL_MODEL(ZCHERRY_FAISTENAU, zcherry/faistenau, 5)
8  #else
9  #ifndef NORMAL_MODEL_ZCHERRY_FAISTENAU_H
10 #define NORMAL_MODEL_ZCHERRY_FAISTENAU_H
11 #include "contact_models.h"
12 #include "fix_property_atom.h"
13 #include <iostream>
14
15 namespace LIGGGHTS {
16     namespace ContactModels
17     {
18         // Functions to create custom pair parameters
19         MatrixProperty* createKrep(PropertyRegistry & registry, const char * caller, bool
            sanity_checks);
20         static const char * KREP = "krep";
21         MatrixProperty* createKrep(PropertyRegistry & registry, const char * caller, bool
            sanity_checks)
22         {
23             return MODEL_PARAMS::createPerTypePairProperty(registry, KREP, caller);
24         }
25         MatrixProperty* createKatt(PropertyRegistry & registry, const char * caller, bool
            sanity_checks);
26         static const char * KATT = "katt";
27         MatrixProperty* createKatt(PropertyRegistry & registry, const char * caller, bool
            sanity_checks)
28         {
29             return MODEL_PARAMS::createPerTypePairProperty(registry, KATT, caller);
30         }
31     }
32
33     template<
34     class NormalModel<ZCHERRY_FAISTENAU> : protected Pointers
35     {
36     public:
37         static const int MASK = CM_REGISTER_SETTINGS | CM_CONNECT_TO_PROPERTIES |
            CM_SURFACES_INTERSECT;
38         NormalModel(LAMMPS * lmp, iContactHistorySetup*, class ContactModelBase *) :
            Pointers(lmp),
39             // Custom pair parameter
40             k_rep(NULL),
41             k_att(NULL),
42             // Custom particle parameter
43             fix_seed(0),
44             // Cases
45             velocity(false)
46         {} // Necessary to compile
47
48         void registerSettings(Settings & settings) {
49             settings.registerOnOff("velocity", velocity, true);
50         }
51
52         inline void postSettings() {} // Necessary to compile
53
54         void connectToProperties(PropertyRegistry & registry) {
55             // Pair parameters
56             // Create parameters
57             registry.registerProperty("k_rep", &createKrep);
58             registry.registerProperty("k_att", &createKatt);
59             // Retrieve value from script
60             registry.connect("k_rep", k_rep, "model zcherry/faistenau");
61             registry.connect("k_att", k_att, "model zcherry/faistenau");
62
63             // Particle parameter, create and retrieve
64             fix_seed = static_cast<FixPropertyAtom*>(modify->find_fix_property("seed", "
                property/atom", "scalar", 0, 0, "normal_model zcherry/faistenau"));
65         }
66
67         // Necessary to compile
68         inline double stressStrainExponent() {return 1.;}
69
70         inline void surfacesIntersect(SurfacesIntersectData & sidata, ForceData & i_forces
            , ForceData & j_forces)
71         {
72             // Used variables
73             const double *seed = fix_seed->vector_atom;
74             const int i = sidata.i;
75             const int j = sidata.j;
76             const int itype = sidata.itype;
77             const int jtype = sidata.jtype;
78             double krep = k_rep[itype][jtype]; // Modified if wall/particle
79             double katt = k_att[itype][jtype];
80
81             const double seedradsum = 0.5*(seed[i]+seed[j]);
82
83             // Compute and apply normal force
84             const double Fn_damping = 0; // No damping implemented
85             double Fn_contact;
86
87             if(sidata.is_wall) { // Wall/particle interaction
88                 double krep = k_rep[itype][itype]; // Set to particle properties
89                 double katt = k_att[itype][itype];
90                 if(sidata.r < 0.5*seedradsum) { // Seed intersection
91                     Fn_contact = krep*(0.5*seedradsum-sidata.r);
92                 }
93                 else { // Crown intersection
94                     if(sidata.vn > 0 or !velocity) { // Tensile relative velocity
95                         Fn_contact = katt*(0.5*seedradsum-sidata.r);
96                     }
97                 }
98             }
99         }
100     };
101 }

```



```

96         }
97         else { // Compressive relative velocity
98             Fn_contact = 0;
99         }
100     }
101     double Fn = Fn_damping + Fn_contact;
102     // Store normal force
103     sidata.Fn = Fn;
104     i_forces.delta_F[0] = Fn * sidata.en[0];
105     i_forces.delta_F[1] = Fn * sidata.en[1];
106     i_forces.delta_F[2] = Fn * sidata.en[2];
107 }
108 else { // Particle/particle interaction
109     if (sidata.r < seedradsum) { // Seed intersection
110         Fn_contact = krep*(seedradsum - sidata.r);
111     }
112     else { // Crown intersection
113         if (sidata.vn > 0 or !velocity) { // Tensile relative velocity
114             Fn_contact = katt*(seedradsum - sidata.r);
115         }
116         else { // Compressive relative velocity
117             Fn_contact = 0;
118         }
119     }
120     double Fn = Fn_damping + Fn_contact;
121     // Store normal force
122     sidata.Fn = Fn;
123     i_forces.delta_F[0] = sidata.Fn * sidata.en[0];
124     i_forces.delta_F[1] = sidata.Fn * sidata.en[1];
125     i_forces.delta_F[2] = sidata.Fn * sidata.en[2];
126     j_forces.delta_F[0] = -i_forces.delta_F[0];
127     j_forces.delta_F[1] = -i_forces.delta_F[1];
128     j_forces.delta_F[2] = -i_forces.delta_F[2];
129 }
130 }
131
132 void surfacesClose(SurfacesCloseData&, ForceData&, ForceData&){}
133 void beginPass(SurfacesIntersectData&, ForceData&, ForceData&){}
134 void endPass(SurfacesIntersectData&, ForceData&, ForceData&){}
135
136 protected:
137     double ** k_rep;
138     double ** k_att;
139     FixPropertyAtom *fix_seed;
140     bool velocity;
141 };
142 }
143 }
144 #endif // NORMAL_MODEL_ZCHERRY_FAISTENAU_H
145 #endif

```

B.3 Sources: Interaction Law *TRILIN*, Contact and Self-Contact

The simulations of Part V are run using the interaction law *TRILIN* and the self-contact algorithm, defined by the file *normal_model_zcherry_outward.h*, *fix_outward.h* and *fix_outward.cpp*.

B.3.1 normal_model_zcherry_outward.h

```

1  /* -----
2  file : normal_model_zcherry_outward.h
3  type : liggghts 3.5.0 contact model
4  -----
5  Contact detection between free interfaces
6  Based on outward normal vector, computed from - Sum (branch vector of neighbors)
7
8  fix_outward.h
9  fix_outward.cpp
10 management of particle-wise variables, with proper
11 mpi communication
12 memory management
13 normal_model_zcherry_outward.h
14 use already computed si.data contact-wise variables
15 avoids loop on neighbors twice
16 make necessary to double variable (current and next) stored (cheap, so its ok)
17 avoids re-code manually contact detection in fix
18 easy contact-wise variables
19 easy potential particle-wall behavior
20
21 Particle-wise variable
22 fix_seed seed (repulsive) diameter of the particle
23 fix_cluster membership to a aggregate for contact detection
24 fix_outward normal outward pointing vector, from previous time step
25 fix_outward_mag magnitude of fix_outward, from previous time step
26 fix_outward_next normal outward pointing vector, contributed to during the current time
27 step
28 fix_outward_numberNeigh number of contacts not across an interface, from previous time step

```

B.3. SOURCES: INTERACTION LAW TRILIN, CONTACT AND SELF-CONTACT²¹³

```

28 fix_outward_number_next number of contacts not across an interface, contributed to during
   the current time step
29 fix_outward_indent sum of the seed indentation on each particle: (seed1+seed2)-distance (
   always whole, particle/particle or particle/wall)
30 fix_outward_indent_sqr sum of the potential energy (square of the seed indentation times
   the rigidity): 0.5*krep*((seed1+seed2)-distance)^2 (half if particle/particle, whole in
   particle/wall)
31 Contact-wise variable
32 past_interface history parameter storing status of the contact:
33 first step of interaction
34 unknown 0
35 n steps of interaction
36 interface n*dt
37 inside -n*dt
38 Interaction parameter
39 mag_out threshold on magnitude of outward vector
40 cos_ij threshold on angle between outward vectors
41 cos_en threshold on angle between outward vector and position difference
42 non_local switch for non-local behavior
43 wall multiplicative factor for wall/particle attraction
44 heal time after which an "interface" interaction is turned into a "bulk" interaction
45 verb output testing data
46
47 Warning
48 also a (currently) hard-coded parameter excluding the new contact from the first 500 steps
   to be interfaces: corresponding to the relaxation time used in the simulations.
49 ----- */
50
51 #ifndef NORMAL_MODEL
52 NORMAL_MODEL(ZCHERRY_OUTWARD, zcherry/outward, 6)
53 #else
54 #ifndef NORMAL_MODEL_ZCHERRY_OUTWARD_H
55 #define NORMAL_MODEL_ZCHERRY_OUTWARD_H
56 #include "contact_models.h"
57 #include "fix_property_atom.h"
58 #include <iostream>
59 #include <math.h>
60
61 namespace LIGGGHTS {
62 namespace ContactModels
63 {
64 // Functions to create custom pair parameters
65 // Piece-wise linear elementary cherry
66 MatrixProperty* createKrep6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks); // createKrep6 has to be unique
67 static const char * KREP6 = "krep"; // KREP has to be unique
68 MatrixProperty* createKrep6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
69 {
70 return MODEL_PARAMS::createPerTypePairProperty(registry, KREP6, caller);
71 }
72 MatrixProperty* createKatt6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks);
73 static const char * KATT6 = "katt";
74 MatrixProperty* createKatt6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
75 {
76 return MODEL_PARAMS::createPerTypePairProperty(registry, KATT6, caller);
77 }
78 // Criteria for interface
79 ScalarProperty* createMagOut(PropertyRegistry & registry, const char * caller, bool
   sanity_checks);
80 static const char * MAGOUT = "magOut";
81 ScalarProperty* createMagOut(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
82 {
83 return MODEL_PARAMS::createScalarProperty(registry, MAGOUT, caller);
84 }
85 ScalarProperty* createCosIJ(PropertyRegistry & registry, const char * caller, bool
   sanity_checks);
86 static const char * COSIJ = "cosIJ";
87 ScalarProperty* createCosIJ(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
88 {
89 return MODEL_PARAMS::createScalarProperty(registry, COSIJ, caller);
90 }
91 ScalarProperty* createCosEN(PropertyRegistry & registry, const char * caller, bool
   sanity_checks);
92 static const char * COSEN = "cosEN";
93 ScalarProperty* createCosEN(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
94 {
95 return MODEL_PARAMS::createScalarProperty(registry, COSEN, caller);
96 }
97 // Traction behavior parameter
98 MatrixProperty* createFatt6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks);
99 static const char * FATT6 = "fatt";
100 MatrixProperty* createFatt6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
101 {
102 return MODEL_PARAMS::createPerTypePairProperty(registry, FATT6, caller);
103 }
104 static const char * WALL6 = "wall";
105 ScalarProperty* createWall6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
106 {
107 return MODEL_PARAMS::createScalarProperty(registry, WALL6, caller);
108 }
109 static const char * HEAL6 = "heal";
110 ScalarProperty* createHeal6(PropertyRegistry & registry, const char * caller, bool
   sanity_checks)
111 {

```

```

112     return MODEL_PARAMS::createScalarProperty(registry, HEAL6, caller);
113 }
114
115
116 template<
117 class NormalModel<ZCHERRY_OUTWARD> : protected Pointers
118 {
119     int history_offset;
120 public:
121     static const int MASK = CM_REGISTER_SETTINGS | CM_CONNECT_TO_PROPERTIES |
122         CM_SURFACES_INTERSECT;
123     NormalModel(LAMMPS * lmp, IContactHistorySetup* hsetup, class ContactModelBase *c)
124         : Pointers(lmp),
125         cmb(c), //not sure it's useful, copied from tangential_model_history.h
126         // Custom pair parameter
127         k_rep(NULL),
128         k_att(NULL),
129         mag_out(0),
130         cos_ij(0),
131         cos_en(0),
132         f_att(NULL),
133         // Custom scalar parameter
134         w_all(0.0),
135         h_eal(0.0),
136         // Custom particle parameter
137         fix_seed(0),
138         fix_cluster(0),
139         fix_outward(0),
140         fix_outward_mag(0),
141         fix_outward_next(0),
142         fix_outward_numberNeigh(0),
143         fix_outward_numberNext(0),
144         fix_outward_indent(0),
145         fix_outward_indent_sqr(0),
146         fix_outward_error_inte(0),
147         fix_outward_error_bulk(0),
148         fix_outward_good_inte(0),
149         fix_outward_good_bulk(0),
150         // Cases
151         velocity(false),
152         non_local(false),
153         verb(false)
154     {
155         history_offset = hsetup->add_history_value("past_interface", "0"); // flag "0",
156         // parameter is symmetric
157     }
158
159     void registerSettings(Settings & settings) {
160         settings.registerOnOff("velocity", velocity, true);
161         settings.registerOnOff("non_local", non_local, true); // default: true
162         settings.registerOnOff("verb", verb); // default: false
163     }
164
165     inline void postSettings() {} // Necessary to compile
166
167     void connectToProperties(PropertyRegistry & registry) {
168         // Pair parameters
169         // Create parameters
170         registry.registerProperty("k_rep", &createKrep6);
171         registry.registerProperty("k_att", &createKatt6);
172         registry.registerProperty("mag_out", &createMagOut);
173         registry.registerProperty("cos_ij", &createCosIJ);
174         registry.registerProperty("cos_en", &createCosEN);
175         registry.registerProperty("f_att", &createFatt6);
176         registry.registerProperty("w_all", &createWall6);
177         registry.registerProperty("h_eal", &createHeal6);
178         // Retrieve value from script
179         registry.connect("k_rep", k_rep, "model zcherry/outward");
180         registry.connect("k_att", k_att, "model zcherry/outward");
181         registry.connect("mag_out", mag_out, "model zcherry/outward");
182         registry.connect("cos_ij", cos_ij, "model zcherry/outward");
183         registry.connect("cos_en", cos_en, "model zcherry/outward");
184         registry.connect("f_att", f_att, "model zcherry/outward");
185         registry.connect("w_all", w_all, "model zcherry/outward");
186         registry.connect("h_eal", h_eal, "model zcherry/outward");
187
188         // Particle parameter, create and retrieve
189         fix_seed = static_cast<FixPropertyAtom*>(modify->find_fix_property("seed", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
190         fix_cluster = static_cast<FixPropertyAtom*>(modify->find_fix_property("cluster", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
191         fix_outward = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward", "property/atom", "vector", 3, 0, "normal_model zcherry/outward"));
192         fix_outward_mag = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_mag", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
193         fix_outward_next = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_next", "property/atom", "vector", 3, 0, "normal_model zcherry/outward"));
194         fix_outward_numberNeigh = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_numberNeigh", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
195         fix_outward_numberNext = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_numberNext", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
196         fix_outward_indent = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_indent", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
197         fix_outward_indent_sqr = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_indent_sqr", "property/atom", "scalar", 0, 0, "normal_model zcherry/outward"));
198         fix_outward_error_inte = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_error_inte", "property/atom", "scalar", 0, 0, "normal_model zcherry/

```

B.3. SOURCES: INTERACTION LAW TRILIN, CONTACT AND SELF-CONTACT 215

```

197         outward));
198     fix_outward_error_bulk = static_cast<FixPropertyAtom*>(modify->find_fix_property(
199         "outward_error_bulk", "property/atom", "scalar", 0, 0, "normal.model zcherry/
200         outward"));
201     fix_outward_good_inte = static_cast<FixPropertyAtom*>(modify->find_fix_property(
202         "outward_good_inte", "property/atom", "scalar", 0, 0, "normal.model zcherry/
203         outward"));
204     fix_outward_good_bulk = static_cast<FixPropertyAtom*>(modify->find_fix_property(
205         "outward_good_bulk", "property/atom", "scalar", 0, 0, "normal.model zcherry/
206         outward"));
207 }
208 // Necessary to compile
209 inline double stressStrainExponent() {return 1.;}
210
211 inline void surfacesIntersect(SurfacesIntersectData & sidata, ForceData & i_forces,
212                             ForceData & j_forces)
213 {
214     ///////////////////////////////////////////////////////////////////
215     // Retrieve parameters
216     const double dt = update->dt;
217     const double dt_zero = dt/10; // time considered negligible
218     bigint step = update->ntimestep;
219     // Interface history
220     if(sidata.contact_flags) *sidata.contact_flags |= CONTACT_NORMAL_MODEL;
221     double * const past_interface = &sidata.contact_history[history_offset];
222     bool new_contact = fabs(past_interface[0]) < dt_zero; // the contact has just
223     // been created
224     double heal = h_eal;
225     bool heal_contact = fabs(past_interface[0]) > heal; // the contact exited
226     // for longer than the healing time
227     //std::cout << std::boolalpha << new_contact << "\t" << past_interface[0] << "\n"
228     // << fabs(past_interface[0]) << "\n";
229     bool interface = past_interface[0] > dt_zero; // particles interacting
230     // through an interface (or particle/wall interface)
231     // Outward data from previous time step
232     double ** outward = fix_outward->array_atom;
233     double *outward_mag = fix_outward_mag->vector_atom;
234     //double *outward_numberNeigh = fix_outward_numberNeigh->vector_atom;
235     // Outward data contributed to during the time step
236     double ** outward_next = fix_outward_next->array_atom;
237     double *outward_number_next = fix_outward_number_next->vector_atom;
238     double *outward_indent = fix_outward_indent->vector_atom;
239     double *outward_indent_sqr = fix_outward_indent_sqr->vector_atom;
240     // Outward verification data
241     double *outward_error_inte = fix_outward_error_inte->vector_atom; // errors on
242     // interface interaction
243     double *outward_error_bulk = fix_outward_error_bulk->vector_atom; // errors on
244     // bulk interaction
245     double *outward_good_inte = fix_outward_good_inte->vector_atom; // goods on
246     // interface interaction
247     double *outward_good_bulk = fix_outward_good_bulk->vector_atom; // goods on bulk
248     // interaction
249     // Surface intersect data
250     const int i = sidata.i;
251     const int j = sidata.j;
252     const int itype = sidata.itype;
253     const int jtype = sidata.jtype;
254     double * delta = sidata.delta; // position Xi-Xj pair_gran_base:257,265
255     const double enx = sidata.en[0]; // unit normal vector from j center to i
256     // center
257     const double eny = sidata.en[1];
258     const double enz = sidata.en[2];
259     // Interaction parameters
260     double krep = krep[itype][jtype]; // Modified if wall/particle, not
261     // set as constant
262     double katt = katt[itype][jtype];
263     const double magnitude = mag_out; // magnitude threshold of the normal
264     // vectors
265     const double cos_outI_outJ = cos_ij; // angle threshold
266     const double cos_out_en = cos_en; // angle threshold
267     double fatt = f_att[itype][jtype];
268     double wall = w_all;
269     // Interaction computed
270     int clu_i, clu_j;
271     double mag_i, mag_j;
272     double cos_ij;
273     double cos_i, cos_j;
274
275     // Double radius cherry model parameter
276     double *seed = fix_seed->vector_atom;
277     double *cluster = fix_cluster->vector_atom;
278     //const double seedradsum = 1.0; // Hard-coded for mpi debugging
279     // purpose
280     const double seedradsum = 0.5*(seed[i]+seed[j]);
281
282     ///////////////////////////////////////////////////////////////////
283     // Type of the new interactions
284     // The defined status is stored
285     if (new_contact) {
286         // Only relaxation time and for particle/particle interactions (exclude
287         // meaningless particle/wall)
288         if ((step > 500) and (not sidata.is_wall)) {
289             // Contact between distinct objects (with cluster membership)
290             clu_i = (int)round(cluster[i]); // (int) cast not necessary. preferred to
291             // state it explicitly
292             clu_j = (int)round(cluster[j]);
293             if (clu_i == clu_j) {
294                 // Particles are in the same cluster
295                 // Self-contact detection
296                 // Magnitude of outward normal
297                 mag_i = outward_mag[i]; // = mag_i;

```

```

278     mag-j = outward_mag[j]; // = mag-j;
279     // Angle (outward[i],outward[j])
280     cos_ij = (outward[i][0]*outward[j][0] + outward[i][1]*outward[j][1] +
281              outward[i][2]*outward[j][2])/(mag_i*mag_j);
282     // Angles (outward[i],IJ) and (outward[j],JI)
283     cos_i = -1*(outward[i][0]*enx + outward[i][1]*eny + outward[i][2]*enz)/
284             mag_i;
285     cos_j = +1*(outward[j][0]*enx + outward[j][1]*eny + outward[j][2]*enz)/
286             mag_j;
287     if (cos_ij<=cos_outI_outJ) {
288         if ((mag_i>=magnitude and cos_i>=cos_out_en) or (mag_j>=magnitude and
289             cos_j>=cos_out_en)) {
290             interface = true;
291         }
292     } else {
293         // Particles are in distinct cluster
294         // Contact between objects
295         interface = true;
296     }
297 }
298
299 ////////////////////////////////////////////////////
300 // Contribute to next outward vector value and history values
301 // The normalization by the number of neighbors could be done at the end of the
302 // time step in fix_outward.cpp
303 // It seems to the normalization leads to too small vector for concave (many
304 // neighbors) zones
305 if (interface) {
306     if (heal_contact){
307         // Healed contact
308         // Not considered as an interface anymore
309         past_interface[0] = -1*dt;
310         interface = false;
311     } else {
312         // Interface contact
313         // Increment the contact time
314         past_interface[0] += dt;
315     }
316 } else {
317     past_interface[0] -= dt;
318     outward_next[i][0] += delta[0];
319     outward_next[i][1] += delta[1];
320     outward_next[i][2] += delta[2];
321     outward_number_next[i] += 1;
322     if (force->newton_pair or j<atom->nlocal) {
323         outward_next[j][0] -= delta[0];
324         outward_next[j][1] -= delta[1];
325         outward_next[j][2] -= delta[2];
326         outward_number_next[j] += 1;
327     }
328 }
329
330 ////////////////////////////////////////////////////
331 // Testing output
332 if ((step>500) and verb and new_contact) {
333     std::string preamble;
334     if (interface) {
335         // Detected as interface
336         if (itype==jtype) { // Incorrect
337             preamble = "[Drawtuo] ii:\t0\t0\t";
338             outward_error_inte[i] += 1;
339             if (force->newton_pair or j<atom->nlocal) {
340                 outward_error_inte[j] += 1;
341             }
342         } else { // Correct
343             preamble = "[Outward] ij:\t1\t1\t";
344             outward_good_inte[i] += 1;
345             if (force->newton_pair or j<atom->nlocal) {
346                 outward_good_inte[j] += 1;
347             }
348         }
349     } else {
350         // Detected as bulk
351         if (itype==jtype) { // Correct
352             preamble = "[Outward] ii:\t1\t0\t";
353             outward_good_bulk[i] += 1;
354             if (force->newton_pair or j<atom->nlocal) {
355                 outward_good_bulk[j] += 1;
356             }
357         } else { // Incorrect
358             preamble = "[Drawtuo] ij:\t0\t1\t";
359             outward_error_bulk[i] += 1;
360             if (force->newton_pair or j<atom->nlocal) {
361                 outward_error_bulk[j] += 1;
362             }
363         }
364     }
365 }
366
367 ////////////////////////////////////////////////////
368 // Compute and apply normal forces
369 // Interface behavior only implemented for particle-particle contact
370 const double Fn_damping = 0; // No damping implemented
371 double Fn_contact;
372 double Fn_elas;
373 double Fn_cons;
374
375 // Wall/particle interaction, interface not implemented
376 if (sidata.is_wall) {
377     double krep = k.rep[itype][itype]; // Set to particle properties
378     double katt = k.att[itype][itype];
379     if (sidata.r<0.5*seedradsum) { // Seed intersection
380         Fn_contact = krep*(0.5*seedradsum-sidata.r);
381     }
382 }

```

B.3. SOURCES: INTERACTION LAW TRILIN, CONTACT AND SELF-CONTACT²¹⁷

```

375         outward_indent[i] += 0.5*seedradsum-sidata.r; // Cumulate the indentation
376         outward_indent_sqr[i] += 0.5*krep*(0.5*seedradsum-sidata.r)*(0.5*seedradsum-
        sidata.r); // Cumulate the seed potential elastic energy
377     }
378     else { // Crown intersection
379         Fn_elas = katt*(0.5*seedradsum-sidata.r);
380         Fn_cons = -1.0*fatt;
381         Fn_contact = max(Fn_elas, Fn_cons); // take the closest to zero
382         Fn_contact *= wall;
383     }
384     double Fn = Fn_damping + Fn_contact;
385     // Store normal force
386     sidata.Fn = Fn;
387     i_forces.delta_F[0] = Fn * sidata.en[0];
388     i_forces.delta_F[1] = Fn * sidata.en[1];
389     i_forces.delta_F[2] = Fn * sidata.en[2];
390 }
391 // Particle/particle interaction
392 else {
393     if (sidata.r < seedradsum) { // Seed intersection
394         Fn_contact = krep*(seedradsum-sidata.r);
395         outward_indent[i] += seedradsum-sidata.r; // Cumulate the indentation
396         outward_indent_sqr[i] += 0.25*krep*(seedradsum-sidata.r)*(seedradsum-sidata.
        r); // Cumulate half the seed potential elastic energy
397         if (force->newton_pair or j<atom->nlocal) {
398             outward_indent[j] += seedradsum-sidata.r;
399             outward_indent_sqr[j] += 0.25*krep*(seedradsum-sidata.r)*(seedradsum-
400                 sidata.r);
401         }
402     }
403     else { // Crown intersection
404         if (interface) { // rough test
405             Fn_contact = 0;
406         }
407         else {
408             if (sidata.vn>0 or (not velocity)) { // Tensile relative velocity
409                 Fn_elas = katt*(seedradsum-sidata.r);
410                 Fn_cons = -1.0*fatt;
411                 Fn_contact = max(Fn_elas, Fn_cons); // take the closest to zero
412             }
413             else { // Compressive relative velocity
414                 Fn_contact = 0;
415             }
416         }
417     }
418     double Fn = Fn_damping + Fn_contact;
419     // Store normal force
420     sidata.Fn = Fn;
421     i_forces.delta_F[0] = sidata.Fn * sidata.en[0];
422     i_forces.delta_F[1] = sidata.Fn * sidata.en[1];
423     i_forces.delta_F[2] = sidata.Fn * sidata.en[2];
424     j_forces.delta_F[0] = -i_forces.delta_F[0];
425     j_forces.delta_F[1] = -i_forces.delta_F[1];
426     j_forces.delta_F[2] = -i_forces.delta_F[2];
427 }
428 }
429 void surfacesClose(SurfacesCloseData & scdata, ForceData&, ForceData&){}
430 void beginPass(SurfacesIntersectData&, ForceData&, ForceData&){}
431 void endPass(SurfacesIntersectData&, ForceData&, ForceData&){}
432
433 protected:
434     class ContactModelBase *cmb;
435     double ** k_rep;
436     double ** k_att;
437     double mag_out;
438     double cos_ij;
439     double cos_en;
440     double ** f_att;
441     double w_all;
442     double h_eal;
443     FixPropertyAtom *fix_seed;
444     FixPropertyAtom *fix_cluster;
445     FixPropertyAtom *fix_outward;
446     FixPropertyAtom *fix_outward_mag;
447     FixPropertyAtom *fix_outward_next;
448     FixPropertyAtom *fix_outward_next;
449     FixPropertyAtom *fix_outward_numberNeigh;
450     FixPropertyAtom *fix_outward_numberNext;
451     FixPropertyAtom *fix_outward_indent;
452     FixPropertyAtom *fix_outward_indent_sqr;
453     FixPropertyAtom *fix_outward_error_inte;
454     FixPropertyAtom *fix_outward_error_bulk;
455     FixPropertyAtom *fix_outward_good_inte;
456     FixPropertyAtom *fix_outward_good_bulk;
457     bool velocity;
458     bool non_local;
459     bool verb;
460 }
461 }
462 }
463 #endif // NORMAL_MODEL_ZCHERRY_OUTWARD_H
464 #endif

```

B.3.2 fix_outward.h

```

1  /* -----
2  file : fix_outward.h
3  type : liggghts 3.5.0 fix
4  ----- */
5

```

```

6
7 #ifndef FIX.CLASS
8
9 FixStyle(outward,FixOutward)
10
11 #else
12
13 #ifndef LMP_FIX.OUTWARD.H
14 #define LMP_FIX.OUTWARD.H
15
16 #include "fix.h"
17
18 namespace LAMMPS_NS {
19
20 class FixOutward : public Fix {
21 public:
22   FixOutward(class LAMMPS *, int, char **);
23   ~FixOutward();
24   void init();
25   int setmask();
26   void initial_integrate(int); // from doc: called at very beginning of each timestep (
27   // optional)
28   void pre_force(int); //
29   void post_force(int); // called after pair & molecular forces are computed and
30   // communicated (optional)
31   void post_create();
32
33 private:
34   FixPropertyAtom * fix_outward;
35   FixPropertyAtom * fix_outward_mag;
36   FixPropertyAtom * fix_outward_next;
37   FixPropertyAtom * fix_numberNeigh;
38   FixPropertyAtom * fix_number_next;
39   FixPropertyAtom * fix_indent;
40   FixPropertyAtom * fix_indent_sqr;
41   FixPropertyAtom * fix_error_inte;
42   FixPropertyAtom * fix_error_bulk;
43   FixPropertyAtom * fix_good_inte;
44   FixPropertyAtom * fix_good_bulk;
45   void updatePtrs();
46   void forward_all();
47   void reverse_all();
48   double ** outward;
49   double * outward_mag;
50   double * outward_next;
51   double * numberNeigh;
52   double * number_next;
53   double * indent;
54   double * indent_sqr;
55   double * error_inte;
56   double * error_bulk;
57   double * good_inte;
58   double * good_bulk;
59   class Pair * pair_gran-;
60   char type[256];
61   char type_mag[256];
62   char type_next[256];
63   char type_numb[256];
64   char type_nuxt[256];
65   char type_inde[256];
66   char type_insq[256];
67   char type_erin[256];
68   char type_erb[256];
69   char type_goin[256];
70   char type_gobu[256];
71   char model[256];
72 };
73 }
74 #endif
75 #endif

```

B.3.3 fix_outward.cpp

```

1 /* -----
2 file : fix_outward.cpp
3 type : liggghts 3.5.0 fix
4 ----- */
5
6 #include <cmath>
7
8 #include "atom.h"
9 #include "error.h"
10 #include "neighbor.h"
11 #include "neigh_list.h"
12 #include "fix_property_atom.h"
13 #include "force.h"
14 #include "pair_gran.h"
15
16 #include "fix_outward.h"
17
18 using namespace LAMMPS_NS;
19 using namespace FixConst;
20
21 FixOutward::FixOutward(LAMMPS *lmp, int narg, char **arg) : Fix(lmp, narg, arg) {
22   if (narg != 6) error->all(FLERR,"Illegal fix outward command");
23   strcpy(type, arg[3]); // retrieve fix outward from input
24   strcpy(model, arg[4]);
25   strcat(model, " ");
26   strcat(model, arg[5]);

```

B.3. SOURCES: INTERACTION LAW TRILIN, CONTACT AND SELF-CONTACT 219

```

27  strcpy(type_mag, type);
28  strcat(type_mag, "_mag");
29  strcpy(type_next, type);
30  strcat(type_next, "_next");
31  strcpy(type_num, type);
32  strcat(type_num, "_numberNeigh");
33  strcpy(type_nuxt, type);
34  strcat(type_nuxt, "_number-next");
35  strcpy(type_inde, type);
36  strcat(type_inde, "_indent");
37  strcpy(type_insq, type);
38  strcat(type_insq, "_indent_sqr");
39  strcpy(type_erin, type);
40  strcat(type_erin, "_error_inte");
41  strcpy(type_erb, type);
42  strcat(type_erb, "_error_bulk");
43  strcpy(type_goin, type);
44  strcat(type_goin, "_good_inte");
45  strcpy(type_gobu, type);
46  strcat(type_gobu, "_good_bulk");
47
48  outward = NULL;
49  outward_mag = NULL;
50  outward_next = NULL;
51  numberNeigh = NULL;
52  number_next = NULL;
53  indent = NULL;
54  indent_sqr = NULL;
55  error_inte = NULL;
56  error_bulk = NULL;
57  good_inte = NULL;
58  good_bulk = NULL;
59 }
60
61 void FixOutward::post_create() { // create fix outward and outward_next if they are missing
62   in input script
63   fix_outward = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward", "property/
64     atom", "vector", 0, 0, this->style, false));
65   if (!fix_outward) {
66     const char* fixarg[11];
67     fixarg[0] = "outward";
68     fixarg[1] = "all";
69     fixarg[2] = "property/atom";
70     fixarg[3] = "outward";
71     fixarg[4] = "vector";
72     fixarg[5] = "no";
73     fixarg[6] = "yes";
74     fixarg[7] = "yes";
75     fixarg[8] = "0.";
76     fixarg[9] = "0.";
77     fixarg[10] = "0.";
78     fix_outward = modify->add_fix_property_atom(11, const_cast<char*>(fixarg), style);
79   }
80
81   fix_outward_mag = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_mag", "
82     property/atom", "scalar", 0, 0, this->style, false));
83   if (!fix_outward_mag) {
84     const char* fixarg[9];
85     fixarg[0] = "outward_mag";
86     fixarg[1] = "all";
87     fixarg[2] = "property/atom";
88     fixarg[3] = "outward_mag";
89     fixarg[4] = "scalar";
90     fixarg[5] = "no";
91     fixarg[6] = "yes";
92     fixarg[7] = "yes";
93     fixarg[8] = "0.";
94     fix_outward_mag = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
95   }
96
97   fix_outward_next = static_cast<FixPropertyAtom*>(modify->find_fix_property("outward_next", "
98     property/atom", "vector", 0, 0, this->style, false));
99   if (!fix_outward_next) {
100     const char* fixarg[11];
101     fixarg[0] = "outward_next";
102     fixarg[1] = "all";
103     fixarg[2] = "property/atom";
104     fixarg[3] = "outward_next";
105     fixarg[4] = "vector";
106     fixarg[5] = "no";
107     fixarg[6] = "yes";
108     fixarg[7] = "yes";
109     fixarg[8] = "0.";
110     fixarg[9] = "0.";
111     fixarg[10] = "0.";
112     fix_outward_next = modify->add_fix_property_atom(11, const_cast<char*>(fixarg), style);
113   }
114
115   fix_numberNeigh = static_cast<FixPropertyAtom*>(modify->find_fix_property("numberNeigh", "
116     property/atom", "scalar", 0, 0, this->style, false));
117   if (!fix_numberNeigh) {
118     const char* fixarg[9];
119     fixarg[0] = "numberNeigh";
120     fixarg[1] = "all";
121     fixarg[2] = "property/atom";
122     fixarg[3] = "numberNeigh";
123     fixarg[4] = "scalar";
124     fixarg[5] = "no";
125     fixarg[6] = "yes";
126     fixarg[7] = "yes";
127     fixarg[8] = "0.";
128     fix_numberNeigh = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);

```



```

125 }
126
127 fix_number_next = static_cast<FixPropertyAtom*>(modify->find_fix_property("number_next",
    property/atom", "scalar", 0, 0, this->style, false));
128 if(!fix_number_next) {
129     const char* fixarg[9];
130     fixarg[0] = "number_next";
131     fixarg[1] = "all";
132     fixarg[2] = "property/atom";
133     fixarg[3] = "number_next";
134     fixarg[4] = "scalar";
135     fixarg[5] = "no";
136     fixarg[6] = "yes";
137     fixarg[7] = "yes";
138     fixarg[8] = "0.";
139     fix_number_next = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
140 }
141
142 fix_indent = static_cast<FixPropertyAtom*>(modify->find_fix_property("indent", "property/atom
    ", "scalar", 0, 0, this->style, false));
143 if(!fix_outward_mag) {
144     const char* fixarg[9];
145     fixarg[0] = "indent";
146     fixarg[1] = "all";
147     fixarg[2] = "property/atom";
148     fixarg[3] = "indent";
149     fixarg[4] = "scalar";
150     fixarg[5] = "no";
151     fixarg[6] = "yes";
152     fixarg[7] = "yes";
153     fixarg[8] = "0.";
154     fix_indent = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
155 }
156
157 fix_indent_sqr = static_cast<FixPropertyAtom*>(modify->find_fix_property("indent_sqr", "
    property/atom", "scalar", 0, 0, this->style, false));
158 if(!fix_outward_mag) {
159     const char* fixarg[9];
160     fixarg[0] = "indent_sqr";
161     fixarg[1] = "all";
162     fixarg[2] = "property/atom";
163     fixarg[3] = "indent_sqr";
164     fixarg[4] = "scalar";
165     fixarg[5] = "no";
166     fixarg[6] = "yes";
167     fixarg[7] = "yes";
168     fixarg[8] = "0.";
169     fix_indent_sqr = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
170 }
171
172 fix_error_inte = static_cast<FixPropertyAtom*>(modify->find_fix_property("error_inte", "
    property/atom", "scalar", 0, 0, this->style, false));
173 if(!fix_outward_mag) {
174     const char* fixarg[9];
175     fixarg[0] = "error_inte";
176     fixarg[1] = "all";
177     fixarg[2] = "property/atom";
178     fixarg[3] = "error_inte";
179     fixarg[4] = "scalar";
180     fixarg[5] = "no";
181     fixarg[6] = "yes";
182     fixarg[7] = "yes";
183     fixarg[8] = "0.";
184     fix_error_inte = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
185 }
186
187 fix_error_bulk = static_cast<FixPropertyAtom*>(modify->find_fix_property("error_bulk", "
    property/atom", "scalar", 0, 0, this->style, false));
188 if(!fix_outward_mag) {
189     const char* fixarg[9];
190     fixarg[0] = "error_bulk";
191     fixarg[1] = "all";
192     fixarg[2] = "property/atom";
193     fixarg[3] = "error_bulk";
194     fixarg[4] = "scalar";
195     fixarg[5] = "no";
196     fixarg[6] = "yes";
197     fixarg[7] = "yes";
198     fixarg[8] = "0.";
199     fix_error_bulk = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
200 }
201
202 fix_good_inte = static_cast<FixPropertyAtom*>(modify->find_fix_property("good_inte", "
    property/atom", "scalar", 0, 0, this->style, false));
203 if(!fix_outward_mag) {
204     const char* fixarg[9];
205     fixarg[0] = "good_inte";
206     fixarg[1] = "all";
207     fixarg[2] = "property/atom";
208     fixarg[3] = "good_inte";
209     fixarg[4] = "scalar";
210     fixarg[5] = "no";
211     fixarg[6] = "yes";
212     fixarg[7] = "yes";
213     fixarg[8] = "0.";
214     fix_good_inte = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
215 }
216
217 fix_good_bulk = static_cast<FixPropertyAtom*>(modify->find_fix_property("good_bulk", "
    property/atom", "scalar", 0, 0, this->style, false));
218 if(!fix_outward_mag) {
219     const char* fixarg[9];
220     fixarg[0] = "good_bulk";

```

B.3. SOURCES: INTERACTION LAW TRILIN, CONTACT AND SELF-CONTACT 221

```

221         fixarg[1]="all";
222         fixarg[2]="property/atom";
223         fixarg[3]="good-bulk";
224         fixarg[4]="scalar";
225         fixarg[5]="no";
226         fixarg[6]="yes";
227         fixarg[7]="yes";
228         fixarg[8]="0.";
229         fix_good_bulk = modify->add_fix_property_atom(9, const_cast<char*>(fixarg), style);
230     }
231 }
232 }
233
234 void FixOutward::forward_all() {
235     fix_outward->do_forward_comm();
236     fix_outward_mag->do_forward_comm();
237     fix_outward_next->do_forward_comm();
238     fix_numberNeigh->do_forward_comm();
239     fix_number_next->do_forward_comm();
240     fix_indent->do_forward_comm();
241     fix_indent_sqr->do_forward_comm();
242     fix_error_inte->do_forward_comm();
243     fix_error_bulk->do_forward_comm();
244     fix_good_inte->do_forward_comm();
245     fix_good_bulk->do_forward_comm();
246 }
247
248 void FixOutward::reverse_all() {
249     fix_outward->do_reverse_comm();
250     fix_outward_mag->do_reverse_comm();
251     fix_outward_next->do_reverse_comm();
252     fix_numberNeigh->do_reverse_comm();
253     fix_number_next->do_reverse_comm();
254     fix_indent->do_reverse_comm();
255     fix_indent_sqr->do_reverse_comm();
256     fix_error_inte->do_reverse_comm();
257     fix_error_bulk->do_reverse_comm();
258     fix_good_inte->do_reverse_comm();
259     fix_good_bulk->do_reverse_comm();
260 }
261
262 void FixOutward::updatePtrs() {
263     outward = fix_outward->array_atom;
264     outward_mag = fix_outward_mag->vector_atom;
265     outward_next = fix_outward_next->array_atom;
266     numberNeigh = fix_numberNeigh->vector_atom;
267     number_next = fix_number_next->vector_atom;
268     indent = fix_indent->vector_atom;
269     indent_sqr = fix_indent_sqr->vector_atom;
270     error_inte = fix_error_inte->vector_atom;
271     error_bulk = fix_error_bulk->vector_atom;
272     good_inte = fix_good_inte->vector_atom;
273     good_bulk = fix_good_bulk->vector_atom;
274     // custom vector created by the fix
275     // hypothesis: has to be used on every processor. If only in init(), only master node will
276     // know the vector and others will cause segmentation faults. To be used also in
277     // post_force().
278 }
279
280 void FixOutward::init() {
281     pair_gran_ = static_cast<PairGran*>(force->pair_match("gran", 0));
282     fix_outward = static_cast<FixPropertyAtom * >(modify->find_fix_property(type, "property/atom",
283     "vector", 0, 0, model));
284     fix_outward_mag = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_mag, "
285     property/atom", "scalar", 0, 0, model));
286     fix_outward_next = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_next, "
287     property/atom", "vector", 0, 0, model));
288     fix_numberNeigh = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_num, "
289     property/atom", "scalar", 0, 0, model));
290     fix_number_next = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_nuxt, "
291     property/atom", "scalar", 0, 0, model));
292     fix_indent = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_inde, "property/
293     atom", "scalar", 0, 0, model));
294     fix_indent_sqr = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_insq, "
295     property/atom", "scalar", 0, 0, model));
296     fix_error_inte = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_erin, "
297     property/atom", "scalar", 0, 0, model));
298     fix_error_bulk = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_erbu, "
299     property/atom", "scalar", 0, 0, model));
300     fix_good_inte = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_goin, "
301     property/atom", "scalar", 0, 0, model));
302     fix_good_bulk = static_cast<FixPropertyAtom * >(modify->find_fix_property(type_gobu, "
303     property/atom", "scalar", 0, 0, model));
304     // "vector", 3 retrieve only if 3 values in vector. Error if less. Silence if more.
305     updatePtrs(); // initialize custom array
306     forward_all();
307 }
308
309 FixOutward::~FixOutward() {}
310
311 int FixOutward::setmask() {
312     int mask = 0;
313     mask |= INITIAL_INTEGRATE;
314     mask |= POST_FORCE;
315     return mask;
316 }
317
318 void FixOutward::initial_integrate(int vflag) {
319     updatePtrs();
320     int nlocal = atom->nlocal;
321     for (int i=0; i<nlocal; i++) {
322         // initialize next data
323         outward_next[i][0] = 0;
324     }
325 }

```

```

311     outward_next[i][1] = 0;
312     outward_next[i][2] = 0;
313     number_next[i] = 0;
314     indent[i] = 0;
315     indent_sqr[i] = 0;
316 }
317 forward_all();
318 }
319
320 void FixOutward::post_force(int vflag) {
321     updatePtrs();
322
323     int nlocal = atom->nlocal; // atom id for the local proc
324     int newton_pair = force->newton_pair;
325
326     forward_all();
327     if (newton_pair) reverse_all();
328
329     updatePtrs();
330     for (int i=0 ; i<nlocal ; i++) {
331         // Store previous step data
332         outward[i][0] = outward_next[i][0];
333         outward[i][1] = outward_next[i][1];
334         outward[i][2] = outward_next[i][2];
335         outward_mag[i] = sqrt(outward[i][0]*outward[i][0] + outward[i][1]*outward[i][1] +
336                               outward[i][2]*outward[i][2]);
337         numberNeigh[i] = number_next[i];
338         // Initialize next data
339         outward_next[i][0] = 0;
340         outward_next[i][1] = 0;
341         outward_next[i][2] = 0;
342         number_next[i] = 0;
343     }
344 }

```

B.4 Typical Input Script

A typical template input script (Appendix B.4.1) is proposed, using the interaction law *TRILIN* and the self-contact algorithm, defined in Section B.3. The parameters to be replaced in the template all start with the symbol @, typical values are given in Table B.2. The paths of the output directory, the initial packing and the boundary condition mesh must be defined respectively in @directory, @file_coord and @mesh.

Script parameter	Variable	Example value
@deps	$\dot{\epsilon}$	$\pm 10^{-3}$
@strain	ϵ	± 1.0
@timeStep	Δt	0.1
@nRelax	Number of initial relaxation steps	500
@epsWin	Width of the averaging window (strain)	$5 \cdot 10^{-3}$
@epsOut	Output period (strain)	$5 \cdot 10^{-2}$
@diam	r_{crown}	1.4
@seed	r_{seed}	1
@mass	m	$3.606 \cdot 10^7$
@krep	k_{rep}	$1.424 \cdot 10^9$
@katt	k_{att}	$7.909 \cdot 10^8$
@fatt	f_{att}	$1.186 \cdot 10^8$
@wall	X_{wall}	3
@magOut	N_{mag}	2.3
@cosIJ	$\cos \alpha_{ij}$	0.4226
@cosEN	$\cos \alpha_{en}$	0.1736
@heal	Healing time	10^{12}

Table B.2: Typical parameters used in Part V. The healing behavior is here deactivated by a numerically infinite healing time. Warning: a hard-coded parameter in the interaction hinders self-contact detection before the 500th step. Proceed with caution to modify the number of initial relaxation steps.

Examples of the formats of the packing and the mesh files are given respectively in

Appendices B.4.2 and B.4.3. A template for the generation of the packings is given in Appendix B.4.4.

B.4.1 Input Script Template

```

1 ##### Single material , moving wall , prescribed strain rate
2 #
3 #
4 ##### file : single.template
5 ##### date : 2017/09/13
6 ##### type : liggghts 3.5.0-robin input script
7 ##### auth : robin.gibaud@simap.grenoble-inp.fr
8 #
9
10 #####
11 ### Parameters
12
13 # Geometry, kinematics
14 variable      deps equal @deps      # Strain rate
15 variable      strain equal @strain   # Final strain
16
17 # Numerical
18 variable      timeStep equal @timeStep      # Time step
19 variable      nRelax equal @nRelax          # Number of relaxation steps
20 variable      epsWin equal @epsWin          # Averaging window in strain
21 variable      epsOut equal @epsOut          # Frequency of outputs in strain
22 variable      relTime equal v_nRelax*v_timeStep
23
24 # Material: directly replaced in script
25 variable      diam equal @diam            # Crown diameter of the particles
26 variable      seed equal @seed            # Seed diameter of the particles
27 variable      mass equal @mass            # Mass of the particles
28
29 #####
30 ### Settings
31
32 # General setting
33 atom_style     granular
34 atom_modify    map array
35 boundary       f f f # See "Compute box dimension"
36 newton         off
37 communicate    single vel yes
38 units          si
39 neighbor        $(v_diam/10) bin
40 neigh_modify    delay 0
41
42 # Import ASCII coordinates file
43 read_data      @file_coord # Initial packing
44 velocity       all set 0 0 0
45
46 # Material
47 set            group all type 1
48 set            group all diameter ${diam} mass ${mass}
49 fix            mySeed all property/atom seed scalar no yes yes ${seed}
50 fix            seedOp all seed seed normal_model zcherry/outward
51 fix            out all property/atom outward vector no yes yes 0 0 0
52 fix            outMag all property/atom outward_mag scalar no yes yes 0
53 fix            outNei all property/atom outward_numberNeigh scalar no yes yes 0.0
54 fix            outNxt all property/atom outward_number_next scalar no yes yes 0.0
55 fix            outInd all property/atom outward_indent scalar no yes yes 0.0
56 fix            outPee all property/atom outward_indent_sqr scalar no yes yes 0.0
57 fix            errInt all property/atom outward_error_inte scalar no yes yes 0.0
58 fix            errBul all property/atom outward_error_bulk scalar no yes yes 0.0
59 fix            gooInt all property/atom outward_good_inte scalar no yes yes 0.0
60 fix            gooBul all property/atom outward_good_bulk scalar no yes yes 0.0
61 fix            outClu all property/atom cluster scalar no yes yes 1.0
62 fix            outOpe all outward outward normal_model zcherry/outward
63 fix            myKrep all property/global krep peratomtypepair 1 @krep # Repulsive stiffness
64 fix            myKatt all property/global katt peratomtypepair 1 @katt # Attractive stiffness
65 fix            myFatt all property/global fatt peratomtypepair 1 @fatt # Attractive force
66
67 threshold      myWall all property/global wall scalar @wall # Multiplicative factor for
68                particle/wall interaction
69 fix            myHeal all property/global heal scalar @heal # Healing time of the "
70                interface" interaction
71 fix            myMagO all property/global magOut scalar @magOu # Magnitude threshold
72 fix            myCosI all property/global cosIJ scalar @cosIJ # Angle threshold
73 fix            myCosE all property/global cosEN scalar @cosEN # Angle threshold
74 pair_style      gran model zcherry/outward velocity on non-local off verb off
75 pair-coeff      * *
76
77 #####
78 ### Control parameters
79
80 ##### Atom (dump output)
81 # Stress image
82 compute        sV all stress/atom pair #pressure*vol units
83 # Others
84 compute        con all contact/atom # seed and crown contacts
85 compute        cutoff all coord/atom ${seed} # seed contacts
86 variable      nInt atom c-con-f_outNei # number of interface contacts
87 variable      pe atom f_outPee # cumulated potential elastic energy
88 compute        ke all ke/atom # kinetic energy
89 variable      peke atom v-pe/c.ke
90 variable      cumdelta atom f_outInd # cumulated indentation for the particle
91
92 ##### Macro (thermo output)
93 variable      ttime equal time-v_relTime

```

```

91 compute      CON all reduce ave c_con
92 compute      SVXX all reduce sum c_sV[1]
93 compute      SVYY all reduce sum c_sV[2]
94 compute      SVZZ all reduce sum c_sV[3]
95 compute      SVVZ all reduce sum c_sV[6]
96 compute      SVXZ all reduce sum c_sV[5]
97 compute      SVXY all reduce sum c_sV[4]
98 variable     myVol equal count(all)*PI*v_seed^3/6
99 variable     P equal (c_SVXX+c_SVYY+c_SVZZ)/(3*v_myVol)
100 compute      KE all reduce sum c_ke
101 compute      PE all reduce sum v_pe
102 variable     KEPE equal c_KE/c_PE
103 compute      CUTOFF all reduce sum c_cutoff
104 compute      CUMDELTA all reduce sum v_cumdelta
105 variable     RELIND equal c_CUMDELTA/c_CUTOFF
106 variable     adimK equal @krep/abs(v_P*v_seed)
107 variable     adimI equal v_deps*sqrt(v_mass/abs(v_P*v_seed))
108 variable     adimE equal (c_KE/count(all))/abs(v_P*v_seed^3)
109
110 # Strain
111 variable     E equal 0
112 compute      maxG all reduce max x y z
113 compute      minG all reduce min x y z
114 variable     lxG equal c_maxG[1]-c_minG[1]
115 variable     lyG equal c_maxG[2]-c_minG[2]
116 variable     lzG equal c_maxG[3]-c_minG[3]
117
118 #####
119 ### Simulation
120
121 ### Integration
122 fix          integ all nve
123
124 # Compute box dimensions
125 variable     tmp equal lx
126 variable     X0 equal ${tmp}
127 variable     tmp equal ly
128 variable     Y0 equal ${tmp}
129 variable     tmp equal lz
130 variable     Z0 equal ${tmp}
131 print        "Initial dimension: (${X0},${Y0},${Z0})"
132
133 variable     security equal 0.6 #0.6 shouldo...
134 # Depend on tensile or compressive load
135 if "${strain}" > 0 then &
136     "variable xSize equal v_security*v_X0" &
137     "variable ySize equal v_security*v_Y0" &
138     "variable zSize equal v_security*v_Z0*exp(v_strain)" &
139 else &
140     "variable xSize equal v_security*v_X0*exp(-0.5*v_strain)" &
141     "variable ySize equal v_security*v_Y0*exp(-0.5*v_strain)" &
142     "variable zSize equal v_security*v_Z0"
143 print "${xSize}, ${ySize}, ${zSize}"
144 #print "${lx}, ${ly}, ${lz}"
145 change-box   all x final $(-1*v_xSize) $(v_xSize) y final $(-1*v_ySize) $(v_ySize) z final
146             $(-1*v_zSize) $(v_zSize) boundary f f f
147 # if mmm, segFault with 8 proc. If scale 1.01, meshes lost.
148
149 ### Boundaries
150 # Import
151 fix          top all mesh/surface/stress file @mesh type 1 move 0 0 $(0.5*v_Z0+0.5*v_seed)
152             reference_point 0 0 $(0.5*v_Z0+0.5*v_seed) # Import and translate the planar mesh
153             defining the boundary conditions, translate reference point
154 fix          bot all mesh/surface/stress file @mesh type 1 move 0 0 $(-0.5*v_Z0-0.5*v_seed)
155             reference_point 0 0 $(-0.5*v_Z0-0.5*v_seed)
156 fix          meshwalls all wall/gran model zcherry/outward mesh n-meshes 2 meshes top bot
157
158 # Define interaction properties
159 # The mesh needs to define a correct box size in x and y direction
160 # A manually computed z box size is computed from strain and Z0
161 # Describe motion of superior boundary
162 variable     veloXY equal 0
163 variable     veloZ equal v_deps*v_Z0*exp(v_deps*v_ttime)
164 variable     veloZtop equal v-veloZ/2
165 variable     veloZbot equal -1*v-veloZ/2
166
167 ### Compute steps
168 variable     Dt equal v_timeStep #abs(v_epsStep/v_deps)
169 timestep     ${timeStep}
170
171 variable     nForming equal ceil(v_strain/(v_deps*v_Dt)) # Forming steps
172 variable     nOutRel equal ceil(v_nRelax/5) # Output frequency for
173             relaxation (steps)
174 variable     stepWin equal ceil(abs(v_epsWin/(v_deps*v_Dt))) # Averaging window (
175             steps)
176 variable     nOut equal ceil(abs(v_epsOut/(v_deps*v_Dt))) # Output frequency (
177             steps)
178
179 ### Output settings
180 # Global
181 #fix          ave all ave/time 1 ${stepWin} ${nOut} f_top[3] mode scalar ave one
182             fzT      fzB      mxT      mxB      myT
183 #
184             myB
185 fix          f_top[5] f_bot[5] mode scalar ave one
186             avat all ave/atom 1 ${stepWin} ${nOut} c_sV[1] c_sV[2] c_sV[3] c_sV[4] c_sV[5]
187             c_sV[6]
188
189 #the timesteps contributing to the average value cannot overlap, i.e. Nfreq > (Nrepeat-1)*
190         Nvery is required
191 # work-around: keyword ave window M
192 thermo      ${nOut}
193 thermo_style custom step c_PE c_KE v_KEPE v_RELIND v_P v_adimK v_adimI v_adimE &
194             zhi zlo f_top[9] f_bot[9] v_E v_ttime v_lxG v_lyG v_lzG c_maxG[3] c_minG[3] &

```

```

182          f_ave[1] f_ave[2] f_ave[3] f_ave[4] f_ave[5] f_ave[6] # fzT      fzB      mxT      mxB
183          myT      myB
184 #v_adimE v_adimK v_P& c_P& c_DELTA c_KEPE
184 thermo_modify      format float %.15e
185
186 ### Initial relaxation
187 write_dump      all custom @directoryatom_0 id type x y z radius # print the initial state in
187           the defined output directory
188 run      ${nRelax}
189 write_dump      all custom @directoryatom_relax id type x y z radius
190
191 ### Prescribed deformation
192 fix      moveTop all move/mesh mesh top linear/variable v_veloXY v_veloXY v_veloZtop
193 fix      moveBot all move/mesh mesh bot linear/variable v_veloXY v_veloXY v_veloZbot
194 variable      E equal v_deps*v_ttime
195 dump      dmpAtom all custom ${nOut} @directoryatom_* id type x y z radius c_con f_avat
195           [1] f_avat[2] f_avat[3] f_avat[6] f_avat[5] f_avat[4] v_cumdelta c_cutoff f_outMag f_out
195           [1] f_out[2] f_out[3] f_outNei v_nInt f_errInt f_errBul
196 run      ${nForming}

```

B.4.2 Test Packing

```

1 LAMMPS data file via write_data, version Version LIGGGHTS-PUBLIC 3.5.0
2
3 8 atoms
4 1 atom types
5
6 -1.0079490209146593e+00 1.0079490209146593e+00 xlo xhi
7 -1.0079490209146593e+00 1.0079490209146593e+00 ylo yhi
8 -1.0079490209146593e+00 1.0079490209146593e+00 zlo zhi
9
10 Atoms
11
12 8 1 1.0e+00 1.0e+14 -2.580785474845e-01 -7.156083105123e-01 -6.171464385032e-01 0 0 0
13 5 1 1.0e+00 1.0e+14 7.368249072898e-01 -6.510293341930e-01 -5.370172428427e-01 0 0 0
14 7 1 1.0e+00 1.0e+14 3.983590840727e-01 4.237290992574e-01 -5.528105627266e-01 0 0 0
15 2 1 1.0e+00 1.0e+14 -5.747668099991e-01 1.714657239578e-01 -2.780545755943e-01 0 0 0
16 6 1 1.0e+00 1.0e+14 -4.859489111352e-01 -5.129999113277e-01 4.463109577725e-01 0 0 0
17 4 1 1.0e+00 1.0e+14 5.003062957558e-01 -3.574610018912e-01 3.897200373051e-01 0 0 0
18 3 1 1.0e+00 1.0e+14 -8.239274048957e-01 5.746822726730e-01 6.026156857010e-01 0 0 0
19 1 1 1.0e+00 1.0e+14 1.617639656358e-01 5.826412040774e-01 4.340756647920e-01 0 0 0

```

B.4.3 Typical Planar Mesh

```

1 solid
2 facet normal 0 0 -1
3   outer loop
4     vertex -3.32365398113 -3.32365398113 00.00
5     vertex -3.32365398113 3.32365398113 00.00
6     vertex 3.32365398113 -3.32365398113 00.00
7   endloop
8 endfacet
9 facet normal 0 0 -1
10  outer loop
11    vertex -3.32365398113 3.32365398113 00.00
12    vertex 3.32365398113 -3.32365398113 00.00
13    vertex 3.32365398113 3.32365398113 00.00
14  endloop
15 endfacet
16 end solid

```

B.4.4 Packing Script Template

```

1 ##### Particle packing by random insertion and isotropic shrinking of the box
2 #-----
3 #
4 ##### file : packing.template
5 ##### date : 2016/07/01
6 ##### type : liggghts 3.4.1-robin input script
7 ##### auth : robin.gibaud@simap.grenoble-inp.fr
8 #-----
9 #
10 # Target parameters
11 variable      diam equal 1
12 variable      part equal @nb # number of particles
13 variable      fraF equal 0.639 # final relative density
14 # Numerical parameters
15 variable      seed0 equal @seed0 # random seeds
16 variable      seed1 equal @seed1
17 variable      seed2 equal @seed2
18 variable      deps equal -1e-5
19 variable      dens equal 1e14 #1e20
20 variable      ksi equal 0.98
21 variable      fraI equal 0.25
22 variable      rigi equal 1e8 #1e10 #1e6
23
24 variable      mass equal v_dens*PI*v_diam^3/6
25 variable      w0 equal (v_rigi/v_mass)^0.5
26 variable      Dt equal 0.01/v_w0 # arbitrary factor
27 variable      visc equal 2*v_mass*v_ksi*v_w0 #5e11 #1e8
28
29 variable      volS equal v_part*PI*v_diam^3/6
30 variable      aFin equal (v_volS/v_fraF)^(1/3)
31 variable      aIni equal (v_volS/v_fraI)^(1/3)
32 variable      eps equal ln(v_aFin/v_aIni)

```

```

33 variable          nStep equal v_eps/(v_deps*v_Dt)
34 #
35 variable          stepOut equal floor(v_nStep/10)
36
37 #####
38 ### Settings
39 # Simulation setting
40 atom_style         granular
41 atom_modify        map array
42 boundary           p p p
43 newton             off
44 communicate        single vel yes
45 units              si
46 neighbor           $(v_diam/10) bin
47 processors         @procX @procY @procZ
48 neigh_modify       delay 0
49 region            reg block $(v_aIni/-2) $(v_aIni/2) $(v_aIni/-2) $(v_aIni/2) $(v_aIni/-2) $(
      v_aIni/2) units box
50 create_box         1 reg
51 timestep           ${Dt}
52 fix                integr all nve
53 fix                comp all deform 1 x trate ${deps} y trate ${deps} z trate ${deps}
54 # Material
55 fix                m4 all property/global coefficientFriction peratomtypepair 1 0
56 fix                m6 all property/global kn peratomtypepair 1 ${rigi}
57 fix                m7 all property/global kt peratomtypepair 1 0
58 fix                m8 all property/global gamman.abs peratomtypepair 1 ${visc}
59 fix                m9 all property/global gammat.abs peratomtypepair 1 0
60 pair_style gran     model hooke/stiffness absolute_damping on limitForce off tangential_damping on
61 pair_coeff          * *
62
63 #####
64 ### Particle insertion
65 # Random gas generation
66 fix                ptsl all particletemplate/sphere ${seed0} atom-type 1 density constant ${dens}
      radius constant $(v_diam/2)
67 fix                pddl all particledistribution/discrete ${seed1} 1 ptsl 1.0
68 fix                ins all insert/pack seed ${seed2} distributiontemplate pddl &
      maxattempt 250 &
70                  insert_every once region reg particles-in-region ${part}
71 run                1 # Insert the particles , before first dump, otherwise empty
72
73 #####
74 ### Control parameters
75 variable           Tosc equal 2*PI*(v_mass/v_rigi)^0.5
76 variable           Tamo equal 2*v_mass/v_visc
77 print              "Tosc = ${Tosc} ; Tamo = ${Tamo} ; dt = ${Dt} (s)"
78
79 compute            con all contact/atom #local coordination number
80 compute            Con all reduce ave c_con #average coordination number
81 #fix               HCon all ave/histo 1 1 $(v_stepOut) 1 11 11 c_con mode vector file @directory/
      contact.@key
82
83 compute            P all pressure thermo-temp pair # -1*pressure units
84 compute            ke all ke/atom
85 compute            MeKe all reduce ave c_ke
86 variable           frac equal atoms*PI*${diam}^3/(6*volt)
87
88 variable           E equal c_MeKe/(c_P*v_diam^2)
89 variable           I equal abs(v_deps*v_diam*(v_dens/c_P)^0.5)
90 variable           K equal abs(v_rigi/(c_P/${diam}))
91
92 #####
93 ### Simulation
94 thermo_style        custom step c_Con c_P v_frac
95 thermo             $(v_stepOut)
96 run                $(v_nStep)
97 #thermo_style        custom c_P ke c_Con v_frac
98 #thermo             $(ceil(v_nStep/10))
99 #run                $(v_nStep/2)
100 unfix              comp
101 #thermo             $(ceil(v_nStep*40/10))
102 run                $(v_nStep/1.5)
103 dump               dmp all custom 1 @directory/atom_@key id type type x y z ix iy iz vx vy vz fx
      fy fz omegax omegay omegaz radius
104 thermo_style        custom step c_P ke c_Con v_E v_I v_K v_frac
105 run                0
106 # Write final state
107 write_data          @file_pack

```

Appendix C

FEM Script

```

1  """ Template for eighth of spherical inclusion test case
2
3  file : sphere134.comm
4  type : code-aster 13.4 input, python 2.7
5  auth : robin.gibaud@simap.grenoble-inp.fr
6
7  """
8
9  from __future__ import division, absolute_import, print_function, unicode_literals
10 import numpy as np
11 import matplotlib.pyplot as plt
12 from Utilitai.Table import Table
13
14 #####
15 ### Python variables
16 #Parameters
17 H0      = @H0          # Initial height [mm]
18 DEPS    = [@deps]      # Strain rate [s-1]
19 de      = @De          # Strain increment between output times [/]
20 mini_de = 1e-10
21 ESTEP   = [0, @eps]    # Global strain [/]
22 young   = @young       # Young's modulus [MPa]
23 poisson = @poisson     # Poisson's ratio [/]
24
25 maxi_dt = np.absolute(1e-3/@deps) # Maximum time step [s] #1e-3 for 0 and 2
26
27 mini_dt = np.absolute(mini_de/@deps)
28 #Time and displacement vectors generation
29 T = [0] # Time [s]
30 E = [0] # Global strain [/]
31 ENDIND = []
32 for i,deps in enumerate(DEPS):
33     n = np.absolute((ESTEP[i+1]-ESTEP[i]) / de)
34     dt = np.absolute(de / deps)
35     e = np.linspace(ESTEP[i],ESTEP[i+1],n)
36     t = np.linspace(T[-1]+dt,T[-1]+n*dt,n)
37     e = e[1:]
38     t = t[1:]
39     E = np.append(E,e)
40     T = np.append(T,t)
41     if i==0:
42         ENDIND = np.append(ENDIND, n)
43     else:
44         ENDIND = np.append(ENDIND, ENDIND[-1]+n)
45 DISP = H0 * ( np.exp(E) - 1 ) # Displacement of face : mm
46
47 #####
48 ### Aster script
49 DEBUT(PAR_LOT='OUI'); #Default safe behavior, blocks python manipulation of
50     table [U1.03.02:3], results are certified
51
52 mesh = LIRE.MAILLAGE( FORMAT = 'MED' );
53
54 matA = DEFLMATERIAU(
55     ELAS=_F(
56         E = young,
57         NU = poisson ),
58     LEMAITRE=_F(
59         N = @n_a,
60         UN_SUR_K = @un_sur_k_a,
61         UN_SUR_M = 0.0) #degenerate Lemaitre behavior to Norton law
62 );
63
64 matB = DEFLMATERIAU(
65     ELAS=_F(
66         E = young,
67         NU = poisson ),
68     LEMAITRE=_F(
69         N = @n_b,
70         UN_SUR_K = @un_sur_k_b,
71         UN_SUR_M = 0.0)
72 );

```



```

73 material = AFPE_MATERIAU(
74     MAILLAGE = mesh,
75     AFPE = (
76         _F( GROUP_MA = 'V-A', MATER = matA ),
77         _F( GROUP_MA = 'V-B', MATER = matB ),
78     )
79 );
80 model = AFPE_MODELE(
81     MAILLAGE = mesh,
82     AFPE = _F( TOUT = 'OUI',
83         #3D.INCO_UP, 3D.INCO_UPO [R3.06.08:13] sec.4.3 Elements finis traitant la quasi-
            incompressibilite, only ELAS or VMIS.ISOT.XXX
84         MODELISATION = '3D.INCO_UPG', # [U2.01.10:4] Notice d'utilisation sur le
            choix des elements
85         # [U2.04.01:6] sec.4.2 Gestion de l'incompressibilite plastique
86         # [R3.06.08:14] sec.4.2 Elements finis traitant la quasi-
            incompressibilite
87         PHENOMENE = 'MECANIQUE' ),
88 );
89
90 boundary = AFPE_CHAR_MECA(
91     MODELE = model,
92     FACE_IMPO = (
93         _F( GROUP_MA = 'S-X0', DX = 0. ),
94         _F( GROUP_MA = 'S-Y0', DY = 0. ),
95         _F( GROUP_MA = 'S-Z0', DZ = 0. ),
96         _F( GROUP_MA = 'S-Z', DZ = 1. ),
97     )
98 );
99
100 f_disp = DEFILFONCTION(
101     NOMPARA = 'INST',
102     PROLDROITE = 'EXCLU',
103     PROLGAUCHE = 'EXCLU',
104     INTERPOL = ('LIN', 'LIN'),
105     ABSCISSE = T,
106     ORDONNEE = DISP,
107 );
108
109 l_inst = DEFILIST_INST( # [U2.04.01:3] sec.2.1 Conseils d'utilisation de STAT_NON_LINE,
    always activate DEFILINST_INST with STAT_NON_LINE
110     DEFILIST = _F(
111         #METHODE = 'MANUEL', # default.
112         #METHODE = 'AUTO', # bug in 13.4
113         VALE = T,
114         #PAS_MAXI = maxi_dt, # [U4.34.03:7] does 'nt apply to the first time step
115         #PAS_MINI = mini_dt, # [U4.34.03:7] sec.3.1.3 DEFILIST_INST. Only
            applied to new time step, may be smaller in subdivision
116     ),
117     ECHEC = (
118         _F( # nb max iteration reach, or integration of constitutive law fails
119             EVENEMENT = 'ERREUR',
120             ACTION = 'DECOUPE',
121             SUBD.METHODE = 'MANUEL',
122             SUBD.PAS = subd.pas,
123             SUBD.NIVEAU = subd.niv,
124         ),
125         _F( # "big" equilibrium residual, meant to prevent Floating Point Exceptions
126             # not very convincing in our case, as FPE can occur even at RESLMAX = 0.1
127             # http://code-aster.org/forum2/viewtopic.php?id=19811
128             EVENEMENT = 'RESLMAXI',
129             RESL.GLOB.MAXI = 1e-1,
130             ACTION = 'DECOUPE',
131             SUBD.METHODE = 'MANUEL',
132             SUBD.PAS = subd.pas,
133             SUBD.NIVEAU = subd.niv,
134         ),
135     ),
136 );
137
138 result = STAT_NON_LINE(
139     MODELE = model,
140     CHAM_MATER = material,
141     EXCIT = (
142         _F(
143             CHARGE = boundary, #define the mechanical charge
144             FONC_MULT = f_disp, #multiply this charge by float
145         ),
146     ),
147     COMPORTEMENT = _F(
148         viscoela, integration schemes # [R5.03.08]
149         RELATION = 'NORTON', # [U4.51.11:17] sec.4.3.3.4 Comportements non
            lineaires. More easily fails to integrate
150         ALGO.INTE = 'NEWTON.PERT', #Norton: theta-methode integration
151         ITER.INTE.MAXI = 50, #50
152         DEFORMATION = 'GDEF.LOG', # [U4.51.11:55] ; [U2.04.01:8]
153         PARM.THETA = 0.5, #Lemaitre or Norton: semi-implicit [U4.51.11:51] sec
            4.8
154     ),
155     NEWTON = _F(
156         PREDICTION = 'TANGENTE', # [U2.04.01:3] Conseils d'utilisation de STAT_NON_LINE
            # [U3.14.06:3] sec.3.3 Modelisation 3D.INCO_UPG,
            necessary because ELASTIQUE not available
157         MATRICE = 'TANGENTE', #default
158         REAC.ITER = 1,
159         REAC.INCR = 1, #default
160     ),
161     INCREMENT = _F( LIST_INST = l_inst ),
162     CONVERGENCE = _F(
163         RESL.GLOB.RELA = @resi, # [U2.08.03:17] Notice d'utilisation des solveurs
            lineaires, might be set to bigger valuer
164         ITER.GLOB.MAXI = 20,

```

```

165     ),
166     SOLVEUR = _F(
167         METHODE = 'MUMPS', #[U2.08.03:6] Notice d'utilisation des solveurs lineaires,
168         recommended for incompressibility problems
169         NPREC = -1, #[ssnv112f.comm:92]
170     ),
171     ARCHIVAGE = _F(
172         PAS_ARCH = @PAS_ARCH, #[U4.51.03:35] sec.3.17 Operateur STAT_NONLINE, save
173         only N step
174     );
175     FIN(RETASSAGE = 'OUI');          #reduce size of saved base [U4.11.02:3]

```

Script parameter	Variable	Example value
@deps	Strain rate	$\pm 10^{-3}$
@eps	Final strain	± 0.3
@H0	Height sample	0.5 mm
@De	Mandatory strain step	$5 \cdot 10^{-4}$
@young	Young's modulus	$1 \cdot 10^5$ MPa
@poisson	Poisson's ratio	0.33
@n_a	Inverse inclusion strain rate sensitivity	1.38
@un_sur_k_a	Inverse inclusion stress level	$3.08 \cdot 10^{-5} \text{ s}^{-M} \cdot \text{MPa}^{-1}$
@n_b	Inverse matrix strain rate sensitivity	32.8
@un_sur_k_b	Inverse matrix stress level	$8.32 \cdot 10^{-3} \text{ s}^{-M} \cdot \text{MPa}^{-1}$
@resi	Numerical tolerance	$1 \cdot 10^{-5}$
@PAS_ARCH	Output stepping	5

Table C.1: Typical parameters used in Part IV for the FEM simulations.

Appendix D

IJMS Article

The article “Modeling Large Viscoplastic Strain in Multi-Material with the Discrete Element Method” was submitted to the International Journal of Mechanical Sciences. Minor revisions were requested for publication and the amended version is proposed here. We are currently waiting for the final decision. The article presents the main results from Part IV.

Modeling Large Viscoplastic Strain in Multi-Material with the Discrete Element Method

Robin Gibaud, Étienne Guesnet, Pierre Lhuissier*, Luc Salvo

Univ. Grenoble Alpes, CNRS, SIMAP, F-38000 Grenoble, France

Abstract

In this paper, the Discrete Element Method (DEM) is used as a tool to phenomenologically model large compressive viscoplastic strain in metallic composites. The model uses pairwise attractive and repulsive forces between spherical particles. Large packings of particles collectively cope with the prescribed strain, the changes of neighbors model the irreversible strain in the material. Using the proposed calibration method of the model parameters, the macroscopic behavior mimics perfect plasticity in compression, with a strain rate sensitivity approximating a viscoplastic Norton law.

The error on flow stress and volume conservation is estimated for single material. Three bi-material geometrical configurations are built: parallel, series and spherical inclusion. Macroscopic metrics (flow stress and shape factor of inclusion) are confronted to Finite Element Method (FEM) simulations.

The potential of the model, from a computing power point of view, is tested on a complex geometry, using a real microstructure of a crystalline/amorphous metallic composite, obtained by X-ray tomography.

Keywords: Discrete Element Method, Large Strain, Viscoplasticity, Multi-Material

1. Introduction

Architected materials display promising properties, allowing fine-tuning of their physical behavior and contriving contradictory functionalities. Typical examples are composite materials, associating complementary phases, whose distribution and topology are controlled towards functional requirements. Such architectures can be elaborated with metallic materials, for example using casting, powder technology or additive manufacturing.

Structural parts often need to be processed from raw architected materials. Many manufacturing processes, like hot forming, rely on deformation

*Corresponding author.

Email address: pierre.lhuissier@simap.grenoble-inp.fr (Pierre Lhuissier)

mechanisms that can involve the motion and interaction of complex 3D interface, large change in shape or topological modification. The typical involved dynamic phenomena – like pore closure, neck creation and phase fragmentation – can be challenging to observe with conventional, or destructive, experimental techniques. Tomography is a non-destructive tool, with active developments toward finer spatial and temporal resolution, allowing *in situ* observations. It is often used in close combination with digital volume correlation and simulation tools. Tomography data serve both as geometrical initial state and as temporal evolution reference for models, for example in calibration and validation procedures. Examples include the study of crack propagation [1] and the study of creep mechanisms in metallic foams [2].

At a macroscopic scale, the physical description of metallic materials as continuous media is often legitimate. However, from a numerical point of view, modeling the typical architectural discontinuities, and more specifically their interactions and topological changes, can be challenging within a continuous framework. Many strategies have been developed to extend the scope of continuous descriptions, among which:

- Dissociating material and mesh motions, with an Eulerian [3, 4, 5] or an Arbitrary Lagrangian-Eulerian [6, 7, 8] kinematical description of the materials.
- Sequentializing a large distortion in smaller steps, periodically re-generating a Lagrangian mesh [9].
- Super-imposing discontinuities description on top of a continuous framework, using additional discontinuous arbitrary shape functions [10, 11] or a set of punctual Lagrangian markers, representing material phases [12, 13] or interfaces [14].
- Discretizing the materials using a cloud of nodes instead of a mesh. The continuum constitutive law can be integrated globally for the whole system [15, 16, 17], or locally in the neighborhood of each node, for example in smooth particle hydrodynamics [18, 19] or non-ordinary state based peridynamics [20, 21] formulations.

A common denominator of these strategies is to derive the local behavior from the macroscopic continuous constitutive law. A distinct route is to describe the material as a set of discrete objects, using *ad hoc* interaction laws between neighboring objects. Such models are innately suited to describe materials where interfaces motions and interactions are predominant with respect to the continuous behavior.

This approach has historically been used in early attempts to numerically solve solid mechanics problems on arbitrary shapes [22]. In the last decade, variants of the Molecular Dynamics methodology, as the Discrete Element Method (DEM) and bond based peridynamics, have successfully been applied to model elastic continuous media. Three-dimensional work in solid mechanics include

the modeling of dynamic brittle failure [23, 24], crack propagation [25] and quasi-static buckling [26]. They all demonstrate the possibility to design and calibrate local interaction DEM laws to display a targeted continuous macroscopic behavior. To our knowledge, these work rely on initially pairwise bonded neighbors, and only allow volumetric strain plasticity [27] [28, p.153], plasticity and viscoplasticity being implemented at the pairs levels. Hence, modeling high strain in metallic alloys is hindered by the total Lagrangian description and the lack of isochoric plasticity mechanisms. Other potential extensions of the Molecular Dynamics, like the Movable Cellular Automaton method [29], rely on many-body interactions for such purposes.

In this paper, we focus on the development of a DEM model describing incompressible bi-materials for large quasi-static compressive strain. In both phases, we assume a perfect viscoplastic behavior, described by the Norton law. One peculiarity of our phenomenological approach is that the local laws have no likeness with the macroscopic behavior. Instead of implementing a continuous behavior at the scale of the numerical discretization, we use the analogy between the motion of a packing of elastic cohesive spheres, collectively sliding on one another, with the plastic shear in continuous media.

The implemented DEM contact law is described in Section 2 and its calibration procedure in Section 3. The macroscopic behavior of a single material is discussed in Section 4. The bi-material behavior of the model is tested, and confronted to continuous models, on elementary geometrical configurations in Section 5. A potential application of the methodology to an experimental microstructure is illustrated in Section 6.

2. Attractive-repulsive model

In this paper, continuous media are discretized by packings of interpenetrated spherical particles. This section describes the contact laws, used to compute interaction forces, between the particles. As in classical DEM implementations, interaction forces $F_{A \rightarrow B} = -F_{B \rightarrow A}$ are computed for each pair of indented particles (A, B) from their distance h and relative velocity $V_B - V_A$ (Figure 1). Time is discretized in constant steps Δt and the motion of the particles are integrated from Newton's second law using a Velocity Verlet explicit scheme. While the interaction forces are computed at the level of each pair, as described in this section, it must be understood that our model can only display the expected behavior for a packing of particles collectively interacting.

The main objective of the contact model is to maintain a cohesive packing, able to re-arrange itself with controlled overall volume change. Thus, at the pair level, interaction must alternatively be attractive and repulsive. Among the possible algorithmic strategies, we chose to use a purely geometrical management of the contacts, with no history parameter stored between the time steps. Each particle is subdivided into two concentric and spherical zones (Figure 1a) with distinct behaviors:

- A repulsive seed, mimicking incompressibility, of radius R_{seed} (Figure 1b);

- An attractive crown, adding cohesiveness, of radius $R_{\text{crown}} > R_{\text{seed}}$.



Figure 1: Pairwise interactions. (a) Geometry and kinematics of a pair: relative position and velocities. Seed contact example. (b) Interaction reciprocal forces: $F_{A \to B} = -F_{B \to A}$. Pairwise forces, computed from the kinematics of the pair (Equation 1).

In both zones, seed and crown, the model is governed by normal elastic forces F_N (Equation 1 and Figure 2). The interaction force is piecewise linear with h , the distance between the centers of the two particles. Two normal stiffnesses are used: k_{rep} for repulsive seed contact and k_{att} for attractive crown contact. Each particle has a numerical mass m , used in the integration of the motion.

The attractive behavior, in the crown zone, depends on the relative normal velocity \dot{h} (time derivative of h). The attractive force is only activated if a pair has a tensile motion ($\dot{h} > 0$), and is canceled in case of compressive motion ($\dot{h} \leq 0$). This behavior helps to smooth the creation of new contacts between particles and introduces a dissipative effect of the total energy, numerically sufficient within the strain rate validity range of the model, linked to the frequency of oscillation of the pairs. At the pair level, no damping, shear or torque interaction laws are implemented. In the tested configurations, such interactions only introduce second-order effects on the macroscopic behavior of packings. Thus, in this paper, interactions between particles are only normal pairwise forces and piecewise linear with the distance between the centers.

$$F_N = \begin{cases} k_{\text{rep}}(2R_{\text{seed}} - h) & \text{if } h \leq 2R_{\text{seed}} \\ k_{\text{att}}(2R_{\text{seed}} - h) & \text{if } 2R_{\text{seed}} < h \leq 2R_{\text{crown}} \text{ and } \dot{h} > 0 \\ 0 & \text{if } 2R_{\text{seed}} < h \leq 2R_{\text{crown}} \text{ and } \dot{h} \leq 0 \end{cases} \quad \begin{matrix} (1a) \\ (1b) \\ (1c) \end{matrix}$$

The possibility for the particles to arbitrarily change neighbors introduces, at a macroscopic scale, a plastic effect. The interaction laws at the pair level differ qualitatively from the targeted physical phenomena. To represent a continuum, a large packing of such particles is generated. Various phases can be represented, assigning in the initial configuration the properties of distinct materials to clusters of particles.

Two types of boundary conditions (Figure 3) are applied to the packings:

- Free boundary, where particles are not constrained by any means;

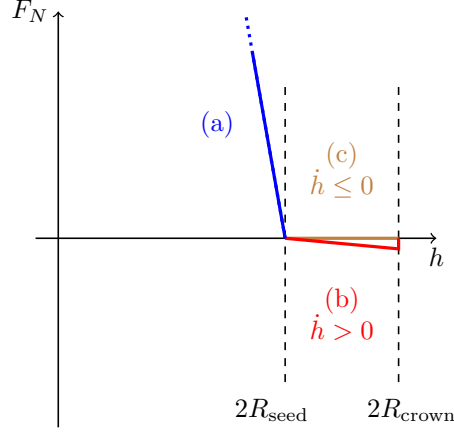


Figure 2: Definition of the piecewise linear normal interaction force. Dependency on the normal relative velocity \dot{h} . Slopes and distances to scale. Cases a, b and c from Equation 1.

- Kinetically constrained boundary, using a rigid mesh following prescribed motion.

For a uniaxial compression, the following boundary conditions are applied: a top and bottom planar meshes and free lateral sides (Figure 3b). In this paper, the meshes are used to apply prescribed macroscopic true strain rate. The forces acting onto the planes are summed to evaluate the macroscopic flow stress, computed using updated or initial cross-section.

Interaction forces between the mesh elements and the particles are computed with a very similar contact law as particle-particle contacts. h is defined as the shortest distance between an element and the center of a particle (Figure 3a), and R is to be used instead of $2R$ in Equation 1. In a mesh/particle contact, the interaction parameters of the particle are used. Any arbitrary geometry meshed with triangular planar elements can be used with the implemented model, in this paper only planar meshes were used.

From their generation to compression tests, the packings go through three main steps:

1. Random packing generation and relaxation [30];
2. Material properties attribution and relaxation;
3. Uniaxial compression.

The initial state and the elaboration route, in the context of the large strains studied here, seemed to have little influence on the compression results, and are not detailed here.

The model was implemented as an independent contact law in the open-source DEM code LIGGGHTS [31]. In this article, all DEM computations were run using LIGGGHTS, and all packing visualizations were rendered using the open-source software OVITO [32].



Figure 3: Boundary conditions. (a) Elementary mesh / particle interaction. Crown contact example. (b) Compression test: top and bottom meshes with prescribed motion; free lateral sides.

Summing-up our model, piecewise linear forces, both attractive and repulsive, are computed between particles and with meshes. No history is stored, and the contact management is geometrical.

3. Calibration procedure

This section describes the calibration procedure of the numerical parameters of the contact model, and the fixed parameters in the scope of this paper. Our objective is to model a perfect viscoplastic behavior, described as a relation between the strain rate $\dot{\epsilon}$ and the flow stress σ , by the Norton law [33, p.106]:

$$\sigma = K \dot{\epsilon}^M \quad (2)$$

Where M is the strain rate sensitivity and K is the stress level. All cases presented in this paper being in compressive state, strain, strain rate and stress will always be given in absolute value.

As the DEM does not rely on a continuous framework, the numerical parameters cannot be derived *a priori* from the targeted macroscopic behavior. We work here at fixed ratio $R_{\text{crown}}/R_{\text{seed}} = 1.5$, to allow a large overlap zone without catching second neighbors. The seed radius is arbitrarily set to a size of $R_{\text{seed}} = 1 \text{ mm}$. The ratio between attractive and repulsive stiffnesses is set to $k_{\text{rep}}/k_{\text{att}} = 10$, to guarantee a numerically predominant repulsion. The time step Δt is fixed to $5 \times 10^{-4} \text{ s}^{-1}$, which is from 20 to 2000 times smaller than the studied natural periods (Equation 3). The force signal, measured on the meshes, is averaged over a sliding window, of typical width 1×10^{-2} in strain. The remaining parameters to be calibrated are the repulsive stiffness of the contacts k_{rep} and the mass of the particles m .

We propose a two-step calibration procedure, based on uniaxial compression test simulations, on cubes of single materials:

Step 1. Calibrate the strain rate sensitivity M , tuning the ratio between mass and repulsive stiffness m/k_{rep} .

Step 2. Calibrate the stress level K , applying a common multiplicative factor to both mass m and repulsive stiffness k_{rep} .

The numerical parameters, obtained independently for each phase, are used in multi-material simulations without further fitting procedure.

3.1. Strain Rate Sensitivity Calibration

The strain rate sensitivity M of a packing depends on its ability to quickly rearrange itself, with regard to the prescribed strain rate.

To quantify an image of the reaction time, we use the natural period t_0 of an ideal spring-mass system of stiffness k_{rep} and mass m :

$$t_0 = 2\pi \sqrt{\frac{m}{k_{\text{rep}}}} \quad (3)$$

This value is not meant to match the actual oscillation period of particles, but to quantitatively compare sets of parameters. Packing of 5×10^3 particles with natural periods ranging from 1×10^{-2} to 1 s are compressed at strain rates from 3×10^{-6} to 1 s^{-1} . The flow stress σ is normalized by the flow stress at the lowest strain rate σ_{low} . The results (Figure 4), are used to build a master curve of the strain rate sensitivity behavior.

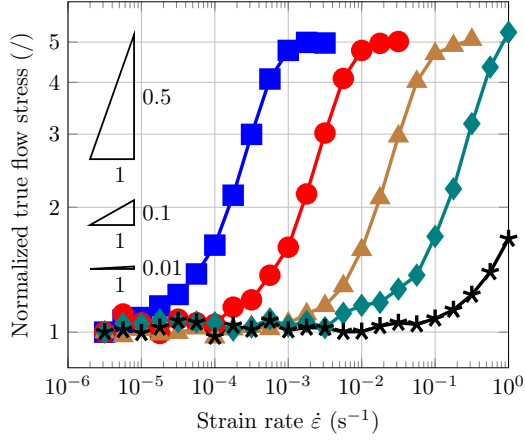
As shown in Figure 4a, the strain rate sensitivity, i.e. the slope in the space $(\dot{\epsilon}, \sigma/\sigma_{\text{low}})$, is driven by the relation between the natural period and the strain. A common trend for all configurations (Figure 4b) is clearly exhibited in the space $(t_0 \sqrt{\dot{\epsilon}}, \sigma/\sigma_{\text{low}})$, and approximated by least-square fitting, using a sigmoid of generic expression:

$$\sigma/\sigma_{\text{low}} = a + b/(1 + \exp(c - d \cdot t_0 \sqrt{\dot{\epsilon}})) \quad (4)$$

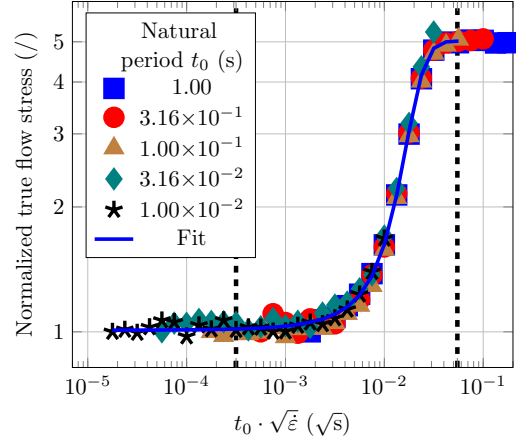
The fitting parameters used here are $(a, b, c, d) \approx (0.9048, 4.116, 3.651, 210.0)$. Using this fitted common trend, a master curve is built in the space $(\dot{\epsilon} \cdot t_0^2, M)$, summing-up this behavior for all sets of tested parameters (Figure 4c).

Three flow regimes, in terms of strain rate sensitivity, can be identified in Figures 4c and 4b:

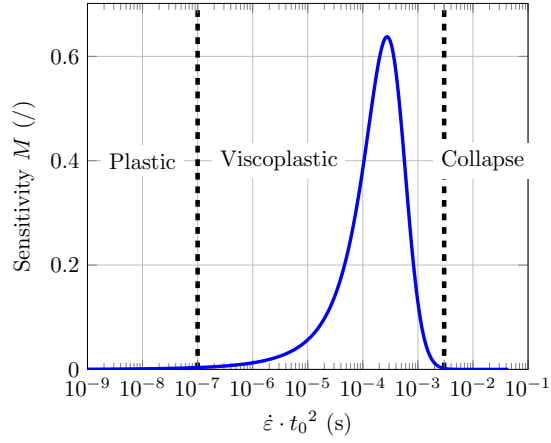
- Plastic: for $\dot{\epsilon} \cdot t_0^2 < 1 \times 10^{-7} \text{ s}$ the strain rate sensitivity is negligible ($M < 4 \times 10^{-3}$). A plastic behavior can thus be represented, with stress variation of the order of magnitude of the expected precision of the model, valid over various orders of magnitude of strain rates. The packing rearranges quickly enough when deformed, so that variations of strain rate does not affect the flow structure.
- Collapse: at higher values than $\dot{\epsilon} \cdot t_0^2 > 3 \times 10^{-3} \text{ s}$, the packing is not reactive enough for the particles to collectively cope with the strain. The strain localizes next to the moving planes, the flow stress drops and the macroscopic equilibrium is lost. Such configurations are not suitable for our purpose.



(a)



(b)



(c)

Figure 4: Calibration of the strain rate sensitivity. Successive steps toward the master curve. (a) Influence of the natural period on strain rate sensitivity. Normalized flow stress at a strain of 0.3 for 5×10^3 particles versus the prescribed strain rate. (b) Normalization in the $(t_0 \sqrt{\dot{\epsilon}}, \sigma/\sigma_{\text{low}})$ space. Common trend for all natural periods. Sigmoidal fit, see Equation 4. (c) Master curve of strain rate sensitivity. Three flow regimes.

- Viscoplastic: in the intermediate window, the $\dot{\epsilon} \cdot t_0^2$ value governs the sensitivity of the packing, up to a maximum of 0.6. In this configuration, when the strain rate increases, the particles are forced to indent more to rearrange, leading to higher flow stress. However, the sensitivity is strongly strain rate dependent. An actual viscoplastic behavior can only be modeled *via* an averaged strain rate sensitivity, with a scope of validity limited to a narrow range of strain rates.

The master curve (Figure 4c) allows to directly choose the natural period approximating the desired sensitivity at the targeted strain rate. The m/k_{rep} ratio is thus fixed. If the strain rate range is known *a priori*, the master curve also gives an approximation of the variation of the strain rate sensitivity within the strain rate range.

3.2. Stress Level Calibration

For a given kinematical behavior of a packing, the stress level can arbitrarily be set. The integration of motion, for each particle, relies on the acceleration computed from Newton's second law. Hence, a multiplicative factor applied to both forces and masses leaves the kinematics of a packing, and its strain rate sensitivity, unchanged. Since our contact laws are linear elastic, we can use a common multiplicative factor on stiffnesses and masses.

The stiffnesses k_{rep} and k_{att} are scaled up to match the desired flow stress at the targeted strain rate. The mass m is proportionally adjusted, in order to maintain the correct strain rate sensitivity.

3.3. Scope of Validity

This two-step calibration allows us to reach arbitrary stress level, but displays limitations regarding the reachable strain rate sensitivity and strain rate.

We cannot model arbitrary strain rates with a given set of parameters. The numerical strain rate sensitivity depends on the strain rate. This effect can be controlled for very low sensitivities: a negligible sensitivity can be respected over various order of magnitudes of strain rate. However, a large tolerance must be accepted on higher sensitivities, which can only be reasonably approximated on narrow ranges of strain rate. The model also has intrinsic limits regarding the reachable strain rate sensitivities. Reaching higher sensitivity would require lower natural periods, for which the packings collapse and are unable to cope with the strain.

As a general conclusion for this section, our calibration procedure allows to choose independently the stress level and the strain rate sensitivity, tuning the mass of the particles and the stiffness of the contacts. The scope of validity, for controlled sensitivity, is limited to narrow strain rate ranges.

4. Application to a Single Material

In this section, we apply our methodology to represent arbitrary homogeneous single materials. Uniaxial compression is performed on initially cubic domains.

As our model relies on a collective motion of particles, a too small packing will not display the expected behavior (Figure 5). The kinematical behavior of single material cube in uniaxial compression is roughly observed with a few dozen of particles. With a few hundred of particles, the stress fails to represent the expected plastic trend, but already exhibits a correct order of magnitude (Figure 5a). A few thousand of particles allow a controlled relative error, around 10 %. Single material configurations are run in this section with 5×10^4 particles, with a typical relative error around 2 % (Figure 5b). The relative error is computed with respect to a packing of 1×10^6 particles, for which spatial convergence is considered to be reached.

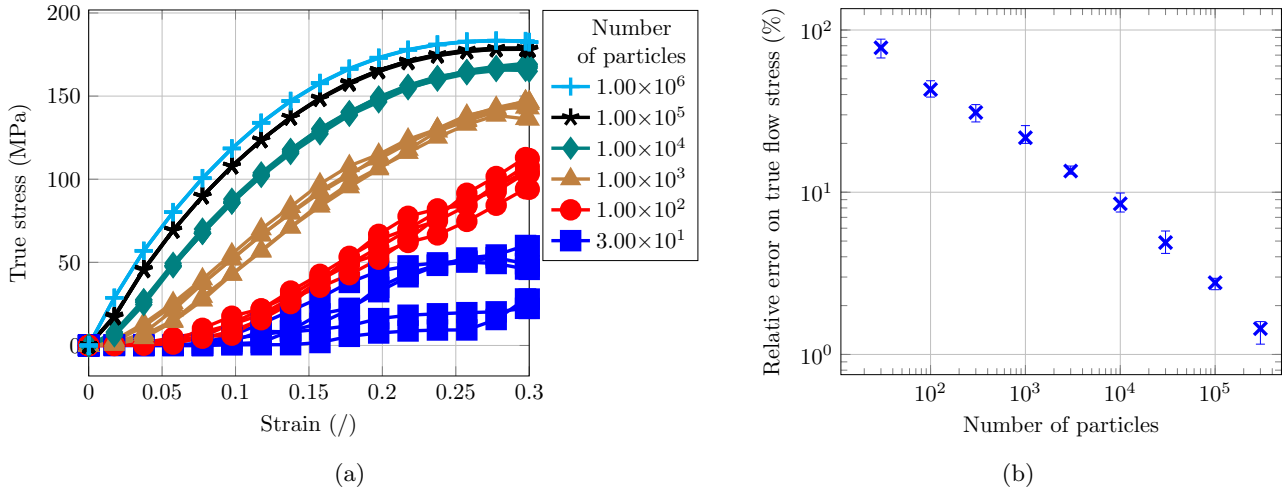


Figure 5: Spatial convergence for single material, from 30 to 1×10^6 particles. (a) Flow stress versus strain, using five distinct initial random packings for each packing size. (b) Relative error versus packing size, in regards to the converged simulation (1×10^6 particles packing).

In this paper, the behavior of the phases are inspired from an experimental setup: the hot forming at 400°C of a metallic composite, composed of a pure copper matrix and spherical inclusions of zirconium based bulk metallic glass. The numerical phases both have a flow stress close to 100 MPa in the strain rate window $1 \times 10^{-4} - 1 \times 10^{-3} \text{ s}^{-1}$, but with drastically distinct strain rate sensitivities. The negligible strain rate sensitivity phase is referred to as A, with a low natural period, the high sensitivity phase is referred to as B, with a high natural period. The corresponding numerical parameters are given in Table 1.

A key feature expected for a set of parameters is the conservation of the packing volume. The volume of the packings is estimated reconstructing a polyhedral mesh, using an algorithm implemented by Stukowski [34], based on the alpha-shape method. For both phases, the volume variation depends little on the strain rate. The prescribed compression decreases the volume, typically

Phase	Discrete parameters			Continuous parameters	
	Scale of the pairs			Macroscopic behavior	
	k_{rep} $\mu\text{N} \cdot \text{mm}^{-1}$	m g	t_0 s	M /	K $\text{MPa} \cdot \text{s}^M$
A	6.23×10^9	1.58×10^4	1×10^{-2}	2.45×10^{-3}	9.59×10^1
B	2.65×10^9	4.29×10^7	8×10^{-1}	4.90×10^{-1}	5.02×10^3

Table 1: Numerical parameters for the two phases. Time step $\Delta t = 5 \times 10^{-4} \text{ s}^{-1}$. Radii $R_{\text{crown}} = 0.75 \text{ mm}$, $R_{\text{seed}} = 0.5 \text{ mm}$. Stiffness ratio $k_{\text{rep}}/k_{\text{att}} = 10$.

about 5 % for A and 10 % for B (Figure 6). Before reaching a somewhat stable flow regime, the packing volume decreases in first 0.2 of strain. Most of the volume variation occurs within this initial stage, the volume then stabilizes on a plateau before a final increase of the error at larger strains, above 0.6. This trend, and its initial transitory regime, will also be observed for the flow stress (Figure 9a).

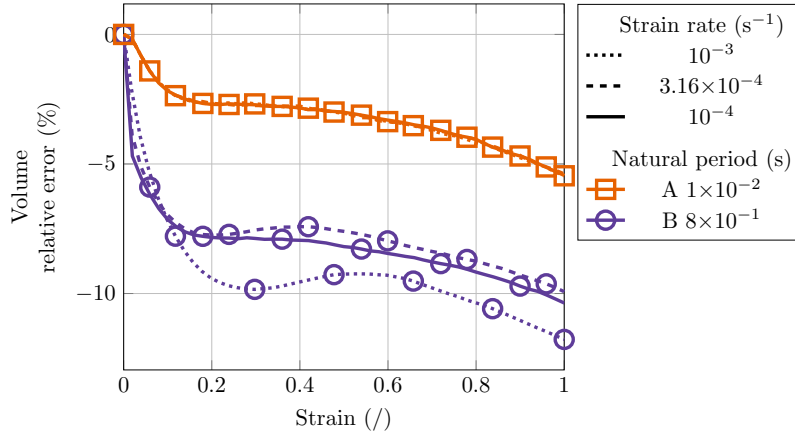


Figure 6: Volume conservation for single material. Relative error on volume versus strain for 5×10^4 particles.

Regarding the kinematical behavior of a packing (Figure 7), the overall cuboidal shape is conserved, but the sharp edges tend to be blurred along with the strain. This is understood as an effect of the surface tension induced by the attractive component of the contact law. As the discretization by particles creates local defects in the geometry, the initially flat faces become slightly wavy.

In order to simulate quasi-static phenomena, the behavior of the packing must be independent from the way the strain is applied. The total forces acting on the boundary conditions, the top and bottom meshes, respectively mobile and fixed, must balance. If a mesh moves too fast, the macroscopic equilibrium is lost and the strain localizes next to the moving plane. At a given strain rate, the equilibrium relative error depends on the natural period, but is of the same

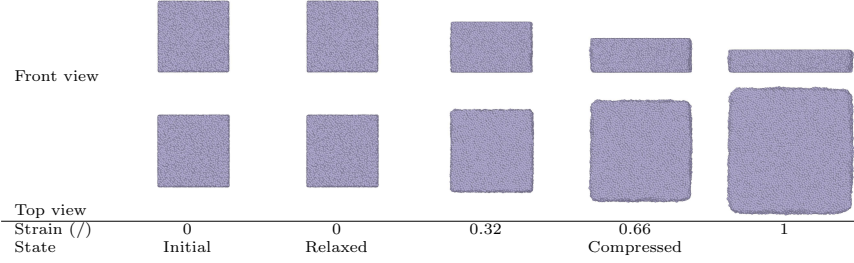


Figure 7: Single material packing. Natural period 0.8 s , strain rate $3.16 \times 10^{-4}\text{ s}^{-1}$, 5×10^4 particles.

order of magnitude for all strains. The equilibrium error (Figure 8) is always inferior to 0.1% for both phases in the studied strain rate range.

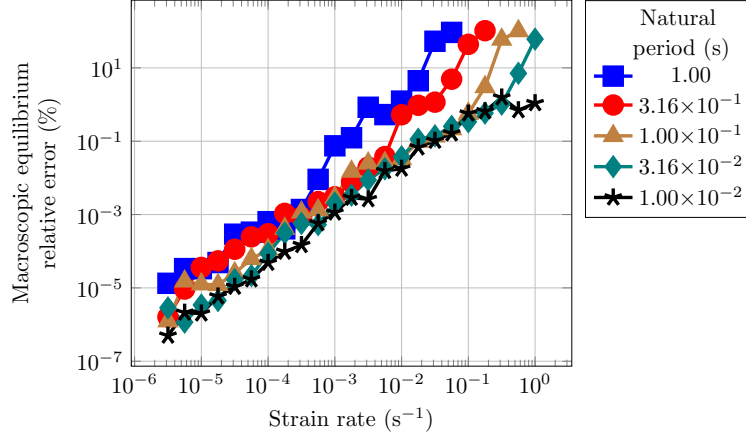


Figure 8: Macroscopic equilibrium for single material. Relative error versus strain rate for 5×10^3 particles, strain 0.3 . Effect of the natural period.

Typical profiles of stress-strain curves are presented Figure 9a. In this section, the true stress is computed using an estimation of the cross-section, based on the current macroscopic strain and the initial volume, assuming its variations (Figure 6) are acceptable. As for the volume evolution, a transitory stage can be observed at the beginning of the deformation, where the stress rises to reach the plastic plateau. The flow stress then oscillates around a fairly constant value. An overshoot effect of the stress can be observed at higher strain rates for the phase B.

For each phase, the values at a strain of 0.3 is used to compute the Norton approximation, by least-square fitting (Figure 9b and Table 1). As discussed in Section 3.3, the high sensitivity phase, B, is only valid within one order of magnitude of strain rate, the approximation is not reasonable when the strain rate is out of the studied range.

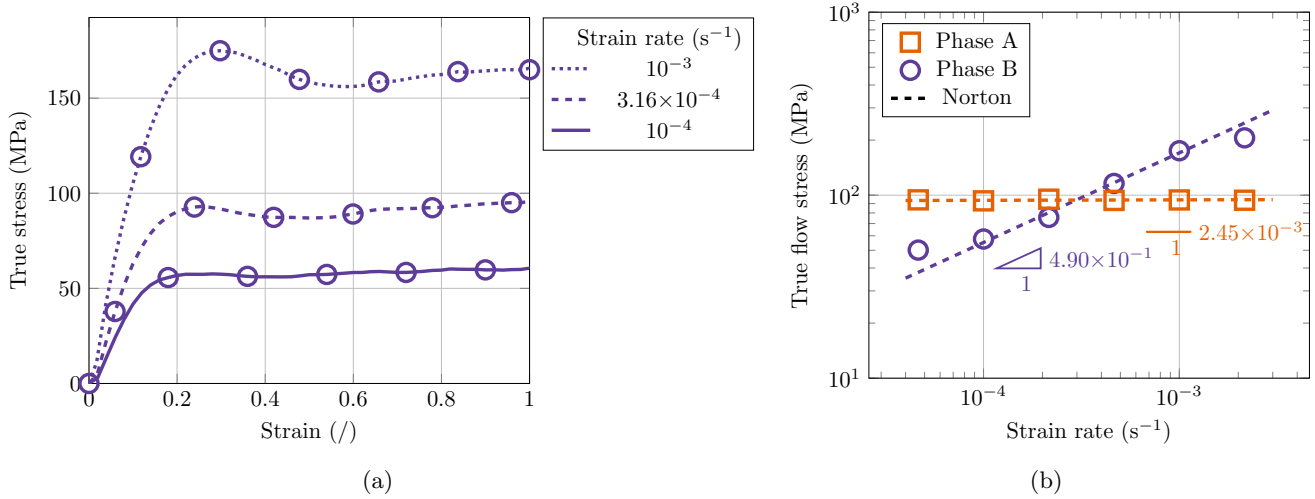


Figure 9: Single materials. Flow stress and strain rate sensitivity. (a) Flow stress versus strain. Effect of the strain rate. Phase B. (b) Norton law approximation for phases A and B, based on flow stress values at a strain of 0.3. Strain rate sensitivity of both phases in the range $1 \times 10^{-4} - 1 \times 10^{-3}$ s $^{-1}$.

To sum-up, two phases, with distinct strain rate sensitivities, are independently defined in this section, with a Norton law approximation of their continuous properties. Numerical properties and precision are evaluated.

5. Application to Bi-Materials

The DEM parameters have been calibrated separately for each phase. Keeping in mind the limitations of the single material model, we here evaluate the reliability of the model for bi-material composites. Three elementary geometrical bi-material configurations are studied: parallel, series and spherical inclusion. The three geometries are discretized with 5×10^4 particles and uniaxially compressed up to a strain of 0.3, at prescribed strain rates. In the studied configurations, interaction parameters at the interfaces had little influence on the macroscopic results. They have been set to the average of the parameters of phase A and B. The shape of the phase and the engineering macroscopic stress are used to compare the results with analytical and FEM references, for various volume fractions. The choice of engineering over true stress allows to use a simple and consistent comparison metric: no unique true stress can be defined for non homogeneous strain configurations.

Total Lagrangian FEM simulations, well suited for our elementary geometrical configurations and limited strains, are run using Code_Aster [35]. The visualization of the FEM results are rendered using PARAVIEW [36]. The quadratic tetrahedral elements used rely on an incompressible finite transformation formulation. Top and bottom nodes follow prescribed vertical motion

and lateral sides deform freely. The geometrical models are reduced using the symmetries of the problems, while the DEM simulate the full geometries. In FEM, at the interface between two phases, the nodes are shared, prohibiting any relative motion, which is the most severe difference with our DEM simulations. In the experimental background of this study, the phases have very little adhesion at the interface. Both materials follow a Norton law. The phase B uses the continuous parameters identified in Section 4 (Table 1). To allow an easier numerical convergence of the model, the numerical strain rate sensibility of the phase A is slightly increased for the FEM simulations ($M = 3.05 \times 10^{-2}$ and $K = 120 \text{ MPa} \cdot \text{s}^M$). In the range $1 \times 10^{-4} - 1 \times 10^{-3} \text{ s}^{-1}$, the induced relative error on the flow stress is $\pm 3\%$.

5.1. Parallel Configuration

A cube is vertically divided into two cuboidal phases, for various volume fractions, and vertically compressed at constant strain rates. The engineering stress is compared to a mixture law, linear with the volume fraction.

In this simple configuration, little interaction should take place between the phases, and in ideal conditions, a homogeneous strain for both phases is expected. In the DEM simulations, the global geometry of each phase remains close to a cuboid along the deformation (Figure 10).

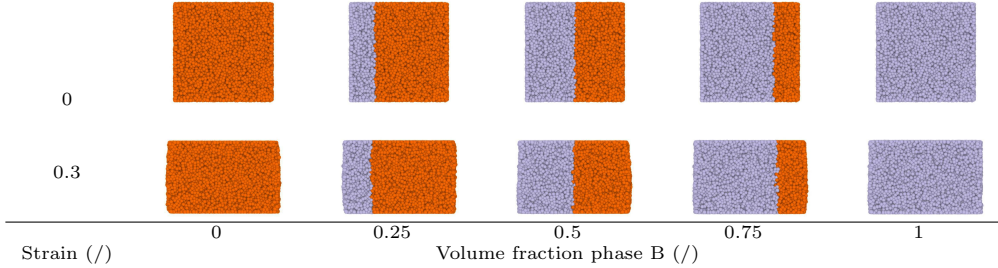


Figure 10: Bi-material parallel configuration. Front view, transverse section. Strain rate $4.64 \times 10^{-4} \text{ s}^{-1}$.

At given strain rate $\dot{\varepsilon}$, the true stress in the phases being independently defined by the Norton law, the global true stress σ_{true} can be computed with an elementary mixture law [37, p.99], linear with the volume fraction f of the phase B:

$$\sigma_{\text{true}}(f, \dot{\varepsilon}) = f \cdot K_B \dot{\varepsilon}^{M_B} + (1 - f) \cdot K_A \dot{\varepsilon}^{M_A} \quad (5)$$

To provide a consistent metric for all configurations, the engineering stress σ_{engineer} is used as reference. It is computed at a given strain ε (Equation 6), based on the true stress and the volume conservation:

$$\sigma_{\text{engineer}}(f, \dot{\varepsilon}, \varepsilon) = \exp(-\varepsilon) \cdot \sigma_{\text{true}}(f, \dot{\varepsilon}) \quad (6)$$

The engineering stress-true strain profile, as in single material configuration, displays a transitory regime, typically in the first 0.15 of strain, with a progressive rise towards the flow stress (Figure 11a).

The DEM model is able to capture, at the precision of the single phases, the linear pattern of flow stress with the volume fraction (Figure 11b). With a rougher discretization, for example only a thousand particles per phase, the result remains qualitatively close, degrading the accuracy by a few percent.

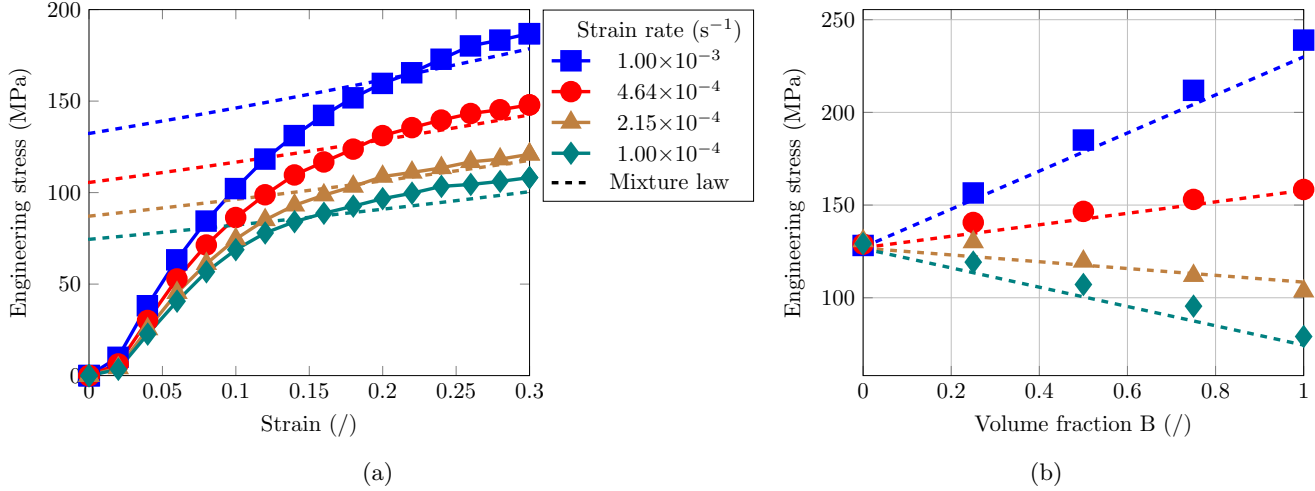


Figure 11: Bi-material parallel configuration. Effect of the strain rate on the flow stress. Theoretical reference: mixture law from Equation 6. (a) Engineering flow stress versus strain. Fixed volume fraction: 0.5. (b) Linear trend of the engineering flow stress, at a strain of 0.3, with the volume fraction.

5.2. Series Configuration

A cube is horizontally divided into two cuboidal phases, for various volume fractions, and vertically compressed at constant strain rate. Using symmetry, one fourth of the geometry is modeled with the FEM, using approximately 1.3×10^3 nodes. For the full geometry, the ratio DEM particles to FEM nodes would be a little under 10.

In this geometrical configuration, the strain is not *a priori* homogeneous anymore. Due to distinct strain rate sensitivities, one phase preferentially deforms depending on the strain rate, which is qualitatively observed both in FEM and DEM simulations. Qualitatively (Figure 12), phase B (bottom phase) deforms more at lower strain rate. Phase A, at high strain rates, deforms more homogeneously in DEM than in FEM, possibly due to more permissive contact conditions between phases. Thus, the "mushroom" shape is slightly blurred in this strain rate range.

FEM and DEM are in good agreement, after the transient regime observed in DEM, within a few percents of relative error (Figure 13a). In the strain rate validity range, the DEM model is thus able to capture the final flow stress evolution with respect to the volume fraction (Figure 13b). As a side note, the

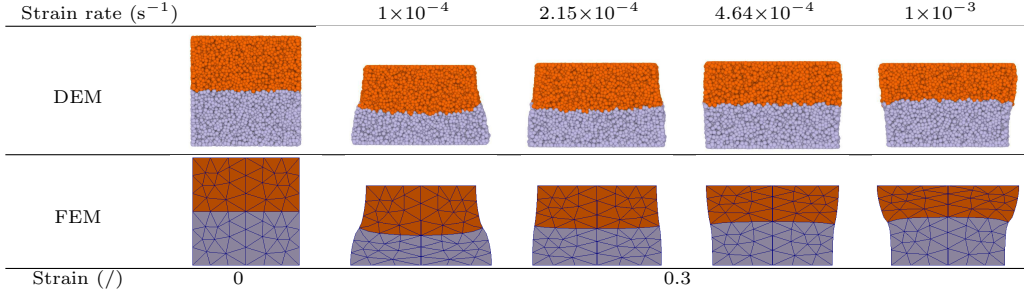


Figure 12: Bi-material series configuration. Volume fraction 0.5. Bottom phase: B (high sensitivity). Front view, transverse section.

heterogeneity of the strain in the series configuration is responsible for a non-linear variation of the flow stress with respect to the volume fraction. This effect of the geometry of the bi-material, clearly displayed at $1 \times 10^{-3} \text{ s}^{-1}$ (Figure 13b), is correctly reproduced.

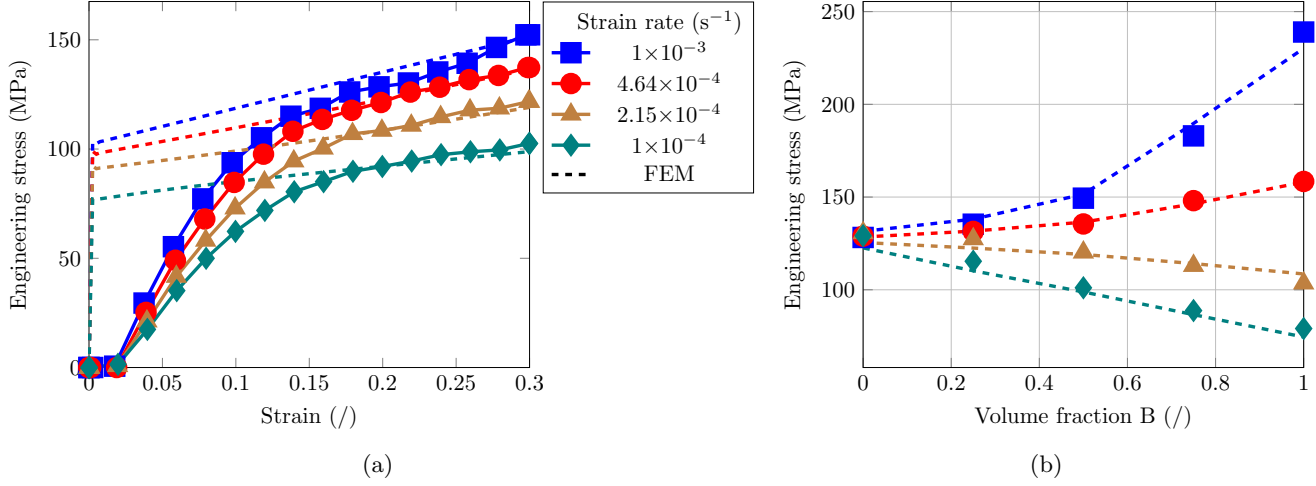


Figure 13: Bi-material series configuration. Effect of the strain rate on the flow stress. Numerical reference: FEM simulations. (a) Engineering flow stress versus strain. Fixed volume fraction: 0.5. (b) Non-linear trend of the engineering flow stress, at a strain of 0.3, with the volume fraction.

5.3. Spherical Inclusion Configuration

A single spherical inclusion of phase B is placed in the center of a phase A cube, with a fixed volume fraction of 20 % of phase B inclusion. Using symmetry, one eighth of the geometry is modeled with the FEM, using 2.1×10^3 nodes. For the full geometry, the ratio DEM particles to FEM nodes would be a little under 3.

Qualitatively, two typical kinematical tendencies of the matrix are displayed in FEM (Figure 14), with an intermediary state of homogeneous co-deformation:

- A barrel shape of the sample, when the flow stress of the inclusion is low, at lower strain rates;
- An hourglass shape, when the flow stress of the inclusion is high, at higher strain rates.

In the FEM simulations, the hourglass shape of the matrix is strongly emphasized by the non-sliding interface between phases. While the barrel shape is easily displayed at low strain rates in DEM, the hourglass shape is only clear at higher strain rates, outside the validity domain studied range.

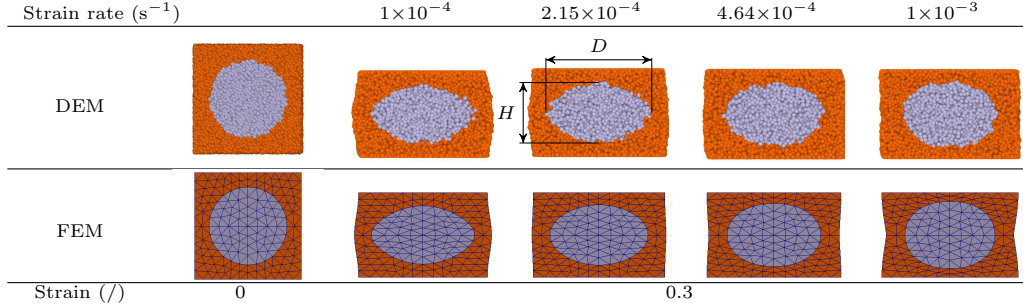


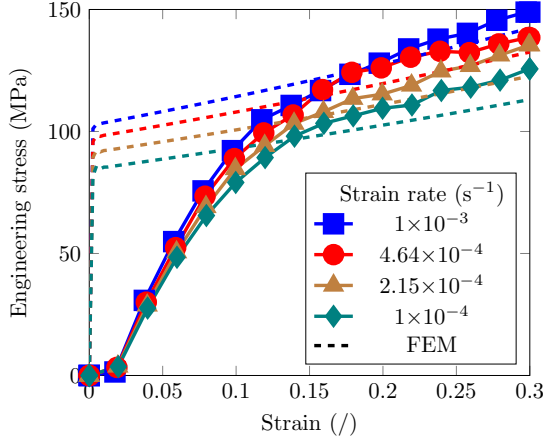
Figure 14: Bi-material spherical inclusion (phase B) configuration. Front view, transverse section. Illustration of the height H and diameter D used to compute the shape factor S_f (Equation 7).

Although we would expect lower stresses with a less constrained system, the flow stress is overestimated (Figure 15b), by about 10% on the studied strain rate range, even if the tendency is acceptable after the transient regime (Figure 15a).

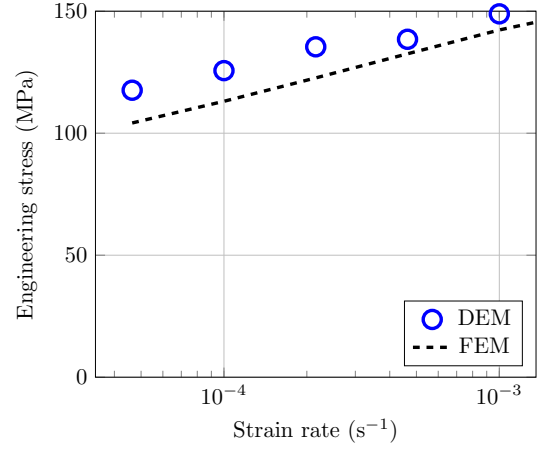
To quantitatively compare the models from a kinematical perspective, we study the macroscopical shape factor S_f of the inclusion, which is less sensitive than the matrix shape to the interface definition. This factor (Equation 7) is the ratio of the inclusion height H , in the compression direction, and diameter D (Figure 14), averaged in all perpendicular directions:

$$S_f = H/D \quad (7)$$

For the DEM simulations, this value is approximated computing the shape factor of an equivalent ellipsoid, having the same inertia matrix as the cloud of particles modeling the inclusion. At all strain rates, at the very beginning of the applied strain (Figure 16a), the inclusion remains roughly spherical for a few percent of strain, and follow a similar trend as in FEM after the transient regime. In the validity range of the phase B, the final shape factor (Figure 16b) is underestimated with a relative error of about 5%. The inclusion deforms more in DEM than in FEM.

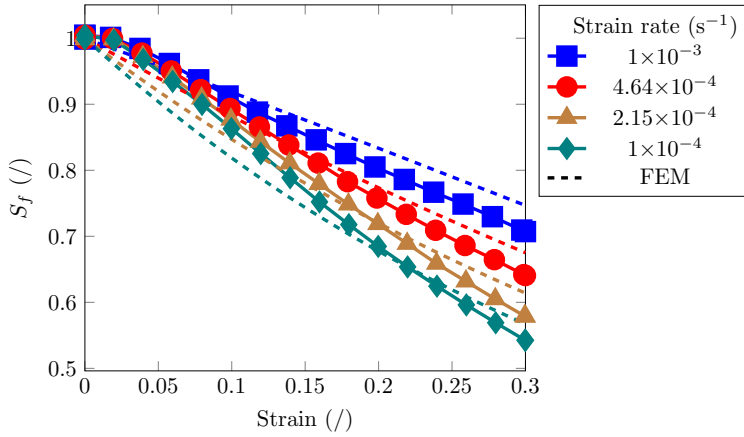


(a)

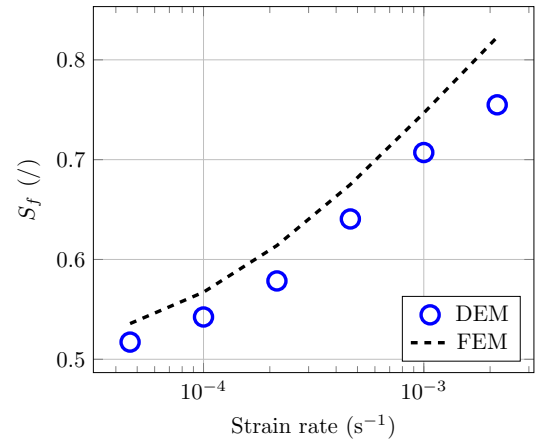


(b)

Figure 15: Bi-material spherical inclusion configuration. Effect of the strain rate on the flow stress. Numerical reference: FEM simulations. Unique volume fraction of inclusion (phase B): 0.2. (a) Engineering flow stress versus strain. (b) Engineering flow stress versus strain rate, at a strain of 0.3.



(a)



(b)

Figure 16: Bi-material spherical inclusion configuration. Effect of the strain rate on the shape factor of the inclusion. Numerical reference: FEM simulations. Unique volume fraction of inclusion (phase B): 0.2. (a) Shape factor versus strain. (b) Shape factor versus strain rate, at a strain of 0.3.

To evaluate the error on the shape factor depending on the roughness of the discretization, an identical geometry is modeled with smaller packings of particles. The relative error for the final shape factor is computed using the 1×10^6 particles configuration as reference, where 1×10^5 particles discretize the inclusion. For each size, five random initial packings are tested. The chosen test case is harsh for our model: for the smaller packings, the meshes may interact directly with the particles of the inclusion at the end of the deformation.

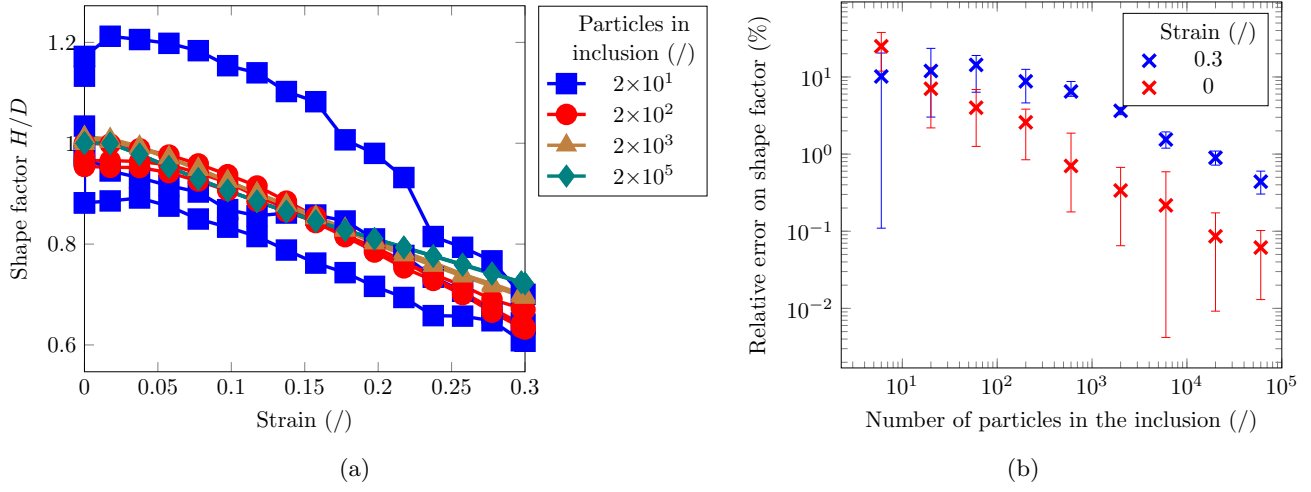


Figure 17: Bi-material spherical inclusion configuration. Spatial convergence: effect of the number of particles discretizing the inclusion on the shape factor. (a) Shape factor versus strain. Typical results for three (out of a total five computed) random packings. (b) Error on the shape factor versus the number of particles used to discretize the inclusion. Reference for relative error: 1×10^5 particles used to discretize the inclusion. Minimum, maximum and average error for five random packings.

A very rough description of the inclusion, with 20 particles for example, remains too inaccurate to catch more than an order of magnitude of the deforming trend (Figure 17a), and the initial shape factor is already far from a perfect sphere, with little repeatability. In a realistic context, such a rough discretization can only reasonably be used to capture the position of an inclusion in a composite. With a finer discretization, starting with a few hundred particles, the qualitative trend can be captured and the repeatability improves: it becomes possible to estimate the necessary discretization for an arbitrary precision (Figure 17b). The purely geometrical error, on the initial state, is about an order of magnitude smaller than the final error, after compression. For a final error under 10 %, more than 200 particles must discretize the inclusion.

To sum up, in this section, three elementary geometrical bi-material configurations were tested. The flow stress and the macroscopic geometrical evolution are compared to FEM simulations. Using a discretization of a few DEM particles per FEM nodes, the error is of the order of magnitude of the expected

precision for a single material. The bi-material simulation does not seem to introduce major error sources. The comparison of the computing time between the numerical methods must be taken with caution and is not detailed here. Indeed, in the FEM approach, a non-linear set of equations must be solved at each time step, the computing time may vary by several orders of magnitude for different strain rates or material parameters. The DEM model computing time is much more calculable, which is discussed at the end of Section 6.

6. Computation on a Real 3D Full Sample

As an illustrative example, the methodology is applied to the real 3D microstructure of a full sample, obtained by X-ray microtomography at the ESRF (beamline ID19). The studied material is a metallic composite, with a crystalline copper matrix and spheroidal inclusions of amorphous zirconium alloy. The total volume of the sample is approximately 0.5 mm^3 , containing a volume fraction of inclusion of 15 %, with diameters up to a few dozens of micrometers. The voxelized image has a size of $594 \times 591 \times 669$ voxels, with a voxel size of $1.3 \mu\text{m}$. The purpose of this section is not to compare quantitatively numerical results and experimental *in situ* results [38], but to underline the potential of the method for large arbitrary data sets.

Starting from the three-dimensional voxelized image, used as a mask on a random packing of particles, the discretization of the geometry has a low algorithmic cost:

1. An 3D image is binarized in two colors: matrix and inclusions;
2. A cuboidal random packing is built, with the same aspect ratio as the image;
3. The image is fitted to the size of the packing, using an affine transformation;
4. For each particle, the color at its center is used to set the material type.

For very large data sets, a smaller periodic packing can be replicated in all directions, minimizing the cost of generation of this initial packing.

The used packing contains about 3.36×10^6 particles, with a number of voxels to number of particles ratio of 70. As shown in Figure 18, about 170 physical inclusions are discretized, using between 500 and 5 discrete element particles each. The rough discretization of the smallest inclusions, for example the further left inclusion in Figure 19, is not precise enough to allow a strain evaluation, only the inclusion position can be tracked. For the bigger inclusions, the expected error on the shape factor after 0.3 of strain is around 10 % (Figure 17b).

The sample is uniaxially compressed up to 0.3 at $1 \times 10^{-3} \text{ s}^{-1}$. Local and global illustrations are given in Figures 18 and 19, with relative displacement and deformation of the inclusions. The computation was run on an Intel Xeon E5520, using 8 processors. The 6×10^5 steps for 3.36×10^6 particles were executed in $8 \times 10^5 \text{ s}$, less than 10 days. The computation time is linear with the number of steps and of particles. As long as the load is properly balanced between

processors and that the geometry of the sub-domains keeps the volume of the communications between processors reasonable, the DEM solver scales properly with the number of processors. On the studied geometries, the roughly cuboidal overall shape of the samples allows simple dynamic balance of the load between processors. The computing time can thus be reliably estimated on a given machine, roughly 3×10^{-6} cpu second per particle and per time step, for a single processor in the given example.

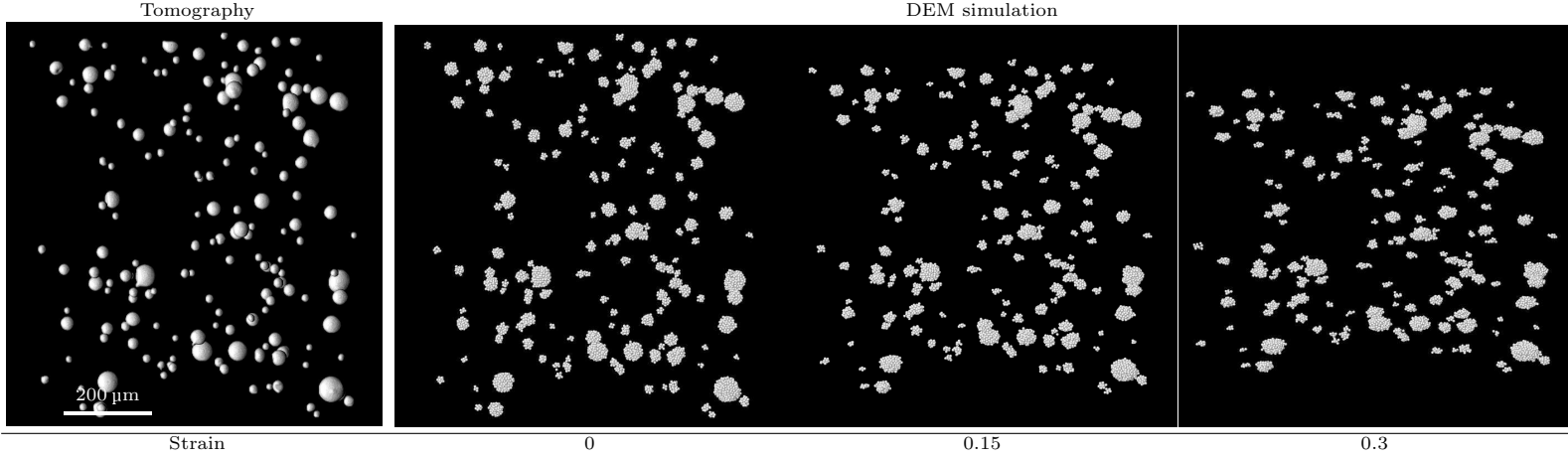


Figure 18: Discretization and compression of the full sample. 3D view of the inclusions only, the matrix is hidden. Vertical compression axis.

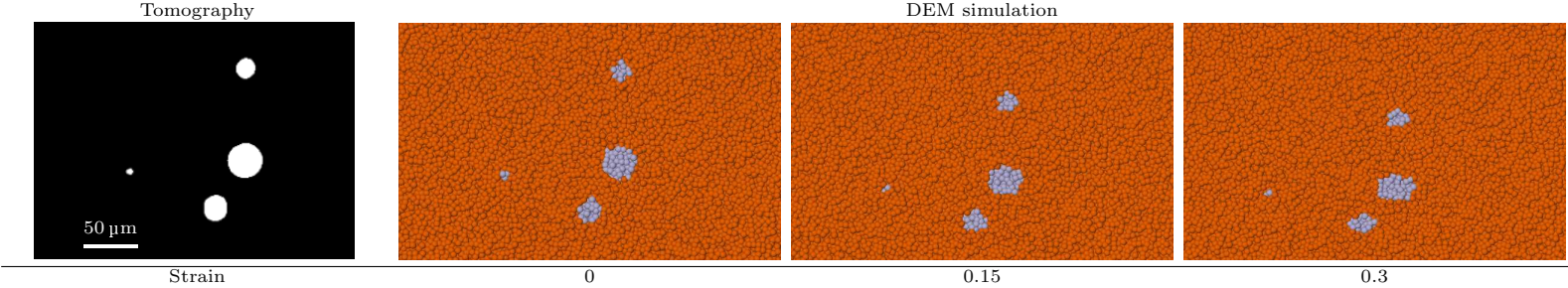


Figure 19: Zoom on a local configuration. Discretization and compression. Cross-section of the matrix and the inclusions. Vertical compression axis.

This section illustrated that the proposed methodology can be applied to large arbitrary real microstructure data. The discretization has low algorithmic cost, but the model does not allow yet a simple way to locally adapt the discretization roughness. The cost of the computation can be reliably estimated as the model does not depend on non-linear resolutions.

7. Conclusion

In this paper, we propose a DEM model for large compressive strains in plastic and viscoplastic continuous media. The continuous materials are discretized with packings of spheres, with attractive and repulsive interaction forces. A double-radius model is implemented, to geometrically manage the history of the contacts. The interaction laws are kept as elementary as possible, with no damping or tangential forces, which provides only second order improvements of the behavior.

A calibration of the numerical parameters is proposed to target continuous parameters, approximating a macroscopic viscoplastic Norton law. This model can represent large strains and shape changes of dense materials under uniaxial compression, with a controlled macroscopic volume variation. After a transitory regime at the beginning of the strain, the flow stress and volume stabilize to a plateau value. Despite limitations, in terms of reachable strain rate sensitivities and strain rate validity range, the viscoplastic approximation is valid over several orders of magnitude of strain rate at low sensitivities.

Elementary bi-material geometries are compared to analytical and FEM references. At the first order, the inaccuracies in bi-material configurations seem to derive directly from the limitations on the single phase materials. The order of magnitude of the error on macroscopic flow stresses is in both cases 10 % in tested configurations. Developing better single material model should lead to better composite description. Ongoing work focus on widening the validity range in strain rate, and on other solicitation types, like tension or torsion.

The discretization of a real full sample geometry, from a voxelized three-dimensional image, has a low algorithmic cost. For the time being, no size adaptivity mechanism has been implemented, as large data sets can be handled through the parallelization of the computing effort. Further developments could allow a local refinement of the discretization of the geometry.

The methodology cannot be as accurate as the FEM to model a continuous medium. However, to describe phenomena where the motions of the discontinuities and their interactions are predominant, the DEM is innately suited and provides a numerically robust alternative. No complex non-linear system has to be solved at each time step, thus the computation time can be predicted and kept under control. An intended extension of the model is the description of self-contact, for example in metallic foam deformation. Other extensions are the description of topological changes, naturally handled in the DEM framework, for example pore closure or phase fragmentation in composites.

References

- [1] Thanh Tung Nguyen, Julien Yvonnet, Michel Bornert, and Camille Chateau. Initiation and propagation of complex 3D networks of cracks in heterogeneous quasi-brittle materials: direct comparison between in situ testing-microCT experiments and phase field simulations. *Journal of the Mechanics and Physics of Solids*, 95:320–350, 2016.

- [2] Anthony Berteau, Jean-Dominique Bartout, Yves Bienvenu, and Samuel Forest. On the creep deformation of nickel foams under compression. *Comptes Rendus Physique*, 15(8):705–718, 2014.
- [3] Diego A Berazategui, Miguel A Cavaliere, Luca Montelatici, and Eduardo N Dvorkin. On the modelling of complex 3D bulk metal forming processes via the pseudo-concentrations technique. Application to the simulation of the Mannesmann piercing process. *International journal for numerical methods in engineering*, 65(7):1113–1144, 2006.
- [4] KS Al-Athel and MS Gadala. Eulerian volume of solid (vos) approach in solid mechanics and metal forming. *Computer methods in applied mechanics and engineering*, 200(25):2145–2159, 2011.
- [5] Ken Kamrin, Chris H Rycroft, and Jean-Christophe Nave. Reference map technique for finite-strain elasticity and fluid–solid interaction. *Journal of the Mechanics and Physics of Solids*, 60(11):1952–1969, 2012.
- [6] Tayfun E Tezduyar, Mittal Behr, S Mittal, and J Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary tests. *Computer methods in applied mechanics and engineering*, 94(3):353–371, 1992.
- [7] Jean Donea, Antonio Huerta, J.-Ph. Ponthot, and A Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian Methods. In Erwin Stein, René de Borst, and Thomas J.R. Hughes, editors, *Encyclopedia of computational mechanics*, volume 1: Fundamentals, chapter 14, pages 413–437. Wiley, 2004.
- [8] Y Avevor, J Vincent, L Faure, A Moufki, and S Philippon. An ALE approach for the chip formation process in high speed machining with transient cutting conditions: modeling and experimental validation. *International Journal of Mechanical Sciences*, 2017.
- [9] Thierry Coupez, Nathalie Soyris, and Jean-Loup Chenot. 3-D finite element modelling of the forging process with automatic remeshing. *Journal of Materials Processing Technology*, 27(1):119–133, 1991.
- [10] Marc Bernacki, Stéphane Andrietti, Jean-Loup Chenot, Pierre-Olivier Bouchard, Lionel Fourment, Elie Hachem, and Etienne Perchat. Recent and future developments in finite element metal forming simulation. *Computer Methods in Materials Science*, 15:265–293, 2015.
- [11] Yixiu Shu, Yazhi Li, Minge Duan, and Fan Yang. An X-FEM approach for simulation of 3-D multiple fatigue cracks and application to double surface crack problems. *International Journal of Mechanical Sciences*, 2017.
- [12] Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87, 1995.

- [13] P Becker, SR Idelsohn, and E Oñate. A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh. *Computational Mechanics*, 55(6):1091–1104, 2015.
- [14] Piotr Breitskopf, Marcela Cruchaga, Diego Celentano, Pierre Villon, and Alain Rassineux. A front remeshing technique for a lagrangian description of moving interfaces in two-fluid flows. *International journal for numerical methods in engineering*, 66(13):2035–2063, 2006.
- [15] B Nayroles, G Touzot, and P Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational mechanics*, 10(5):307–318, 1992.
- [16] Ted Belytschko, Lei Gu, and YY Lu. Fracture and crack growth by element free galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, 2(3A):519, 1994.
- [17] Mohit Pant, I.V. Singh, and B.K. Mishra. A novel enrichment criterion for modeling kinked cracks using element free galerkin method. *International Journal of Mechanical Sciences*, 68, 3 2013.
- [18] Larry D. Libersky, Firooz A. Allahdadi, Theodore C. Carney, Jim R. Hipp, and Albert G. Petschek. High stain lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response. *Journal of Computational Physics*, 109:67–75, 1993.
- [19] Kirk A. Fraser, Lyne St-Georges, and Laszlo Istvan Kiss. A Mesh-Free Solid-Mechanics Approach for Simulating the Friction Stir-Welding Process. In Mahadzir Ishak, editor, *Joining Technologies*, chapter 3, pages 27–52. InTech, 2016.
- [20] John T. Foster, Stewart Andrew Silling, and Wayne W. Chen. Viscoplasticity using peridynamics. *International journal for numerical methods in engineering*, 81(10):1242–1258, 2010.
- [21] Pranesh Roy, Anil Pathrikar, S.P. Deepu, and Debasish Roy. Peridynamics damage model through phase field theory. *International Journal of Mechanical Sciences*, 2017.
- [22] Alexander Hrennikoff. Solution of problems of elasticity by the framework method. *Journal of applied mechanics*, 8(4):169–175, 1941.
- [23] Stewart A. Silling and Ebrahim Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures*, 83(17):1526–1535, 2005.
- [24] Damien André, Ivan Iordanoff, Jean-Luc Charles, and Jérôme Néauport. Discrete element method to simulate continuous material by using the cohesive beam model. *Computer Methods in Applied Mechanics and Engineering*, 213–216:113–125, 2012.

- [25] Yunteng Wang, Xiaoping Zhou, and Yundong Shou. The modeling of crack propagation and coalescence in rocks under uniaxial compression using the novel conjugated bond-based peridynamics. *International Journal of Mechanical Sciences*, 2017.
- [26] Rishi Kumar, Sarshad Rommel, David Jauffrès, Pierre Lhuissier, and Christophe L. Martin. Effect of packing characteristics on the discrete element simulation of elasticity and buckling. *International Journal of Mechanical Sciences*, 110:14–21, 2016.
- [27] Mohamed Jebahi, Damien André, Frédéric Dau, Jean-Luc Charles, and Ivan Iordanoff. Simulation of Vickers indentation of silica glass. *Journal of Non-Crystalline Solids*, 378:15–24, 2013.
- [28] Stewart A. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88(2):151–184, 2007.
- [29] Sergey Grigorievich Psakhie, A Yu Smolin, EV Shilko, GM Anikeeva, Yu S Pogozhev, MI Petrzhik, and EA Levashov. Modeling nanoindentation of TiCCaPON coating on Ti substrate using movable cellular automaton method. *Computational materials science*, 76:89–98, 2013.
- [30] Charles Voivret, Farhang Radjai, and Jean-Yves Delenne. Particle assembling methods. In Farhang Radjai and Frédéric Dubois, editors, *Discrete-element modeling of granular materials*, chapter 5, pages 123–150. Wiley-Iste, 2011.
- [31] Christoph Kloss, Christoph Goniva, Alice Hager, Stefan Amberger, and Stefan Pirker. Models, algorithms and validation for opensource DEM and CFD-DEM. *Progress in Computational Fluid Dynamics, An Int. J.*, 12(2/3):140–152, 2012.
- [32] Alexander Stukowski. Visualization and analysis of atomistic simulation data with OVITO - the Open Visualization Tool. *Modelling Simul. Mater. Sci. Eng.*, 18(1):015012, 2010. www.ovito.org.
- [33] Jean Lemaitre and Jean-Louis Chaboche. *Mechanics of solid materials*. Cambridge university press, 1994.
- [34] Alexander Stukowski. Computational Analysis Methods in Atomistic Modeling of Crystals. *JOM*, 66(3):399–407, 2014.
- [35] Électricité De France R&D. Code_Aster Open Source - general FEA software. www.code-aster.org.
- [36] A. Henderson. *ParaView Guide, A Parallel Visualization Application*. Kitware Inc., 2007.

- [37] Dominique François, André Pineau, and André Zaoui. *Comportement mécanique des matériaux. Viscoplasticité, endommagement, mécanique de la rupture, mécanique du contact*, volume 2. 1993.
- [38] Therese Bormann, Robin Gibaud, Pierre Lhuissier, Elodie Boller, Alexander Rack, and Luc Salvo. Investigation of the deformation process of amorphous/crystalline metal composites by in situ x-ray micro-tomography and dem modelling. *Conférence matériaux et grands instruments & matériaux numériques – Analyses in situ : expérience / modélisation, Paris, France*, September 2016.

Notation

General convention

Z scalar

\overline{Z} vector

$\overline{\overline{Z}}$ tensor of order two

Main Introduced Notation

\overline{a} particle acceleration

α_{e_n} angle threshold between \overline{n} and $\overline{e_n}$ in self-contact detection; Section 9.5

α_{ij} angle threshold between both \overline{n} in self-contact detection; Section 9.5

d particle diameter

\mathcal{D} density of particle; Section 8.6

Δt time step

E continuous medium Young's modulus

$\overline{e_n}$ unit vector from a particle to its neighbor

E_k kinetic energy; Section 8.6

E_p potential elastic energy; Section 8.6

ε uniaxial macroscopic strain

$\overline{\varepsilon}$ strain

$\dot{\varepsilon}$ uniaxial macroscopic strain rate

f signed norm of the interaction force

\overline{f} interaction force

f_{att} attractive pair force threshold; Section 9.1

h distance between two particles

i indentation of a pair

\mathcal{I} inertial numer; Section 8.6

i_r relative indentation of a pair (indentation/radius)

- k generic interaction stiffness
- K stress level in the Norton law; Equation 11.1 on page 120
- \mathcal{K} stiffness level; Section 8.6
- k_{att} attractive pair stiffness; Section 9.1
- k_{rep} repulsive stiffness; Section 9.1
- \bar{l} branch vector, pointing from the center of the particle to the center of its neighbor
- m particle mass
- M strain rate sensitivity in the Norton law; Equation 11.1 on page 120
- N number of particles in the system
- \bar{n} particle outward vector; Section 9.5
- N_{mag} magnitude threshold on \bar{n} in self-contact detection; Section 9.5
- N_n inverse of the strain rate sensitivity in the Norton law; Equation 11.1 on page 120
- ν continuous medium Poisson's ratio
- P confinement pressure; Section 8.6
- Q energy ratio; Section 8.6
- r generic particle radius
- r_{crown} particle crown radius; Section 9.1
- r_{seed} particle seed radius; Section 9.1
- ρ continuous medium density
- σ uniaxial macroscopic stress
- $\bar{\bar{\sigma}}$ macroscopic stress
- t time
- t_0 image of natural period; Equation 8.4 on page 90
- T_g amorphous metallic alloy glass transition temperature
- T_x amorphous metallic alloy crystallization temperature
- \bar{v} particle velocity
- V_{tot} volume of the system
- \bar{x} particle position
- X_{wall} multiplicative factor for mesh/particle interactions; Section 9.1

Glossary

BILIN custom interaction law; defined in Section 9.1 and Equation 9.1; applied in Part IV.

TRILIN custom interaction law; defined in Section 9.1 and Equation 9.2; applied in Part V.

ALE arbitrary Lagrangian Eulerian; Chapter 4 [58].

bondPD bond-based peridynamics; Section 5.3.2.2 [209].

BSD-3 3-clause Berkeley Software Distribution (BSD) license.

C compiled programming language.

C++ compiled programming language; isocpp.org.

CA cellular automata; Chapter 4 [174, Chap. 6].

CD contact dynamics; Section 5.3.2.1.

code_aster code d'analyse des structures et thermomécanique pour des études et des recherches; FEM solver; released under GPL [245].

contact mechanical interaction by contact of interfaces of distinct physical objects.

CPU central processing unit.

CZM cohesive zone model; Section 5.2.1.1.1 [65].

DEM discrete element method; Section 8 [50]; not to be mistaken for the difEM, sharing the acronym DEM in the literature.

difEM diffuse element method; Section 5.2.1.2 [158]; referred to as the DEM in the literature.

DSD/SST deforming spatial domain/stabilized space time; Chapter 4 [230].

EDM event-driven method; Section 5.3.2.1 [174, Chap. 3].

EFG element-free Galerkin method; Section 5.2.1.2 [23].

ESRF European synchrotron radiation facility; Section 2.4.

FEM finite element method; Section 5.2.1.1.1 [55].

finite transformation mechanical transformations involving large (but not infinite) strain, displacement or rotation; for which the difference between reference and deformed configurations is not considered negligible [192, p.59]; in contrast with infinitesimal transformations.

fortran compiled programming language; www.nag.co.uk/sc22wg5.

FVM finite volume method; Chapter 4.

GFDM generalized finite difference method; Section 5.2.1.2 [135].

GPGPU general-purpose computing on graphics processing units.

GPL GNU's Not Unix (GNU) general public license.

GPU graphics processing unit.

lammmps large-scale atomic/molecular massively parallel simulator; MD solver software [172]; released under GPL.

LBM lattice Boltzmann method; Chapter 4.

LGPL GNU's Not Unix (GNU) lesser general public license.

liggghts lammmps improved for general granular and granular heat transfer simulations; DEM solver software forked from [120]; we refer in our work to the public version, released under GPL.

MCA movable cellular automata; Section 5.3.2.2 [177].

MD molecular dynamics; Section 5.3.2.2 [183].

Mka3D discrete formalism for continuous media; Section 5.2.2 [154].

MLIRT moving Lagrangian interface remeshing technique; Section 5.2.1.1.2 [37].

MPI message passing interface.

MPM material point method; Section 5.2.1.1.2 [225].

NEM natural element method; Section 5.2.1.2 [223].

NSCD nonsmooth contact dynamics; Section 5.3.2.1 [110].

OpenMP open multi-processing.

ovito open visualization tool; particle data analysis and visualization software [220]; released under GPL.

PAF particle and force method; Section 5.2.2 [52].

paraview data analysis and visualization software [100]; released under 3-Clause BSD License.

PDE partial differential equation.

PFEM particle finite element method with moving mesh; referred to in the literature as the PFEM; Section 5.2.1.1.1 [107].

- PFEM2** particle finite element method with fixed mesh; referred to in the literature as the PFEM-2; Section 5.2.1.1.2 [190].
- PIC** particle in cell method; Section 5.2.1.1.2 [225].
- PSFL** python software foundation license.
- PUM** partition of unity method; Section 5.2.1.2 [150].
- python** interpreted programming language; released under PSFL; www.python.org.
- RPKM** reproducing particle kernel method; Section 5.2.2 [136].
- salome** computer aided design and pre/post processing softwares for FEM; including a fork of paraview; released under LGPL; <http://www.salome-platform.org/>.
- salome-meca** generic FEM framework, including and salome; released under LGPL.
- self-contact** mechanical interaction by contact of the interfaces of a single physical object.
- SEM** scanning electron microscopy.
- SPH** smooth particle hydrodynamics; Section 5.2.2 [130].
- SPS** spark plasma sintering.
- statePD** state-based peridynamics; Section 5.2.2 [210].
- STZ** shear transformation zone; proposed mechanism for inelastic strain in amorphous metallic alloys; [11].
- XFEM** extended finite element method; Section 5.2.1.1.3 [157].

Bibliography

- [1] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.
- [2] Andrew Adamatzky, R. Alonso-Sanz, and A. Lawniczak. *Automata-2008: Theory and applications of cellular automata*. Luniver Press, 2008.
- [3] Ayesha Afzal. The cost of computation - Metrics and models for modern multicore-based systems in scientific computing. Master’s thesis, FAU Erlangen, 2015.
- [4] Andreas Aigner. Liggghts 1.4 version history. www.cfdem.com/node/42, August 2011.
- [5] K. S. Al-Athel and M. S. Gadala. Eulerian volume of solid (VOS) approach in solid mechanics and metal forming. *Computer methods in applied mechanics and engineering*, 200(25):2145–2159, 2011.
- [6] Alain. *Propos sur les pouvoirs – Éléments d’éthique politique*. Gallimard, 1985.
- [7] Berni J. Alder and T. E. Wainwright. Studies in molecular dynamics. I. General method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.
- [8] E. W. Andrews, L. J. Gibson, and M. F. Ashby. The creep of cellular solids. *Acta materialia*, 47(10):2853–2863, 1999.
- [9] Damien André, Jean-Luc Charles, and Ivan Iordanoff. A new C++ workbench to develop discrete element simulations: GranOO. *2nd ECCOMAS Young Investigators Conference, Bordeaux, France*, 2013.
- [10] Damien André, Ivan Iordanoff, Jean-Luc Charles, and Jérôme Néauport. Discrete element method to simulate continuous material by using the cohesive beam model. *Computer Methods in Applied Mechanics and Engineering*, 213–216:113–125, 2012.
- [11] A. S. Argon. Plastic deformation in metallic glasses. *Acta Metallurgica*, 27, 1979.
- [12] Mike F. Ashby. Designing architected materials. *Scripta Materialia*, 68, January 2013.
- [13] Neil W. Ashcroft and N. David Mermin. *Solid State Physics*. Holt, Rinehart and Winston, 1976.
- [14] Michel Atten and Dominique Pestre. *Heinrich Hertz: l’administration de la preuve*. Presses universitaires de France, 2002.

- [15] Laurent Babout, Eric Maire, Jean-Yves Buffière, and Roger Fougères. Characterization by X-ray computed tomography of decohesion, porosity growth and coalescence in model metal matrix composites. *Acta Materialia*, 49(11):2055–2063, 2001.
- [16] David H. Bailey. Twelve ways to fool the masses in performance evaluation. *Supercomputing Review*, pages 54–55, August 1991.
- [17] Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. Deal.II – a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)*, 33(4):24, 2007.
- [18] P. Baque, E. Felder, J. Hyafil, and Y. Descatha. *Mise en forme des métaux - Calculs par la plasticité*. Dunod, 1973.
- [19] J. Bass. *Cours de mathématiques. Équations différentielles et aux dérivées partielles. Optimisation. Groupes de transformations. Méthodes numériques*, volume 2. Masson, 5 edition, 1978.
- [20] Klaus-Jürgen Bathe. *Finite element procedures in engineering analysis*. Civil engineering and engineering mechanics. Prentice-Hall, 1982.
- [21] P. Becker, S. R. Idelsohn, and E. Oñate. A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh. *Computational Mechanics*, 55(6):1091–1104, 2015.
- [22] T. Belytschko and K. Mish. Computability in non-linear solid mechanics. *International Journal for Numerical Methods in Engineering*, 52(1-2):3–21, 2001.
- [23] Ted Belytschko, Lei Gu, and Y. Y. Lu. Fracture and crack growth by element free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, 2(3A):519, 1994.
- [24] Ted Belytschko, Yury Krongauz, Daniel Organ, Mark Fleming, and Petr Krysl. Meshless methods: an overview and recent developments. *Computer methods in applied mechanics and engineering*, 139(1-4):3–47, 1996.
- [25] Diego A Berazategui, Miguel A. Cavaliere, Luca Montelatici, and Eduardo N. Dvorkin. On the modelling of complex 3D bulk metal forming processes via the pseudo-concentrations technique. application to the simulation of the mannesmann piercing process. *International journal for numerical methods in engineering*, 65(7):1113–1144, 2006.
- [26] Guy L. Bergel and Shaofan Li. The total and updated Lagrangian formulations of state-based peridynamics. *Computational Mechanics*, 58(2):351–370, 2016.
- [27] R. Berger, C. Kloss, A. Kohlmeyer, and S. Pirker. Hybrid parallelization of the LIGGGHTS open-source DEM code. *Powder Technology*, 278:234–247, 2015.
- [28] Marc Bernacki, Stéphane Andrietti, Jean-Loup Chenot, Pierre-Olivier Bouchard, Lionel Fourment, Elie Hachem, and Etienne Perchat. Recent and future developments in finite element metal forming simulation. *Computer Methods in Materials Science*, 15:265–293, 2015.
- [29] John Desmond Bernal. A geometrical approach to the structure of liquids. *Nature*, 183(4655):141–147, 1959.

- [30] John Desmond Bernal. The Bakerian lecture, 1962. The structure of liquids. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 280(1382):299–322, 1964.
- [31] M. A. Bessa, J. T. Foster, T. Belytschko, and Wing Kam Liu. A meshfree unification: reproducing kernel peridynamics. *Computational Mechanics*, 53(6):1251–1264, 2014.
- [32] Jacques Besson, Georges Cailletaud, Jean-Louis Chaboche, and Samuel Forest. *Non-linear mechanics of materials*, volume 167 of *Solid mechanics and its applications*. Springer, 2009.
- [33] Edwin Black. *IBM and the Holocaust: The strategic alliance between Nazi Germany and America's most powerful corporation*. Random House Inc., 2001.
- [34] Bruno Blais, Manon Lassaigne, Christoph Goniva, Louis Fradette, and François Bertrand. Development of an unresolved CFD–DEM model for the flow of viscous suspensions and its application to solid–liquid mixing. *Journal of Computational Physics*, 318:201–221, 2016.
- [35] Therese Bormann, Robin Gibaud, Pierre Lhuissier, Elodie Boller, Alexander Rack, and Luc Salvo. Investigation of the deformation process of amorphous/crystalline metal composites by in situ X-ray micro-tomography and DEM modelling. *Conférence matériaux et grands instruments & matériaux numériques – Analyses in situ : expérience / modélisation, Paris, France*, September 2016.
- [36] V. Boulos, L. Salvo, V. Fristot, P. Lhuissier, and D. Houzet. Investigating performance variations of an optimized GPU-ported granulometry algorithm. *18th International European Conference on Parallel and Distributed Computing, Rhodes Island, Greece*, 2012.
- [37] Piotr Breitskopf, Marcela Cruchaga, Diego Celentano, Pierre Villon, and Alain Rassineux. A front remeshing technique for a Lagrangian description of moving interfaces in two-fluid flows. *International journal for numerical methods in engineering*, 66(13):2035–2063, 2006.
- [38] J.-Y. Buffiere, E. Maire, J. Adrien, J.-P. Masse, and E. Boller. In situ experiments with X ray tomography: an attractive tool for experimental mechanics. *Experimental Mechanics*, 50, March 2010.
- [39] Arthur W. Burks. The invention of the universal electronic computer—how the electronic computer revolution began. *Future Generation Computer Systems*, 18(7):871–892, 2002.
- [40] J. C. Chapman, Burton Erickson, and N. J. Hoff. A theoretical and experimental investigation of creep buckling. *International Journal of Mechanical Sciences*, 1(2-3):145–174, 1960.
- [41] Shiyi Chen and Gary D. Doolen. Lattice Boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364, 1998.
- [42] A. C. F. Cocks and M. F. Ashby. Creep-buckling of cellular solids. *Acta materialia*, 48(13):3395–3400, 2000.
- [43] Alain Col. *L'emboutissage des aciers*. Dunod, 2010.

- [44] Gaël Combe and Jean-Noël Roux. Construction of granular assemblies under static loading. In Radjai and Dubois [181], chapter 6, pages 153–198.
- [45] Thierry Coupez, Nathalie Soyris, and Jean-Loup Chenot. 3-D finite element modelling of the forging process with automatic remeshing. *Journal of Materials Processing Technology*, 27(1):119–133, 1991.
- [46] Richard Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49, January 1943.
- [47] Richard Courant, Kurt Friedrichs, and Hans Lewy. On the partial difference equations of mathematical physics. *IBM journal of Research and Development*, 11(2):215–234, 1967.
- [48] Hardy Cross. Analysis of continuous frames by distributing fixed-end moments. volume 1, pages 919–928. May 1930.
- [49] Peter A. Cundall. A computer model for simulating progressive, large scale movement in blocky rock systems. In *Proceedings Symposium Int. Soc. Rock Mech., Nancy, France, Proc.*, volume 1, pages 129–136, 1971.
- [50] Peter A. Cundall and Otto D. L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47–65, 1979.
- [51] Pierre Daigneres. Les méthodes expérimentales. In Bernard Baudalet, editor, *Mise en forme des métaux et alliages*, chapter XIII, pages 225–231. Centre National de la Recherche Scientifique, 1976.
- [52] Bart J. Daly, Francis H. Harlow, James E. Welch, Everett E. Sanmann, and Edward N. Wilson. Numerical fluid dynamics using the particle-and-force method. Technical Report LA-3144, Los Alamos scientific laboratory, 1965.
- [53] Philippe De Bonnières. Intégration des relations de comportement viscoélastiques dans l’opérateur STAT_NON_LINE. Technical Report R5.03.08, révision 08af9e956266, Électricité De France R&D, August 2017. Documentation Code-Aster version 13.
- [54] Thomas De Soza. Notice d’utilisation du contact. Technical Report U2.04.04, révision 41bc29d2caf8, Électricité De France R&D, August 2016. Documentation Code-Aster version 13.
- [55] Gouri Dhatt, Gilbert Touzot, and Emmanuel Lefrançois. *Méthode des éléments finis*. Hermes Science, 2005.
- [56] Stéphane Di Iorio, Laurent Briottet, Edgar F. Rauch, and Didier Guichard. Plastic deformation, damage and rupture of PM Ti–6Al–4V at 20K under monotonic loading. *Acta materialia*, 55(1):105–118, 2007.
- [57] Florin Diacu. The solution of the n-body problem. *The mathematical intelligencer*, 18(3):66–70, 1996.
- [58] Jean Donea, Antonio Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. In Erwin Stein, René de Borst, and Thomas J.R. Hughes, editors, *Encyclopedia of computational mechanics*, volume 1: Fundamentals, chapter 14, pages 413–437. Wiley, 2004.
- [59] Frédéric Donzé and Sophie-Adélaïde Magnier. Formulation of a 3-D numerical model of brittle behaviour. *Geophysical Journal International*, 122(3):790–802, 1995.

- [60] Frédéric Donzé, Peter Mora, and Sophie-Adélie Magnier. Numerical simulation of faults and shear zones. *Geophysical Journal International*, 116(1):46–52, 1994.
- [61] Mathias Dratt and André Katterfeld. Coupling of FEM and DEM simulations to consider dynamic deformations under particle load. *Granular Matter*, 19(3):49, 2017.
- [62] Frédéric Dubois. Numerical modeling of granular media composed of polyhedral particles. In Radjai and Dubois [181], chapter 9, pages 233–62.
- [63] Stephen A. Edwards. Fundamentals of computer systems – advanced microarchitectures. www.cs.columbia.edu/~sedwards/classes/2012/3827-spring/advanced-arch-2011.pdf, 2011.
- [64] Anna Effeindzourou, Bruno Chareyre, Klaus Thoeni, Anna Giacomini, and François Kneib. Modelling of deformable structures in the general framework of the discrete element method. *Geotextiles and Geomembranes*, 44(2):143–156, 2016.
- [65] M. G. G. V. Elices, G. V. Guinea, J. Gomez, and J. Planas. The cohesive zone model: advantages, limitations and challenges. *Engineering fracture mechanics*, 69(2):137–163, 2002.
- [66] Martha W. Evans, Francis H. Harlow, and Eleazer Bromberg. The particle-in-cell method for hydrodynamic calculations. Technical Report LA-2139, Los Alamos scientific laboratory, 1957.
- [67] Experts Exchange. Processing power compared. <http://pages.experts-exchange.com/processing-power-compared>, 2015.
- [68] Michael L. Falk and James S. Langer. Dynamics of viscoplastic deformation in amorphous solids. *Physical Review E*, 57(6):7192–7205, 1998.
- [69] N. A. Fallah, C. Bailey, M. Cross, and G. A. Taylor. Comparison of finite element and finite volume methods application in geometrically nonlinear stress analysis. *Applied Mathematical Modelling*, 24(7):439–455, 2000.
- [70] Carlos A. Felippa. A historical outline of matrix structural analysis: a play in three acts. *Computers & Structures*, 79(14):1313–1324, 2001.
- [71] D. F. Ferguson. Value of π . *Nature*, 157, March 1946.
- [72] E. Fermi, J. Pasta, S. Ulam, and M. Tsingou. Studies of non linear problems. Technical Report LA-1940, Los Alamos scientific laboratory, 1955.
- [73] Antoine Ferré. *Élaboration et caractérisation 3D de l’endommagement dans les composites amorphes-cristallins à matrice aluminium*. PhD thesis, INSA Lyon, 2015.
- [74] Laurie J. Flynn. Intel halts development of 2 new microprocessors. *The New York Times*, May 2004.
- [75] Samuel Forest, Michel Amestoy, Gilles Damamme, Serge Kruch, Vincent Maurel, and Matthieu Mazière. *Mécanique des milieux continus*. Écoles des Mines de Paris, 2009-2010.
- [76] Thomas Forgber, Julian R. Tolchard, Abdelghafour Zaabout, Paul Inge Dahl, and Stefan Radl. Optimal particle parameters for CLC and CLR processes-predictions by intra-particle transport models and experimental validation. *IV International Conference on Particle-based Methods – Fundamentals and Applications*, 2015.

- [77] John T. Foster, Stewart Andrew Silling, and Wayne W. Chen. Viscoplasticity using peridynamics. *International journal for numerical methods in engineering*, 81(10):1242–1258, 2010.
- [78] Dominique François, André Pineau, and André Zaoui. *Comportement mécanique des matériaux. Viscoplasticité, endommagement, mécanique de la rupture, mécanique du contact*, volume 2. 1993.
- [79] Kirk A. Fraser, Lyne St-Georges, and Laszlo Istvan Kiss. A Mesh-Free Solid-Mechanics Approach for Simulating the Friction Stir-Welding Process. In Mahadzir Ishak, editor, *Joining Technologies*, chapter 3, pages 27–52. InTech, 2016.
- [80] Harold J. Frost and Mike F. Ashby. *Deformation mechanism maps: the plasticity and creep of metals and ceramics*. Pergamon press, 1982.
- [81] Jacques Gal and Claude Lemasson. *Mécanique: statique graphique, résistance des matériaux*, volume 2. Dunod, 1971.
- [82] David Gale. An indeterminate problem in classical mechanics. *American Mathematical Monthly*, 59, May 1952.
- [83] Georg C. Ganzenmüller, Stefan Hiermaier, and Michael May. On the similarity of meshless discretizations of peridynamics and smooth-particle hydrodynamics. *Computers & Structures*, 150:71–78, 2015.
- [84] Daniel Gay. *Matériaux composites*. Hermès, 1991.
- [85] Robin Gibaud. Déformation à chaud de composites métal cristallin / métal amorphe. Master’s thesis, Université de Technologie de Compiègne, 2014.
- [86] Robert A. Gingold and Joseph J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.
- [87] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48, 1991.
- [88] Herman H. Goldstine. *The computer from Pascal to von Neumann*. Princeton University Press, 1993.
- [89] Christoph Goniva, Christoph Kloss, Niels G. Deen, Johannes A. M. Kuipers, and Stefan Pirker. Influence of rolling friction on single spout fluidized bed simulation. *Particuology*, 10(5):582–591, 2012.
- [90] Nicolas Greffet. Utilisation de méthodes de résolution transitoires pour les problèmes quasi-statiques fortement non linéaires. Technical Report U2.04.07, révision 44d67daa72fd, Électricité De France R&D, July 2015. Documentation Code_Aster version 13.
- [91] David Alan Grier. Human computers: the first pioneers of the information age. *Endeavour*, 25(1):28–32, 2001.
- [92] David Alan Grier. *When computers were human*. Princeton University Press, 2013.
- [93] Alexandre Grothendieck. Allons-nous continuer la recherche scientifique?, 1972. Transcrit par Jacqueline Picard de la conférence-débat donnée au CERN le 27 janvier 1972.

- [94] T. Gruening, G. Kunze, and A. Katterfeld. Simulating the working process of construction machines. *3rd International Conference & Exhibition BulkSolids Europe*, 2010.
- [95] Torsten Gröger and André Katterfeld. On the numerical calibration of discrete element models for the simulation of bulk solids. *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, 2006.
- [96] Etienne Guesnet and Robin Gibaud. A heavvvy tutorial for liggghths. Technical report, Université Grenoble Alpes, 2017.
- [97] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, pages 49–52, 1902.
- [98] Manfred Hahn, Mohamed Bouriga, Bernd-H Kröplin, and Thomas Wallmersperger. Life time prediction of metallic materials with the discrete-element-method. *Computational Materials Science*, 71:146–156, 2013.
- [99] Barthelemy Harthong. *Modélisation du comportement des poudres métalliques à l'échelle du grain*. PhD thesis, Institut National Polytechnique de Grenoble, 2010.
- [100] A. Henderson. *ParaView Guide, A Parallel Visualization Application*. Kitware Inc., 2007.
- [101] John L. Hennessy and David A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, 4 edition, 2006.
- [102] Heinrich Hertz. Ueber die Berührung fester elastischer Körper. *Journal für die reine und angewandte Mathematik*, 92:156–171, 1882.
- [103] Georg Hess, Christian Richter, and André Katterfeld. Simulation of the dynamic interaction between bulk material and heavy equipment: Calibration and validation. *12th International Conference on Bulk Materials Storage, Handling and Transportation (ICBMH)*, page 427, 2016.
- [104] Douglas C. Hofmann, Jin-Yoo Suh, Aaron Wiest, Gang Duan, Mary-Laura Lind, Marios D. Demetriou, and William L. Johnson. Designing metallic glass matrix composites with high toughness and tensile ductility. *Nature*, 451(7182):1085–1089, 2008.
- [105] Alexander Hrennikoff. Solution of problems of elasticity by the framework method. *Journal of applied mechanics*, 8(4):169–175, 1941.
- [106] J. H. Hubbell and S. M. Seltzer. X-ray mass attenuation coefficients. <http://www.nist.gov/pml/data/xraycoef/index.cfm>. National Institute of Standards and Technology.
- [107] Sergio R. Idelsohn, Julio Marti, A. Limache, and Eugenio Oñate. Unified Lagrangian formulation for elastic solids and incompressible fluids: application to fluid–structure interaction problems via the PFEM. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1762–1776, 2008.
- [108] Sergio R. Idelsohn and Eugenio Oñate. Finite volumes and finite elements: two ‘good friends’. *International journal for numerical methods in engineering*, 37(19):3323–3341, 1994.

- [109] David Jauffrès, Christophe L. Martin, Aaron Lichtner, and Rajendra K. Bordia. Simulation of the toughness of partially sintered ceramics with realistic microstructures. *Acta Materialia*, 60(12):4685–4694, 2012.
- [110] Michel Jean. Frictional contact in rigid or deformable bodies: numerical simulation of geomaterials, 1995.
- [111] Michel Jean. Contact dynamics method. In Radjai and Dubois [181], chapter 2, pages 27–66.
- [112] Michel Jean, Vincent Acary, and Yann Monerie. Non-smooth contact dynamics approach of cohesive materials. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 359(1789):2497–2518, 2001.
- [113] Mohamed Jebahi, Damien André, Frédéric Dau, Jean-Luc Charles, and Ivan Iordanoff. Simulation of Vickers indentation of silica glass. *Journal of Non-Crystalline Solids*, 378:15–24, 2013.
- [114] Mohamed Jebahi, Damien André, Inigo Terreros, and Ivan Iordanoff. *Discrete element method to model 3D continuous materials*. Wiley, 2015.
- [115] Avinash C. Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. Classics in Applied Mathematics. Society of Industrial and Applied Mathematics, 2001.
- [116] Ken Kamrin, Chris H. Rycroft, and Jean-Christophe Nave. Reference map technique for finite-strain elasticity and fluid–solid interaction. *Journal of the Mechanics and Physics of Solids*, 60(11):1952–1969, 2012.
- [117] M. E. Kassner and M.-T. Pérez-Prado. Five-power-law creep in single phase metals and alloys. *Progress in materials science*, 45:1–102, 2000.
- [118] Tadahiko Kawai. New element models in discrete structural analysis. *Journal of the Society of Naval Architects of Japan*, 1977(141):174–180, 1977.
- [119] Ernst Gustav Kirsch. Die Fundamentalgleichungen der Theorie der Elasticität fester Körper, hergeleitet aus der Betrachtung eines Systems von Punkten, welche durch elastische Streben verbunden sind. *Zeitschrift des Vereines Deutscher Ingenieure*, 7(8), 1868.
- [120] Christoph Kloss, Christoph Goniva, Alice Hager, Stefan Amberger, and Stefan Pirker. Models, algorithms and validation for opensource DEM and CFD-DEM. *Progress in Computational Fluid Dynamics, An Int. J.*, 12(2/3):140–152, 2012.
- [121] Andrea Komoróczy, Steffen Abe, and Janos L. Urai. Meshless numerical modeling of brittle–viscous deformation: first results on boudinage and hydrofracturing using a coupling of discrete element method (DEM) and smoothed particle hydrodynamics (SPH). *Computational Geosciences*, 17(2):373–390, 2013.
- [122] Seiichi Koshizuka and Yoshiaki Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering*, 123(3):421–434, 1996.
- [123] Rishi Kumar, Sarshad Rommel, David Jauffrès, Pierre Lhuissier, and Christophe L. Martin. Effect of packing characteristics on the discrete element simulation of elasticity and buckling. *International Journal of Mechanical Sciences*, 110:14–21, 2016.

- [124] Ferenc Kun and Hans J. Herrmann. Transition from damage to fragmentation in collision of solids. *Physical Review E*, 59(3):2623–2632, 1999.
- [125] Frédy Lelong and Francis Cossiaux. *Le traçage en structures métalliques*. Casteilla, 1999.
- [126] Jean Lemaitre and Jean-Louis Chaboche. *Mechanics of solid materials*. Cambridge university press, 1994.
- [127] Xavier Leroy. How I found a bug in Intel Skylake processors. <http://gallium.inria.fr/blog/intel-skylake-bug/>, July 2017.
- [128] P. Lhuissier, A. Villanueva Fernandez, L. Salvo, and J.-J. Blandin. High temperature deformation and associated 3D characterisation of damage in magnesium alloys. *Materials Science Forum*, 706-709, 1 2012.
- [129] Qing-Feng Li, Hui-Ru Weng, Zhong-Yuan Suo, Ying-Lei Ren, Xiao-Guang Yuan, and Ke-Qiang Qiu. Microstructure and mechanical properties of bulk Mg–Zn–Ca amorphous alloys and amorphous matrix composites. *Materials Science and Engineering: A*, 487(1):301–308, 2008.
- [130] Larry D. Libersky, Firooz A. Allahdadi, Theodore C. Carney, Jim R. Hipp, and Albert G. Petschek. High strain lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response. *Journal of Computational Physics*, 109:67–75, 1993.
- [131] Larry D. Libersky and Albert G. Petschek. Smooth particle hydrodynamics with strength of materials. In Harold E. Trease, Martin F. Fritts, and William P. Crowley, editors, *Advances in the Free-Lagrange method: including contributions on adaptive gridding and the smooth particle hydrodynamics method*, volume 395. Springer, 1991.
- [132] R. Lichtenheldt. A stable, implicit time integration scheme for discrete element method and contact problems in dynamics. *V International Conference on Particle-Based Methods, Hannover*, September 2017.
- [133] Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. Résolution d’EDP par un schéma en temps ”pararéel”. *Comptes Rendus de l’Académie des Sciences*, 332(7):661–668, 2001.
- [134] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures*, 11, 1980.
- [135] T. J. Liszka, C. A. M. Duarte, and W. W. Tworzydło. Hp-meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):263–288, 1996.
- [136] Wing Kam Liu, Sukky Jun, Shaofan Li, Jonathan Adee, and Ted Belytschko. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, 38(10):1655–1679, 1995.
- [137] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International journal for numerical methods in fluids*, 20(8-9):1081–1106, 1995.
- [138] Eli Livne. An implicit method for two-dimensional hydrodynamics. *The Astrophysical Journal*, 412:634–647, 1993.

- [139] A. Longuet, Y. Robert, E. Aeby-Gautier, B. Appolaire, J. F. Mariage, C. Colin, and G. Cailletaud. A multiphase mechanical model for Ti-6Al-4V: Application to the modeling of laser assisted processing. *Computational Materials Science*, 46(3):761–766, 2009.
- [140] Augustus Edward Hough Love. *A treatise on the mathematical theory of elasticity*, volume 1. Cambridge university press, 1892.
- [141] Leon B. Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977.
- [142] A. Létalnet, R. Pasquier, and J. Triouleyre. *Chaudronnerie - Constructions métalliques*. Techniques et normalisation - Technor. Delagrave, 1962.
- [143] Lucien Malavard. The use of rheo-electrical analogies in certain aerodynamical problems. *The Aeronautical Journal*, 51(441):739–756, 1947.
- [144] Louis Marciliac. Étude de la déformation à chaud d'un composite métal amorphe / métal cristallin. Master's thesis, Grenoble-INP PHELMMA, July 2013.
- [145] C. L. Martin, D. Bouvard, and S. Shima. Study of particle rearrangement during powder compaction by the discrete element method. *Journal of the Mechanics and Physics of Solids*, 51(4):667–693, 2003.
- [146] James Clerk Maxwell. On the dynamical theory of gases. *Philosophical transactions of the Royal Society of London*, 157:49–88, 1867.
- [147] James Clerk Maxwell. On reciprocal figures, frames, and diagrams of forces. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, 26(1), January 1870.
- [148] W. Barksdale Maynard. Daybreak of the digital age. *Princeton Alumni Weekly*, 112(10):28–33, April 2012.
- [149] Sean McNamara. Molecular dynamics method. In Radjai and Dubois [181], chapter 1, pages 1–23.
- [150] Jens M. Melenk and Ivo Babuška. The partition of unity finite element method: basic theory and applications. *Computer methods in applied mechanics and engineering*, 139(1-4):289–314, 1996.
- [151] Mark Michael. *A discrete approach to describe the kinematics between snow and a tire tread*. PhD thesis, Université du Luxembourg, 2014.
- [152] GDR Midi. On dense granular flows. *European Physical Journal E-Soft Matter*, 14(4), 2004.
- [153] Joseph J. Monaghan. SPH without a tensile instability. *Journal of Computational Physics*, 159(2):290–311, 2000.
- [154] Laurent Monasse and Christian Mariotti. An energy-preserving discrete element method for elastodynamics. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(6):1527–1553, 2012.
- [155] David Morin. <http://www.people.fas.harvard.edu/~djmorin/waves/>, 2010. A potential book on Waves.

- [156] F. Moukalled, L. Mangani, and M. Darwish. The finite volume method in computational fluid dynamics. 113, 2016.
- [157] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150, 1999.
- [158] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational mechanics*, 10(5):307–318, 1992.
- [159] Thomas R. Nicely. Pentium FDIV flaw. <http://www.trnicely.net/pentbug/pentbug.html>, August 2011.
- [160] Sophie Nowak. *Effet de la composition et de la technique d’élaboration sur le comportement mécanique des verres métalliques base zirconium*. PhD thesis, Université Paris-Est, 2009.
- [161] Sophie Nowak, Loïc Perrière, Lucas Dembinski, Sandrine Tusseau-Nenez, and Yannick Champion. Approach of the spark plasma sintering mechanism in $\text{Zr}_{57}\text{Cu}_{20}\text{Al}_{10}\text{Ni}_8\text{Ti}_5$ metallic glass. *Journal of Alloys and Compounds*, 509(3):1011–1019, 2011.
- [162] William L. Oberkampf, Timothy G. Trucano, and Charles Hirsch. Verification, validation, and predictive capability in computational engineering and physics. *Applied Mechanics Reviews*, 57(5):345–384, 2004.
- [163] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer, 2006.
- [164] Martin Ostojca-Starzewski. Lattice models in micromechanics. *Applied Mechanics Reviews*, 55(1):35–60, 2002.
- [165] Michael L. Parks, Richard B. Lehoucq, Steven J. Plimpton, and Stewart A. Silling. Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications*, 179(11):777–783, 2008.
- [166] John R. Pasta and Stanislaw Ulam. Heuristic numerical work in some problems of hydrodynamics. *Mathematical tables and other aids to computation*, 13(65):1–12, 1959.
- [167] Ch. PavanaChand and R. KrishnaKumar. Remeshing issues in the finite element analysis of metal forming problems. *Journal of Materials Processing Technology*, 75(1):63–74, 1998.
- [168] Alan J. Perlis. Epigrams on programming. *SIGPLAN Notices*, 17(9):7–13, 1982.
- [169] Loïc Perrière and Yannick Champion. Phases distribution dependent strength in metallic glass–aluminium composites prepared by spark plasma sintering. *Materials Science and Engineering: A*, 548:112–117, 2012.
- [170] Charles S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of computational physics*, 10(2):252–271, 1972.
- [171] Yuri P. Petrov and Valery S. Sizikov. *Well-posed, ill-posed, and intermediate problems with applications*, volume 49 of *Inverse and ill-posed problems*. Walter de Gruyter, 2005.

- [172] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.
- [173] Alexander Podlozhnyuk, Stefan Pirker, and Christoph Kloss. Efficient implementation of superquadric particles in discrete element method within an open-source framework. *Computational Particle Mechanics*, 4(1):101–118, 2017.
- [174] Thorsten Pöschel and Thomas Schwager. *Computational granular dynamics: models and algorithms*. Springer, 2005.
- [175] Michel Provost, Philippe De Kemmeter, and David Attas. *Comment tout ça tient? Voyage au pays des structures*. Alice Éditions, 2011.
- [176] S. G. Psakhie, Y. Horie, G. P. Ostermeyer, S. Yu. Korostelev, A. Yu. Smolin, E. V. Shilko, A. I. Dmitriev, S. Blatnik, M. Špegel, and S. Zavšek. Movable cellular automata method for simulating materials with mesostructure. *Theoretical and applied fracture mechanics*, 37(1):311–334, 2001.
- [177] Sergey Grigorievich Psakhie, Ya. Horie, S. Yu. Korostelev, A. Yu. Smolin, A. I. Dmitriev, E. V. Shilko, and S. V. Alekseev. Method of movable cellular automata as a tool for simulation within the framework of mesomechanics. *Russian Physics Journal*, 38(11):1157–1168, 1995.
- [178] Sergey Grigorievich Psakhie, A. Yu. Smolin, E. V. Shilko, G. M. Anikeeva, Yu. S. Pogozhev, M. I. Petrzhhik, and E. A. Levashov. Modeling nanoindentation of TiC-CaPON coating on Ti substrate using movable cellular automaton method. *Computational materials science*, 76:89–98, 2013.
- [179] J. W. Qiao, S. Wang, Y. Zhang, P. K. Liaw, and G. L. Chen. Large plasticity and tensile necking of Zr-based bulk-metallic-glass-matrix composites synthesized by the Bridgman solidification. *Applied Physics Letters*, 94(15):151905, 2009.
- [180] Junwei Qiao. In-situ dendrite/metallic glass matrix composites: A review. *Journal of Materials Science and Technology*, 29(8):685–701, 2013.
- [181] Farhang Radjai and Frédéric Dubois, editors. *Discrete-element modeling of granular materials*. Wiley-Iste, 2011.
- [182] Jennifer Ragani. *Élaboration par co-déformation de matériaux stratifiés alliage léger/verre métallique*. PhD thesis, Université de Grenoble, 2011.
- [183] A. Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A):A405, 1964.
- [184] John William Strutt Baron Rayleigh. *The theory of sound*, volume 1. Macmillan, 1894.
- [185] Nathalie Revol and Philippe Théveny. Numerical reproducibility and parallel computations: Issues for interval algorithms. *IEEE Transactions on Computers*, 63(8):1915–1924, 2014.
- [186] Thomas Richards. *At Work with Grotowski on Physical Actions*. Routledge, 1995.
- [187] Wolfgang Riedel. Beiträge zur Lösung des ebenen Problems eines elastischen Körpers mittels der Airyschen Spannungsfunktion. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 7(3):169–188, 1927.

- [188] Walter Ritz. Theorie der Transversalschwingungen einer quadratischen Platte mit freien Rändern. *Annalen der Physik*, 333(4):737–786, 1909.
- [189] Yvan Rivierre. *Self-Stabilizing Algorithms for Constructing Distributed Spanning Structures*. PhD thesis, Université de Grenoble, 2013.
- [190] Sergio Rodolfo Idelsohn, Norberto Marcelo Nigro, Juan Marcelo Gimenez, Riccardo Rossi, and Julio Marcelo Marti. A fast and accurate method to solve the incompressible Navier-Stokes equations. *Engineering Computations*, 30(2):197–222, 2013.
- [191] E. Rougier, A. Munjiza, and N. W. M. John. Numerical comparison of some explicit time integration schemes used in DEM, FEM/DEM and molecular dynamics. *International journal for numerical methods in engineering*, 61(6):856–879, 2004.
- [192] Paul Rougée. *Mécanique des grandes transformations*, volume 25 of *Mathématiques & Applications*. Springer, 1997.
- [193] Denis Roussel, Aaron Lichtner, David Jauffrès, Julie Villanova, Rajendra K. Bordia, and Christophe L. Martin. Strength of hierarchically porous ceramics: Discrete simulations on X-ray nanotomography images. *Scripta Materialia*, 113:250–253, 2016.
- [194] Jean-Noël Roux and François Chevoir. Dimensional analysis and control parameters. In Radjai and Dubois [181], chapter 8, pages 199–232.
- [195] Jean-Noël Roux and Gaël Combe. Quasistatic rheology and the origins of strain. *Comptes Rendus Physique*, 3(2):131–140, 2002.
- [196] Jean-Noël Roux and Gaël Combe. Quasi-static methods. In Radjai and Dubois [181], chapter 3, pages 67–102.
- [197] Ulrich Rüde. Algorithmic efficiency and the energy wall. *2nd workshop on power-aware computing*, July 2017.
- [198] Michel Saby, P.-O. Bouchard, and Marc Bernacki. Void closure criteria for hot metal forming: a review. *Journal of Manufacturing Processes*, 19:239–250, 2015.
- [199] Malcolm Sambridge, Jean Braun, and Herbert McQueen. Geophysical parametrization and interpolation of irregular data using natural neighbours. *Geophysical Journal International*, 122(3):837–857, 1995.
- [200] Sandia National Laboratories. LAMMPS benchmarks. lammmps.sandia.gov/bench.html, June 2017.
- [201] Sandia National Laboratories. LAMMPS FAQ (frequently asked questions). lammmps.sandia.gov/FAQ.html, June 2017.
- [202] N. Sator, S. Mechkov, and F. Sausset. Generic behaviours in impact fragmentation. *Europhysics Letters*, 81(4):44002, 2008.
- [203] Stewart Schlesinger. Terminology for model credibility. *Simulation: Transactions of the Society for Modeling and Simulation*, 32, March 1979.
- [204] Christopher A. Schuh, Todd C. Hufnagel, and Upadrasta Ramamurty. Mechanical behavior of amorphous alloys. *Acta Materialia*, 55(12):4067–4109, 2007.
- [205] Samuel M. Selby, editor. *Standard mathematical tables*. The Chemical Rubber Co., 17 edition, 1969.

- [206] Brigitte Servatius and Herman Servatius. Generic and abstract rigidity. In Michael F. Thorpe and Phillip M. Duxbury, editors, *Rigidity theory and applications*, Fundamental Materials Research, pages 1–20. Springer, 1999.
- [207] H. W. Sheng, W. K. Luo, F. M. Alamgir, J. M. Bai, and E. Ma. Atomic packing and short-to-medium-range order in metallic glasses. *Nature*, 439(7075):419–425, 2006.
- [208] Robin Sibson. A brief description of natural neighbor interpolation. *Interpreting multivariate data*, pages 21–36, 1981.
- [209] Stewart A. Silling and Ebrahim Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures*, 83(17):1526–1535, 2005.
- [210] Stewart A. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88(2):151–184, 2007.
- [211] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*, volume 7 of *Interdisciplinary Applied Mathematics*. Springer, 1998.
- [212] Gérard Simon. L’astrologie de Kepler: le sens d’une réforme. *L’astronomie*, 86:325, 1972.
- [213] Bart Smeets, Tim Odenhal, Janos Keresztes, Simon Vanmaercke, Paul Van Liedekerke, Engelbert Tijskens, Wouter Saeys, Hans Van Oosterwyck, and Herman Ramon. Modeling contact interactions between triangulated rounded bodies for the discrete element method. *Computer Methods in Applied Mechanics and Engineering*, 277:219–238, 2014.
- [214] Václav Šmilauer. *Woo Documentation, Release 1.0+rev4036-git-052c18c*. <https://woodem.org/Woo.pdf>, 2016.
- [215] Václav Šmilauer, Emanuele Catalano, Bruno Chareyre, Sergei Dorofeenko, Jerome Duriez, Anton Gladky, Janek Kozicki, Chiara Modenese, Luc Scholtès, Luc Sibille, et al. *Yade documentation*. yade-dem.org/doc/, 2 edition, 2015.
- [216] Alvy Ray Smith. A pixel is not a little square! (and a voxel is not a little cube). Technical Report 6, Microsoft Computer Graphics, July 1995. Technical Memo.
- [217] A. Yu. Smolin, E. V. Shilko, S. V. Astafurov, I. S. Konovalenko, S. P. Buyakova, and S. G. Psakhie. Modeling mechanical behaviors of composites with various ratios of matrix–inclusion properties using movable cellular automaton method. *Defence Technology*, 11(1):18–34, 2015.
- [218] Robert Speck, Daniel Ruprecht, Rolf Krause, Matthew Emmett, Michael Minion, Mathias Winkel, and Paul Gibbon. A massively space-time parallel N-body solver. *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 92, 2012.
- [219] Erich Strohmaier, Jack Dongarra, Horst Simon, and Martin Meuer. TOP500 list. www.top500.org, August 2017.
- [220] Alexander Stukowski. Visualization and analysis of atomistic simulation data with OVITO - the open visualization tool. *Modelling Simul. Mater. Sci. Eng.*, 18(1):015012, 2010. www.ovito.org.
- [221] Alexander Stukowski. Computational analysis methods in atomistic modeling of crystals. *JOM*, 66(3):399–407, 2014.

- [222] Natarajan Sukumar. *The natural element method in solid mechanics*. PhD thesis, Northwestern University, June 1998.
- [223] Natarajan Sukumar, B. Moran, and T. Belytschko. The natural element method in solid mechanics. *International journal for numerical methods in engineering*, 43(5), 1998.
- [224] Deborah Sulsky, Zhen Chen, and Howard L. Schreyer. A particle method for history-dependent materials. *Computer methods in applied mechanics and engineering*, 118(1-2):179–196, 1994.
- [225] Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87, 1995.
- [226] C. Suryanarayana and Akihisa Inoue. *Bulk metallic glasses*, volume 361. Taylor & Francis, 2011.
- [227] Robert P. Swift, Carl R. Hagelberg, Ted C. Carney, Doren Greening, and Michael Hilt. Modeling stress-induced damage from impact recovery experiments. In *Proceedings of the ETCE/OMAE 2000 Joint Conference: Energy for the New Millennium*, 2000.
- [228] G. I. Taylor. The mechanism of plastic deformation of crystals. part I. theoretical. *Proceedings Mathematical Physical & Engineering Sciences*, 145, July 1934.
- [229] S. Terzi, L. Salvo, M. Suéry, E. Boller, and A. K. Dahle. Control and characterisation of nucleation in liquids and solids. *The 12th International Conference of Aluminium Alloys, Yokohama, Japan*, September 2010.
- [230] Tayfun E. Tezduyar, Mittal Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary tests. *Computer methods in applied mechanics and engineering*, 94(3):353–371, 1992.
- [231] Stephen P. Timoshenko. *Résistance des matériaux*, volume 2. Théorie développée et problèmes. Dunod, 1968.
- [232] L. Traversoni. Natural neighbour finite elements. *WIT Transactions on Ecology and the Environment*, 8, 1994.
- [233] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp. Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences*, 23(9), September 1956.
- [234] Michael D. Uchic, Lorenz Holzer, Beverley J. Inkson, Edward L. Principe, and Paul Munroe. Three-dimensional microstructural characterization using focused ion beam tomography. *Mrs Bulletin*, 32(5):408–416, 2007.
- [235] O. Ya. Viro, O. A. Ivanov, N. Yu. Netsvetaev, and V. M. Kharlamov. *Elementary Topology: Problem Textbook*. American Mathematical Society, 2008. Abridged version.
- [236] Charles Voivret, Farhang Radjaï, and Jean-Yves Delenne. Particle assembling methods. In Radjaï and Dubois [181], chapter 5, pages 123–150.
- [237] Richard von Mises. On Saint Venant’s principle. *Bulletin of the American Mathematical Society*, 51(8):555–562, 1945.

- [238] John von Neumann. The principles of large-scale computing machines. *Annals of the History of Computing*, 10, 1988.
- [239] A. Wang, P. F. Thomson, and P. D. Hodgson. A study of pore closure and welding in hot rolling process. *Journal of Materials Processing Technology*, 60(1-4):95–102, 1996.
- [240] D. Weatherley, W. Hancock, and V. Boros. ESyS-particle tutorial and user’s guide version 2.3.1. https://launchpadlibrarian.net/187793286/ESyS-Particle_Tutorial.pdf, 2014.
- [241] Benjamin Lee Whorf. *Language, Thought and Reality. Selected Writings of Benjamin Lee Whorf*. The M.I.T. press, 1956.
- [242] K. Wieghardt. Über einen Grenzübergang der Elastizitätslehre und seine Anwendung auf die Statik hochgradig statisch unbestimmter Fachwerke. *Verhandlungen des Vereins zur Beförderung des Gewerbefleißes*, 85:139–176, 1906.
- [243] Tao Zhang, Eric Maire, Jerome Adrien, Patrick R. Onck, and Luc Salvo. Local tomography study of the fracture of an ERG metal foam. *Advanced Engineering Materials*, 15(8):767–772, 2013.
- [244] Olek C. Zienkiewicz and Robert L. Taylor. *The finite element method for solid and structural mechanics*. Elsevier, 6 edition, 2005.
- [245] Électricité De France R&D. Code.aster open source - general FEA software. www.code-aster.org.